2-2014

Exploiting Energy Harvesting for Passive Embedded Computing Systems

Jeremy Joel Gummeson
University of Massachusetts - Amherst, jgummeso@student.umass.edu

Recommended Citation

Gummeson, Jeremy Joel, "Exploiting Energy Harvesting for Passive Embedded Computing Systems" (2014). *Doctoral Dissertations 2014-current.* Paper 8.

EXPLOITING ENERGY HARVESTING FOR PASSIVE EMBEDDED COMPUTING SYSTEMS

A Dissertation Presented

by

JEREMY GUMMESON

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

February 2014

Department of Electrical and Computer Engineering

EXPLOITING ENERGY HARVESTING FOR PASSIVE EMBEDDED COMPUTING SYSTEMS

A Dissertation Presented

by

JEREMY GUMMESON

Approved as to style and content by:

Tilman Wolf, Co-chair

Deepak Ganesan, Co-chair

Lixin Gao, Member

Bodhi Priyantha, Member

Michael Zink, Member

C.V. Hollot, Department Head
Department of Electrical and Computer Engineering

To my Mom, Dad, and the rest of the Js

ACKNOWLEDGMENTS

This thesis would not have been possible without the guidance of my advisors and the support of my friends and family. Professor Deepak Ganesan helped shape me from a novice embedded systems engineer to a researcher capable of defining specific problems, implementing solutions, exploring tradeoffs, and quantifying performance. As each paper deadline approached, his clarity in understanding the difference between important research problems and minute implementation details kept our research on track and presented in the best light possible. All of these lessons learned will be completely invaluable during the course of my career.

During my graduate and undergraduate careers, I've had the pleasure of learning from Professors Tilman Wolf, Lixin Gao, and Michael Zink. I particularly appreciate Tilman for teaching me the fundamentals of networks during the first year of my graduate studies and being such a flexible advisor. Lixin Gao and Michael Zink were also both very influential in shaping me as a researcher. In addition to providing valuable insights into preparing my dissertation, Lixin was also my undergraduate advisor. I first got to know Mike while he was a research scientist in the Computer Science department and his experience as a systems researcher helped guide the way I approach systems design.

I've also had the unique opportunity of being funded by, and having extensive collaborations with, the Computer Science Department. Several projects in collaboration with Professor Kevin Fu got me interested in passive systems and RFID. The foundations of this thesis were forged by interactions with him and his lab over the course of several years. During the first several years of my graduate work, Professor Prashant Shenoy served a significant role as an advisor and served on my Master's

thesis committee. I've had the opportunity to publish several papers with him that were outside the scope of this thesis. His patience and advice throughout my graduate career were essential. While working with Prashant, I also worked closely with David Irwin and Emmanuel Cecchet. I am forever grateful for the time they spent helping me formulate several research projects and assist with practical systems implementations. During my Master's thesis work I also had the pleasure of working with Professor Mark Corner, whose approach towards systems building was highly influential.

I was extremely fortunate to have spent the better part of a year working with Bodhi Priyantha at Microsoft Research. During my first eight months of my time at Microsoft, I had the truly unique opportunity of transitioning a research demonstration into a tangible product. This was an experience that would have been impossible in a strictly academic setting. After returning to the university, I continued working with Bodhi on an NFC energy harvesting project that became one of the core pieces of this dissertation. I again had the opportunity to return to Microsoft for a summer internship that was more research focused and worked on a project that partially built on our collaboration at UMass. I also would like to acknowledge Jie Liu and the rest of the Sensors and Energy Research Group at Microsoft Research for giving me invaluable insight into the world of industrial research.

After having spent several years following a body of research on passive embedded computing from the University of Washington and Intel Research, I had the distinct pleasure of collaborating with Alanson Sample, Artem Dementyev, Aaron Parks, and Professor Joshua Smith. We spent many hours together building an open source platform for experimentation with Near Field Communications. Their experience in building well-supported, well-documented, and robust research platforms is knowledge I am very grateful to take with me in the future.

I've met many peer researchers through my time at UMass. First and foremost, I need to thank my frequent collaborators: Shane Clark and Pengyu Zhang. Shane and Pengyu worked with me on a large portion of this thesis and without their contributions, this work would not have been possible. Next I need to thank the Sensors and LASS labs in the Computer Science Department. Specifically I'd like to thank (in no particular order) Ming Li, Tingxin Yan, Abhinav Parate, Moaj Musthag, Addison Mayberry, Sean Barker, Pan Hu, Tian Guo, Navin Sharma, Vijay Kumar, Aditya Misra, Siddarth Gupta, Vikas Kumar, Aditya Nemmaluri, Tim Wood, Devesh Agarwal, Rahul Singh, Upendra Sharma, Peter Desnoyers, and Gaurav Mathur. I had many productive discussions in the office and at lab meetings that greatly impacted my approach towards research. I also want to acknowledge the SPQR and PRISM labs in the Computer Science Department. During the early years of my graduate studies Nilanjan Banerjee, Jacob Sorber, and Aruna Balasubramanian served as great role models and showed me how hard work and perseverance lead to success. I also want to acknowledge Negin Salajegheh, Andres Molina, and Benjamin Ransford. We frequently put our heads together in understanding a whole host of issues in passive computing research.

I also had the opportunity of advising and having a long term collaboration with an exceptional undergraduate student. Derek Thrasher was critical in the completion of two of the topics covered in this thesis – without his contributions, I would have had many more sleepless nights.

During my undergraduate career, I never would have imagined I'd end up going down a path that led to a Ph.D. I'd like to thank Professor Maciej Ciesielski for getting me started – my summer research experience with him and his lab was a transformative experience. I'd also like to acknowledge Baird Soules who has been an incredible advisor and friend over the years. It was during my time working on electronic music and MIDI projects with him in Keith's trailer that I decided hands on embedded systems research is what I wanted to do with the rest of my life.

Throughout my graduate studies I've made some great friends that provided much needed support when times got tough. I'd specifically like to thank Gal Niv, David and Erin Cooper Megan Olson, Shiraj Sen, and Brandon McPhail. It would have been difficult to keep going without such good company.

Last, but certainly not least, I'd like to thank all my of my family and local friends. Unlike many graduate students who pursued advanced degrees far from home, I had access to a unique support network right down the road in Belchertown.

ABSTRACT

EXPLOITING ENERGY HARVESTING FOR PASSIVE EMBEDDED COMPUTING SYSTEMS

FEBRUARY 2014

JEREMY GUMMESON

B.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Tilman Wolf and Professor Deepak Ganesan

The key limitation in mobile computing systems is energy - without a stable power supply, these systems cannot process, store, or communicate data. This problem is of particular interest since the storage density of battery technologies do not follow scaling trends similar to Moore's law. This means that depending on application performance requirements and lifetime objectives, a battery may dominate the overall system weight and form factor; this could result in an overall size that is either inconvenient or unacceptable for a particular application. As device features have scaled down in size, entire embedded systems have been implemented on a single die or chip, resulting in the battery becoming the form factor bottleneck.

One way to diminish the impact that batteries have on mobile embedded system design is to decrease reliance on buffered energy by providing the ability to harvest power from the environment or infrastructure. There are a spectrum of design choices available that utilize harvested power, but of particular interest are those that use small energy buffers and depend almost entirely on harvested power; by minimizing buffer size, we decrease form factor and mitigate reliance on batteries. Since harvested power is not continuously available in embedded computing systems, this brings forth a unique set of design challenges.

First, we address the design challenges that emerge from mobile computing systems that use minimal energy buffers. Specifically, we explore the design space of a computational radio frequency identification (RFID) platform that uses a small solar harvesting unit to replenish a capacitor-based energy storage unit. We show that such a system's performance can be enhanced while in a reader's field of interrogation and also allows for device operation while completely decoupled from reader infrastructure. We also provide a toolset that simulates system performance using a set of experimentally obtained light intensity traces gathered from a mobile subject.

Next, we show how energy buffered from such a harvesting-based system can be used to implement an efficient burst protocol that allows a computational RFID to quickly offload buffered data while in contact with a reader. The burst mechanism is implemented by re-purposing existing RFID protocol primitives, which allows for compatibility with existing reader infrastructure. We show that bursts provide significant improvements to individual tag throughput, while co-existing with tags that do not use the burst protocol.

Next, we show that energy harvesting can be used to enable a novel security mechanism for embedded devices equipped with Near Field Communications (NFC). NFC is growing in pervasiveness, especially on mobile phones, but many open security questions remain. We enable NFC security by harvesting energy via magnetic induction, use the harvested energy to power an integrated reader chip, and selectively block malicious messages via passive load modulation after sniffing message contents.

We show that such a platform is feasible based on energy harvested opportunistically from mobile phones, successfully blocking a class of messages while allowing others through.

Finally, we demonstrate that energy harvested from mobile phones can be used to implement wirelessly powered ubiquitous displays. One drawback of illuminated displays is that they need a continuous source of power to maintain their state – this is an undesirable property, especially since the display is typically the highest power consumption system component of embedded devices. Electronic paper technologies eliminate this drawback by providing a display that requires no energy to maintain state. By combining NFC energy harvesting and communication, and electronic paper technologies, we implement a companion display for mobile phones that obtains all the energy required for a display update while communicating with a user application running on a mobile phone. The companion display assists the phone in displaying static information while the power hungry display remains unpowered.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1
INTRODUCTION

Embedded computing systems are an essential component of modern infrastructure; these systems are deployed in a diverse array of applications including continuous environmental sensing, consumer products, industrial manufacturing, avionics, and mobile health. As trends in microelectronic circuit design continue to diminish device feature size and power consumption, applications that were previously thought to be infeasible are becoming tantalizingly close to reality. While these trends are certainly a boon to any embedded computing system, there remain significant challenges for applications that require untethered, mobile operation as a result of energy storage limitations.

This thesis advocates for *passive* embedded computing systems that are passively powered by energy harvesting and passively communicate by modifying a signal transmitted by an initiator. While the core technology exists that allows such systems to be implemented, we need to fundamentally change the ways we approach system design to provide support for robust applications.

Traditionally, untethered systems are designed around a large energy buffer, typically in the form of a battery. To ensure that a given application meets a set of requirements, the energy buffer is sized to accommodate a particular workload given system power consumption and desired lifetime. Since *passive* systems rely on harvested energy, which varies over time, we must rethink this design process and provide new hardware and software building blocks that provide the tools necessary to implement applications. Exploring alternative design processes and providing these tools and building blocks are the goals of this thesis.

1.1 Background and Motivation

Over the past decade, there has been an intense push towards more deeply embedding computing and sensing systems. At the smallest scale, these embedded computing systems take the form of tiny invisible sensors that continuously monitor our infrastructure [33], internal health [19], or the world we live in [32]. In some of these scenarios, it might make sense to power these devices directly from the grid [52], but many will benefit from untethered operation – even those embedded near powered infrastructure [41].

Generally speaking, there are two approaches used in the design of untethered embedded systems: *active* and *passive.* Active systems use batteries for power and active radios for communication, while passive systems harvest the energy required for operation during deployment and use passive radios to mitigate the energy burden of wireless communication. First we make a case for passive embedded system design, then go on to highlight the challenges specific to passive system design.

1.1.1 A case for passive embedded system design

Untethered embedded computing systems need to achieve long deployment lifetimes, operate continuously and wirelessly communicate while maintaining small form factor. There are a variety of approaches that may be used to achieve these objectives, however we can split them into two main categories: *active* and *passive.* Active systems use batteries as a power source that slowly depletes over their operational lifetime and wirelessly communicate using an active radio that generates an RF carrier. Passive systems harvest energy from their surroundings or infrastructure and use passive radios that selectively reflect a carrier generated by an infrastructure powered reader. We now highlight the differences in these approaches.

Power. Untethered embedded computing systems that are actively powered need to carry the energy with them that allows them to operate continuously and communicate wirelessly. The most obvious way to power these systems is with a battery. The key problem with this approach is that to achieve long deployment lifetimes, we will have to increase form factor because more energy will be required. Increases in battery energy storage density could partially address this problem, however, these improvements are dictated by improvements to battery chemistry [20] and do not follow the predictable scaling trends observed in the semiconductor industry [69]. Even though these densities could improve with the development of new battery chemistries or alternative storage mechanisms, the form factor of a mobile device's battery will always be proportional to the desired deployment lifetime – if a mobile device can be made to work with less stored energy, it can be made smaller and lighter.

Another approach towards powering these systems is to passively power them using a variety of energy harvesting techniques. Rather than relying on the battery to store the requisite energy for continuous operation and communications, this energy may be harvested over the course of deployment. Instead of sizing the battery proportional to the desired operational lifetime, the battery may instead be sized according to the particulars of the energy harvesting technique used. By adhering to these principles, energy harvesting can be used to achieve continuous operation while retaining a small form factor.

Communications. Untethered embedded systems using active communication techniques generate an RF carrier used to send data to peer devices or infrastructure. Fundamentally, this is a power hungry way to achieve wireless communication – one could consider an active radio transmission as an extremely low efficiency energy transfer system, where the transmitter sends energy to a receiver that discards this transmitted energy after receiving the encoded data. While there have been a variety of techniques proposed to reduce the amount of energy consumed in sending and receiving data using active radios, this still remains the most power hungry operation of low-power, untethered embedded systems.

Another way to achieve wireless communications is using *passive* techniques. Instead of actively generating an RF carrier, passive communication is achieved by selectively reflecting an RF carrier back to a reader. Since readers are either connected to infrastructure or are significantly less energy-constrained than passive devices, the energy burden of communication is shifted from the low-power embedded device to the unconstrained reader. Using such an approach allows one to achieve wireless communications with essentially zero additional power consumption.

Combining passive power and passive communication techniques allow an untethered embedded device to achieve all of our desired design objectives simultaneously. Passive power allows for long deployment lifetimes and continuous operation while retaining small form factor; the use of passive communication enables communication without sacrificing the other objectives. Despite the advantages of passive design principles, there are a number of challenges we must overcome to effectively use these technologies.

1.1.2 Challenges in passive system design

Passive embedded systems are able to achieve small form factors by using a combination of energy harvesting and passive communication techniques; using this combination of technologies gives rise to two primary challenges: power and communication throughput. As a consequence of limitations in existing passive communication infrastructure, there are additional challenges associated with implementing applications for these devices. We now explore each of these challenges in more detail.

Power. Passive embedded computing systems use two types of energy harvesting for power: *ambient* and *active*. Passive devices use ambient harvesting to gather energy from their surroundings, while active harvesting is used to transmit energy wirelessly from powered infrastructure. Each of these techniques has its own set of limitations.

Ambient harvesting can be difficult to use effectively because energy availability from harvesting sources varies both spatially and temporally. As passive devices move through their environment or stationary devices are subjected to time-varying harvesting dynamics, they need to properly manage their energy reserves to operate continuously; since passive embedded systems seek to minimize their form factor, and subsequently energy buffer size, this problem is particularly challenging.

Active harvesting is another way to power passive embedded systems; unlike ambient harvesting, these sources primarily suffer from spacial variations in harvesting rates. While close to the active transmission source, devices can harvest significant amounts of energy, but this rapidly diminishes with the square of the distance. Since existing infrastructure for active harvesting is extremely limited, harvesting opportunities for these devices remain scarce and only feasible for niche applications.

Throughput. Passive embedded systems currently suffer from poor throughput as a result of the physical and protocol layers they use for communications. Passive devices depend on a reader for connectivity and have no way to communicate with each other; this has several implications.

Passive communication is achieved using a technique called RF backscatter, where a passive device modifies a signal sent by an active transmitter. Since the passive device selectively reflects this same signal back to the sender, communication throughput rapidly decays with the fourth power of distance. This means that devices at the edge of the effective communication range suffer from excessively poor throughput.

Since passive devices lack the signal processing capabilities to decode each others transmissions, they lack the ability to coordinate communication between themselves and a reader; as a consequence the reader must manage the channel, resulting in communication protocols for passive sensors that use centralized algorithms to manage the wireless medium. To quickly identify large populations of tags, readers generate slots, each to be used by different passive devices. Each device randomly chooses a slot for its own transmission; to minimize the probability of collision, the reader will typically over-provision slots. This protocol is simple to implement for passive devices, but only works well when each tag only needs to send a small amount of data due to the large fraction of reader generated slots that go unused.

Applications. Implementing applications for passive devices is particularly challenging because of their dependence on readers for communications and power. Much like challenges found in ambient harvesting, the availability of a reader's resources can vary both temporally and spatially. To understand both of these challenges we consider applications that stress each of these dimensions.

In applications where passive embedded devices remain in close proximity to a reader during harvesting and communication, we must consider cases where these readers operate from batteries and do not offer continuous power output for the passive device. In other words, to ensure that passive embedded devices get the power they need, we must consider the harvesting variability induced by the energy management policies adopted by mobile readers. In particular, NFC readers integrated in mobile phones exhibit harvesting dynamics that depend on user behavior as well as the operating system's energy management policy.

Applications harvesting power from mobile phones in particular, may be implemented in two fundamentally different ways. First, the passive device may opportunistically harvest energy emitted from the phone during normal use. In this scenario, such a device may be integrated as an accessory for the mobile phone that augments its capabilities. Second, the passive device may be intentionally brought into the vicinity of the mobile phone for an explicit interaction. This use case requires the mobile device to harvest all the energy it needs for the interaction, as well as the energy needed until the next interaction. Each of these application scenarios introduce unique challenges in embedded hardware and software design.

In short, existing technologies make implementing robust passive computing systems possible, however many challenges remain. The central theme of this thesis is to explore how we make these systems a reality by carefully considering their power and communication needs. In particular, we seek to answer the following questions:

- What are the design tradeoffs available for passive embedded computing systems that combine RF and ambient energy harvesting modalities?
- How can we achieve high throughput passive communications while remaining compatible with existing communications infrastructure?
- Can we implement robust applications for passive embedded systems that harvest energy opportunistically from an attached mobile phone?
- Can we implement energy neutral applications for passive devices that harvest all the energy they require from normal user interactions?

The small scale of passive devices, limited platform visibility for debugging and evaluation, and uncertainty in availability of harvested energy make these questions difficult to answer. By providing energy harvesting traces, simulation tools, software, and hardware architectures and implementations we provide a means to reliably solve these problems, improve *passive* system performance, and provide toolsets to other researchers working in the area.

1.2 Thesis Contributions

The challenges addressed in this thesis focus in particular on power and throughput enhancements to UHF RFID systems and applications implemented for HF RFID systems. However, the general principles applied towards solving these problems are applicable to a much broader class of *passive* systems. In each of these systems, we harvest energy and efficiently communicate data by employing a novel set of hardware and software techniques that address the difficulties inherent in passive embedded computing systems.

1.2.1 Contribution Summary

This thesis advocates for hardware and software techniques to provide new design insights, communication performance improvements, security enhancements, and novel user interactions with *passive* embedded computing systems that rely only on harvested power. The core thesis of this dissertation is that *passive embedded computing systems can provide significant utility to target applications by utilizing limited amounts of harvested energy.* To support this statement, we propose four significant contributions that provide new hardware and software techniques that utilize harvested energy to realize such *passive* systems:

- *Hybrid Energy Harvesting*: An end-to-end design exploration study that examines solar harvesting as a supplemental energy source for computational RFIDs. [23, 37]

- *Flit*: A burst transfer protocol that utilizes existing UHF RFID protocol primitives to provide high throughput, energy efficient communication between mobile RFID sensors and sparsely deployed readers. [39]

- *EnGarde*: A passive, programmable HF RFID embedded security peripheral that shields mobile phones from malicious Near Field Communications (NFC) transactions. [38]

- *Bistable Display Tags*: An energy neutral, bistable display device that receives updates and harvests energy via NFC from mobile phones. [28]

1.3 Thesis Outline

The following sections of this thesis are organized as follows. Chapter 2 provides background on energy harvesting and RFID systems to set the context of our work. The proposal starts in chapter 3, which describes how to use a combination of ambient and active energy harvesting modalities to achieve the continuous operation of UHF RFID-based passive embedded systems. Chapter 4 describes a UHF RFID protocol that enables high throughput bursts of communication between mobile passive devices and fixed readers. In chapter 5, we present the design of a passively powered HF RFID peripheral that protects a mobile phone from malicious NFC transactions. Chapter 6 presents the design and implementation of a passively powered bistable display device that harvests the energy needed for an update while receiving the data for the new display image. Finally, Chapter 7 concludes and describes potential future research directions that build on the work presented in this thesis.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter presents background material on energy harvesting techniques and RFID technologies to set the context for our contributions. More detailed related work sections are also provided in the remaining chapters.

2.1 Energy Harvesting

Energy harvesting has gained a considerable amount of exposure over the past decade as energy costs have surged and global warming has become an increasing concern. Computing systems that employ energy harvesting to mitigate these concerns can range in scale from the largest data centers to the smallest passive sensors. For each of these types of systems, appropriate harvesting techniques and energy management techniques should be applied.

Of particular interest to this thesis are minimal passive systems that use micro-energy harvesters and minimal energy buffers. A plethora of harvesting modalities are starting to approach the stage of maturity where they can be included in passive embedded systems [62]. One thing that differentiates these systems from those previously outlined, is that even a single misprediction in harvested energy can result in a power outage. There have been a variety of techniques proposed to solve this problem, including adaptive runtimes [15], operating system like abstractions [92], and checkpointing mechanisms [64].

We can also consider a set of applications where it makes sense to use energy harvesting for smaller-scale systems that seek to have long deployment lifetimes. These lifetimes are dictated by rechargeable battery longevity rather than by what can be obtained from a single battery charge; since energy varies from day-to-day, energy must be used carefully. There have been a variety of approaches used towards achieving these objectives. One way is to use an adaptive runtime that matches workload to locally predicted daily harvesting rates [80], other techniques have used various methods of adapting duty-cycling rates of energy hungry system components [45, 86], or techniques that use tiered-architectures that only activate power hungry system components when needed [10, 73].

One particular set of applications that could have large global impact on these sustainability issues are where large computing systems run exclusively from electricity generated from ambient harvesting. At the largest scale, entire data centers can be powered from green energy sources [1]. Harvesting energy at this scale allows the conventional architecture of a data center to remain largely intact, but load balancing will need to be used to match data center power consumption with harvesting rates. At smaller scales, one can imagine individual servers that operate "off-the-grid" and rely on a combination of energy harvesters combined with battery-based energy storage to smooth out harvesting outages. Many techniques have been proposed to ensure that these systems can schedule activities accordingly, including leveraging weather forecasts [72], duty-cycling [71], and virtualization [74].

2.2 Radio Frequency Identification

Radio frequency identification (RFID) technology is most often associated with large scale inventorying applications used in large volume supply chains. However, during the last few years, there has been increasing interest in using this technology for a range of applications including activity monitoring [16], mobile health [29], and sensing [87]. In the last couple years, HF RFID has even found its way into mobile phones under the trade name "near field communication"(NFC) [7] .

One area of research, studies conventional ultra-high frequency (UHF) RFID systems implementing the EPC Gen 2 standard [18]. One class of applications that uses this technology are RFID-based locationing systems including building-scale activity monitoring [89] and room-scale object localization [60].

Going beyond systems that are implemented using conventional tags and readers, there have been a number of systems that modify the reader side of the RFID protocol to make performance improvements to throughput. Software radios [13] are typically used to modify the Gen 2 protocol to explore a variety of improvements including throughput [93] and collision resolution [75].

RFID-like systems may also be implemented without the need for explicit readers. By adding some of the signal processing capabilities of readers to passive devices, peer-to-peer communications can be achieved between passive devices, removing the requirement of a powered reader as an intermediary. To eliminate this requirement, passive devices selectively absorb and reflect transmissions from remote sources such as television and radio towers. Each device can detect another's absorption pattern as a change in its own harvesting voltage from the remote source [55]. Since this communication methodology requires that the two devices observe nearly identical versions of the remotely transmitted signal, it only works effectively if the two communicating devices are in close proximity to one another.

Another way to use RFIDs, is in more general computation and sensing workloads [14]. This class of device is referred to as a computational RFID or CRFID and is perhaps best exemplified by the Intel WISP [78] which contains an RF front end, microcontroller, and several sensors. CRFIDs have been used in a variety of applications ranging from medical security applications [26] to daily activity monitoring of tagged individuals [16].

There has been a large body of work that focuses on high frequency (HF) RFID readers and tags [34]. One example of an HF RFID system, called the Proxmark, was implemented using the combination of a field programmable gate array (FPGA) and microcontroller and emulates both readers and tags; this tool was used to expose a security vulnerability in MiFare tags [27]. Another type of system that protects against attacks and provides privacy guarantees for tags in its vicinity is the RFID Guardian [66]. HF RFID has also been used to implement a variety of commercially available sensors [2].

CHAPTER 3

HYBRID ENERGY HARVESTING FOR PASSIVE EMBEDDED SYSTEMS

Passive embedded computing systems currently suffer from energy starvation while decoupled from powered infrastructure and poor communication throughput while at the edge of a reader's communication range. In this chapter, we propose the use of ambient solar harvesting to boost the effective communication range of a computational RFID (CRFID) and provide power when none is available from a reader.

3.1 Background and Motivation

An emerging model of RFID requires tags that go beyond mere identification, also carrying out sensing, computation, and storage duties. Such Computational RFIDs (CRFIDs) [63] come equipped with ultra low-power microcontrollers and a suite of low-power sensors such as those found in the Intel Wireless Sensing Platform (WISP) [78]. A CRFID resembles a sensor mote stripped of its battery and active radio, but augmented with an RFID front end for RF energy harvesting and backscatter communication.

Limitations. A CRFID cannot sustain operation for long outside of the effective harvesting range of a reader; this lack of autonomy necessitates carefully planned deployments of RFID readers relative to tags, making such networks expensive to deploy and maintain. A CRFID's lack of autonomy is exacerbated by the observation that its performance varies with distance from a reader. For example, our experiments show that an Intel WISP [78] within a few feet of an RFID reader receives sufficient energy to sample and transmit hundreds of times per *second*, whereas one that is near its maximum reliable distance (a few meters) may be able to sample and transmit only a few times per *minute*. This rapid performance degradation limits the efficacy of CRFIDs for applications requiring complex processing or frequent sensing.

Approach. Our work exploits supplemental energy harvested using a miniature solar panel attached to the Intel WISP that makes the WISP more autonomous and therefore more useful for mobile sensing and computation. Even a tiny amount of ambient harvesting enables a CRFID to better tolerate power interruptions. Our measurements show that the SolarWISP prototype is able to retain memory state in all but the most difficult lighting conditions that we encountered. Modest modifications to the WISP enable ambient energy harvesting and provide the platform with new capabilities, but there are many issues associated with batteryless harvesting.

3.1.1 Design Considerations

There are several design considerations for CRFIDs that leverage ambient energy harvesting. In this section, we present some of these considerations, as well as the key questions that motivate the approaches in this chapter.

Using limited energy. A central challenge in designing hybrid harvesting CRFIDs is dealing with the limited amount of harvested energy. CRFIDs have smaller footprints than motes, restricting harvesting units to only a few cubic centimeters in size. Micro harvesters of this size typically generate very little power. For example, our experiments show that solar harvesting provides power ranging from less than a μW to a few mW depending on the panel size and lighting conditions. Thus, the first question that we ask is:

Question 1: *What are the lower limits of usable energy for current CRFIDs?*

Handling harvesting variability. Ambient energy harvesting exhibits spatial and temporal variability, sometimes over very short time scales. Harvesting output can be relatively predictable for a static outdoor deployment, for example, but unpredictable and bursty when CRFIDs are attached to mobile objects or persons constantly moving past occlusions and altering the panel's orientation. Because CRFIDs use capacitors with potentially low storage capacity, even a few seconds without any incoming energy can have disastrous consequences. The second question that we address is then:

Question 2: *What impact do dynamics in the ambient energy source have on hybrid harvesting CRFIDs?*

Making component choices. Capacitor size plays an important role in the design of a CRFID system. Larger capacitors are desirable because they can more effectively buffer excess harvested energy, allowing a CRFID to survive periods when there is no power from a light source or reader. Smaller capacitors have the advantage of short charge times. This leads to our third question:

Question 3: *What capacitor size maximizes CRFID performance for a given application?*

Choosing suitable workloads. While CRFIDs have unique advantages for some applications, they have no clear benefit for others. Past work has focused almost entirely on applications requiring low throughput communication or near-constant reader interactions [17, 91]. No work has yet been done with applications specifically suited to hybrid harvesting CRFIDs. The final question is then:

Question 4: *For what applications are hybrid harvesting CRFIDs most suitable?*

Motivated by the questions above, we aim to quantify the performance of hybrid harvesting CRFIDs and determine an appropriate set of environmental conditions and use cases where these devices prosper while passive CRFIDs (such as WISPs) and active battery-powered sensors (such as motes) may be less appropriate.

3.1.2 Goals & Contributions

Our research seeks to discover how to design hybrid energy harvesting CRFID platforms that match hardware choices with application workloads. Thus, our work aims to:

- Determine the boundary between useful and unworkable energy for hybrid CRFIDs.
- Explore the design tradeoffs in hardware components that influence CRFID performance.
- Understand the effectiveness of CRFIDs for an appropriate set of applications.

Our contributions that address these goals include:

Tools. We created two tools to assist in the evaluation of CRFIDs. Our software contribution is the trace-driven CRFID Crash Test Simulator (CCTS) that models the expected behavior of a CRFID's capacitor under varying workloads with solar harvesting. Our hardware contributions are the SolarWISP and FrankenWISP.

Measurements. A comprehensive measurement study is important in determining how miniature solar panels can be used to improve the performance of passive CRFIDs.

Applications. We present an end-to-end study of two applications that are enabled by ambient solar harvesting. *Path Reconstruction* is an application that needs to communicate frequently with readers while harvesting minimal amounts energy from dynamic, indoor lighting sources. *Greenhouse Monitoring* is a sensing application that relies on the predictable nature of static, outdoor deployments to buffer large amounts of energy in order to survive long harvesting outages.

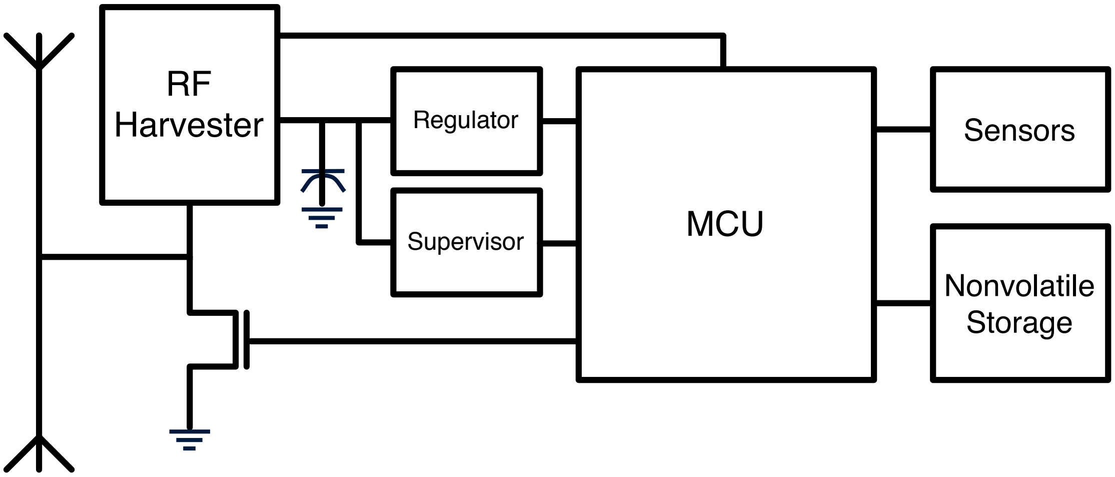

Figure 3.1. A CRFID strives to combine the versatility of a microcomputer with the energy efficiency of an RFID. With sensing capabilities, CRFIDs enable a host of new ubiquitous applications that go beyond mere identification.

3.2 A CRFID Primer

Traditional passive RFIDs use small amounts of harvested RF energy from reader infrastructure to report static identifiers. Computational RFIDs, or CRFIDs, use the same basic operating principles, but provide general-purpose computation. The core components of CRFIDs are as follows (see Figure 3.1 for a block-level illustration):

- **RF harvester:** CRFIDs harvest small amounts of RF energy provided by an RFID reader. The energy is rectified to produce DC voltage and boosted to an appropriate level by a charge pump The RF harvester also includes an analog comparator circuit to decode reader transmissions.

- **Backscatter circuit:** To transmit data, a CRFID toggles the state of a transistor to detune its antenna and reflect a different signal back to the reader. This method of transmission requires little energy from the tag when compared to active communication circuits.

- **Storage capacitor:** Energy harvested from a reader needs to be accumulated before any computation or sensing can begin. A CRFID uses a small storage capacitor for this purpose.

- **MCU:** A microcontroller is used to participate in the RFID protocol, as well as to perform arbitrary computation. Ultra low-power MCUs are limited in capability but provide fine-grained duty-cycling options with low power state transition costs.

- **Supervisor:** The microcontroller does not know the state of its capacitor, thus additional hardware is required for energy awareness. A supervisor circuit generates an interrupt or releases the MCU from a reset state after climbing above a preset voltage threshold.

- **Regulator:** The voltage stored on a CRFID's capacitor is highly variable. A regulation circuit is used to provide a stable supply voltage for the MCU.

- **Sensors:** Sensors give a CRFID awareness of the world around it, unlike a traditional RFID. Sensors that are considered low power for other devices can dominate the power budget of a CRFID.

- **Nonvolatile storage:** Frequent power failure is common for CRFIDs, making nonvolatile storage a necessity. MCUs typically contain a small amount of onboard storage, but this may be augmented with off-chip storage.

3.3 A Trace-Driven CRFID Simulator

We created the CRFID Crash Test Simulator (CCTS) in order to more fully explore the space of possible parameters and applications for CRFIDs — particularly the points of failure. It is difficult to profile a platform with power production and consumption values as small as those of CRFIDs. The difference of a microwatt is significant in many cases, leaving little room for error. Additionally, solar-powered devices that use capacitors for energy storage require the modeling of non-linear solar panel and capacitor outputs. Some platforms also use voltage regulators, which are yet another non-linear component to model. With these modeling difficulties in mind, the goal for CCTS is to track the charge and discharge of the storage capacitor *closely enough to inform platform provisioning decisions.*

CCTS requires a number of parameters to adequately describe the platform and application to evaluate:

- A recorded or generated illuminance trace
- Maximum and minimum VI curve approximations
- Capacitor size and initial voltage
- Application power consumption characteristics

With the exception of a recorded illuminance trace, each of these inputs can be found in a datasheet or measured empirically with little more than a multimeter.

Internally, CCTS models the platform and mobility in the following logical order:

Illuminance trace. CCTS illuminance traces are time-stamped lists of lux readings used to estimate solar panel output. The simulator looks up the most recent illuminance measurement as time progresses and provides this as input to the simulated solar panel.

A potential limitation with the use of illuminance traces as input is that illuminance is weighted toward portions of the spectrum perceptible by humans using the luminosity function. Irradiance, measured in W/m^2, would be the ideal metric for light intensity but we did not have ready access to any compact device capable of measuring irradiance. In order to keep conversions from illuminance to power output from being overly optimistic, we make the assumption that there is very little light available outside of humans' perceptible range. This assumption makes conversions from sources such as fluorescent bulbs the most accurate, while underestimating the power available from sources with greater bandwidth, such as incandescent bulbs or the sun. In practice, this inaccuracy may not have much of an impact because the efficiency of a given solar cell also varies with wavelength and it is impractical to model this efficiency curve for a large subset of those available.

Harvesting unit. The harvester is characterized using maximum and minimum functions to approximate the solar panel's family of VI curves. Given an illuminance value and voltage, CCTS interpolates between the two curves and estimates a current output value. We chose this approach because of the relationship between a solar panel's voltage and current output. As the load resistance of the platform increases, the solar panel's voltage output approaches its maximum and the current output approaches its minimum. A different VI curve exists for each light level that the panel could possibly observe and the curves do not simply scale with incident radiation. Their shapes change as well.

Storage capacitor. The energy output by the solar panel is next applied to the storage capacitor. The capacitor is the most difficult component to model accurately because of its dynamism and complex behavior. Changes in voltage on a capacitor depend on present voltage, effective resistance of the load, incoming energy, capacitor size, and internal leakage. None of these factors are modeled precisely in our discrete-time simulator. In order to approximate continuous behavior, the simulator re-calculates the voltage on the capacitor at a rate of 10 kHz using the standard equation for voltage at time i given a load R (which scales with present voltage) applied for t seconds:

$$V(i) \quad = \quad V_f + Ae^{-t/RC}$$

t is equal to Δi so that all changes in load are taken into account. The effective resistance of the load, R, is calculated using the voltage at time $i - \Delta i$ and the difference between incoming and outgoing current. The constant A is similarly recalculated based on conditions at time $i - \Delta i$. Internal capacitor leakage is difficult to even approximate, as it depends on factors including: temperature, present voltage, time stable at present voltage, storage capacity, and capacitor type (ceramic, electrolytic, etc.). CCTS does not yet model internal leakage.

Platform power states. Finally, CCTS removes energy from the storage capacitor based on the platform's consumption. Platform power state information is a time-stamped list of operations used in the application and their associated consumption data. At runtime, CCTS applies the appropriate current drain for the given operation. Modeling the WISP, for example, required that we measure EEPROM operations, computation, and two low-power modes. Unfortunately the WISP uses a linear voltage regulator, which changes current consumption dependent upon voltage. Rather than using a single consumption number, we chose to measure current consumption at a variety of voltages and to use a regression function to scale the simulated consumption with voltage.

3.3.0.1 Simulator Validation

In order to gauge the accuracy of CCTS, we compared simulated versus actual SolarWISP performance. As previously noted, CCTS does not attempt to achieve perfect accuracy. The goal for CCTS is to produce results accurate enough to inform platform provisioning decisions. The comparison was performed using a Franken-WISP to monitor a SolarWISP continuously while it sat stationary and the lights were turned on and off to vary incoming energy. While this trace was captured, a TelosB mote was positioned as close as possible without shading the WISP's solar panel and set to sample illuminance continuously. Despite being placed close together,

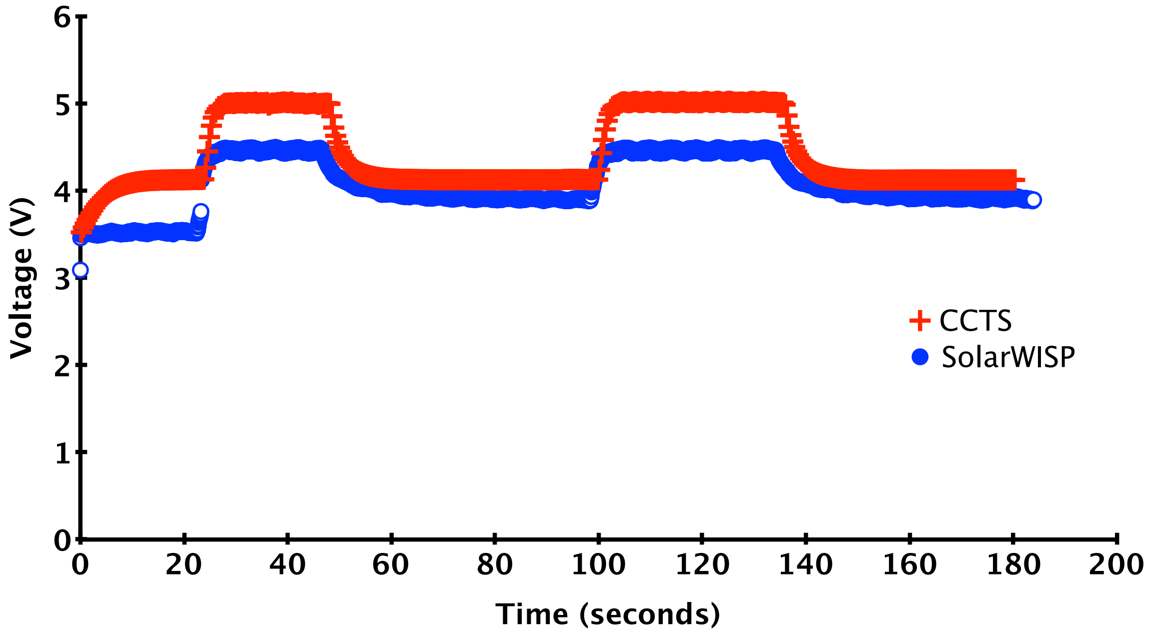

Figure 3.2. The two series in this plot are of a SolarWISP (as measured by the FrankenWISP) and a voltage trace produced by CCTS using a concurrently gathered illuminance trace. Notice that CCTS consistently overestimates the voltage, but the shape of the curve is similar. We believe that CCTS represents an upper-bound on empirical performance.

it is difficult to measure the illuminance at exactly the same angle and position as the solar panel. Other potential sources of error include the accuracy of the TelosB's illuminance sensor. We chose to use a 100 μF capacitor for this experiment.

As Figure 3.2 shows, CCTS consistently overestimates the voltage on the capacitor but stays within ~0.5 V of the empirical result. Because CCTS consistently overestimates the voltage, we believe that it is an upper-bound for an empirical deployment.

3.3.1 Real-World Illuminance Traces

To ensure that our platform benchmarks and simulation results translate into real-world performance, we gathered a set of illuminance traces from diverse environments, including mobile indoor and outdoor deployments from all times of day.

Environment	Trace length	Mean / Std. dev.
office1	2.5 hours	241.3 / 78.82 lux
office2	2.5 hours	34.71 / 24.25 lux
residential1	2.5 hours	49 / 62.91 lux
residential2	2.5 hours	124 / 161 lux
campus	15 minutes	3200 / 97.62 lux
yard	15 days	1076 / 1407 lux

Table 3.1. The length, mean, and standard deviation for each illuminance trace. The standard deviation varies widely based on mobility and diurnal cycles. See Section 3.3.1 for more detailed discussion.

These traces allow us to calculate the amount of solar energy available to a CRFID in each scenario. Each trace was captured using a TelosB's photodiode to take periodic illuminance measurements. For mobile traces, the TelosB was carried on the shoulder. To accurately capture mobility-induced dynamics, all mobile traces were collected at a sampling rate of 20 Hz, providing a maximum trace length of 2.5 hours given TelosB flash memory constraints. The static outdoor trace was collected using a sampling period of 30 seconds; this is suitable because of slower dynamics and allowed the capture of a trace 15 days in length. See Table 3.1 for a summary of the traces, with mean and standard deviation values for each illuminance trace. The traces fall into a few rough qualitative categories.

Indoor office traces. The first set of traces, *office1* and *office2*, represents two different office buildings. In each case, the TelosB was carried during daylight hours. These mobile scenarios illustrate how changes in panel orientation and location affect harvestable light.

Indoor residential. The second set of traces, *residential1* and *residential2*, gauges available light in a residential environment. The same experimental setup was used as in the *office* traces. This environment is of interest because there is little light available from incandescent bulbs but ample light available from windows.

Outdoor mobile. There is one mobile trace gathered outdoors — labeled as *campus*. The trace chronicles a 15-minute walk, which includes periods when the sensor was obstructed by occlusions such as trees and buildings. Of particular interest is the availability of much more intense light outdoors.

Outdoor static. The final trace, *yard*, uses a static sensor with a low sampling rate placed outdoors. It is intended to capture dynamics primarily induced by diurnal and weather variations. To assist in correlating light variations with weather patterns, we took note of conditions during the deployment. Our data represent days ranging from cloudy to sunny as well as several rainstorms.

3.4 Evaluation Summary

We now evaluate our hybrid CRFID prototype using platform benchmarks, real-world illuminance traces, and two application studies supplemented with CCTS simulation results. The first part of our evaluation explores several new capabilities enabled by ambient energy harvesting and quantifies them with platform benchmarks. Our measurements show that solar harvesting extends a WISP's effective communication range from 2 m to more than 7 m. We also show that the SolarWISP prototype can achieve perpetual with local timekeeping at an illuminance of 35 lux, while a solar harvesting mote prototype requires 200 lux to achieve perpetual operation. Next, we look at the implications of harvesting dynamics on platform performance with a simulation study. Our trace-driven simulation results show that the SolarWISP, even with the smallest capacitor size, can achieve nearly 95% uptime in RAM retention mode during a period of low light, while a standard WISP would fail completely in fewer than 15 seconds in the absence of an RFID reader. After examining the impact of harvesting dynamics, we provide a table to help platform designers predict response time and read rate. Finally, we present two case studies to validate our recommendations.

Conditions	Illuminance	Harvested power
Full shading	28 lux	6.6 μW
Partial shading	85 lux	35.9 μW
Diffuse	340 lux	62.5 μW
Direct	1300 lux	192.0 μW

Table 3.2. The amount of actual harvested power depends greatly on illuminance. A fully-shaded 11.4 cm^2 solar panel produces 29 times less power than the same panel under bright indoor lighting conditions.

3.4.1 Is Micro-Harvesting Sufficient for CRFIDs?

Harvesting energy from an ambient source, such as indoor lighting, is fundamentally different than intentionally delivering RF energy to a conventional RFID tag. Most important to consider when assessing the viability of solar harvesting for a CRFID are scenarios where a CRFID is disconnected from reader infrastructure.

Energy harvesting CRFIDs, such as the SolarWISP, must provide a reasonable level of performance when operating autonomously and must simultaneously be energy efficient enough to survive short-lived interruptions to harvestable energy. To assess whether the SolarWISP is viable for typical sensing and computational workloads, we present microbenchmarks that quantify the performance of critical system components. Additionally, we show that solar harvesting increases the effective communication rate and range beyond that of a conventional tag, thus improving performance for basic identification applications. Finally, we present a comparison demonstrating that the SolarWISP can achieve higher uptime and communication rates than a mote prototype that uses a battery for energy storage.

3.4.1.1 Micro-Energy Harvester Benchmarks

To evaluate the effectiveness of solar harvesting for CRFIDs, we first benchmarked the SolarWISP's small photovoltaic cell (see Table 3.2). The particular panel used for this measurement study is optimized for artificial, indoor light sources. The power

Power state	Power draw	Energy consumption
Active	467.1 μW	-
LPM3	4.5 μW	-
ADC read	-	0.244 μJ
EEPROM read	-	0.216 μJ
EEPROM write	-	0.125 μJ

Table 3.3. WISP power consumption benchmarks. LPM3 is the lowest power mode that allows the WISP to maintain a persistent clock. The WISP can achieve a long lifetime despite limited energy reserves by leveraging low-power states.

harvested varies widely with illuminance. In an indoor setting, the panel produces 6.6 μW of power while under full shading (28 lux) and 192.0 μW of power while under bright indoor light (1300 lux).

3.4.1.2 Computation, Sensing and Storage Benchmarks

Micro-power harvesting and capacitor-based energy storage together provide small amounts of buffered energy. This harvested energy is primarily used for three CRFID tasks: computation, sensing and storage. To evaluate the costs of these tasks, we benchmarked computation cost in terms of platform power consumption while in different power states and evaluated sensing and storage in terms of the energy required for an individual operation. A summary of these benchmarks appears in Table 3.3. The WISP consumes 4.5 μW in the lowest power mode that allows internal clocks (LPM3). This power state is sufficiently low to achieve perpetual operation while harvesting at a light level of 28 lux. The platform consumes 467 μW of power while in a fully active state. By combining low-power states with periods of active operation, the platform can achieve duty-cycles of 0.5% to 40.5% for illuminance values ranging from 28 to 1300 lux. Sensor readings require the WISP to sample the MCU's 12-bit ADC, with each read costing 0.244 μJ. Storage is accomplished by writing data over the MCU's I^2C bus to an off-chip EEPROM, with reads and writes costing 0.216 and 0.125 μJ respectively. These measurements agree with our claim that micro-power harvesting is sufficient for performing several computation, sensing, and storage operations on a CRFID.

3.4.1.3 Exploiting Hybrid-Harvesting for Improved Tag-Reader Interactions

CRFIDs must be able to communicate data to reader infrastructure in order to function effectively as autonomous sensors. We now consider computation and sensing in conjunction with communication and determine whether hybrid harvesting can improve performance while a CRFID interacts with a reader.

A CRFID generates responses to reader queries by modulating and reflecting a carrier waveform. Because the tags themselves do not generate the RF signal, wireless communication can occur at extremely low energy cost relative to an active radio circuit. Interestingly, the range at which a passive tag's RF circuit can correctly decode messages is longer than the range at which a tag harvests sufficient energy to generate a reply [23]. This imbalance suggests that a passive tag harvesting relatively continuous solar power has the potential for significantly improved communication ranges.

To investigate this performance enhancement, we measured the communication range of an RF harvesting WISP running the default firmware provided by Intel, as well as an identical SolarWISP. To find the maximum reliable communication range, we recorded the read rate for each device at a series of distances progressively farther from the reader. Figure 3.3 shows that read rates for a conventional WISP start at ∼150 reads/second, but quickly reduce to 0 at a distance of 2 m, with constructive multipath interference allowing intermittent communication at greater ranges. This is not the case for the SolarWISP, which sustains 23 reads/second at a distance of 7.1 m while observing an illuminance of ∼300 lux. According to our measurements,

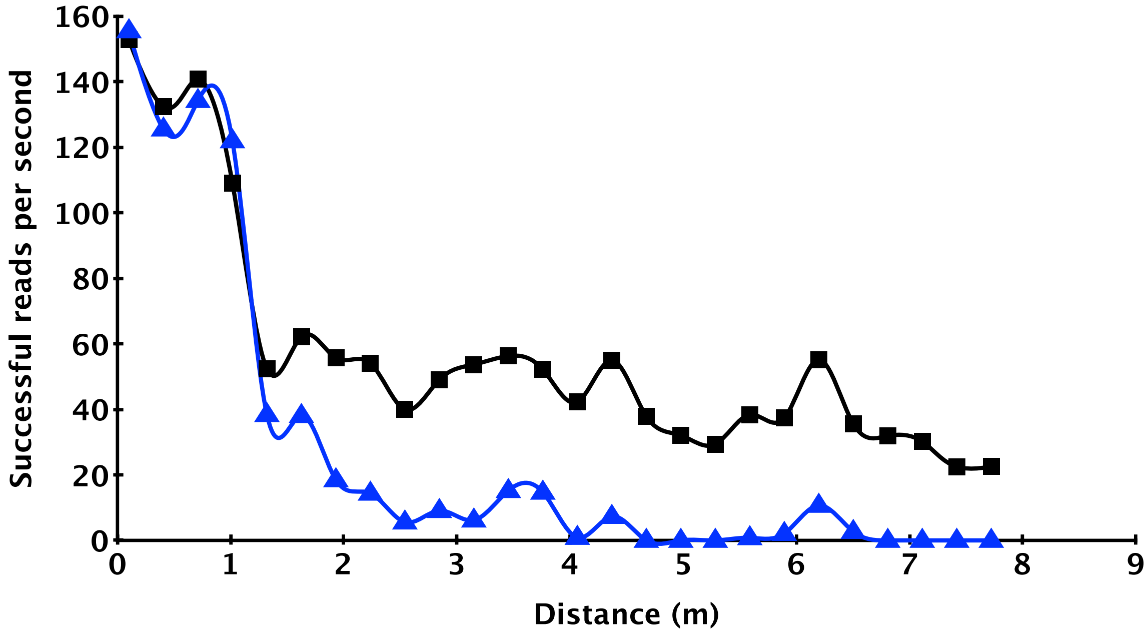

Figure 3.3. The number of successful tagID reads/sec indoors at a variety of distances with and without a 11.4 cm^2 solar panel. Note that the SolarWISP's energy harvesting gives it a consistent advantage at most ranges whereas the non-solar WISP encounters read rates of nearly zero beyond two meters.

the SolarWISP is able to sustain more than four times the read rate of a standard WISP at any range greater than 2.5 m.

The SolarWISP has more than *triple the effective communication range of a standard WISP*, providing the following benefits for potential applications:

1. Fewer readers required to cover a given area.

2. More communication opportunities for mobile CRFIDs.

3.4.2 Capacitor Sizing for CRFIDs

Capacitor size is an essential design parameter for CRFIDs and capacitors have two key characteristics that must be taken into account during selection.

1. **Exponential charge curve.** Capacitors reach a useful voltage level slowly. As discussed in Section 3.3, the charge curve is described by an exponential

function. This is in contrast to batteries, which reach nearly maximal voltage early in their charge cycles. Hardware solutions, such as boost converters, could potentially reduce this problem but have the potential to significantly increase quiescent current consumption. A low-input boost converter from TI, for example, consumes 5 μA of current while boosting voltages as low as 0.9 V [84]. This would more than double the timekeeping current consumption of a WISP.

2. **Low energy densities.** Capacitors, including supercapacitors, have lower energy densities than batteries. Due to size and weight limits, we only consider capacitors which will provide sufficient energy for at most *hours* of operation without incoming energy.

These characteristics create a tradeoff between responsiveness and survivability. Responsiveness is how quickly a capacitor can reach a suitable voltage to respond to reader contact events. Survivability is how long a capacitor can sustain operation without incoming energy. To demonstrate how capacitor size impacts responsiveness, we look at two benchmarks. First, we look at single capacitors' charge and discharge times. Next, we determine the performance of the SolarWISP in terms of the number of reads at different distances from a reader and for differently sized energy buffers.

Impact on tag-reader interactions. To demonstrate the effect that capacitor size has on aggregate read rate, we benchmarked tag performance in an environment similar to that which produced Figure 3.3. The SolarWISP was programmed with an application that ignores any reader queries when its voltage is below a threshold of 2 V and responds to the reader otherwise. Intuitively, a SolarWISP should generate a burst of reads when voltage surpassed this threshold, followed by a period of time spent recovering when it crossed below.

Figure 3.4 shows the average number of reads per second for several different capacitors at varying distances. The smallest capacitor size (10 μF) and the highest capacitor size (9400 μF) have the two worst read rates among all capacitor sizes. The intermediate sizes perform better with 100 μF performing the best of the lot.

This behavior results from the combination of two factors. At one end of the spectrum, larger capacitors take more time to charge and therefore spend long periods of time with voltage below 2 V, thereby missing many read opportunities. At the other end of the spectrum, a small capacitor size triggers pathological behavior wherein a SolarWISP attempts to wake prematurely, starving itself of energy. An intermediate capacitor size (100 μF) works best since it balances the two factors.

Figure 3.6 provides further insight into our results. The figure shows the average burst size and recovery time for all four capacitance values at a distance of 5 meters. The burst size monotonically increases with capacitor size because of the increasing amount of energy available to the platform during one burst interval. Response time generally increases with capacitance, as it takes more energy to reach the same voltage.

Impact on survivability. While larger capacitors are less responsive to rapid changes in energy, they provide greater survivability, The larger the capacitor, the longer a SolarWISP can survive. As shown in Table 3.3, the SolarWISP consumes only a few μW for basic timekeeping operation, hence even the energy in a reasonably sized supercapacitor can last a long time. In our experiments, we find that the smallest capacitor (10 μF) lasts only seconds whereas the large capacitor (9400μF) lasts $\sim$11.5 hours. Thus, for the SolarWISP, a modestly sized supercapacitor is sufficient as a power source for tens of hours while leveraging low-power states.

3.4.3 Coping with Harvesting Dynamics

While we have addressed the feasibility of autonomous operation, we have yet to account for the important issue of energy harvesting dynamics. Mobile devices that rely on harvested power for continuous operation are at the mercy of harvesting

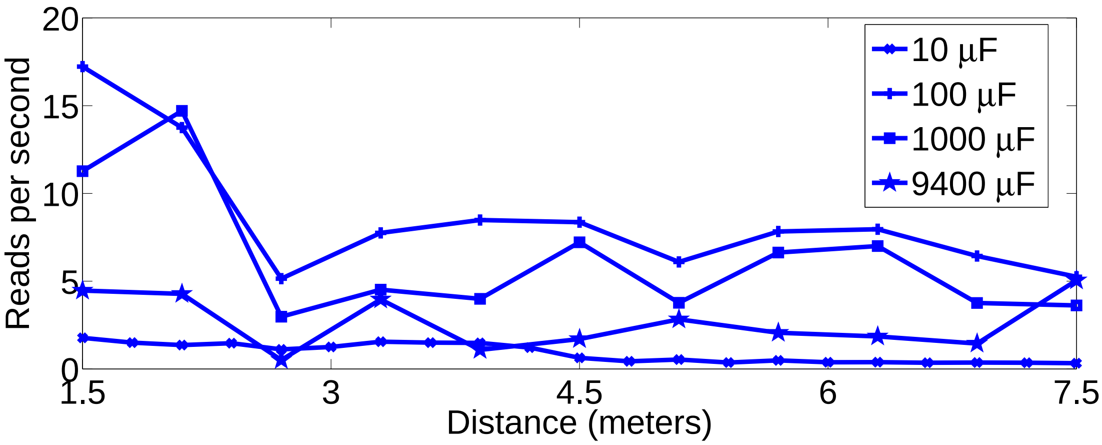

Figure 3.4. At distances greater than 3 meters, read rate becomes more dependent on harvestable solar energy than harvestable RF energy. A 100 μF capacitor provides the maximal read rate in this experiment.

dynamics while trying to achieve robustness. Energy harvested by the SolarWISP varies greatly with time and movement. Indoors, this can be attributed to lighting distribution and short-term changes in panel orientation. Outdoors, variations are due mainly to temporary occlusions and diurnal variations.

To illustrate the impact of harvesting dynamics on the operation of a CRFID, we performed a simulation study using a 30-minute portion of the *office2* trace with little available light and significant dynamics (see Figure 3.5). The trace excerpt stresses the ability of the SolarWISP to cope with dynamics and limited harvesting rates. The simulation also uses the WISP's standard 10 μF storage capacitor, which can only buffer enough energy for a several seconds of operation. As a result, harvesting dynamics are a major factor.

We considered several workloads for the SolarWISP. The lowest power state is RAM retention mode, which is only sufficient for refreshing SRAM between reader contact. A slightly higher power mode is RAM retention with periodic EEPROM writes every 5 minutes. Next, timekeeping requires waking up every ten seconds to bump a counter in order to maintain a persistent clock. We also simulated differ-

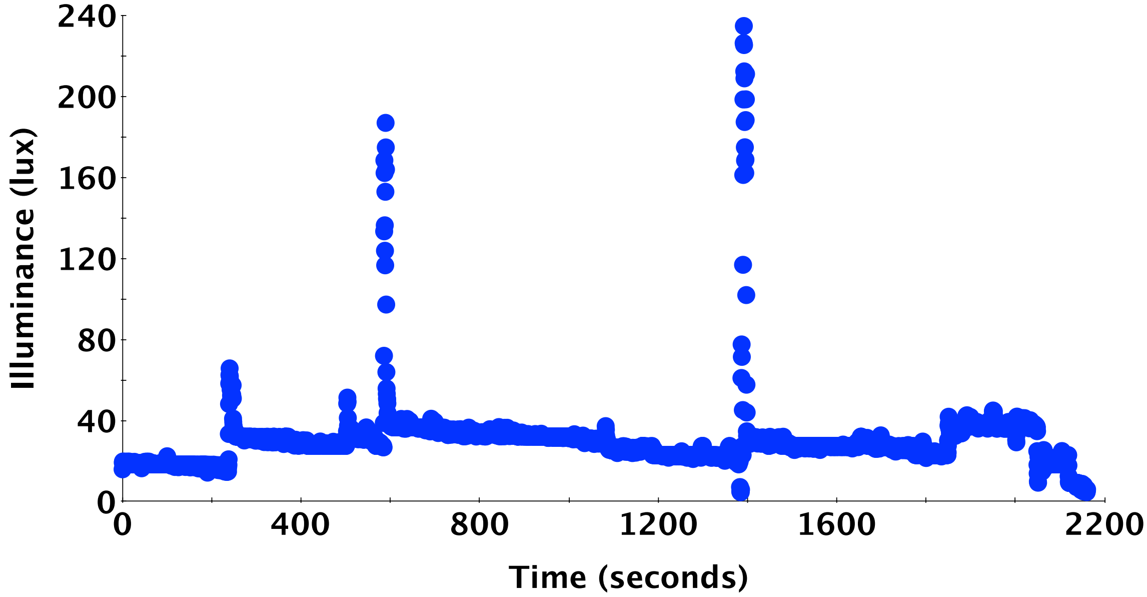

Figure 3.5. Time series plot of the *office2* trace excerpt used to illustrate dynamism tradeoffs. The mean illuminance is only 24 lux, but the maximum illuminance is more than 200 lux and the minimum is zero.

Workload	# Failures	% Uptime
RAM retention	1	99.19
RAM retention + record	1	99.19
Timekeeping	1	98.51
1% Duty-cycle	1	98.14
2% Duty-cycle	174	95.14
5% Duty-cycle	169	91.69

Table 3.4. The total number of outages predicted by CCTS for the *office2* excerpt. "Record" is the process of writing one byte to EEPROM. The number of outages in these CCTS results highlight how little excess energy the SolarWISP can store in a 10 μF capacitor.

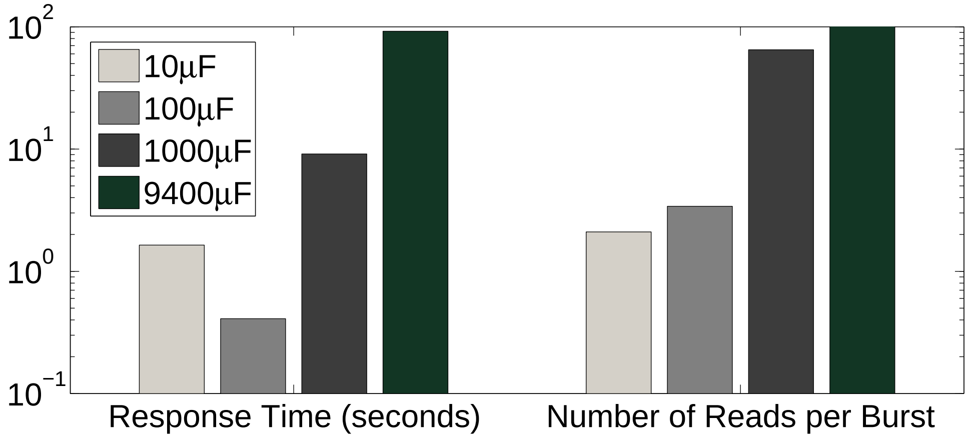

Figure 3.6. Progressively larger capacitors allow for increasingly large bursts of reads at the cost of responsiveness. We attribute the poor response time for the 10 μF capacitor to a pathology where the WISP attempts to wake prematurely, thus starving itself of energy.

ent duty-cycles for the SolarWISP, where the device periodically transitions from the RAM retention state to an active state. For a 1% duty-cycle, for example, the SolarWISP is up for 100 ms out of every 10 s. A summary of the CCTS results appears in Table 3.4.

The results show that the SolarWISP achieves an uptime of more than 90% in all cases, whereas a standard WISP would fail completely in fewer than 15 seconds away from an RFID reader. In this particular trace, writing data to EEPROM memory every five minutes has no impact on the WISP's time of death and saves most of the data recorded. This behavior is due to the 10 μF capacitor's high responsiveness. It typically achieves maximal voltage within the five-minute duration between EEPROM writes. The WISP is then able to perform the energy-expensive EEPROM writes without failing. The performance remains similar up to a duty-cycle of 2%, at which point performance degrades significantly. In terms of uptime, performance reduces only by about 3%, but the SolarWISP fails more far more frequently (174 failures in 30 minutes). Finally, a duty-cycle of 5% results in slightly fewer failures than 2% because of longer individual failures, but the overall uptime decreases as expected.

The simulation study has important implications for designing systems using SolarWISPs. Despite the fact that the trace represents bad lighting conditions with heavy dynamics, the SolarWISP achieves uptimes of close to 100% across a range of workloads. This result suggests that it is feasible to design systems that can exploit the SolarWISPs ability to do useful work between reader contact events. A designer may even be able to assume that the device rarely loses power or state between reader contacts. However, the results also show that it is difficult to guarantee that a SolarWISP will stay awake. For example, it is not possible for the device to survive the full 30 minutes — even if it remains in the lowest power mode at all times. Thus, it is critical to design under the assumption that, even with ambient harvesting, a small number of failures are inevitable and will need to be somehow mitigated.

While our study focused on a small capacitor size, dynamics are less of a concern with larger capacitor sizes. As mentioned in Section 3.4.2, larger capacitors can sustain a CRFID for more than 10 hours, but there is a price to be paid in responsiveness. For some applications a large capacitor will allow a CRFID to ignore harvesting dynamics, but for others, the rate of change in energy availability largely determines the minimal appropriate capacitor.

3.4.4 Application Studies

We now present case studies that validate our recommendations for balancing responsiveness and survivability. Each application stresses a different aspect of the *responsiveness* vs. *survivability* tradeoff. The first mobile application interacts with readers in frequent bursts of activity, making *responsiveness* critical. For the second application, a SolarWISP is statically deployed in an outdoor environment. The long harvesting outages caused by night make *survivability* the critical metric for success. We evaluate each application using empirical measurements of a deployment. For one application, we supplement the results with CCTS predictions to quantify the expected performance with a minimal capacitor size during a long deployment. For example, CCTS predicts that a SolarWISP can achieve 84% uptime in the worst case scenario that occurs in our traces.

3.4.4.1 Path Reconstruction

A mobile RFID tag or CRFID that periodically visits a set of networked readers generates data as a collection of individual read events. Unfortunately, these individual read events can be quite noisy. For example, Welbourne et al. report less than a 40% median success rate in detecting contact events using carefully placed readers and a variety of tag positions [89]. Heavy post processing was required in order to achieve reasonable inference accuracy for path reconstruction applications.

An attractive alternative to inferring tag behavior by examining reader logs is to directly report the desired application level measurement. Because CRFIDs can execute arbitrary instructions and store arbitrary state in memory, they are capable of reporting a result directly to a reader. In the case of path reconstruction, a single CRFID can report a timestamped list of all encountered readers. Providing mobile CRFIDs the ability to directly report pathing information is attractive because this removes the burden from the backend to infer paths based on large databases of tag visitations. Additionally, security may be added by reporting a cryptographically secure hash of path information instead of reporting a list of visitations in plain text.

Tag–reader interactions. To accurately reconstruct a path, at least one successful read must be recorded per reader contact event. The additional range and higher read rates enabled by ambient energy harvesting give the SolarWISP a higher chance of being read when passing by a reader.

Table 3.5 shows a comparison between the standard and SolarWISP in terms of successful reads per contact event. The experimental setup consisted of a mobile WISP moving past each deployed reader 30 times. Note that the SolarWISP can

Reader	Reads (WISP)	Reads (SolarWISP)	Improvement
0	25.9	31.8	1.23x
1	7.2	18.3	2.54x
2	7.7	31.5	4.09x
3	6.9	26.6	3.86x

Table 3.5. The average number of times a mobile tag can be read increases by as much as a factor of four when harvesting energy from ambient indoor lighting.

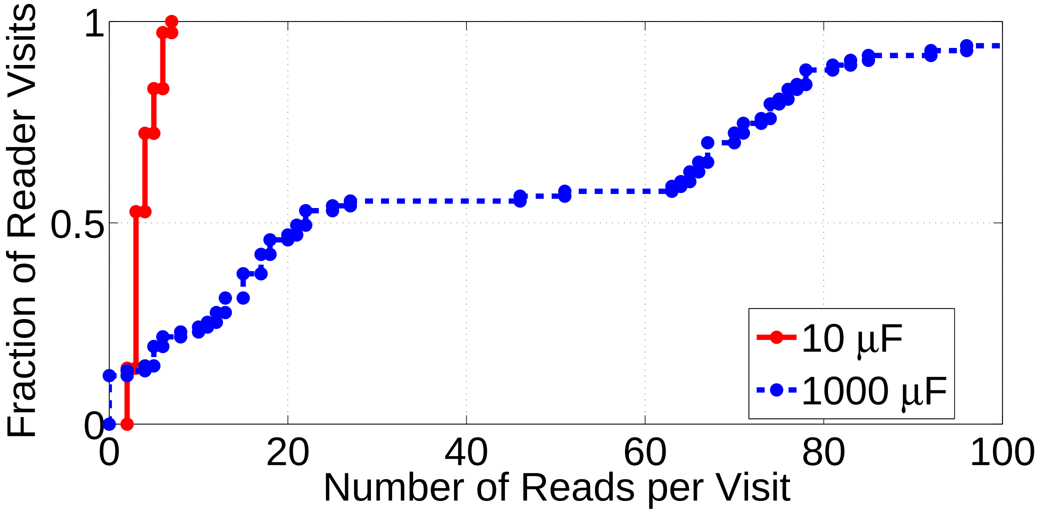

Figure 3.7. Large capacitors allow many reads per visit, but reduce responsiveness. A SolarWISP equipped with a 1000 μF capacitor misses 16% of reader visitations in this experiment.

achieve up to four times the number of reads as compared to a standard WISP. Also of significance is that the SolarWISP performs *consistently* better at each reader location, in spite of different multipath environments, due to its greater energy availability.

Responsiveness. The responsiveness of a tag to reader queries is key to the performance of path reconstruction. Choosing an excessively large capacitor will result in large recovery times after responding to a burst of queries. If these recovery times become too large, it is possible that reader visitations could be missed because of insufficient voltage on the SolarWISP's capacitor.

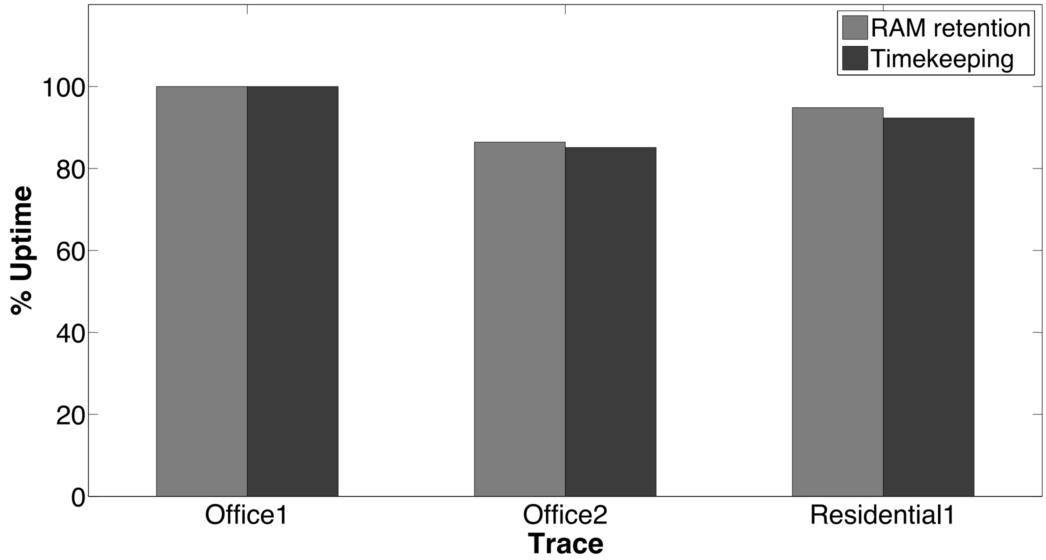

Figure 3.8. The percent uptime for a SolarWISP with two different workloads as predicted by CCTS. There are no outages predicted for the *office1* trace, but there are for the others. The percent difference in performance between the RAM retention and timekeeping workloads is ∼2%.

To test responsiveness, we moved a SolarWISP through the field of a reader ∼50 times with two different sized capacitors. Figure 3.7 shows how the number of reads per visitation contributes to the fraction of total reader visitation events. A SolarWISP equipped with the stock 10 μF capacitor was read 2–7 times per visitation. This small capacitor size results in few reads per visit, but no visitations are missed completely. The SolarWISP augmented with a 1 mF capacitor is read 0–558 times per visit. This capacitor size, however, takes longer to charge and causes the SolarWISP to miss 16% of reader visitations.

Based on this experiment, it is clear that dense reader deployments or high rates of mobility require careful consideration of capacitor size. An excessively large capacitor will cause a CRFID to miss reader visitations. A capacitor that is too small will result in few reads per visitation, but may respond more consistently.

Survivability. Survivability is also desirable for path reconstruction, but is not essential since readers could supply timestamps while reader IDs and timestamps may be potentially stored in EEPROM. Whether a CRFID keeps a local clock or relies on reader infrastructure, the power consumption of this application is minimal in between reader contact events. In the worst case, the application only needs to increment a timestamp once every 10 seconds and retain the state of SRAM, which contains past pathing data. These application characteristics suggest that a very small capacitor should be sufficient in many cases.

To gauge the performance of a large variety of capacitor sizes for disconnected operation in the path reconstruction application, we implemented the application in CCTS and tested it against the indoor illuminance traces discussed in Section 3.3.1. To test the application fully, we implemented both the version we measured empirically which records time locally, and the simple RAM retention version which assumes timestamps from the reader. Figure 3.8 shows the most interesting results from these simulation runs.

The only capacitor sizes to experience any outages were 10 μF and 100 μF, with the 10 μF performance being worse in all cases. Because these runs represent the worst performance, we present the percent uptimes for a 10 μF SolarWISP in Figure 3.8. For the *office1* simulation, there are no predicted outages for either workload. For the *residential1* simulation, the predicted percent uptime is between 92% and 95%. Finally, the predicted performance for the *office2* trace is significantly worse—with only 85% and 84% uptime for the timekeeping and RAM retention workloads respectively.

3.4.4.2 Greenhouse Monitoring

The temperature, relative humidity, and incident radiation observed in a greenhouse over time are of great interest to scientists studying biological responses to these factors. The sensing rate used for the experiment is 0.1 Hz. This sampling rate was suggested by a biologist who monitors greenhouse conditions.

From a sensing perspective, greenhouses are an environment of interest for solar-assisted CRFIDs because they can harvest light for long durations during the day and little to no artificial light at night. The sampling rates are also low enough that a mote-class device is greatly over-provisioned for the task. At the same time, greenhouse monitoring is an application for which a perpetual deployment is desirable, making batteryless computation a concrete maintenance advantage.

The SolarWISP contains an onboard temperature sensor, allowing the platform to achieve some of the functionality required by biologists. We implemented an application that records a 1-byte ADC sample once every 8 seconds and stores the result to a 1kB EEPROM. This limited amount of nonvolatile storage only provides sufficient space for 2.8 hours of temperature readings at this sampling rate, but this does not change the accuracy of the application power profile. A next generation platform could easily select a larger EEPROM while maintaining a similar form factor.

To survive the long power outages caused by night, we augment the SolarWISP with a supercapacitor. We also leverage the FrankenWISP to record the supercapacitor voltage once every 30 seconds to test survivability. Figure 3.9 shows voltage traces for a two-day application deployment using two different supercapacitor sizes. While both the 100 mF and 220 mF capacitor survive this deployment, they do decline to ∼2 V at night.

3.5 Related Work

RFIDs and RFID sensors. As the first example of a CRFID, Intel's WISP has defined the class of devices [78]. Emerging variants on the WISP platform include Duke's Blue Devil WISP [65] and the UW SoCWISP [61]. Yeager et al. propose the use of supercapacitors to extend the fleeting lifetime of the WISP given a full

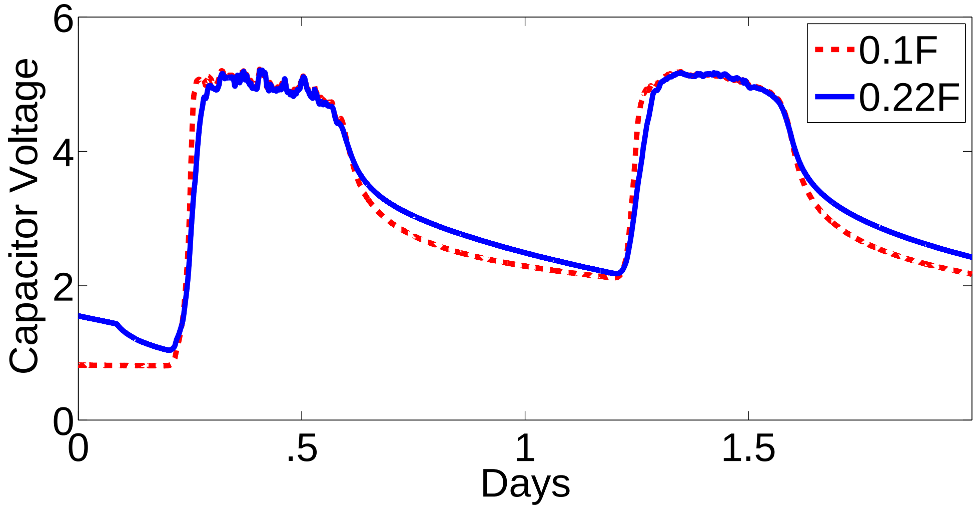

Figure 3.9. Supercapacitors of varying sizes store sufficient energy during daytime periods to prevent outages during night intervals. On an overcast snowy day, the SolarWISP still manages to store enough energy to survive.

charge [91]. Buettner et al. demonstrate activity inference enabled by the WISP's sensors [17]. This body of work is mostly concerned with designing RFID sensors, and does not consider how they can be augmented with ambient energy for autonomy.

Energy harvesting. There has been significant work on energy harvesting in sensor networks. Recent work [12, 44, 46, 70, 83, 86, 36] has explored scenarios in which nodes can harvest energy from their environment (e.g., from the sun) and use it to recharge their batteries. In the absence of such energy, nodes can then subsist on their replenished battery supply. There are also a growing number of solar-powered sensor network deployments, such as James reserve in Irvine, CA [43], Berkeley Angelo reserve, CA [83], CSIRO's Fleck sensor network in Australia [25], and the LUSTER system in Virginia [70]).

These systems are predominantly of the *macro-harvesting* type, wherein solar panel and energy buffer sizes are chosen to smooth out the short time scale variations in incoming energy. This greatly simplifies the design of a harvesting-aware sensor network, and enables a priori provisioning of resources. In many of these deployments, prior measurement studies of incident solar energy at the deployment location are used to select appropriate solar panel sizes for the sensor devices and to set system parameters such as the duty-cycling rate [90]. In contrast, our work tackles *micro-harvesting*, where the device, panels, and buffer are small. As a consequence, small-scale variations in energy conditions across seconds, minutes, or hours have a significant impact on the design of our system.

More broadly, the viability of various energy harvesting sources for computation and sensing has been considered many times in the past [62], but we focus exclusively on RFID harvesters that produce power at the μW scale. Sample et al. have demonstrated a WISP retrofitted with a directional TV antenna capable of harvesting energy from a TV transmitter over two miles away when positioned carefully [67]. The energy, however, was used to power a static load (small thermometer with LCD) rather than a WISP or other computation device.

Energy storage. At the core of our work is an understanding of the tradeoffs presented by capacitors for hybrid harvesting CRFIDs. The use of supercapacitors in sensor platforms is relatively common. For example, Prometheus is a harvesting-based sensor platform that integrates a conventional LiOn rechargeable battery and supercapacitors [44]. Capacitors are used in RFID systems as well. However, to the best of our knowledge, the tradeoff between survivability and responsiveness due to the use of capacitors has not been studied in any of the prior work.

Energy scheduling. Energy management for harvesting-based sensor networks has been studied in the past. Moser et al. [59] present optimal scheduling algorithms for harvesting networks that must meet deadlines; Vigorito et al. [86] present algorithms for adaptive duty-cycling based on harvested energy. In addition to adaptive duty-cycling, Kansal et al. [46] present a methodology for sizing energy buffers. While our work does not directly relate to such scheduling schemes, we believe that our work can inform such approaches. Many existing scheduling techniques are designed with implicit assumptions about the energy buffer and harvesting rate in mind. For example, [59] uses predictions of harvesting rates for its scheduling. Our work shows that for micro-harvesting, one needs to consider extremely small windows of time, and cannot assume that harvesting is smoothed by the energy buffer.

3.6 Status and Conclusion

A small amount of ambient energy can grant autonomy to CRFID sensors. The SolarWISP augments the traditional WISP with a 11.2 cm^2 solar panel to supplement the energy harvested by the RF charge pump. This small change increases effective communication range threefold and quadruples read rate. The SolarWISP requires only 35 lux to remain in perpetual timekeeping mode. Our trace-driven simulation results show that the SolarWISP, even with only a 10 μF capacitor, can achieve nearly 95% uptime in RAM retention mode during a period of low light, while a standard WISP would fail completely in less than 15 seconds in the absence of an RFID reader.

Our tools include a trace-driven simulator and an energy monitoring subsystem. The CRFID Crash Test Simulator estimates the survivability of a parameterized platform under a variety of lighting conditions. The FrankenWISP allows real-time monitoring of extremely resource-limited devices without greatly disturbing mobility or deployability. We hope that our tools and traces will help developers more easily evaluate the design space for future hybrid micro-energy harvesting CRFIDs.

CHAPTER 4

INCREASING PASSIVE COMMUNICATION THROUGHPUT WITH THE BURST PRIMITIVE

Despite additional energy from ambient harvesting, passive embedded computing systems still suffer from poor communication throughput while in the field of an RFID reader. The root cause of this poor performance stems from inefficiencies in the implementation of the defacto standard protocol used by UHF RFID devices. In this chapter, we describe the design, implementation, and evaluation of a burst protocol built on existing primitives that provides small populations of CRFIDs the ability to send high throughput bursts of data.

4.1 Introduction

In this chapter, we investigate how to efficiently utilize the energy buffer of an energy harvesting CRFID node for burst message exchange. We focus our attention on mobile CRFIDs [89], whose movements result in two communication states: connected and disconnected. While tags traverse their environment they perform a series of sensing and computation operations. As time elapses, these devices buffer some amount of data during disconnected operation; occasionally they encounter a reader, resulting in a variable connection interval during which a tag may offload buffered data.

Our goal is to optimize the bulk transfer of this buffered data from the CRFID sensor to an RFID reader. Because CRFID sensors have small energy buffers, it is imperative that communications maximize throughput while minimizing the amount of energy per unit data. This presents several challenges. First, commercial RFID readers follow the EPC Gen 2 protocol which is optimized for large numbers of tags that each transfer a small amount of data (tag identifier). This protocol is inefficient when considering sparsely deployed CRFID sensors that each potentially need to transfer large amounts of buffered data to a reader. EPC Gen 2 also makes duty-cycling for CRFIDs very difficult to implement because they must listen to a potentially large number of messages while waiting to transmit; this is unacceptable as CRFIDs treat energy as a precious commodity. While a complete re-design of the protocol stack is possible, this would mean that CRFIDs could not take advantage of existing commercial RFID readers, making them far less attractive for widespread use. Rather, we seek to support efficient bulk data transfer while still being compatible with commercially available EPC Gen 2 RFID readers. Second, when compared with other sensing platforms, CRFIDs have different hardware components, use different energy sources, use different energy buffers, and follow a different communication protocol. Thus, designing an energy-optimized bulk transfer protocol for RFID sensors requires an entirely new set of bandwidth and energy-optimization mechanisms. Third, CRFIDs present different usage scenarios since they are largely deployed indoors, and often on mobile objects or people. Thus, any data transfer protocol should operate effectively under scenarios where there are short contact durations with readers, and considerable changes in link characteristics during mobility.

Our protocol, Flit, provides a fast and efficient alternative to the existing EPC Gen 2 protocol for bulk data transfer from sensors, while still remaining compatible with existing RFID readers. Flit makes three fundamental changes to the protocol stack. First, it enables each sensor to transfer data in a burst by responding to all slots in a query round rather than just its assigned slot. This design choice improves goodput and energy-efficiency by reducing wasted slots, and takes advantage of extended query rounds with less control overhead. Second, Flit coordinates across sensors by using explicit burst notifiers that are echoed by RFID readers, rather than devices randomly picking a slot in which to transmit. This approach serializes burst transfers across nodes, thereby allowing greater goodput while reducing potential for collisions. Third, Flit improves energy-efficiency by duty-cycling the RFID sensor when another CRFID is in the middle of a burst. This avoids wasted energy due to overhearing of reader messages during the burst, thereby enabling better use of a small buffer of stored energy on the CRFID sensor.

Our results show that:

- Flit achieves 60% greater goodput than EPC Gen 2 for a single tag at different distances from the reader, and for different mobility conditions. A breakdown shows that much of these gains are due to the use of larger query rounds, and avoiding wasted slots in the round.

- Flit achieves 4.5x more goodput than an EPC Gen 2 tag when three tags are transferring data concurrently, and 9.2x goodput when five tags are transferring simultaneously. In addition, Flit has considerably higher fairness than EPC Gen 2, which is skewed towards the node with highest SNR to the reader. This allows sensors to take advantage of shorter contact durations with readers.

- Duty-cycling in Flit achieves up 6.04x better average power efficiency than a non-duty cycled implementation. These power savings are acheived by minimizing energy consumpted from listening to other CRFIDs' transmissions.

4.2 An EPC Class 1 Gen 2 Primer

The Gen 2 protocol for RFID tags is designed to inventory large tag populations over a number of communication rounds. To realize this protocol, an RFID must traverse a simple state machine and respond appropriately to a set of reader commands. Throughout this discussion, refer to Figure 4.1 to understand how a sequence

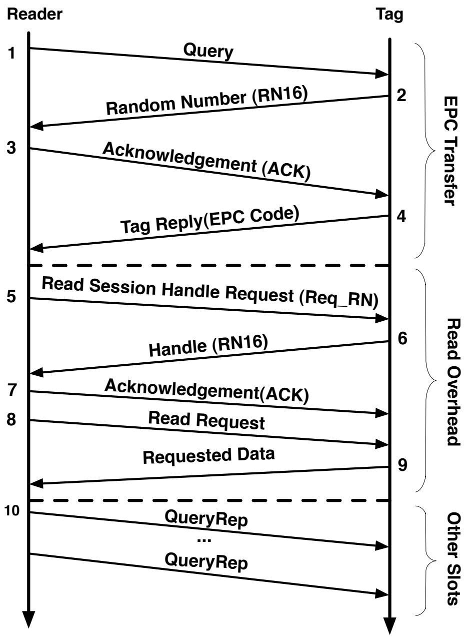

Figure 4.1. A series of message exchanges are required between a reader and tag to read the tag's EPC code or user memory.

of reader commands and tag responses are used to transmit data to a reader. The critical subset of EPC commands a CRFID must implement are:

Query, QueryRep, and QueryAdjust. A *Query* message (1) initiates a round of communication. This message specifies several round parameters. The most critical of these parameters is Q, which defines the number of slots in a round to be $2^Q - 1$ where $0 \leq Q \leq 15$. Tags generate a non-negative, random slot counter within the range specified by Q. The reader chooses Q such that collisions between tags are minimized.

Slots after a *Query* are occupied by *QueryRep* (10) and *QueryAdjust* messages. *QueryRep* messages indicate a successive slot for this round; *QueryAdjust* messages indicate a successive slot and additionally adjust the current Q value by +/- 1. After receiving either message, the tag decrements its slot counter; when the counter reaches 0, the tag proceeds.

Ack(RN16). To disambiguate tags in the event of collision, an *RN16* (2) message is used. The RN16 is a 16-bit value randomly generated by the tag. Upon decoding an RN16 from tag(s), the reader will echo one of the RN16s it received as an ACK

EPC. A tag knows it was chosen for communication if the received ACK matches the sent RN16; if this is not the case, the tag gives up on this round of communication to avoid further collision. After receiving its own RN16, the tag may backscatters its *EPC* (4) code to the reader. After sending its EPC, the tag will not respond to subsequent QueryReps or QueryAdjusts during this round of communication.

Req_RN. A reader that wants to further investigate a tag's state may send a *Req_RN* message (5). This message establishes a 16-bit handle to be use for subsequent communication. The tag echoes the handle back to the reader as an ACK (7).

Read. After establishing a session handle, the reader may send a *read* command (8) to request a segment of the tag's memory; after receiving a Read command, the tag responds with the requested data (9). The tag appends the session handle and a two byte CRC computed across the payload.

4.3 Limitations of Gen 2 for CRFIDs

In this section, we discuss several limitations the EPC Gen 2 protocol has on designing an energy-efficient bulk data transfer protocol that transfers data from a CRFID sensor to a reader. To understand these limits, we conducted a benchmark study that quantifies the timing and energy requirements of relevant Gen 2 messages for the Intel WISP 4.1; the results of this study are presented in Table 4.1. For each message type, we report the time required to finish sending or receiving a particular message in the *Active* column. The amount of time between a message and subsequent message is reported in the *Idle* column. We compute the energy for a particular operation by multiplying the platform power consumption by the sum of the *Idle* and *active* durations. For each message type, we also note whether WISP sends(TX) or receives(RX) the message, as the WISP consumes more power when receiving a message because it increases its clock frequency.

4.3.1 Singulation Inefficiency

The Gen 2 RFID protocol is designed around inventorying large numbers of tags that need only report a static identifier. It is therefore primarily focused on collision avoidance for a large number of passive tags. In this section, we show that EPC Gen 2 is inefficient for bulk transfer both in terms of throughput and energy-efficiency.

A key parameter that controls the efficiency of the EPC Gen 2 protocol is the window size, Q. The window size is a parameter that is set by a reader based on the tag population that it observes, as described in §4.2; during a round a number of slots are chosen such that the probability of collision between two tag responses is negligible. The EPC Gen 2 standard provides some general guidelines as opposed to a specific algorithm for how to set Q, so the implementation of the algorithm is vendor specific and is typically unavailable to the customer. In addition, there is often no way to control the Q values set by a reader since modern RFID readers are designed for ease of use and hide low-level protocol parameters from the operator. In particular, the Impinj Speedway reader we used offers no visbility into the chosen Q value; the resulting window size is completely decided by the reader's proprietary algorithm.

The efficiency of the EPC protocol depends on the value of Q set by the reader in each round. For example, if a reader picks $Q = 3$ and there is only one tag present, then there are 8 slots in this round, including the Query, one of which is utilized by the tag. In addition to the obvious throughput inefficiency, this is also inefficient energy-wise as a tag incurs the energy overhead of listening to the QueryRep or QueryAdj messages for the slots that it does not respond to.

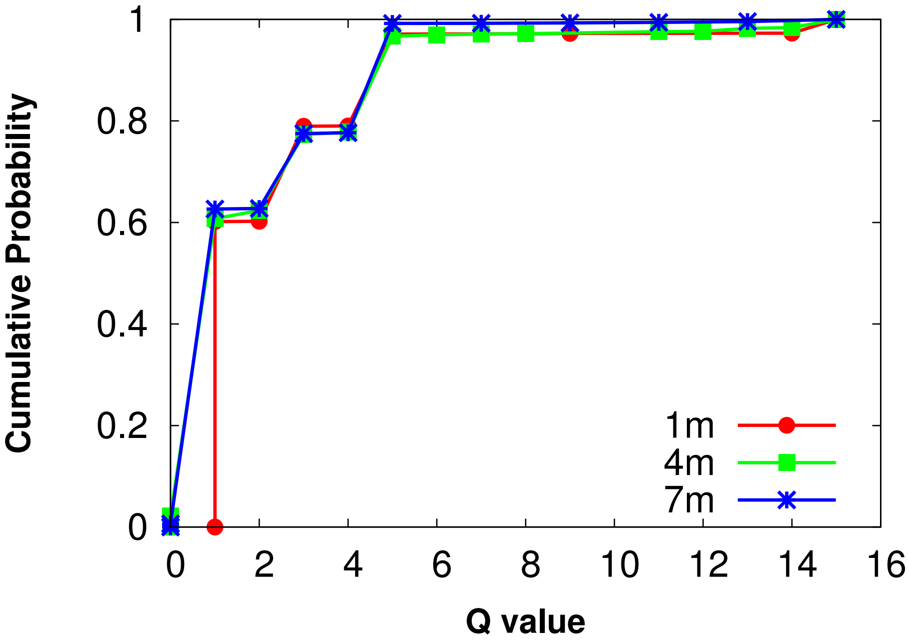

Figure 4.2. This plot shows what Q value a reader actually chooses when non-burst communication is used

To understand the practical inefficiencies of Gen 2 singulation, we looked at the round lengths selected by an Impinj Speedway reader when a single WISP tag is placed in front of it. While the reader does not provide an interface to obtain the chosen Q value, the WISP is programmable, therefore we were able to obtain the numbers by transmitting this information in place of the EPC code. Figure 4.2 shows that the reader typically chose a Q value between 1 and 6, with a mean of 2.5; this behavior held for distance up to 7 m. These Q values indicate that the number of slots in a round varies between 2 and 64 slots despite only a single tag being present, clearly a major source of inefficiency.

To further drive this point, we refer to Table 4.1. Based on a mean Q value that varies between 2 and 6 as in Figure 4.2, the extra communication slots result in degradation throughput that varies between 9.7 - 294.2% and energy consumption that increases by between 34.9 - 546.4% as compared to a single tag communication during a single slot round. It is also important to note that this is a lower bound

Operation	# bits	Time Active	Time Idle	Energy
Query(RX)	22	983 μs	52 μs	648 nJ
QueryRep(RX)	4	273 μs	50 μs	210 nJ
QueryAdj(RX)	9	415 μs	51 μs	319 nJ
Read(RX)	52	2100 μs	50 μs	1615 nJ
RN16(TX)	16	641 μs	2390 μs	422 nJ
Ack(RX)	18	660 μs	36 μs	508 nJ
Req_RN(RX)	40	1616 μs	51 μs	1241 nJ
EPC(TX)	128	2450 μs	2360 μs	1615 nJ
CRC16	–	452 μs	–	307 nJ

Table 4.1. A CRFID emulates Gen 2 in software leading to widely varying amounts of energy consumption depending on the command.

on the amount of energy required as CRFIDs will likely remain in an active state between received messages. These performance penalties change as a function of Q, which is in turn a function of the number of tags present.

4.3.2 Inefficiency of Read Messages

Gen 2 supports tag user memory operations in addition to simple EPC queries. Of particular interest is the Read command, which allows a reader to request a region of the tag's user memory. While at first glance, Read message seem to ideal for transmitting sensor data from a tag, we show that they are inefficient in terms of channel utilization and energy consumption.

Read messages are attractive because they support variable response lengths; a long read message could potentially overcome the singulation inefficiencies we previously highlighted, in addition to allowing CRFIDs to transmit large amounts of data to a reader. In theory, large Read messages are possible since Gen 2 specifies that an upper limit of 255 bytes on their size, but in practice, the size of read messages is limited by factors such as bit error rate, hardware limitations, and timing drift. For example, we found that the Impinj Speedway reader supports Read requests of lengths upto 60 bytes. For the Intel WISP, we found that read error rates sharply approached 100%, when 16 bytes of data were requested via Reads. Our hypothesis is that these practical limitations stem from three reasons: a) long messages are vulnerable to high bit-error rate (BER) at longer distances, particularly since the path loss on a backscatter link drops as the fourth power of distance [88], b) longer messages incur more timing drift, and RFID-scale devices often do not have real-time clocks to adjust for these, and c) large messages incur high memory overhead, which is limiting for RFID-scale devices. On the Intel WISP, both BER at higher distances and the timing drift were issues that made it difficult to get longer Read messages across to the reader.

Read messages also incur significant control overhead, which results in considerable throughput and energy inefficiency. As seen in Figure 4.1, a tag needs to be singulated prior to handling a read request, and depending on the Q value chosen for the round, may need to listen to several slots before it can set up a Read with the reader. This design clearly outlines the priorities of EPC Gen 2 — it is designed for obtaining identifiers from tags, and Reads are a second-class citizen that is intended to be used sparingly. The overhead is compounded by the fact that long Read messages are not practical, and is inefficient energy-wise since the tag is forced to listen to a long series of messages before its turn.

4.3.3 Lack of Duty-Cycling Support

Another major limitation of EPC Gen 2 is its lack of support for duty-cycling. While duty-cycling of a CRFID may seem unimportant for communication since the device receives power from the reader, this is not entirely true. The distance at which a CRFID can communicate with a reader is far more than the distance at which a tag can receive power from a reader. It is for this reason that passive tags have operating distances of a few feet from a reader, whereas a hybrid-powered CRFID

(RF + ambient harvesting) or a battery-powered active tag can have communication ranges of 50-70 feet [40]. At longer distances, a CRFID needs to duty-cycle and leverage low-power states since they are using precious reserves of stored energy or are operating on small amounts of ambient power.

Wireless MAC protocols designed with duty-cycling in mind typically use a number of mechanisms to synchronize senders and receivers, and buffer packets while waiting for synchronization to occur. For example, the 802.15.4 MAC layers uses preambles to synchronize sleeping senders and receivers, the 802.11 power save mode (PSM) relies on the access point buffering, and TDMA MACs have fixed slots, allowing a device to sleep for a fixed duration without the risk of missing messages. In contrast, EPC Gen 2 has non-deterministic arrival times of messages and variable round lengths. While Q determines the length of a round, this length is often not set at the beginning of a round. Instead, Q can be dynamically changed using QueryAdjust messages (based on estimated tag density), and a tag would not know the current value of Q if it misses a QueryAdjust message. In addition, slot lengths can be different since slots can terminate at different times due to timeouts after different steps of the protocol. The consequence of lack of duty-cycling support is that Gen 2 can cause CRFIDs to waste excessive energy on idle listening while waiting for their communication slot.

4.4 Flit Design

There are a number of factors to consider when designing a bulk data transfer mechanism for Gen 2. Such a mechanism must strive to: 1) maximize data transfer rates so that sensor tags can transfer their data quickly and efficiently to a reader during short contact events, 2) minimize power consumption so that a CRFID can maximize the amount of data transferred using its small energy buffer, and 3) inter-operate with standard commercial RFID readers, so that CRFIDs can leverage existing RFID reader infrastructure.

To realize these goals, we present the design of a burst protocol for CRFID sensors. First, we discuss the design tradeoffs in using an EPC Query versus the Read command as the data transfer primitive. Next, we demonstrate how sensors can achieve high levels of goodput using burst-mode data transfer that leverages unused slots in the EPC Gen 2 protocol. Third, we show a coordination mechanism that uses burst notifiers to avoid collisions among bursting tags. Fourth, we present a duty-cycling mechanism that minimizes the energy lost to idle listening. Finally, we discuss implications of the design choices that we make when there are a mix of sensor tags that are bursting and standard EPC Gen 2 tags that are only transmitting their identifier.

4.4.1 Read vs EPC for Burst Transfer

The first question in designing a burst data transfer protocol is which EPC Gen 2 message primitive to use as the building block for transferring data. Two options present themselves in terms of adapting the EPC Gen 2 protocol for bulk data transfer from the sensor to the reader. The first option is to use EPC Read command which allows a variable amount of data to be transmitted from a tag to reader, but has several inefficiencies as described above. The second is to use the *EPC* message, and send application data instead of the 12 byte static identifier within this message.

We first look at the energy efficiency of Reads vs EPC messages using the set of energy benchmarks in Table 4.1. The energy efficiency of the read command varies with the length of the data sent in response to the read request, while an EPC message is always 12 bytes. These benchmarks were captured using the Intel WISP 4.1 [78] (more details in §4.5).

From this breakdown, we compute the amount of energy consumed per byte of data transfer. Each Read command incurs energy overhead for steps 1–8 in Figure 4.1

that precede the read payload. The energy consumed for each EPC command varies a small degree based on whether it is in response to a Query, QueryRep or QueryAdjust since they have different sizes.

Suppose that EPC codes are used to transfer data to the RFID reader. An analysis based on a round with 4 slots (Q = 2) will result in 12 bytes of data arriving at the reader per slot, for a total of 48 bytes of data. This process takes 39.77 μs.

Now, suppose that a Read message is used to request data from a tag's user memory. Since we must now use the EPC data to singulate an individual tag for the subsequent Read command, a larger Read message is required to compensate for this additional overhead. When considering the same 4 slots as in the EPC based approach, the Read command incurs the previously computed time delay as overhead, as well as steps 5 - 8 from Figure 4.1. In order to match the throughput of a pure EPC-based approach under this scenario, a Read request of at least 116 bytes is required. This result indicates that Reads are a poor choice for high throughput tag to reader communications based on the reasons outlined in §4.3.2. A similar analysis of the energy required per byte of transmitted data shows that a Read with 143 byte request size is required to match the energy efficiency of the pure EPC approach. A similar conclusion applies if energy efficient tag to reader communication is needed.

One advantage of Gen 2 Reads, is that they have built in options for security. After a tag receives the handle message depicted in Figure 4.1, the tag may optionally be sent an *access* message that contains a password; reception of a correct password moves the tag into a logical state called *Secured*. This state may be utilized to protect portions of tag memory targeted by a subsequent Read command. However, since computational RFIDs can implement cryptography, they could instead encrypt data locally if a particular application requires it.

4.4.2 Burst-Mode EPC Transfer

Having selected the 12 byte EPC message as the building block for bulk transfer, we turn to the question of improving efficiency when several hundreds of bytes of data need to be transferred using this message primitive. If the Gen 2 protocol were followed, data transfer would need to be over several tens or hundreds of rounds, and a CRFID would receive only one slot in each round. As described in §4.3.1, this would be extremely inefficient due to poor choices of Q at the reader.

The central idea in burst transfer is to ignore Gen 2 semantics of rounds, and to treat the protocol simply as a sequence of unassigned request/response slots. Each of these slots can be initiated by a Query, QueryRep, or QueryAdjust, but the burst protocol does not treat them differently. Instead, a CRFID sensor assumes that every slot is available to it for burst transfer, and just transfers its data in a sequence of consecutive slots. Before discussing issues of coordination across multiple tags (§4.4.4), we look at the benefits that this offers to a single tag.

The key benefit of burst transfer from a single tag is that we are no longer limited by poor selection of Q by a reader (§4.3.1). In fact, we turn a drawback into an advantage. To obtain the full benefits of burst transfer, we want the reader to choose a large Q. As previously shown, a round is initiated by a Query message for the first slot, and the other $2^Q - 1$ slots are initiated by QueryReps. A few slots are initiated by QueryAdjust messages, whose purpose is to increment or decrement Q in the middle of a round. A subtle benefit of QueryReps and QueryAdjs, as opposed to Queries, is their brevity. Based on protocol specs, Query, QueryRep and QueryAdjust messages have lengths of 22, 4, and 9 bits respectively. As Q grows, the energy expended during a round of communication becomes dominated by round trips involving reps and adjusts.

The energy and throughput benefits of using longer Q are quantified in the following equations:

$$\mathrm{E}_{\mathrm{round}} = \mathrm{E}_{\mathrm{query}} + \left(2^{Q+A} - 1\right) \cdot \mathrm{E}_{\mathrm{rep}} + n \cdot \ \mathrm{E}_{\mathrm{adjust}} \tag{4.1}$$

$$\mathrm{Goodput}_{\mathrm{rnd}} = \frac{12}{\mathrm{T}_{\mathrm{query}}} + \frac{12 \cdot \left(2^{Q+A} - 1\right)}{\mathrm{T}_{\mathrm{rep}}} + \frac{12 \cdot n}{\mathrm{T}_{\mathrm{adj}}} \tag{4.2}$$

Equation 4.1 shows the total energy spent on a round of communication, which includes the energy spent on listening and replying to the first query slot($\mathrm{E}_{\mathrm{query}}$), the energy spent listening/replying to subsequent slots initiated by QueryReps (Q is the initial value assigned by the reader, and A indicates how it was adjusted during the round), and finally the energy spent on n QueryAdj messages that were sent during the round. Equation 4.2 shows the goodput counterpart, which takes into account the length of an EPC message (12 bytes), and the time for the three types of queries.

Using numbers from our microbenchmarks in Table 4.1, we see that a long round with $Q = 15$ can give about 10% benefit in both energy and goodput over a short round with $Q = 0$.

In summary, treating the Gen 2 protocol as a sequence of unassigned slots enables us to a) limit inefficiency due to empty slots caused by poor selection of Q, and b) improve efficiency by taking advantage of shorter slots initiated by QueryRep messages.

4.4.3 Coordination via Burst Notifier

Responding in every slot has a severe limitation: if multiple CRFIDs are present, they will suffer from collisions and see reduced energy efficiency and goodput instead of the improvements. An active radio system could solve this problem using control messages such as RTS/CTS or an overhearing-based approach such as CSMA to coordinate transfers between peer nodes. These approaches are not suitable for backscatter communication circuits because they are unable to decode messages transmitted by peers. An alternative would be for the reader to explicitly select a tag in the Query or QueryRep message, and all other tags that receive the message can ignore the slot. However, as mentioned earlier, QueryReps are only *4 bits* long, and leaves no room for such addressing. Besides there is the limitation that readers do not allow modifications of Query messages, making any such approach impractical. Thus, we ask the question: *How can CRFID sensors use the existing EPC protocol to efficiently coordinate bursts?*

A closer look at the Gen 2 Query/EPC exchange reveals that there is a two-way handshake being performed, which presents a solution to this problem. As shown in Step 3 of Figure 4.1, the reader echoes the RN16 of the RFID it chooses to occupy a given communication slot. Our strategy is to overload the RN16 to signify that a particular CRFID is currently bursting. We accomplish this by providing a special interpretation of a segment of reserved RN16s; we partition the space of RN16s as $0 < n < 2^{16}$, where n is the number of CRFID sensors deployed and values less than n are considered burst notifiers. The value of n is statically selected at compile time and is chosen based on the maximum number of CRFIDs envisioned for a particular application. A sensor that wishes to send a burst of EPCs will use its statically selected burst notifier chosen from the available pool, instead of a random value. Note that the sensor selects a notifier just once for an entire burst, rather than once per slot as is done by a standard tag.

The RN16 burst notifier is used in the following way: prior to initiating a burst, a CRFID sensor listens to the channel after decoding a query, rep, or adjust message. If the sensor observes an Ack within the range of burst RN16s, it should remain silent to avoid colliding with an ongoing burst. If the slot contains an RN16 outside of this range, it can go ahead and start a burst transfer after the current slot using its own burst notifier, as non-burst EPC messages occupy only one slot.

It is, of course, possible that another CRFID sensor is in the middle of its burst and either the reader might have missed the burst notifier or the listening sensor may not have received the notifier echoed by the reader due to channel error. Both cases would lead the listening sensor to conclude that the channel is free and start to burst, resulting in collisions at the reader. A collision at the reader typically results in the reader receiving the stronger signal among the colliding tags due to capture effect. The reader echoes the burst notifier that it receives, which results in only the sensor with stronger signal continuing to burst. While a collision could also result in neither signal being received by the reader, we handle this case by assigning a random back-off interval after hearing no Ack when one is expected.

To prevent the sensor from holding the channel indefinitely, the burst will terminate after a small, fixed amount of time that is large enough to amortize coordination overheads, but small enough to allow mobile tags with limited communication opportunities a chance to offload a burst of data to the reader.

4.4.4 Duty-cycled Coordination

Burst transfer is a natural fit for CRFID duty-cycling for two reasons: a) transfer is in large chunks of consecutive slots, enabling other nodes to sleep for longer durations and re-charge while waiting for a burst to end, and b) inefficiencies incurred due to duty-cycling such as wasted slots because a tag is asleep or wasted energy for listening because it is awake too early can be amortized over the longer sleep durations.

While bursts are convenient for duty-cycling, the lack of enough bits in the Query/QueryRep messages impacts duty-cycling efficiency as well. If a waiting tag knew precisely how much longer a burst from another tag would last, it could sleep for exactly that duration. However, this information is unavailable since a sensor tag relies on the burst notifier from the reader to detect a burst, which provides no information on the time remaining for the burst. Thus, a tag needs to periodically wakeup to check the channel and detect if a burst has ended. Thus, a key challenge for a duty-cycling strategy is to efficiently find the end of a burst so that sensors can capture the channel from another sensor between bursts and react quickly to mobility dynamics while avoiding most of the energy wastage caused by overhearing.

Thus, there are two questions that remain regarding how to duty-cycle an CRFID sensor: a) the amount of time a sensor should probe the channel and b) how much time a sensor should sleep. Since tag-to-tag communication is impossible, we do not consider adaptive policies for determining these intervals since the new probe and sleep intervals would need to be shared with all tags. We now describe how these intervals should be statically selected.

Probe Duration. The probe duration should be long enough such that a tag can detect whether another tag is continuing to burst. This duration is equivalent to a single slot in a query round. A sensor tag wakes up, listens to the first Query, QueryRep, or QueryAdjust slot, and sees whether a burst notifier is echoed by the reader during this slot. If so, it concludes that another tag is bursting and goes to sleep; if not, it concludes that it can initiate its own burst and starts transmission in the next slot. In the middle of a burst, if a tag detects that the reader has echoed a different burst notifier it concludes that another CRFID sensor is bursting and goes to sleep to save energy.

A potential issue here is that the duration of a slot can vary because a) Query, QueryRep, and QueryAdjust messages are of different lengths, b) a slot can terminate at different times depending on whether the reader times out after the RN16, Ack or EPC steps in its state machine and c) mobility can introduce additional dynamics. To address this, we look at the probe duration empirically by measuring the inter-arrival time of Query, QueryRep or QueryAdjust messages for a continuous exchange between an Intel WISP programmed with the EPC Gen 2 protocol and a reader. We look at this distribution for different distances from the reader, as well as for different mobility patterns. Figure 4.3 shows the CDF of the inter-message duration. The results show that the inter-query intervals do not depend significantly on the

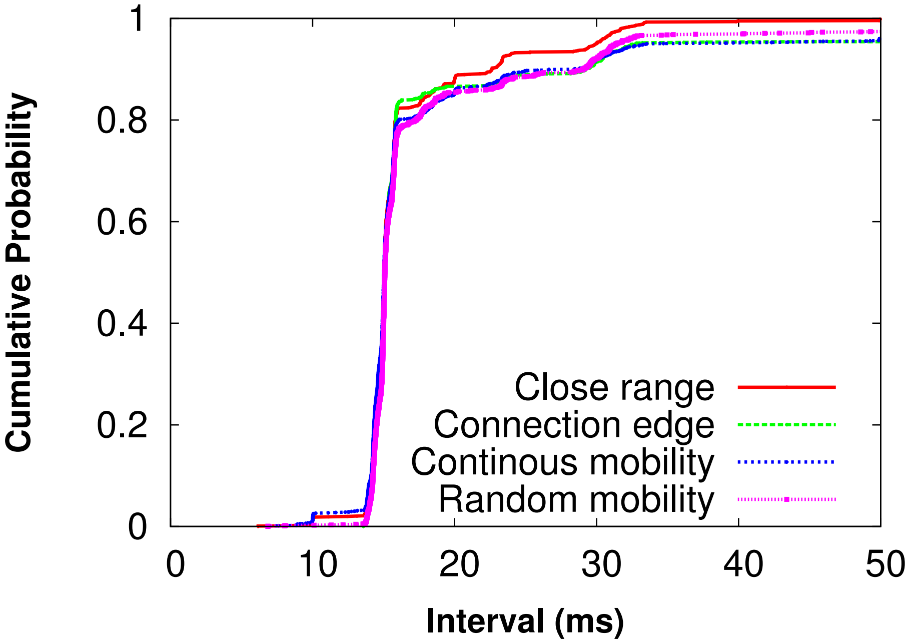

Figure 4.3. Most Query, QueryRep, and QueryAdj messages have inter-arrival times of less than 20 ms.

distance, and are impacted a little ,but not a lot, by mobility. The knee of the curves is in the 15-20 ms range, thus we select 20 ms as our probe duration; this probe duration is short and provides a reasonable guarantee that a query will be heard by the WISP during the period it is awake.

Sleep Interval. There are several considerations in determining the sleep interval. First, the sleep interval must be long enough that we get significant energy benefits from duty-cycling. Second, it should be short enough that a tag can quickly react to mobility-induced channel dynamics. Third, it should have sufficient randomization so that we avoid unwanted synchronization issues that can result from multiple tags waking up at the same time.

In terms of the energy consumption, we want a duty-cycle of lower than 10%, hence the sleep duration should be at least 200 ms when the probe duration is 20 ms. To understand the typical contact duration at walking speed, we use 1 reader in a cor-

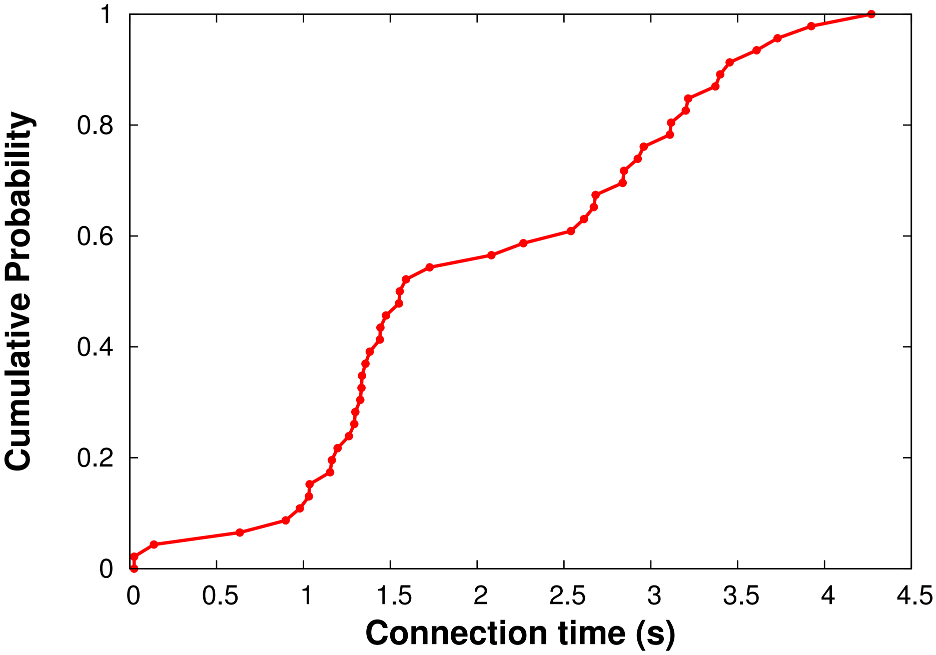

Figure 4.4. For human-scale mobility rates, the connection time between a tag and reader typically lasts several seconds.

ridor, and walk in circles around it. We found that a typical contact duration is a few seconds in duration (see Figure 4.4), hence the sleep duration should be much smaller than this number. To prevent synchronization issues, the tag can randomize the sleep time within a tolerable range that provides a desired amount of energy savings while maintaining reactivity to expected mobility patterns for a given deployment.

4.4.5 Coexistence with Non-burst Tags

While our discussion thus far has assumed the tag population comprises solely of sensor tags that have to transfer data in a burst, we now look at the implications when a mix of sensor tags and standard EPC Gen 2 tags are communicating with the same reader infrastructure. Not surprisingly, the net effect is that standard EPC Gen 2 tags incur more delay in communicating with a reader infrastructure. However, there are mitigating factors that can enable better coordination across tags.

The burst mode transfer mechanism that fills up all slots of a round impacts standard Gen 2 tags in two ways. First, a standard tag which picks a slot within a round will collide with a burst tag, resulting in loss of one of the messages. In practice, we find that because of reader sensitivity, the reader gets one of the CRFID's messages with high probability (due to capture effect), hence it is still possible that the standard tag gets its data through. However, if the burst tag has the stronger signal, the standard tag suffers. Second, the burst transfer approach results in large Q values, which makes a round long; since standard tags only respond in one slot within a round, this makes their response slow. This effect is mitigated by the fact that we limit bursts to a relatively short duration of time (1sec in our implementation), after which a sensor goes to sleep for a short window of time before bursting again. The duration between bursts is sufficient for a few short communication rounds, enabling standard tags to get their data through.

The use of burst notifier facilitates co-ordination across CRFIDs, however, passive tags are free to choose any value from 0 to 2^{16} as its RN16, so it is possible that a passive tag could choose an RN16 that conflicts with a burst notifier. However, we choose a small part of the space for burst notifiers since we expect the number of sensors in the vicinity of a reader to be in the tens (equivalent to the number of objects in the vicinity of a reader) as opposed to thousands. Thus, the probability of collision is low. In addition, the RN16s are chosen anew in each round, hence a standard tag would likely choose a non-colliding RN16 in the next round.

4.5 Implementation

Our bulk transmission protocol is well suited for implementation on CRFID sensors because it is a modification of the EPC Gen 2 protocol they already support. CRFID sensors that want to implement the protocol need only modify their state machine to properly handle burst-mode transmission, burst notifers, and duty cycle

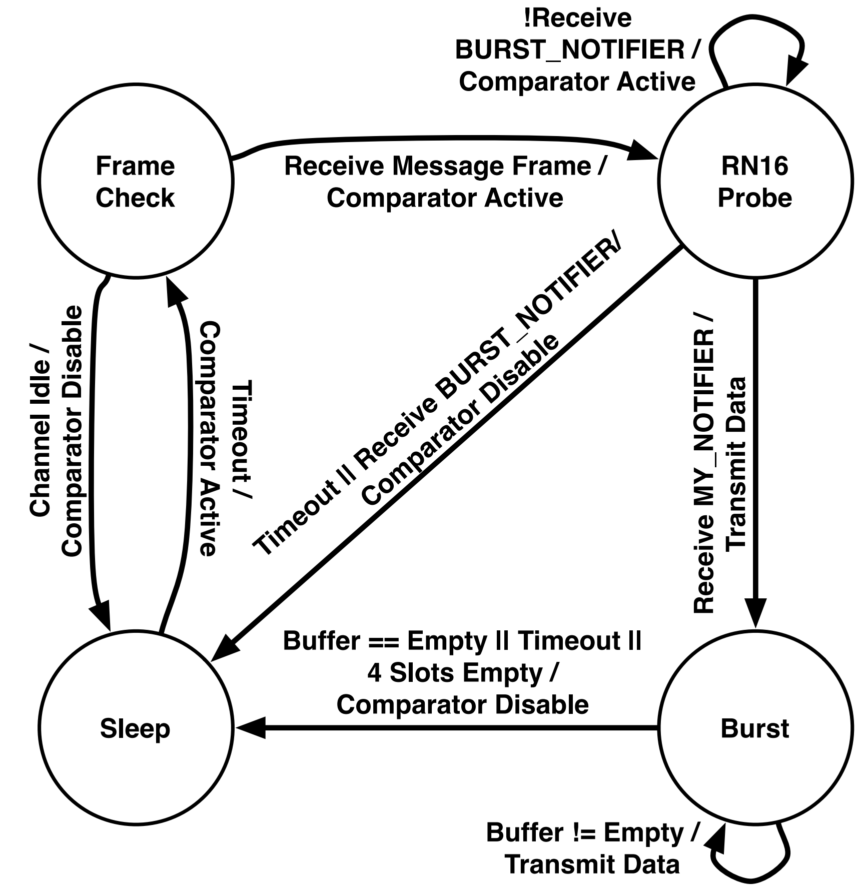

Figure 4.5. A coordinated bursting protocol for CRFID sensors can be implemented as a state machine. The protocol uses sleep states to both avoid contention and achieve energy efficiency.

appropriately in response to received message frames from a reader. In this section, we show the state machine we used to implement our Bulk Transmission Protocol for the Intel WISP. Next, we describe how state machine parameters can be defined based on results from channel measurements and mobility experiments.

Figure 4.5 shows the state machine used to implement our Bulk Transmission Protocol for the Intel WISP. Sensors that have data to send initialize a timer interrupt and begin operation in the *Sleep* state. After this timer expires, the sensor activates its comparator and enters state *Frame Check*; after initializing another timeout value, the microcontroller enters a low-power mode, only waking up to handle an incoming message frame. Upon receiving a valid message frame, the sensor will enter state *RN16 Probe*; else, if the sensor does not hear a valid delimiter from a reader, it goes back to state *Sleep* after timing out. While in state *RN16 Probe*, the sensor initializes its timer with another timeout value; after hearing at least one empty frame from the reader, in which the sensor does not hear another sensor's BURST_NOTIFIER, it will send its own BURST_NOTIFIER in response to a Query, QueryRep, or QueryAdjust message. If the sensor hears its own BURST_NOTIFIER, it enters state *Burst*; if another sensor's BURST_NOTIFIER is heard, instead of an empty slot or if the timeout value is reached, the sensor enters state *Sleep.* Upon entering state *Burst*, the sensor will again initialize a timer, then begin transferring the contents of its buffered data as an EPC message in response to Query,QueryRep, or QueryAdjust messages; the sensor uses its BURST_NOTIFIER to send every message within the burst. Upon completion, timeout, or detecting 4 slots during which it finds no acknowledgement, the sensor returns to state *Sleep.*

4.5.1 Parameter Selection

While the state machine we previously described is a useful framework to constuct our protocol, implementation of the state machine was not straightforward, and needed several parameters to be carefully chosen and implementation aspects to be carefully addressed. We describe a few of these challenges in this section.

Timeouts. The timeout values used in our state machine are chosen based on Figures 4.3 and 4.4 in §4.4 that give good insight into expected connection intervals and message inter-arrival times respectively. In practice, these timeout values are used as comparison values for Timer A_1 on the WISP's MSP430 microcontroller. When considering hardware constraints and initialization overheads, one must also be careful to not choose a set of timeout values that generate too many interrupts that interfere with the WISP's ability to timely respond to reader messages. In practice, timeouts > 2 ms give the WISP sufficient time to listen for messages, while also providing the time needed to for timer initialization.

Duty-cycling. To implement the state machine, we also need to understand how the RF subsystem operates, and how to achieve maximal duty-cycling benefits. The RF subsystem comprises two components: a) the analog comparator that senses the channel to detect the presence of a bit, and b) the microcontroller that wakes up upon each interrupt from the comparator to process the bit and check if a valid message is present. The duty-cycling strategy is straightforward — shutting off the comparator avoids any energy lost from responding to interrupts and idle listening.

Burst Length. We choose one second as the length of a burst since it is long enough to obtain substantial duty-cycling benefits. After using up a burst, a tag pause 250 ms before trying to capture the channel for another burst. This duration provides a window for other tags to capture the channel or passive tags to transmit their identifier.

Burst Notifiers. The final parameter we consider is the burst notifier used by tags to coordinate their burst transfers. When modifying the WISP firmware, we found it can be difficult to get the state machine to stay within the tight timing constraints specified by EPC Gen 2 protocol. Complex operations in the firmware diminish responsiveness and in the end manifest as a reduction in goodput. For example, choosing a poor ordering of comparisons while looking for a burst notifier can lead to a 30% reduction in goodput. We also found that messages sent from the reader to the WISP can contain bit errors; one example is the RN16 field, which in actuality contains only 15 bits of consistent data. Thus, a careful implementation was needed to make burst notifiers operate correctly.

4.6 Evaluation

In this section, we evaluate the implementation of our bulk transmission protocol. The evaluation consists of four parts: 1) quantifying the goodput achievable by burst-mode EPC transfer, 2) showing that our burst notifier based coordination mechanism retains most of the goodput achieved by bursts by avoiding collisions, 3) demonstrating the energy benefits of duty-cycling, and 4) evaluating the interaction between bursting tags and standard EPC tags.

All evaluation results are obtained empirically; we feel this is important because of the complex RF environment that affects tag performance. We present results for small numbers of CRFIDs due to the limited number of prototypes available; the class of applications we consider (described in §4.1) are well aligned to the number of tags we use. In all of our experiments we use an Impinj Speedway reader with a fixed set of configuration parameters. The reader paramters used were: 1) Type A reference interval (Tari) = 25.0 μs, 2) Pulse Interval encoding (PIE) = 2.0:1, 3) Forward link = PR-ASK, 4) Pulse width = 0.5 5) Link Frequency = 256 KHz 6) Reverse Modulation = Miller 4, 6) Transmit power = 30.00 dBm, 7) channel = frequency hopping.

4.6.1 Burst mode transmission

The burst mode transmission protocol that we described in Section 4.4 ensures that all slots created for a round of communication are utilized by CRFIDs. In this section, we evaluate our burst transmission protocol in three ways: 1) We measure the window size allocated by Impinj reader when burst mode transmission is utilized, 2) We evaluate the goodput benefit gained by burst mode transmission, and 3) We provide a breakdown to show where the goodput benefits comes from.

Window size. In §4.4.2, we argued that the burst mode transmission strategy results in an RFID reader choosing a large window size (Q value) within a round of communication, leading to greater opportunities for using shorter messages, such as

QueryRep or QueryAdjust, for data delivery. To validate this argument, we design an experiment that compares the Q value chosen by an Impinj reader for standard EPC transfer vs burst transfer. Our experiment setup places an Intel WISP in line-of-sight of an Impinj reader's antenna. The WISP piggybacks the Q value in place of the EPC code that it backscatters to the reader. We found that the Impinj reader consistently selects Q $\approx$ 10 for burst transmission independently of distance. As described in §4.4.2, a large Q value is good for burst transmission. In contrast, the Impinj reader selects $Q \approx 2$ when the standard EPC Gen 2 protocol is used. Any Q value larger than 0 leads to un-utilized slots, therefore, this choice results in a significant fraction of wasted slots.

Goodput. To quantify how burst-mode transfer of EPC codes better utilizes slots, we design an experiment that compares the goodput of a burst-optimized version where a single tag responds within every slot versus EPC Gen 2. In this experiment, we measure the goodput of an Intel WISP tag programmed to act as a conventional EPC Gen 2 tag vs a WISP programmed for burst mode operation. We log the average throughput observed by the reader at different distances for several minutes and compare the results. Figure 4.6 shows that burst mode communication achieves 60% higher goodput than standard EPC Gen 2, and that these benefits are sustained across different distances. It is also notable that, in practice, the benefits of burst transfer are considerably larger than those that we predicted in §4.4.2. Our hypothesis is that additional gains are a result of the reader over-allocating slots for passive tags, resulting in comparatively low goodput.

Goodput breakdown. The increase in goodput for burst transmissions stems from two factors. First, we utilize every slot rather than one slot in each inventory round to transmit data. As shown in Figure 4.7, a standard EPC Gen 2 protocol only utilize only 64.8% of slots in each inventory round at 1m and 68.0% at 7m since it responds only to either a Query message or a QueryRep message, but not both – the other

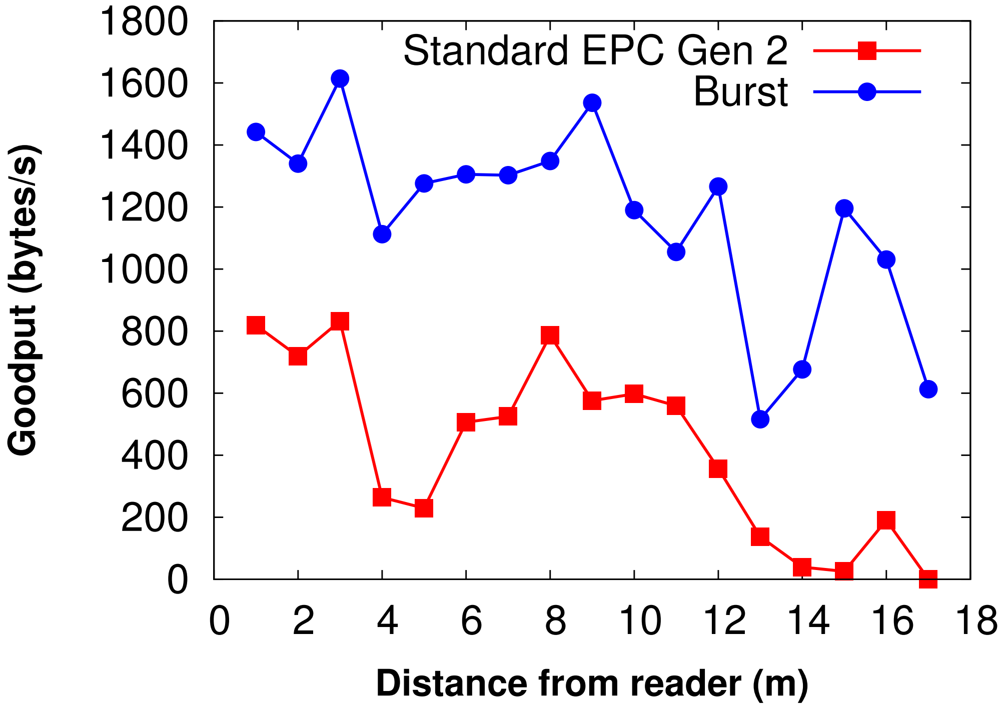

Figure 4.6. A CRFID sensor can achieve a 60% improvement in goodput by utilizing all slots in a communication round.

slots will go unutilized. In contrast, bursts will utilizes 100% of the slots. Second, in burst mode, we get more opportunities to exploit the shorter inventory messages: QueryReps and QueryAdjusts. Shorter messages in the forward link have lower loss rates; this leads to tags successfully capturing slots for data transfer with higher probability. For large Q, communication is dominated by slots that are initiated by QueryRep and QueryAdjust messages; as shown in Figure 4.7, the percentage of QueryReps at 1 m while bursting increases to 76.6%, as compared to the 34.2% slots in standard EPC Gen 2. This result holds true at a distance of 7 m as well. By exploiting QueryRep and QueryAdjust messages, tags get more opportunities for data transfer that contribute to higher goodput.

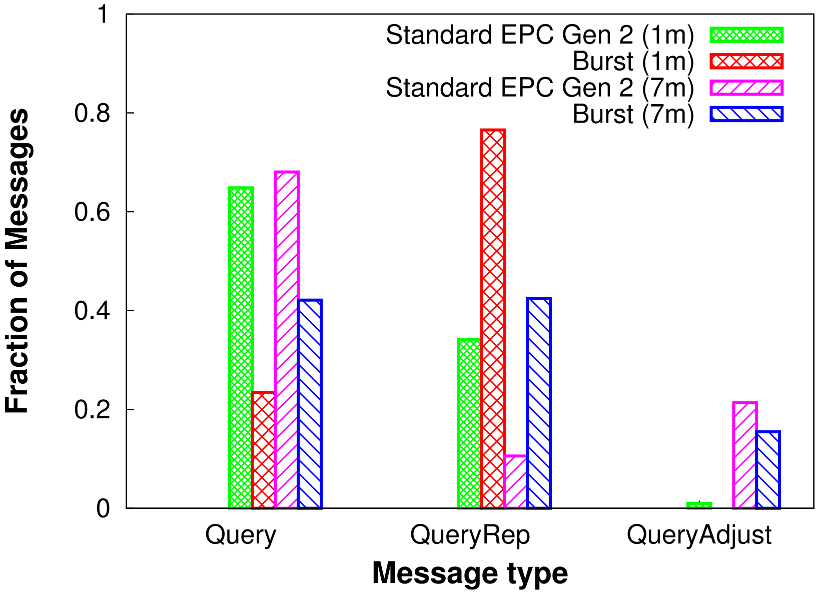

Figure 4.7. Bursting CRFIDs cause the number of round slots to increase. We observe this as an increase in the fraction of QueryReps received.

4.6.2 Coordinating bursts

While burst-mode transfer provides significant improvements to goodput, it also introduces problems when multiple RFID sensors wish to use the channel simultaneously. In this experiment, we evaluate how much the co-ordination mechanism benefits goodput when multiple tags are transmitting.

We consider a baseline case when a single CRFID is present, a low-contention case when three CRFIDs are simultaneously transferring data and a high-contention case when five tags are simultaneously transferring data. The number of tags in our experiment is limited by the WISPs that we have available, however, we expect that the number of sensors transferring simultaneously will be a relatively small number. All tags are placed in a line with their antennas placed parallel to the Impinj reader's antenna within line-of-sight at a distance of 1 meter. Figure 4.8 shows a breakdown of the goodput for three protocols: a) standard EPC Gen 2, b) Burst mode without coordination, and c) Burst mode transfer with cooordination.

First, we look at the case where there is a single tag. We see that the standard EPC Gen 2 tag performs significantly worse than the bursting sensor tags, as expected. We also see that the co-ordination scheme performs about 9.2% worse than the case without co-ordination. This is because of the overhead required for coordination. After finishing a burst transmission, a coordinated tag releases the channel for a 50 ms to allow other waiting tags to acquire the channel, which reduces goodput. Next, we increase the tag population to three; we see a similar behavior in terms of overall goodput, but the split across nodes is very different. Overall, we see that burst mode transfer with coordination is still a little lower (7.7%) in total goodput than the uncoordinated case. However, the un-coordinated case gives out more than 87.3% of the channel to one of the three tags. The burst protocol with co-ordination is considerably fairer — all three tags receive a good chunk of the overall goodput. Finally, when the tag population reaches five, the impact of collisions becomes significant and adversely impacts the performance of un-coordinated sensors. For bursts without coordination, the total goodput reduces by 65.7% as compared to the single tag and the three tag cases. In addition, the uncoordinated scheme is highly unfair and allocates 90.0% of the goodput to one of the five tags. In contrast, when the five tags coordinate, they acheive similar goodput as observed in the one and three node cases.

Table 4.2 shows the Received Signal Strength Indicator (RSSI) values logged at the reader for the backscattered data from the five tags and their respective goodputs. Despite having similar distance from the reader, the tags observe different RSSI values as a result of differences in antenna orientation. As a result the tag(Tag A) with strongest RSSI(-39 dBm) captures the channel entirely and achieves a goodput of 540.5 bytes/second. The other four tags cumulatively get 10.0% of the total goodput, each achieving less than 30 bytes/second. In contrast, coordination results in a more even partitioning of goodput. Table 4.2 shows the co-ordination mechanism doesn't

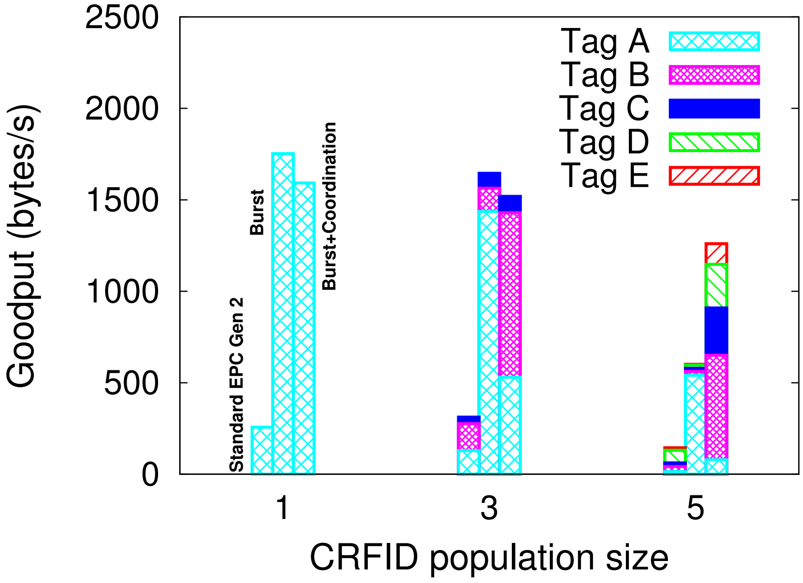

Figure 4.8. Coordination improves throughput and fairness as more bursting CRFIDs compete for the channel. For small tag populations, coordination incurs a small protocol overhead.

ID	Coordination		No Coordination	
	RSSI (dBm)	Goodput	RSSI (dBm)	Goodput
A	-41.7	78.9 Bps	-39.6	540.5 Bps
B	-37.7	571.1 Bps	-45.1	21.5 Bps
C	-39.9	258.8 Bps	-47.2	16.9 Bps
D	-46.0	238.2 Bps	-45.9	17.8 Bps
E	-50.1	112.8 Bps	-54.9	4.0 Bps

Table 4.2. A breakdown of the goodput from Fig 4.8 for the 5 tag case, shows that the amount of goodput a tag achieves is related to the average RSSI value at the reader.

necessarily favor the tag with highest RSSI — in fact although Tag D's RSSI is lower than Tag A, it gets a large fraction of the goodput.

4.6.3 Coordination-Aware Duty-Cycling

Coordination allows tags to avoid collisions between each others bursts, but does nothing to prevent energy consumed due to idle listening. We now evaluate our duty cycling mechanism which duty-cycles the comparator to reduce energy consumed due to idle listening. Our evaluation answers three questions: a) how much power is saved due to duty-cycling?, B) does duty-cycling result in degradation in goodput? and C) how does duty cycling affect the distribution of burst length among tags?

To quantify duty-cycling benefits, five tags continuously transmit EPCs for 10 minutes while deployed in a linear topology several feet from the reader with their antennas perpendicular to the reader's antenna. We set the probe and slew durations as defined in §4.4.4 as 20 and 200 ms respectively and look at the power consumed by each tag.

Figure 4.9 shows that duty-cycling reduces the average power consumption by 3.24x to 6.04x across the five tags when considering the amount of time the MCU spends in sleep vs active modes. The benefits of duty cycling differ depending on how often an individual tag gets access to the channel.

While duty-cycling improves overall energy consumption, a natural question is whether these gains come at the cost of goodput. Figure 4.10 shows that duty cycle based coordinated bursting tags have the highest goodput. This demonstrates that our duty-cycling mechanism does not sacrifice goodput to achieve its energy gains because of reduced channel contention that makes burst notifiers even more effective.

Finally, to better understand the dynamics of coordination and how frequently nodes switch among each other, we look at a breakdown of the duration that each tag holds the channel for the same dataset. We define a burst period as a contiguous segment when a single tag is bursting; at the end of the period, some other tag acquires the channel and starts bursting. These results are plotted in Figure 4.11. The results show that bursts are variable in length, even though they are allowed a maximum length of 1 second.

Fragmentation of bursts can result in unfair channel sharing because tags with better placement will see fewer spurious losses and hold the channel longer more

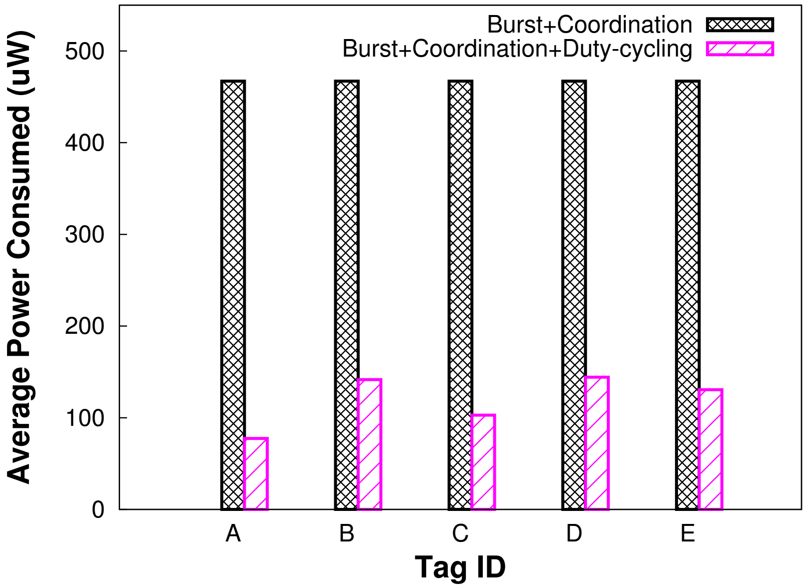

Figure 4.9. The Average power consumption of a CRFID is reduced considerably by reducing the energy lost to idle listening.

frequently. We see this phenomena manifest prominently in Figure 4.11, as Tag A was placed in an unintentionally poor orientation. Although Tag A's bursts are shorter on average, they maintain throughput (Figure 4.10 – when the channel becomes poor according to tag A, another node grabs the channel. Upon hearing another tag's burst notifier, tag A immediately goes to sleep, resulting in very high power efficiency.

4.6.4 Coexistence with passive tags

In this section, we investigate the interaction between our burst protocol and standard EPC Gen 2. We answer two questions when both bursting tags and standard EPC tags are transmitting to an Impinj reader: 1) How does the goodput of CRFIDs change when they compete for the channel with multiple standard EPC tags? 2) What is the time between inventorying standard EPC tag when CRFIDs are bursting? We setup experiments where both bursting tags and passive tags are collectively within the field of an Impinj Reader's antenna. We measure the goodput achieved by the

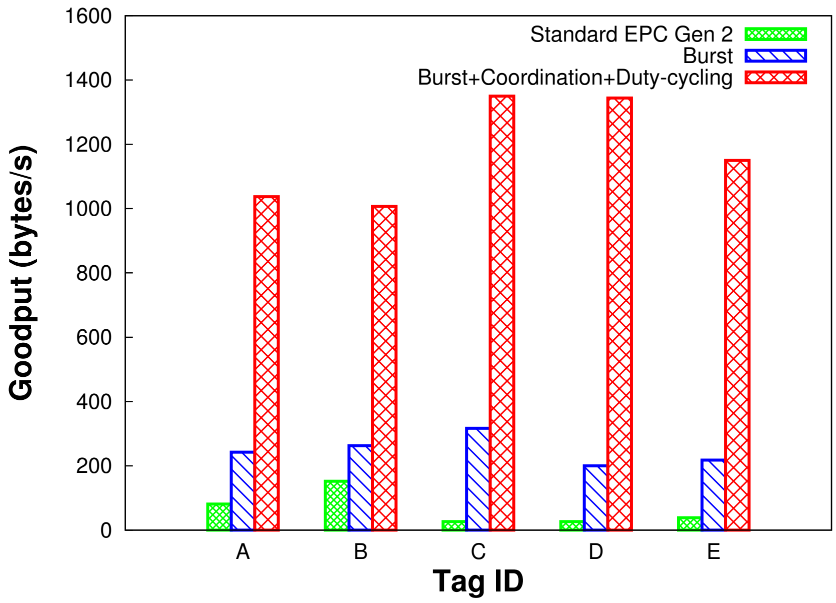

Figure 4.10. Nodes that duty cycle achieve considerably higher goodput. This is a result of reduced channel contention while non-bursting tags sleep.

tags that run the Flit protocol, and the interval required to inventory passive Gen 2 tags.

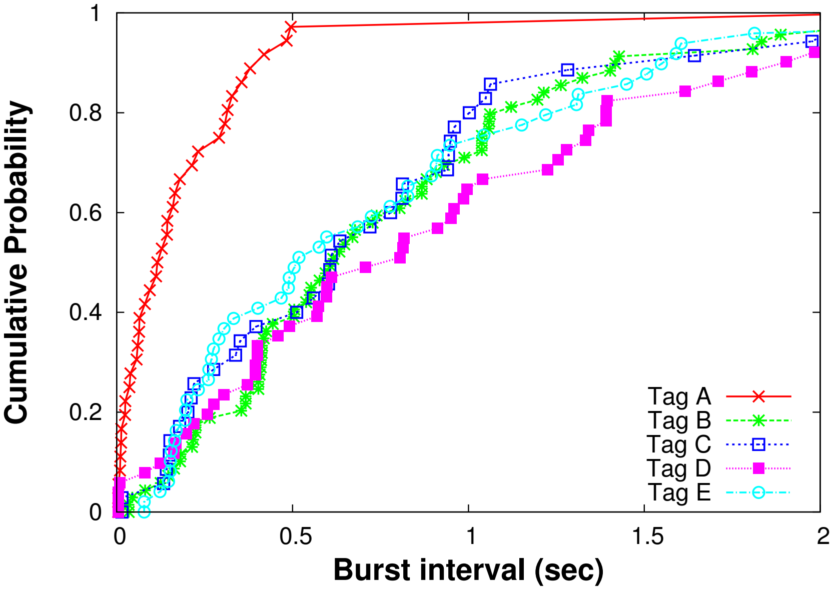

Figure 4.11. When tags use coordination, they achieve similar burst lengths; in this case Tag A is an exception because of particularly poor antenna orientation.

Goodput. Our goal in this experiment is to understand how increasing passive tag populations impact the goodput of bursting CRFIDs. Since we did not have enough Intel WISPs to use as passive tags, we use commercial passive tags for this experiment. We deploy five CRFIDs that are continually bursting to a reader, and increase the commercial passive tag population from 5 to 30. All commercial passive tags and CRFIDs are placed 1 meter from reader and spread in a line parallel with the reader antenna. Table 4.3 shows the average goodput as the passive tag population increases. The results show that there is only a small effect until about 20 tags (the average goodput is 253.3 bytes/second which is only a bit lower than 264.4 bytes/second when there are no passive commercial tags). For larger populations of passive tags, there are more collisions in slots which impacts goodput of burst CRFIDs. However, we see that despite a relatively large passive tag population, CRFIDs continue to perform well while bursting.

# Passive Tags	Avg Goodput (B/s)
0	264.4
5	222.1
10	238.0
15	270.7
20	253.3
25	184.2
30	184.9

Table 4.3. Deploying passive tags alongside a bursting CRFID causes the goodput of the CRFID to degrade as the number of passive tags increases.

Time between inventory. In this experiment, we look at how the time to inventory a population of passive tags is impacted by the presence of a burst CRFID. We consider two variants of Flit — one with the standard 50 ms sleep period between bursts, and one where this duration is increased to 100 ms. Figure 4.12 shows the average inventory time and associated 95% confidence intervals when there's a population solely comprising commercial passive tags vs a mix of passive tags and bursting CRFIDs. While the average inventory time increases with increasing population, the increase is greater when a bursting CRFID is in the mix. However, the total time is still of the order of a few seconds, showing that passive tags still see opportunities to get data through. In addition, increasing the sleep duration between bursts to 100 ms dramatically reduces the inventory time to be only slightly larger than the case when there are only passive tags. This provides a simple knob in deployments where inventorying time for passive tags needs to be low.

4.7 Related Work

Empirical Wireless Measurements. In recent years there has been significant work in measuring the characteristics of wireless communication channels for a variety of communication mechanisms (e.g. [82]). Perhaps most relevant to our work is [**?**],

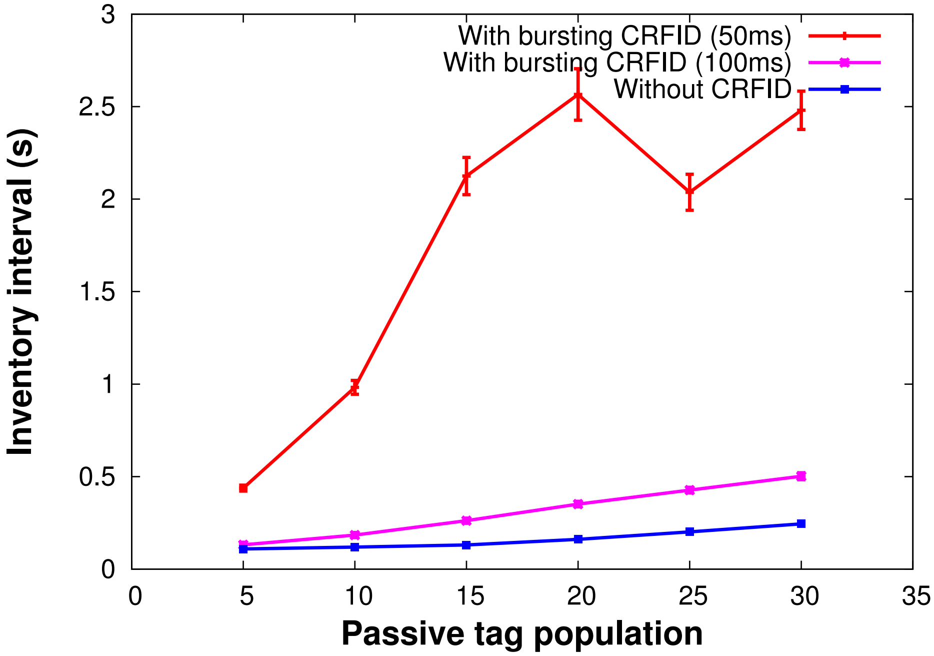

Figure 4.12. The average interval for a passive tag to be read increases with the tag population.

which quantifies wireless performance of Gen 2 through an empirical study that looks only at messages sent by a reader. In contrast, our focus is on improving tag to reader communications for CRFIDs. Additionly, our measurements provide visibility into the backscatter link by piggybacking statistics in EPC codes.

Bulk Data Transfer. The bulk transmission of data through a wireless channel has been studied in a variety of contexts including hardware and software systems. In an 802.11 setting [51] describes several mechanisms that together provide a transport layer optimized for bulk data transmission. Our work looks at RFID backscatter as opposed to 802.11; specifically, we focus on optimizations at the MAC layer for improving the throughput of bulk data transfer. Also of interest is [47], which provides a bulk transport protocol for 802.15.4 based wireless sensor networks. This work uses an end-to-end acknowledgement-based protocol to provide reliability, a rate control mechanism to minimize transfer time and a metric derived from combined signal strengths to avoiding hidden terminal issues. Instead, we focus on detecting other CRFID bursts using reader messages rather than explicitly avoiding them with a collision avoidance metric. Most similar to our work is [13], which proposes the use of persistent read handles in EPC Gen 2 to offload large amounts of data from an individual tag. While more efficient than requiring singulation for each read command, this approach is limited to data transfer from a single tag and cannot take advantage of the relatively shorter QueryRep and QueryAdjust commands.

While Flit provides bulk data transfer at the MAC layer, it does not preclude performance improvements at the PHY layer as defined by EPC Gen 2. In [93], the authors present a system that optimizes the channel and bit-rate selected by a reader, such that tag goodput is maximized. This approach is completely compatible with Flit; a CRFID communicating with an optimized reader would only see further goodput gains.

The elimination of idle slots introduced by coordination is not a new idea and has been looked at in wired contexts, where a common data bus is arbitrated for use by multiple entities. One good example of this is the bus parking mechanism used in the PowerPC 60x [9]. Here, the bus arbiter speculatively grants a bus master before receiving a master request; this grant message is coordinated by broadcasting to all attached devices. This is a very similar concept to the coordination mechanism we use in Flit, where the reader broadcasts a burst notifier to all tags within range.

Energy Management. There has been some recent work on energy management for CRFID systems. The work presented in [64] instruments code at compile time to enable checkpointing software state to non-volatile storage; the goal is to avoid energy wasted on work that was lost to power outages. In [15], a run-time system is presented that adaptively schedules a task to avoid energy wastage. Wastage is defined as work that does not complete due to of energy limitations, as well as harvested energy that cannot be stored. [40] looks at tradeoffs when ambient harvesting is used with RF harvesting, and explores the choice of hardware components to satisfy application requirements given an anticipated amount of harvested energy. Our work is complementary to these efforts, as we improve the bandwidth and energy-efficiency of the communication stack and efficiencies would only further improve with a run-time management strategy.

Looking beyond CRFID research, there has been substantial work on energy management in harvesting-based sensors. For example, [80] achieves perpetual operation by scheduling tasks to match predicted energy harvesting rates, [46] and [86] looks at adaptive duty-cycling strategies for harvesting-based systems, [36] looks at using efficient solar harvesting in combination with an ultra-wideband impulse radio to balance energy usage at an even smaller scale, and commercial efforts have looked at micro-energy harvesting from miniature solar panels, thermal differences, vibrations, and wireless power-over-distance technology (e.g. Powercast [3]). Such techniques may be useful to ensure a suitable amount of buffered energy for bursts is available during reader contact periods, and is complementary to this work.

4.8 Conclusion

In this chapter we presented the design, implementation, and evaluation of Flit, a bulk transmission protocol for RFID-scale sensors. Through a careful analysis of the EPC Gen 2 protocol for passive RFIDs, we identified several opportunities for improvements to both throughput and energy efficiency when considering small numbers of CRFIDs that have large amounts of data to send. Through empirical evaluation, we demonstrated that significant gains in goodput are possible over a variety of distances when compared to a tag that implements vanilla EPC Gen 2. To enable the simultaneous bulk transfer of data from multiple CRFIDs, we designed a simple coordination mechanism that works well in practice and through an experimental evaluation, showed the complete system retains most of the performance improvements we observed for a single, uncoordinated CRFID. Finally, we demonstrated that our protocol can coexist with passive RFIDs can, but are inventoried with increased latency.

CHAPTER 5

USING A PASSIVE EMBEDDED SYSTEM TO PROTECT THE MOBILE PHONE FROM MALICIOUS NFC INTERACTIONS

One of the greatest challenges that passive embedded systems currently face is lack of available communication infrastructure. The previous two chapters described systems that target UHF RFID protocols; one major drawback of UHF RFID is the reader infrastructure used for communications is primarily deployed in industry supply chain networks and is not accessible for general purpose consumer applications. In this chapter we shift our focus to HF RFID systems since readers for HF are commonly found in mobile phones. In particular, we look at an application scenario where a phone-based reader is used to power a passive peripheral that protects the phone's NFC interface by selectively blocking malicious NFC messages.

5.1 Introduction

Near Field Communication (NFC) has begun to make its way into major mobile phones, with several Android, Blackberry, and Nokia phones already providing such functionality. The proliferation of NFC on phones can open up a range of applications, from being able to interact with NFC-tagged smart posters to revolutionizing the payment industry, where phones are expected to replace credit cards as the most convenient way to pay for products at the point-of-sale.

These benefits of NFC come at a price — security becomes particularly challenging since phones are general-purpose computing devices that expose a relatively large attack surface that can be exploited by unscrupulous individuals. These issues were exposed in a recent security breach that leveraged the fact that NFC tags can be registered to open applications on a phone such as images, contacts, or web pages *without requiring user consent*. In this attack, a mobile phone was directed to a URL that hosted code that exploited a vulnerability in Android 4.1's web browser [58].

In addition to technical issues, there are also non-technical challenges at play — NFC mobile payments involve interaction between mobile phone manufacturers and OS vendors (Blackberry, Google), mobile phone operators (ATT, Verizon, etc), and banking organizations (VISA), leading to a complex and intertwined web of control. For example, viaForensics [85] announced a Google Wallet vulnerability almost a year ago, but it has yet to be patched because the fix would require a "change of agency" rather than a quick OS patch. Thus, the outcome of business interests and complex business dealings are likely to provide more opportunities for attacks that target the fuzzy boundaries between these entities.

Existing efforts attempt to address these security concerns in several ways. First, many mobile operating systems turn off the NFC interface when the screen is locked. But if the OS is compromised, a malicious rootkit can keep the NFC interface turned on when the screen is locked, thereby thwarting this defense. Second, mobile payments ask the user to provide a four digit pin before an NFC-initiated payment. However, this is also vulnerable to attacks such as the one demonstrated in [22], where the pin code was inferred by looking at data stored by an NFC payment application. Once the pin-code is cracked, a rootkit can potentially bypass user input entirely and make a mobile payment that the user is completely unaware of. Third, phones can use hardware (secure elements) that provides security guarantees for mobile payments ([54, 81]), but such hardware is not available for phones acting as readers. Thus, none of the mechanisms fully address the scope of security issues presented by NFC.

We argue that there is a need for a hardware-based "NFC guardian", *EnGarde*, that is perpetually attached to the phone, and acts as an NFC firewall that allows legitimate interactions to occur as normal, while blocking unwanted NFC interactions through jamming. While the idea of jamming unwanted interactions is reminiscent of RFID blockers [66], practical instantiations of such ideas are bulky systems with large power draw, and consequently not in wide use. In contrast, our design is small, passively powered, and can be fully integrated on a mobile phone, thereby making it entirely practical.

In addition to jamming, one of our goals is to design a tool that can be invaluable to security concerned individuals that seek more insight into the low-level behavior of their phone's NFC interface. For example, unexpected data usage by an official NFC application, Google Wallet, has been reported in several forums [4] but the lack of visibility makes it difficult to determine whether this is the result of interactions with external NFC devices. Creating a tool that puts the user in control of the interface, rather than the operating system, could satiate some of these concerns until the security implications of NFC on mobile phones are better understood.

Our design contributions are four-fold. First, *EnGarde* has the form-factor of a self-contained and self-powered thin pad that attaches to the back of the phone, and is agnostic of mobile operating system differences as well as idiosyncrasies of different dock connectors. Second, *EnGarde* is easy to use since it operates entirely through power scavenged from the NFC reader on mobile phones (or external readers accessing the phone). Thus, it requires zero effort on the part of user to change batteries, and only has a small effect on the phone in terms of overall harvesting needs. Third, *EnGarde* defends against a wide range of passive tag and active reader based attacks that cover the spectrum of NFC protocols and operational modes including those that target the phone a) in reader mode interacting with a malicious tag, b) in tag mode interacting with a malicious reader, and c) in active peer-to-peer mode interacting with a malicious phone. Fourth, *EnGarde* can be programmed to trigger upon detecting specific types of messages, protocols, or transactions that are indicative of security violations, and disrupt these interactions through jamming.

Our design presents a range of technical challenges that we address in this work. First, we dramatically reduce power consumption during jamming by requiring no active transmission in most cases; rather we leverage the NFC carrier wave to generate an interfering subcarrier while scavenging energy. Second, we design algorithms that maximize the energy scavenging efficiency from the phone while simultaneously minimizing the power footprint on the phone. Third, we design an early warning mechanism that detects presence or absence of an NFC device in the vicinity without any communication occurring between the phone and the device, thereby enabling *EnGarde* to stay out of the way when there is a legitimate transaction as well as to prime itself to thwart an illegitimate one. Fourth, we prototype the complete system and hardware, and demonstrate that all of the outlined capabilities can fit in a flat form-factor of roughly five square inches, demonstrating its practicality.

Our results show that:

- We can jam tag responses with 100% success rate while consuming only 6.4 μW of power, which is considerably more efficient than prior approaches that have used active jamming.
- We can accurately detect tag presence with an accuracy of 95% under a wide range of conditions, while having negligible impact on legitimate communications.
- We can continuously power *EnGarde* solely through NFC-based power scavenging, while being 4× more efficient than a naive harvesting approach that does not consider the host phone's power consumption.
- We can defend successfully against attacks similar to a known URL attack scenario, and show that we can detect and block a particular NDEF URL type

with 100% accuracy, while allowing other NDEF messages to reach the phone unimpeded.

5.2 Design Requirements

In this section, we discuss some of the requirements that we used as the rationale for our design choices in *EnGarde*.

Protect all NFC Modes: A central design goal is protecting all NFC modes implemented on a mobile phone. This means that *EnGarde* should be able to block NFC messages between the phone and external entity whether the phone is acting as a reader or in tag emulation mode. The specific types of attacks *EnGarde* should protect against are:

- ▶ Malicious tags deployed in infrastructure (such as tags with URLs). When the phone discovers and interprets the tag's information, it is instructed to take some action that compromises the phone's security; this could be the phone being directed to a malicious web site. In this attack, we need to protect the phone while it acts as a reader and block communication before the phone receives the malicious data. This type of attack is well known in the security community as "fuzzing".
- ▶ A Phone encounters a malicious device in peer-to-peer mode. Since this type of interaction can support arbitrary file transfer, the phone is vulnerable to whatever content is transferred from its peer. *EnGarde* would need to detect and block these malicious data transfers.
- ▶ An external reader reads and discovers the ID used by the phone while in tag emulation mode. This would mean that an external entity would be able to track the location of a particular phone user each time the ID is read, compromising their privacy. Additionally, a similar attack could result in the user's financial information being compromised if it were sent in the clear. *EnGarde* should

block the release of this information. This type of attack is well known in the security community as “identity theft”.

- A user inadvertently installs an application on their mobile phone that uses the NFC interface for malicious purposes. The user’s phone may then subsequently be used by an attacker to perform any of the attacks above.

Transparently Powered: *EnGarde* should integrate with a mobile phone in the most transparent way possible. For example, one way to power *EnGarde* would be to connect it via the phone’s dock connector which could then be used to supply power. However, this would leave *EnGarde* completely unpowered if a user connects a different peripheral to the dock or charges the phone and forgets to plug in *EnGarde*; instead, we advocate a passively powered mechanism where *EnGarde* harvests power from the NFC interface while the phone is actively being used. We also note that powering the phone in this way would allow *EnGarde* to function independently from a *potentially* compromised operating system. Thus, *EnGarde* should be a physically separate piece of hardware that does not rely on a wired interface for power.

No impact on usability: We wanted *EnGarde* to be almost invisible, both in terms of physical form factor as well as in its effect on the usability of the phone for legitimate NFC transactions. This meant that *EnGarde* should be small enough that it can be a patch stuck on a phone (or eventually integrated with a phone’s battery). Additionally, *EnGarde* should not diminish user experience for NFC transactions that a user wishes to make. In other words, there should be negligible effect in terms of packet loss rates or distance at which NFC transactions are possible if a legitimate NFC device is communicating with the phone.

Programmable Rules: Given that the types of NFC vulnerabilities will almost certainly evolve as new attacks are discovered and known attacks are patched, we want to have a fully programmable platform where the rules upon which to jam can be specified. These rules can range from blocking all exchanges of a certain type (e.g. payments), blocking exchanges with tags that contain a URL and use the browser, blocking when certain sensitive information is transmitted in clear text, and so on. Thus, *EnGarde* should be programmable, and block only those interactions that are known to be vulnerable, while allowing other messages to get through to the phone.

Fail Safe: Since *EnGarde* relies on energy scavenging, one question is what happens if it runs out of power; this may occur after a long period after the phone is idle. In particular, since *EnGarde* decodes messages and jams only when malicious interactions were detected, what would happen if the microcontroller that makes this decision is unable to operate? The duration when *EnGarde* is charging provides a window of opportunity to an attacker. Thus, a key requirement is that *EnGarde* should fail safely, i.e. when the MCU does not have sufficient power to make intelligent jamming decisions, it should default to a mode where it jams NFC interactions as soon as the phone initiates NFC discovery until the MCU is able to operate and make a more judicious decision. In this manner, the phone is protected whether or not *EnGarde* has charge.

5.3 An Overview of NFC

NFC is a relatively new technology. In this section, we give an overview of NFC by examining the underlying communication standards, protocols, and physical layer characteristics. The design of *EnGarde* is heavily influenced by these details.

5.3.1 NFC Communication Layer

NFC uses High Frequency (HF) RFID as its communication layer. The NFC standard requires that a compliant device be compatible with all existing HF RFID communication layer standards. HF RFID is used to communicate between a *tag* and

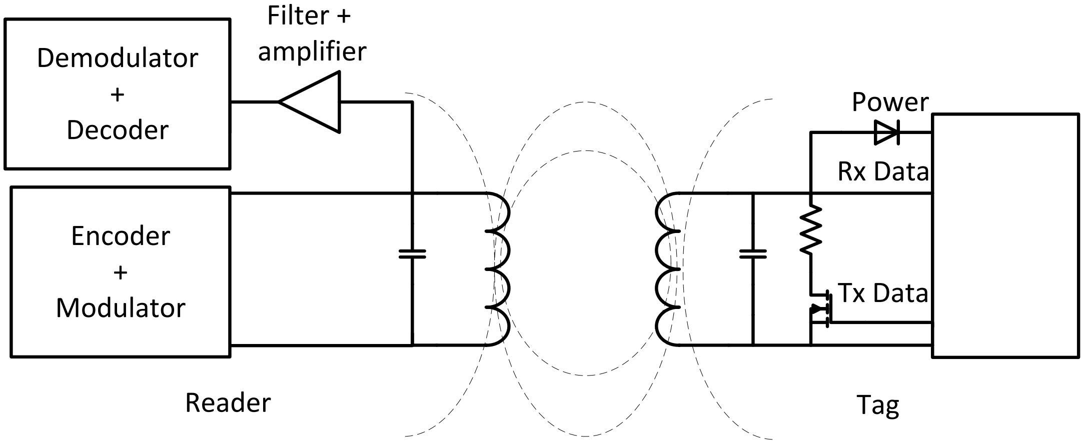

Figure 5.1. Functional block diagram of HF RFID reader and tag.

a *reader*. The tag contains a globally unique ID and some data which is optionally writable by a reader. The tag is a passive electronic device which is powered by the reader during communication.

The reader powers tags in its vicinity using a magnetic field. Before communicating with a tag, the reader runs a discovery protocol to discover the tags in its vicinity. If multiple tags are found, a collision avoidance protocol is used to identify individual tags. Once a tag is discovered, the reader uses the tag ID to uniquely address the tag for reading and writing tag data.

Since the reader generates the magnetic field to power the tag, the communication is always reader initiated. During each interaction, the reader generates the field and sends a message addressed to a specific tag; the tag, after interpreting the reader message, sends a reply.

Figure 5.1 shows the basic components of the HF reader and tag. The reader generates a magnetic field at 13.56MHz using a tuned reader coil; the tag has a coil tuned to the same frequency. Due to the magnetic coupling between these coils, similar to the operation of an electrical transformer, the reader coil induces a voltage in the tag coil. This AC voltage is converted to a DC voltage to power the tag electronics. Since magnetic field strength decays rapidly with distance, NFC systems have a typical range of a few centimeters (with larger reader antennas and high-power readers, the communication range can go up to 1 meter).

Reader to tag communication. Reader to tag communications use Amplitude Modulation (AM) of the 13.56MHz carrier. The carrier amplitude variation causes a corresponding variation of the voltage induced at the tag's coil.

The tag decodes this signal variation using a simple circuit. Different communication protocol standards use AM as a primitive to encode data using different coding techniques. Table 5.1 shows various protocol standards and modulation formats.

Tag to reader communication. Tag to reader communications use load modulation, where the load across the tag coil is varied by switching on and off a parallel resistor (or a capacitor). Since the tag coil receives its power from the reader coil, the varying load causes a varying current and a voltage at the reader coil.

The load modulation is used to generate a 847.5kHz sub carrier which is encoded using different coding techniques (Table 5.1).

5.3.2 NFC Device-Level Interactions

Although NFC is based on HF RFID technology, NFC has more capabilities than discovery, reading, and writing of RFID tags by a reader.

Communication modes Unlike a traditional reader or a tag, an an NFC-enabled phone can take on multiple roles:

Phone as a reader. In this mode, the NFC enabled mobile phone behaves as an RFID reader. The phone periodically runs a tag discovery loop to identify compatible tags in its vicinity, and establishes communication with them. This mode is typically used to scan QR-code like tags that contain a short piece of information such as phone numbers and URLs.

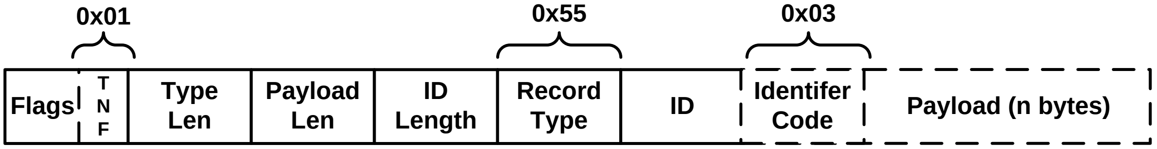

Figure 5.2. An NDEF message has a regular structure and holds an NDEF record. This message contains a URL that uses a prefix of "http://"

Phone as an emulated tag. In this mode, called *tag emulation* mode, the mobile phone behaves like an RFID tag – an external reader can discover and interact with the phone. Since the NFC-related circuitry is powered by the reader's magnetic field, this mode can be active even when the phone has no power. This mode is typically used for mobile payments in transit card-like applications.

Phone as a peer. Here the phone communicates in a peer to peer mode with another NFC enabled device such as a phone. In this mode, which is typically entered after one device discovers the other, both parties take turns generating the carrier. This communication mode supports the highest rate of communication and is used to share small files between mobile phones.

NDEF standard The NFC Data Exchange Format (NDEF) provides a common language that enables HF RFID tags, which could be based on different HF standards, to exchange data. The well known NDEF message has a regular structure – In Figure 5.2, we show an example of one such NDEF message. This particular message contains a record that conforms to a well-defined type, as indicated by the TNF field being set to `0x01`. The ID field of the message increases the degree of specificity – the particular well defined type is a URI, as indicated by the Record Type field being set to `0x55`. The last field in an NDEF message contains the NDEF record; in a URI record, the first byte contains a prefix that is applied to the message. This particular URI uses an identifier code of `0x03` to apply the prefix `http://`; other options could have been `0x00` or `0x04` for example, which correspond to `no prefix` and `https://` respectively.

	Coding Forward	**Coding Reverse**	**Bit Rate kbps**
ISO 15693	1 out of 4/256	Manchester	1.65, 6.62, 26.48
ISO 14443-A	Mod. Miller	Manchester	106, 212, 424
ISO 14443-B	NRZ-L	BPSK	106, 212, 424
Sony FeliCa	ASK	Manchester	212, 424
ISO 18092	Mod. Miller Manchester	Manchester	106, 212,424

Table 5.1. Summary of NFC Forum Supported Protocols

Platform support In addition to the various protocols and messaging formats available, there are also differences in how NFC is supported across platforms. We summarize some of these differences in Table 5.2. All platforms we looked at disallow use of the phone as a reader while the screen is locked. A couple of key differences are that Blackberry 7 and Windows Phone 8 allow card emulation mode to work while the screen is locked. In fact, Blackberry 7 allows any user application to access card emulation mode; however, only core applications have access to the secure element. *EnGarde* is designed to operate across all these platforms and operating systems.

Platform	**Card Emulation Support**
Android 4.1	While screen unlocked; only Google Wallet
Windows Phone 8	While screen unlocked/locked; Restricted applications
Blackberry 7	While screen unlocked/locked/off; Any user application supported

Table 5.2. The management of tag emulation mode varies across platforms.

5.4 Identifying NFC protocols

One of our design requirements is that *EnGarde* supports programmable blacklisting rules, which implies that it should be able to both listen to, and interpret all possible NFC message exchanges in real time, and decide which ones to block and which ones to allow. However, this requires that *EnGarde* be able to decode a wide range of NFC modulation formats to determine which of the NFC protocols is being used so that it can determine the information content in them.

While this may seem similar to what an NFC reader does to read tags that support different NFC formats, there is a key difference. When an NFC reader establishes communication with another NFC device, it first goes through a discovery phase composed of multiple RFID protocol-dependent discovery messages. Hence, when a reader discovers a tag, the reader identifies and agrees upon the modulation protocol to be used with that tag. In contrast, *EnGarde* does not know what protocol is currently being used, and needs to search through all possible protocols to determine which one is correct.

One option to perform such a search might be to use a software radio, but this has significant limitations. The first column in Table 5.3 shows the modulation pulse width for different protocols varies by more than an order of magnitude; this implies that a software radio would need to sample the carrier at the highest rate required to decode all these protocols, and then search through the signal to identify the current protocol. However, this would result in considerable energy overhead, both because of the high rate of carrier sampling (e.g. detecting NFC 15693 requires 31× lower sampling rate than for NFC 14443-A), and because of the substantial processing overhead of performing the search. Thus, it is critical to find a cheaper option for identifying the protocol.

Leverage reader-to-tag messages Our key idea is to leverage the reader to tag portion of the each communication round. During each reader and tag interaction,

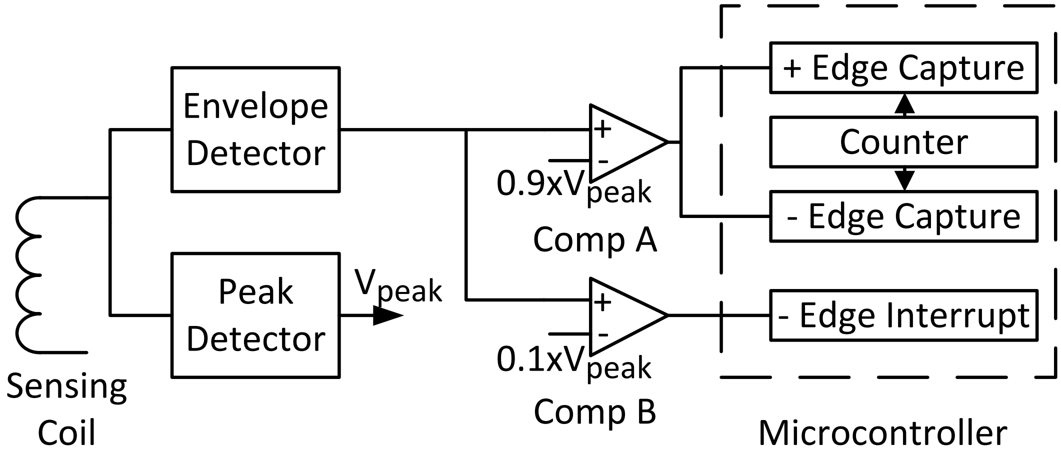

Figure 5.3. *EnGarde* classifies different NFC protocols by examing a signal's carrier pulse characteristics.

the reader initiates the communication with an ASK modulated signal, while tags respond back with a subcarrier modulated signal. The ASK modulated carrier signal requires very limited energy resources to decode. We can simply examine the first pulse of the amplitude modulated carrier at the start of an NFC message to group the protocol into several categories. Table 5.3 shows how different pulse characteristics map to different protocols.

Low-power protocol detector Figure 5.3 shows the HW implementation of such a detector that does the first level of protocol classification. *EnGarde* uses a small "sampling coil", consisting of a couple of turns, to sample the RF signal. Two comparators are used to detect the pulse edges and the modulation depth of the envelope of the modulated carrier signal. A HW timer-based capture units of a microcontroller enable the measuring of this pulse width with an accuracy of 0.5 μs; an interrupt pin captures the ASK modulation type. The power consumption of the analog portion of this circuit is only 34 μW.

Pulse (μs)	OOK	ISO Protocol and speed (kbps)
0.29	N	14443A-848
0.59	N	14443A-424
1.18	N	18092-424, 14443A-212, Felica-424
2.36	N	18092-212, 14443B-424, Felica-212
2.36	Y	18092-106, 14443A-106
4.72	N	14443B-212
9.44	N	14443B-106, 15693
9.44	Y	15693

Table 5.3. Different protocols that map to given characteristics of the 1st carrier modulation pulse of a NFC data packet.

Once the protocol is assigned to one of the subgroups, a lightweight software solution can uniquely identify the specific protocol by examining the first few starting bytes. Once the RFID protocol is identified, we use an off-the-shelf NFC reader chip for decoding the data.

5.5 Jamming NFC communication

Once malicious activity is suspected by examining ongoing message exchanges, *EnGarde* should disrupt the communication by jamming. While past work on protecting RFID transactions has used active jamming techniques, this requires several 100 mWs of power, which is much higher than what we can afford on *EnGarde*. Our goal is to design a cheaper jamming mechanism, that operates within the constraints of the energy that we can scavenge.

Since NFC communication has two distinct phases — reader communication and tag response–we look at these cases separately, and design jamming primitives for each of them. We then show how these primitives can be used to effectively jam the different NFC protocols.

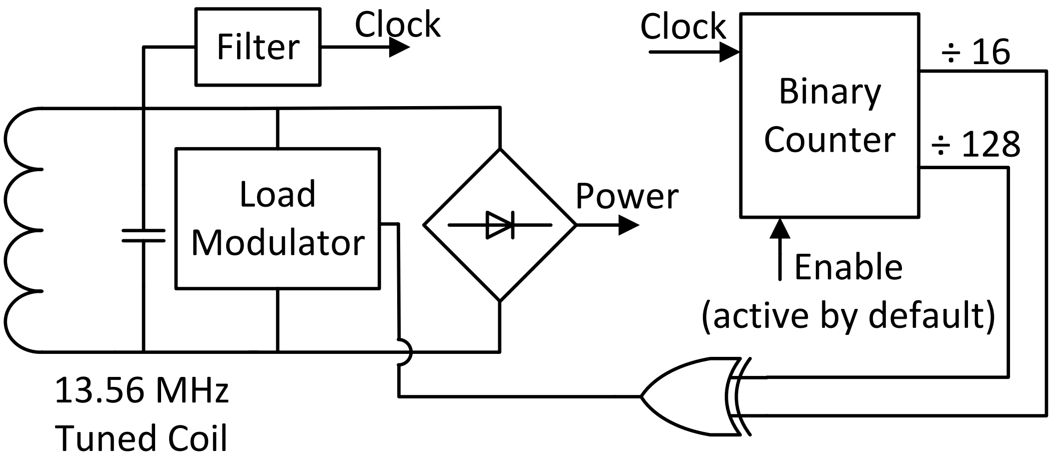

Figure 5.4. *EnGarde* harvests energy and blocks malicious tags using a load modulation-based tag jamming circuit.

5.5.1 Jamming Primitives

There are two jamming primitives, which we call *reflective jamming* and *pulse jamming* based on what communication modality is being jammed.

Reflective jamming Tag-to-reader communication uses load modulation of the tag antenna using a subcarrier frequency. Our key observation is that all NFC protocols use a *common 847.5kHz subcarrier*, irrespective of the communication data rate used. Hence, to jam ongoing tag response communications, *EnGarde* only needs to generate a 847.5kHz subcarrier using load modulation of the tuned coil. Such a load-based modulation is particularly attractive since this is exactly what a typical NFC tag needs to perform, and hence can be easily done using energy scavenged from the ongoing communication that is being jammed.

A hardware implementation of such a subcarrier-based jammer is shown in Figure 5.4 (based on an NFC tag reference design in [34]). This circuit is similar to an NFC tag in that the jamming electronics are completely powered by the energy scavenged from the reader. The circuit has minimal components and therefore consumes little power. Our measurements show that subcarrier generation only consumes 6.4 μW of power.

Pulse jamming Unlike jamming a tag response, jamming a reader (or a peer device) requires *EnGarde* to generate an active magnetic field transmission that interferes with an ongoing reader transmission. However, as we described earlier, generating active transmissions that are capable of swamping the signal from the reader would require 100s of mW of power (e.g. the TI TRF7970A RFID reader consumes as much as 250 mW). This would be almost an order of magnitude more power than what can be scavenged on *EnGarde*, making it infeasible for our purposes.

Our approach to address this problem is to generate a targeted pulse that disrupts an ongoing communication. Since different ASK-based carrier modulation schemes require carrier modulations at bit-time durations, a carrier pulse only needs to be $\simeq$ 20 μs (2 bit durations at the lowest data rate) long in-order to corrupt the message. Such a pulse-based jamming mechanism is orders of magnitude shorter than the duration of the shortest valid NFC message, which means that it can easily be supported via scavenged energy.

Our pulse-based jamming approach has one drawback — it is possible that a high-powered NFC reader generates a strong enough signal that our attempts at corrupting the signal does not result in a sufficiently high signal-strength difference, and is therefore unsuccessful. While this is a weakness of the technique, we think that we would block a large fraction of reader transmissions, particularly because *EnGarde* would be much closer to the phone than an external NFC device that is initiating such a signal.

5.5.2 Jamming During NFC Communication

Given the two primitives, we now look at how to use them to jam different NFC communication modes.

- ► *Tag Reader Mode:* In this mode, the phone acts as an RFID reader and reads passive tag content. A possible attack could be a malicious tag that directs the mobile phone to a malicious website. In this mode, *EnGarde* uses subcarrier-based jamming to disrupt the data exchange with the tag.
- ► *Tag Emulation Mode:* Here, the phone acts as a tag and responds to queries from another NFC device (a phone or an infrastructure reader). In this mode, *EnGarde* can use the subcarrier-based jamming to prevent leakage of sensitive information from the phone.
- ► *Peer-to-Peer Mode:* During this mode, the phone and an external NFC device exchanges information by both actively transmitting signals. *EnGarde* needs to transmit a jamming pulse signal to block malicious interactions in this instance. While *EnGarde* may not be able to block malicious high power transmitters, we note that peer-to-peer interaction starts with a discovery step during which nearby devices are discovered. Hence, subcarrier-based jamming can be used at this stage to disrupt the establishment of a peer-to-peer communication thereby nipping such a transaction in the bud.

5.6 Energy Harvesting

Energy scavenging is central to the design of *EnGarde* since it allows the device to operate perpetually despite having a small energy buffer. Our approach is to leverage the same inductive coupling based harvesting mechanism by which NFC tags harvest power to communicate with a phone. This gives *EnGarde* the unique ability to jam communications while at the same time scavenging energy from the source.

While NFC enables energy transfer, one question is how much power can be harvested by *EnGarde* from the phone, and how much power is expended by the phone for this transfer. To understand this, we measure the power draw of the phone using a Monsoon power meter on a Samsung Galaxy Nexus phone running Android

Harvesting Strategy	Phone's NFC Duty Cycle	Phone Power Consumption Overhead	Energy Transfer Efficiency
Opportunistic	10.1%	0 mW	17.30%
Tag-Spoofing	33.3%	27.2 mW	19.16%
Subcarrier	86.6%	154.1 mW	12.49%
Full NFC	100%	287.38 mW	8.04%

Table 5.4. Tradeoff between harvesting efficiency and phone's transfer efficiency

4.1 (Jelly Bean) when an NFC tag is in front of the phone vs when NFC is completely turned off. We also measure the peak AC power harvested on *EnGarde* during NFC activity. Our results show that the phone's carrier is continuously switched on, hence we are able to harvest 30mW of power at *EnGarde*. This means that *EnGarde* can potentially have a fairly significant power profile. However, we also see that the phone consumes 301.5mW of power during this process, i.e. the transfer efficiency is only 9.95%. In this section, we ask how to balance the need to buffer sufficient energy in *EnGarde*'s energy buffer while maximizing transfer efficiency so that *EnGarde* has only a small impact on the phone's battery.

5.6.1 Harvesting mechanisms

We now outline three harvesting alternatives that have better power transfer efficiency than the full-NFC mode for harvesting power from the phone.

Opportunistic The first harvesting mode, which we refer to as *opportunistic*, is essentially for the tag to do nothing special, and just opportunistically harvest energy coming from discovery messages that are transmitted by the phone. When a phone is unlocked, the phone sends out discovery messages periodically (10% duty-cycle) to check for the presence of nearby devices. The advantage of this mode is that the phone incurs no additional overhead beyond what it is already incurring.

The transfer efficiency in this mode reveals a surprising result — the average power consumed by the phone increases only by 14.1 mW for transmitting discovery messages, which gives us a transfer efficiency of 17.3%, which is almost double of the transfer efficiency when the phone is in full NFC active mode. The likely cause for the difference in efficiency seems to be that the bulk of the NFC protocol is implemented on a dedicated NFC reader chip that resides in the phone's battery. Only valid responses from an NFC tag results in interrupts that are handled by the operating system. Thus, when an NFC tag is transmitting valid responses, additional CPU cycles must be spent handling the read events, resulting in the extra power consumption. Thus, opportunistic harvesting is efficient but *EnGarde* only gets 10% of the power that it would have received were the phone's carrier continually active.

Tag-Spoofing The second harvesting mode, which we refer to as *tag-spoofing*, tries to trick the dedicated NFC chip into delivering more power without interrupting the Android OS as frequently as in full NFC mode. To implement this strategy, we look at how an NFC reader performs initial detection of tag presence. After sending energy to a potential tag, the first hint that a tag may actually be present is looking for a change in the voltage of the carrier it sent to a tag. A transponder influences this voltage by *modulating* its transponder coil. If a reader observes this change in voltage, it may decide to send additional energy for subsequent communications.

We test this theory by modulating the harvesting coil via a resistor using a short pulse that is 10 us in length. Indeed, we found that this did result in the phone providing more power for a short period, after which it times out and reverts to discovery mode. The process can be repeated to ensure that the reader continues to provide additional power.

This harvesting strategy results in the phone's subcarrier being active for 33.3% of the time, so *EnGarde* gets about three times the power that was harvested in opportunistic mode, but it also incurs 41mW overhead on the phone (i.e. 27mW more than for discovery messages). However, the transfer efficiency is high at about 19%, which is even higher than what was obtained in opportunistic mode.

Subcarrier Our third harvesting mode, *subcarrier*, closes the gap between tag-spoofing and full-NFC in terms of amount of harvested power at *EnGarde*. As discussed in §**??**, jamming is performed by generating a 848 kHz subcarrier. Our measurements show that subcarrier is able to increase the amount of time the carrier is active to 86.6%, and results in 168.2 mW of harvesting overhead on the phone. This gives us a transfer efficiency of 12.49%, which, while not as good as the opportunistic and tag-spoofing modes, is about 50% better in efficiency than the full-NFC mode while giving *EnGarde* close to the maximum average power.

5.6.2 Demand Harvesting Algorithm

We now have three harvesting schemes that are more efficient than full NFC mode — opportunistic, tag spoofing, and subcarrier — that give us different options in terms of amount of scavenged energy at *EnGarde* and transfer efficiency.

We now describe a demand-based harvesting algorithm that runs on *EnGarde* and ensures that sufficient power is harvested from the phone to maintain its energy buffer at close to its maximum capacity, while minimizing the energy cost incurred by the mobile phone to transfer power. The algorithm works in two steps: First, it estimates the length of the next unlock interval by observing history of phone use. We use a simple EWMA filter over the history of unlock durations in our implementation. Second, it uses the estimated unlock duration to determine the fraction of time to use each harvesting mode needs to be used. Intuitively, if the buffer can be filled up just using opportunistic harvesting, then this is the cheapest approach since discovery messages are transmitted by the phone whether or not *EnGarde* is harvesting this power. If this is not sufficient to replenish energy in the buffer, then the algorithm needs to decide how to use a combination of the other two harvesting modes to ensure that the buffer is filled while maximizing transfer efficiency.

We formally define the parameters described in the model as: a) T is the current estimate of unlock duration from an EWMA-based estimator, b) B is the current energy buffer level, and B_{max} is the desired energy level, c) $\{E_{opp}, E_{so}, E_{sub}\}$ are the Energy harvested from {opportunistic, tag-spoofing, subcarrier} modes if they were exclusively used for the time T, d) $\{C_{so}, C_{sub}\}$ represents the phone energy overhead for tag spoofing and subcarrier modes; note that the overhead in opportunistic mode, $C_{opp} = 0$, since the phone is expending this energy whether or not *EnGarde* is present, and e) $\{f_{opp}, f_{so}, f_{sub}\}$ are the fraction of time opportunistic, tag-spoofing and subcarrier modes are used within the interval T, where $f_{opp} + f_{so} + f_{sub} = 1$.

Our optimization problem can now be formulated as minimizing the overhead on the phone, where overhead is defined as the additional energy that the phone needs to use above and beyond what it is expending in the opportunistic mode, given the constraints that the total energy harvested in time T should be sufficient to get the buffer to B_{max}.

$$\begin{aligned} &\textit{min:} && Tf_{so}(C_{so} - C_{opp}) + Tf_{sub}(C_{sub} - C_{opp})) \\ &\textit{subject to:} && E_{opp}f_{opp} + E_{so}f_{so} + E_{sub}f_{sub} + B = B_{max} \\ & && f_{opp} + f_{so} + f_{sub} = 1 \\ & && f_{opp} > 0, f_{so} > 0, f_{sub} > 0 \end{aligned}$$

This linear optimization can be simplified using well-known approximation methods to run in real-time on *EnGarde*. We do not provide a complete algorithm in the interest of space. The intuition behind the approximation is that $f_i, i = \{so, sub\}$ should be chosen to maximize the harvested energy E_i while minimizing the cost C_i. So the ratio of $\frac{E_i}{C_i}$ determines the selection between tag-spoofing and subcarrier modes. Because opportunistic mode has zero overhead, it will be used whenever possible.

5.7 NFC Device Detection

The ability to detect the presence or absence of an NFC device in the vicinity of the phone is important in two ways: a) *EnGarde* can avoid disrupting legitimate interactions between the phone and an NFC device (smart tag, payment station, etc), and b) *EnGarde* can stop jamming when it detects that the offending device is no longer in the phone's vicinity, and is thus no longer a threat.

One approach to solving this problem would be to look at the message interactions to determine whether or not there is another NFC device present. The phone switches from discovery mode to active mode (or software tag emulation mode) once it starts communicating with another device in the vicinity. Since *EnGarde* has the capability to decode messages, it can detect a message that indicates the start of an interaction with another device.

However, this solution has a problem. If *EnGarde* is harvesting energy in any of the three modes, even if just harvesting opportunistically, it hampers the coupling between the phone and the external device. This means that we need to detect devices prior to communication occurring between them. Similarly, while we are jamming, we cannot decode messages to detect when the NFC device leaves the vicinity of the phone, and therefore when we should stop jamming.

Our solution to this problem has two key contributions: a) a reliable and fast NFC device detector that leverages changes in mutual coupling, and b) a dual-coil hardware design that includes a harvesting coil and a sampling coil that are tailored to different needs.

5.7.1 Mutual Coupling-based NFC Detection

Our key idea to detect the presence of an NFC device leverages the manner in which inductive coupling works when several coils are present. NFC coils operate using the property of electromagnetic induction *i.e.* one coil induces a voltage in the other coil (mutual inductance). If multiple coils are present in the vicinity of an inductor, then the mutual inductance is split across these two coils. Therefore, the voltage induced in each of the coils reduces. Our idea is to detect this change in voltage at the output of the rectifier, and use it as an indicator of the presence of another NFC device.

One drawback of such a detector is that nearby metallic materials that couple, may have the same effect on voltage. When a coil generating a magnetic field is brought near a conductive material such as aluminum, it induces eddy currents that reduce the amount of flux detected in *EnGarde*. However, we argue that false positives are not a significant concern since if *EnGarde* detects no NFC interaction for a time period, it can revert to harvesting mode.

To test this theory we attach a tuned coil and voltage regulator circuit to a Galaxy Nexus phone and bring tags of various technologies in proximity of the phone / harvester pair. In Figure 5.5, we plot the voltage across the rectifier. The plot shows two interesting observations. First, we see that the decrease in voltage is proportional to the amount of power the tag draws. A simple tag, such as an ISO 14443-A Charlie card transportation transponder, has a small impact, while a more complex tag, such as an ISO 14443-B EEPROM tag, has a much more noticeable impact. Second, we see that, as expected, other metallic objects (in this case a large aluminum plane) also causes large voltage changes.

To ensure reliable NFC device detection, we tune the detection threshold such that even a slight dip in the voltage compared to no tag being present causes *EnGarde* to backoff. To test our detector, we placed a set of tags (same as those used in Figure 5.5) in and out of the proximity of the phone and turned the phone's screen on and off. The results are over 100 such tag presence events, and we observe a detection accuracy of 95%, which shows that we only miss a small fraction of the cases. Note that even in these cases where a tag is not detected, *EnGarde* is still securing the phone since it is continuously listening for any message interaction that could be indicative of malicious behavior. The only downside of missed detection is a diminished user experience since the phone might need to be moved closer to the tag to ensure that *EnGarde* backs off and enables communication to occur.

5.7.2 Dual-coil Design

What should *EnGarde* do when an NFC device is detected in the vicinity? One option is to have a switch and detach the load from the coil, but in doing so, *EnGarde* loses the ability to listen to messages and decide when to jam based on message content.

Our key insight is that we can decode communications by using a small “sampling” coil that has fewer turns and is detuned to the carrier, and use a “harvesting” coil solely for harvesting and jamming purposes. The sampling coil would reduce the level of interference to be small enough not to impact communication between the tag and external device while still retaining the ability to decode messages.

To understand how well our dual-coil design works, we look at the cases when the coil is connected and disconnected from our harvesting circuit. With the harvesting coil disconnected, we measured an average communication latency of 20 ms across a set of ISO 14443-A, ISO 14443-B, and ISO 15693 tags. In all test cases, we found that the phone was able to read the tags even though *EnGarde* was physically present. We then connect it to our harvesting coil and repeat the previous experiment. We found that while harvesting power, tags had an increase in communication latency of 3 ms. We also found that in a handful of test cases (15% of the cases we tested), ISO 14443-B EEPROM based tags could not successfully read. This emphasizes the importance of the tag detection as described above.

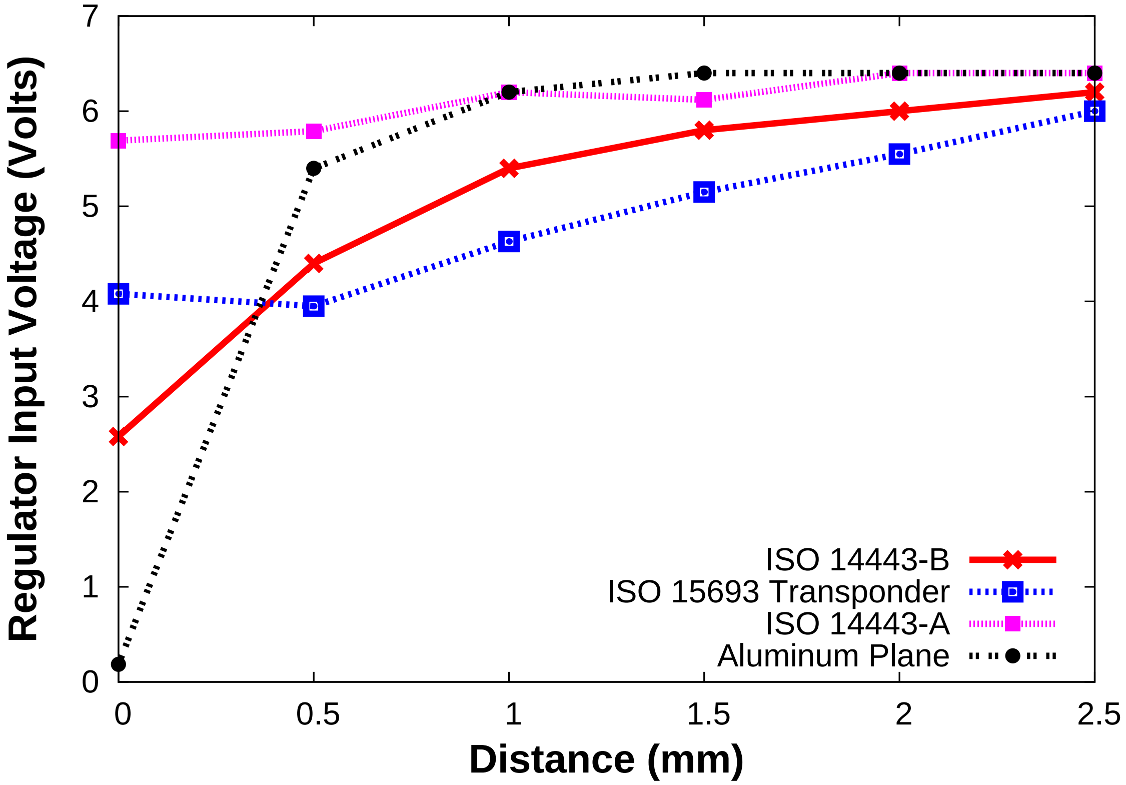

Figure 5.5. The presence of NFC transponders can be identified by observing a change in the output of a voltage rectifier; tag technologies that draw more power see a larger change in voltage. The presence of metal causes a huge change in voltage over a short range.

5.8 EnGarde Implementation

Figure 5.7 shows a prototype version of *EnGarde*; our current hardware implements all the design elements, except for pulse jamming, described in §5.5.1. The current prototype measures 2.0 by 2.6 inches, and is well within the form-factor of a typical smartphone. We believe that future revisions can shrink this even further. We now briefly describe the key hardware sub-components used in the prototype and describe its operation using a state machine abstraction that uses the hardware primitives to enable selective jamming.

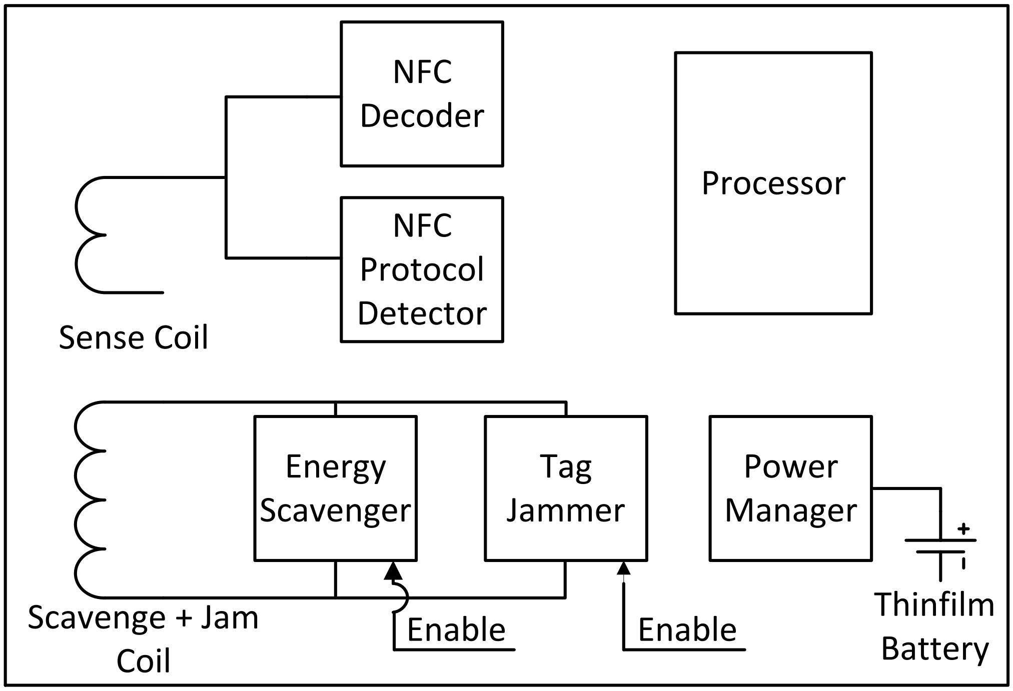

Figure 5.6. *EnGarde* is implemented across several subsystems that provide power, jam, and enforce a set of programmable blocking rules.

5.8.1 Hardware

The goal of our *EnGarde* implementation was to build a form-factor prototype that can actually be attached to the back of a mobile phone. We show how hardware subcomponents are interconnected in Figure 5.6.

The first key hardware element is a small "sensing" coil that is used to sense the magnetic field in vicinity of the phone. The NFC protocol detector module uses this coil's output to detect the active NFC protocol type. The NFC decoder block uses the sense coil's output and the Rx chain of a TI TRF7970A NFC reader; the reader is configured in software by the microcontroller to decode a particular RFID protocol. The sense coil's output is also used by the microcontroller for tag presence detection.

The next key design element is a tuned coil and a capacitor arranged in parallel; this coil is used for both jamming and energy scavenging. The jamming module is

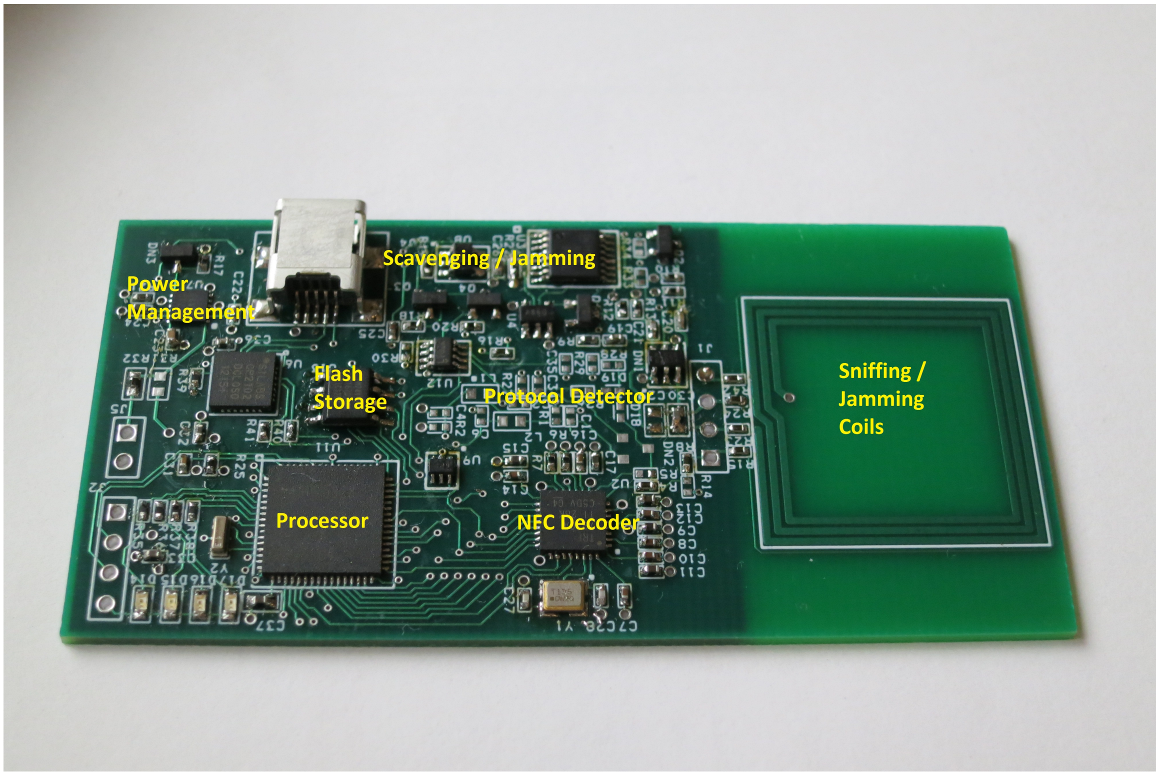

Figure 5.7. *EnGarde* is implemented as a low-profile printed circuit board with small form factor.

controlled by the onboard microcontroller and may be enabled or disabled depending on security or harvesting needs. One important characteristic of this circuit is that it fails safe if *EnGarde*'s energy buffer is depleted – this enables protection against malicious transactions and also improves the energy available via scavenging (Section 5.6).

A critical element of our hardware design is the energy scavenging module used to harvest energy from active reader transmissions. This module can be disabled to reduce the impact on the phone's NFC communications (Section 5.7). Since the microcontroller needs energy to boot, the scavenging module, much like the jamming module, defaults to active mode in the event that the energy buffer is depleted. Since jamming is based on load modulation, jamming is automatically disabled when the scavenging module is disabled.

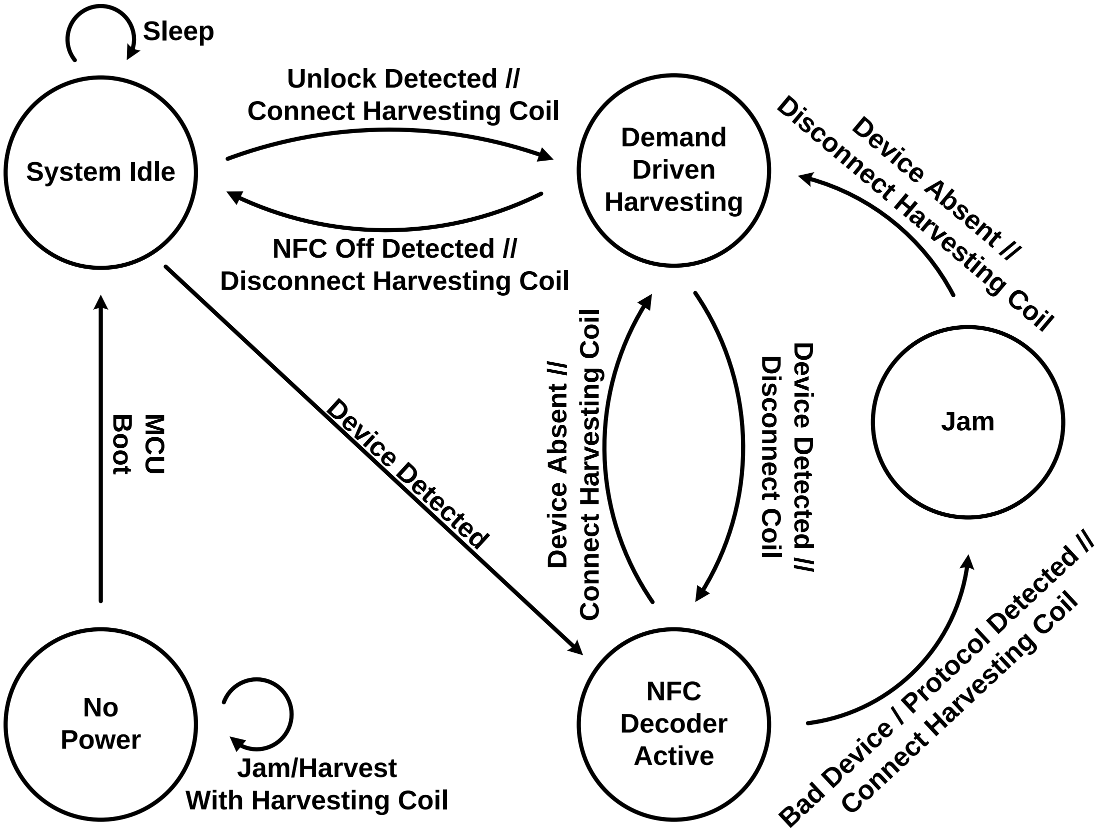

Figure 5.8. *EnGarde* switches between several different operational states to harvest energy, detect NFC devices, decode messages, and jam malicious NFC devices.

To condition harvested energy for storage, a MAX17710 battery manager chip manages the charging of the on-board Thinergy MEC201 1 mAH thinfilm battery. The use of a diminutive thin-film battery is particularly compelling, since *EnGarde* needs to minimize thickness in addition to length and width.

Finally, an MSP430F2274 16-bit low power microcontroller manages the various sub-components of *EnGarde*. This particular microcontroller was chosen because it has an ADC that enables tag detection, has low power operating modes and can transition between power states quickly.

5.8.2 State Machine

EnGarde follows the state machine shown in Figure 5.8. When *EnGarde* is drained of power or when its energy reserve is depleted, the device is in the state *no power* where the microcontroller is not active. However, our device fails safe, so the jamming module is used in conjunction with the tuned coil in this mode of operation. Whenever an NFC signal is seen either from the phone, or an external device, this circuit simultaneously jams the signal while increasing power transfer from the reader. After accumulating sufficient energy, control is relinquished to *EnGarde*'s microcontroller.

After the microcontroller boots, it enters a low power state referred to as *System Idle*; while in this mode, the microcontroller listens for interrupts from the sensing coil / tag presence detector. If an NFC device is found to be present, it uses the protocol detector module to decode reader-side messages.

If an external device has entered the vicinity of the mobile phone, *EnGarde* also switches on the decoder and enters its highest power state, *NFC Decoder Active*, where it decodes NFC transactions; before entering this state *EnGarde* detaches its harvesting coil and listens with the sensing coil.

After activating its NFC decoder, *EnGarde* decodes messages sent by the phone, as well as messages coming from the external device. It goes through its list of blacklisting rules, and if there is a match, it enters state *Jam*. If no such blacklist entry is matched, *EnGarde* will continue to listen to message exchanges until the external device exits the vicinity of the mobile phone at which point it reverts to demand-based harvesting using its tuned coil.

While in state *Jam*, *EnGarde* continuously generates a subcarrier that makes communication with external passive devices impossible for the phone to decode. If *EnGarde* detects a message from an active external device, as in peer-to-peer mode, it can generate an active subcarrier pulse for two bit durations per frame to disrupt active communications. As in the previous case, *EnGarde* continues to jam until it detects the external device has left the vicinity and returns to state *Demand Driven Harvesting.*

5.8.3 Blocking Rules

In addition to harvesting sufficient power and jamming effectively, *EnGarde* also needs to know when to block and when not to block particular NFC message exchanges. Since NFC is an emerging technology rather than a well established one, our work should be viewed as a preemptive mechanism that addresses potential threats rather than a reactive one that addresses existing attacks (NFC is one of the threat predictions for 2013 released by McAfee Labs [57]). Thus, instead of focusing on particular attacks, we provide a framework under which a rich sets of rules may be constructed.

Block protocol. The first level of rule-based filtering occurs in hardware – tag responses are sorted according to their respective protocols by using information provided by the protocol detection circuit presented in Figure 5.3 and subsequently handled by protocol-specific code. The highest granularity control a user has over *EnGarde* is to block all messages belonging to an entire protocol. An example where this might be used is where a more concerned user would like to prevent their phone's unique NFC ID from being read, so it blocks the entire ISO 14443-A protocol.

Block tag IDs. The next level of rule-based filtering occurs during the anti-collision phase of a particular protocol's anti-collision message exchange. During each of the protocols, a tag responds with a unique or pseudo-unique identifier that belongs to a particular tag. Thus, a user can block some subset of tag IDs that could correspond to a particular manufacturer or set of compromised tags.

Message content. The finest granularity of rule-based filtering occurs based on the content of the messages themselves. These rules are specified in a software graph structure using Aho and Corasick's keyword tree [8]. This structure has been used

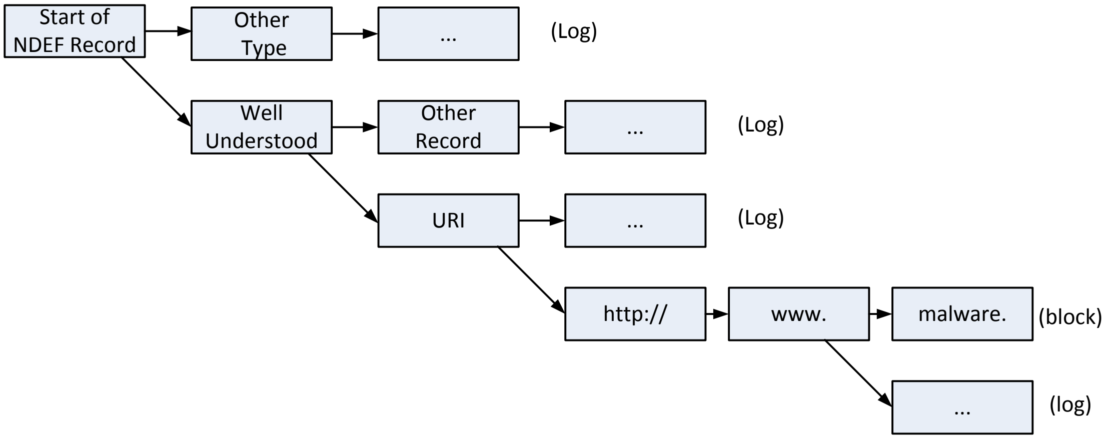

Figure 5.9. *EnGarde* implements a set of flexible rules based on a keyword tree structure specified in software. In this example, all well-understood NDEF messages corresponding to a URI type with prefix "http://www.malware.*" are blocked. Other sniffed tag responses are logged.

widely in pattern search algorithms and is also used in a popular packet tracing program, Snort [30]. We illustrate an example using this approach in Figure 5.9 – this rule definition logs all NDEF messages and proactively blocks those that correspond to well-known NDEF messages of the type URI that start with the substring "http://www.malware."

In our current implementation, these rule sets are decided at software compile time and programmed into the Microcontroller using a wired JTAG interface. In principle, these rules can be updated via NFC from a secure application, however we have not yet implemented this functionality on our current hardware.

5.9 Experimental Evaluation

Our evaluation covers three major aspects of our system: a) how effective is our jamming scheme in blocking interaction between the phone and other NFC devices, and b) a demonstration of *EnGarde*'s capability to perform targeted jamming of malicious tags while allowing benign ones to interact with the phone, and c) how well does our scavenging scheme perform over a long-term phone usage dataset,

5.9.1 Jamming Effectiveness

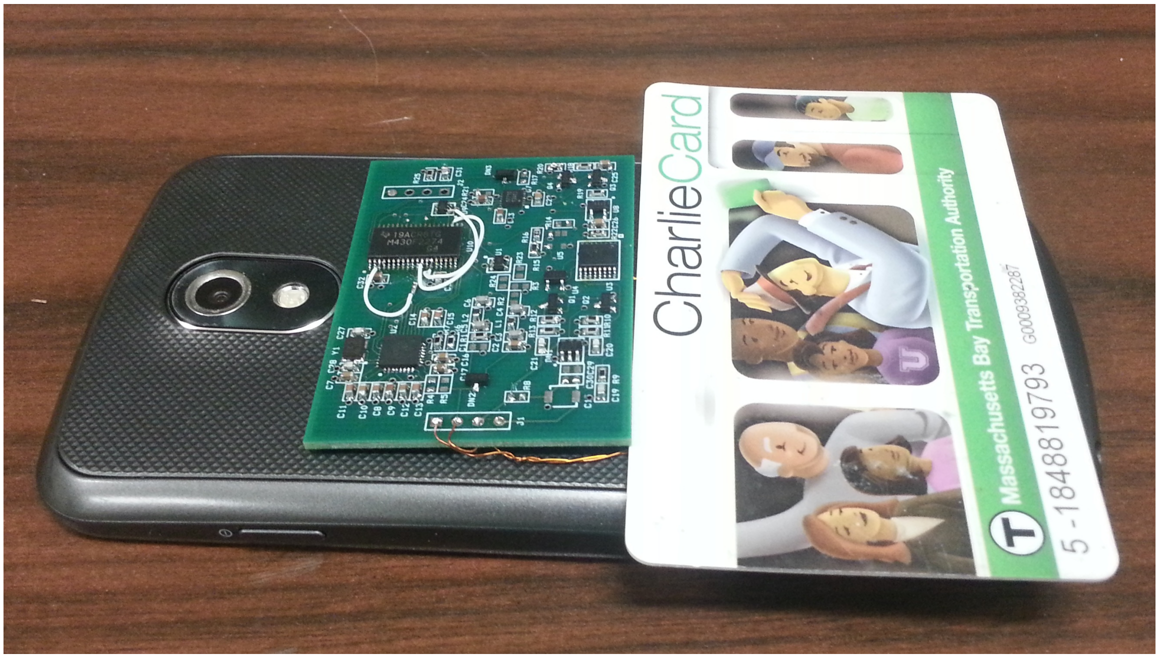

Figure 5.10. Our *EnGarde* prototype meets the form factor needs required for semi-permanent attachment to a mobile phone. Here, we show it on the back of a Galaxy Nexus

An understanding of how effectively *EnGarde* is capable of jamming NFC devices is critical towards showing that it sufficiently protects a mobile phone from external NFC threats. In particular, we want to understand what types of tags can circumvent our jamming signal and which types of tags the phone might be more vulnerable to. Figure 5.10 shows an image of our jamming effectiveness test setup.

Jamming malicious tags. We installed *EnGarde* on the back of a Galaxy Nexus phone and moved several different tags towards the phone, such that they were in direct contact with the back of the phone. The types of tags that we looked at were: ISO 14443-A. ISO 14443-B, ISO 15693, and a TI TRF7970 operating in ISO 14443-B tag emulation mode. We found that *none* of these tags could successfully communicate with the phone while the subcarrier was active. While we don't want to make any claims that communication with the phone is impossible, we haven't been able to find a tag that can get past our jamming signal.

Jamming malicious readers. Another important jamming on *EnGarde* is when an NFC reader, such as a mobile payment station, tries to read the mobile phone while in card emulation mode. We program a TRF7970 as a general purpose NFC reader, sending queries at its highest power level (200 mW). We found that when *EnGarde* is installed on the back of the phone, we effectively block 100% of the phone's ISO 14443-A response.

***EnGarde* Versus RFID Guardian [66].** While a direct comparison against active jamming approaches, such as the RFID Guardian, would require designing another hardware platform, we briefly discuss the key differences. NFC Guardian actively generates two 424 KHz sub bands around the 13.56 MHz, which can block NFC tags within a half meter radius. Since we are only interested in protecting the mobile phone, we are able to passively generate a similar signal at negligible energy cost. For example, in the above experiments, if we change the setup by moving *EnGarde* some distance away from the phone, and place a tag directly on the back of the phone where *EnGarde* would normally be installed, we find that *EnGarde* blocks all communication provided that it is within 1.0 mm of the phone, but has limited effect after that distance. Thus, our jamming is extremely targeted, which improves our efficiency.

5.9.2 Targeted blocking of malicious interactions

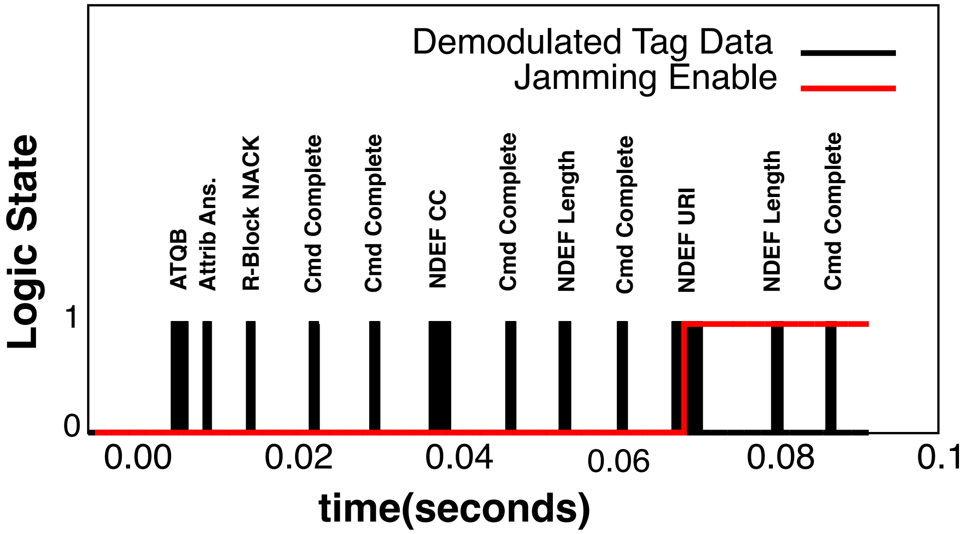

Figure 5.11. *EnGarde* monitors the messages sent from an emulated ISO14443-B tag, detects a malicious URL type and jams all subsequent communications

We now look at a case where there is a malicious tag and other non-malicious ones, and show that *EnGarde* can be programmed with blacklisting rules that allows real-time decoding of NFC interactions and targeted jamming of malicious ones. Specifically, we look at a case study where *EnGarde* is programmed to block a particular set of URLs on an ISO 14443-B NDEF tag.

In our study, we program a TRF7970A evaluation module to behave as an emulated ISO-14443-B NDEF tag. This emulated tag approaches a Galaxy Nexus phone; in a scenario when *EnGarde* is not present, the phone uses the discovery phase to identify a tag is present. The phone then sends a series of messages that select the NDEF message stored on the emulated tag, leading up to where the tag sends its reply that contains the requested NDEF message.

After successfully decoding the NDEF response, the phone takes action according to the contents of the NDEF message. In this case, the NDEF message has its TNF field set to 0x01, which means that it is a well understood type. After checking the

ID type field, it finds that this message is a URI type message that contains a URL, Phone Number, or other address from a variety of different protocols. In the first byte of the NDEF record, the phone finds the value 0x01, which corresponds to the string “http://www.” The subsequent characters correspond to the rest of the URL “malware.com”. The phone automatically loads this webpage in its web browser.

Next let’s look at the case where *EnGarde* is installed on the back of the phone. *EnGarde* decodes all of the bits corresponding to the emulated tag’s reply; we show the bits actually decoded by *EnGarde* the time series shown in Figure 5.11. We can see that the tag first responds to the phone’s REQB discovery message with an ATQB that contains the tags pseudo unique ID. After identifying the emulated tag, the phone sends an Atrrib message that indicates this particular tag has been selected for further communication, after which the tag replies with a standard Attrib answer message.

EnGarde next observes the sequence of messages corresponding to the NDEF message selection. After observing that the tag has sent its capability container (NDEF CC) and subsequent NDEF record length value, *EnGarde* knows where to find the NDEF message. It looks in the byte location that contains the URI identifier code 0x01, which corresponds to “http://www/” and immediately activates its subcarrier jamming circuit to block the rest of the message. It is also important to note that *EnGarde* will parse individual characters if the URI identifier code contains 0x00, which means that no compressed prefix is applied to the URI. If the characters correspond to “http://”, again the rest of the message is blocked. We tried to get the phone to read the tag 20 times and the phone was never successful.

Finally, we show that *EnGarde* allows transactions that don’t satisfy our blocking rules. To prove this, we use another emulated tag, but program it with the URL “https://www.cs.umass.edu”. In this case, the URL is not blocked and the page opens in the phone’s web browser. Again, we found that this was robust to various placements of the tag. While we did not quantify the impact *EnGarde* had on the benign tag's read range, it wasn't noticeably different than during a typical NFC interaction.

5.9.3 Scavenging performance

Our final evaluation looks at the performance of the scavenging subsystem. Since this evaluation depends on the actual time for which the phone is unlocked, and the duration between unlock events, we look at what impact these dynamics have on *EnGarde*. To accomplish this, we look at a set of traces provide by the LiveLab project at Rice University[76]. These traces were collected from 35 users over the span of a year and contain the screen unlock data needed to understand variability in available harvested energy.

Simulation Parameter	Value
Quiescent Power Consumption	38.8 μW
Reader Power Consumption	32.7 mW
Opportunistic Power Harvesting	3.03 mW
Semi-Opp Power Harvesting	10.0 mW
Subcarrier Power Harvesting	26.0 mW
Thin-film battery capacity	14.4 J
Duration of NDEF exchange	0.56 seconds

Table 5.5. A summary of the parameters used in our simulation study

Harvesting Study: Before looking at the behavior of the adaptive algorithm presented in §5.6.2, we first seek an understanding of the performance tradeoffs given the screen unlock interval dynamics present in the Livelab traces. In particular we ask the question: *What performance can EnGarde achieve despite variability in the amount of energy harvested from the mobile phone and what impact does this harvesting have on the mobile phone's battery lifetime?* To answer this question, we look at how many messages *EnGarde* can sniff on a given day, as well as the impact

EnGarde has on a mobile phone's battery irrespective of energy storage limitations on *EnGarde*. The CDFs we plot in Figures 5.12 and 5.13 are computed across all days and all users.

In Figure 5.12, we show *EnGarde*'s harvesting potential in terms of the number of NDEF interactions we sniff. To get a sense of how much each of these individual messages cost in terms of energy, we looked at the amount of time such an NDEF interaction takes to complete by measuring such an interaction between an Android Galaxy Nexus and an ISO 14443-A MiFare DESFire transponder card. We found that these interactions take 0.56 seconds, averaged across 20 trials; given the power consumption of *EnGarde*'s reader hardware, each interaction consumes 18.3 mJ of energy. Our three naive harvesting strategies – Opportunistic, Tag-Spoofing, and Subcarrier – are each capable of sniffing large numbers of these interactions. For example, the opportunistic harvesting strategy is capable of sniffing 100 such interactions for 86% of the days across all traces and 2413 interactions for 50% of the days; performance vastly improves when using the other two naive strategies.

Although *EnGarde* is capable of harvesting sufficient energy to sniff a substantial number of NDEF interactions, this comes at a cost. In Figure 5.13 we show the impact of NDEF sniffing on the phone's battery consumption for each naive harvesting strategy (We note that the x-axis % battery consumed is truncated to 100% because of the impact of several outlying data points; our goal is to show the impact on a single battery given no recharging). We note that the opportunistic-only strategy consumes 8.3% of the phone's battery in half the cases, while the more aggressive subcarrier-only strategy uses 95.7% of the phone's battery in half the cases. While we have shown that sufficient energy may be harvested to sniff substantial numbers of NFC transactions using all of the strategies, these naive strategies have a large impact on the phone's battery life. Thus, we need to show how this energy may be used for realistic workloads while also taking into account *EnGarde*'s energy buffer constraints.

Harvesting Strategy	% Missed NDEF interactions / day	% Phone battery consumed / day
Opportunistic	1.76	0.096
Tag-spoofing	0.68	0.281
Subcarrier	0.37	1.29
Adaptive	0.42	0.145

Table 5.6. Given a daily workload of 100 interactions with an NDEF tag, *EnGarde* is able to sniff most of these transactions with negligible miss-rate and with little impact on the host phone's battery across all users and all days by using an adaptive harvesting strategy.

Adaptive Harvesting Simulation: As a consequence of the issues raised in the prior study, we ask the question: *Does our demand driven harvesting algorithm achieve low NFC sniffing miss rates with a limited energy buffer, while simultaneously having minimal impact on its host phone's battery lifetime?* We answer this question through a trace-driven simulation of our adaptive harvesting algorithm; our simulator implements a complete *EnGarde* state machine, shown in Figure 5.8, and uses the measured values shown in Table 5.5. We compare the demand-driven harvesting algorithm against exclusively using one of the other three strategies. In the previous results, we showed *EnGarde*'s potential by removing restrictions on energy storage; however, in our simulation study, we simulate the behavior of a thin film battery with 14.4 Joules of energy storage capacity (same battery as used on *EnGarde* implementation). We look at *EnGarde*'s performance for a single workload that corresponds to 100 simulated NDEF interactions whose energy consumption is spread uniformly throughout the day; we do not simulate these interactions as discrete events, as no traces are currently available that show NFC user behavior. This workload is de-

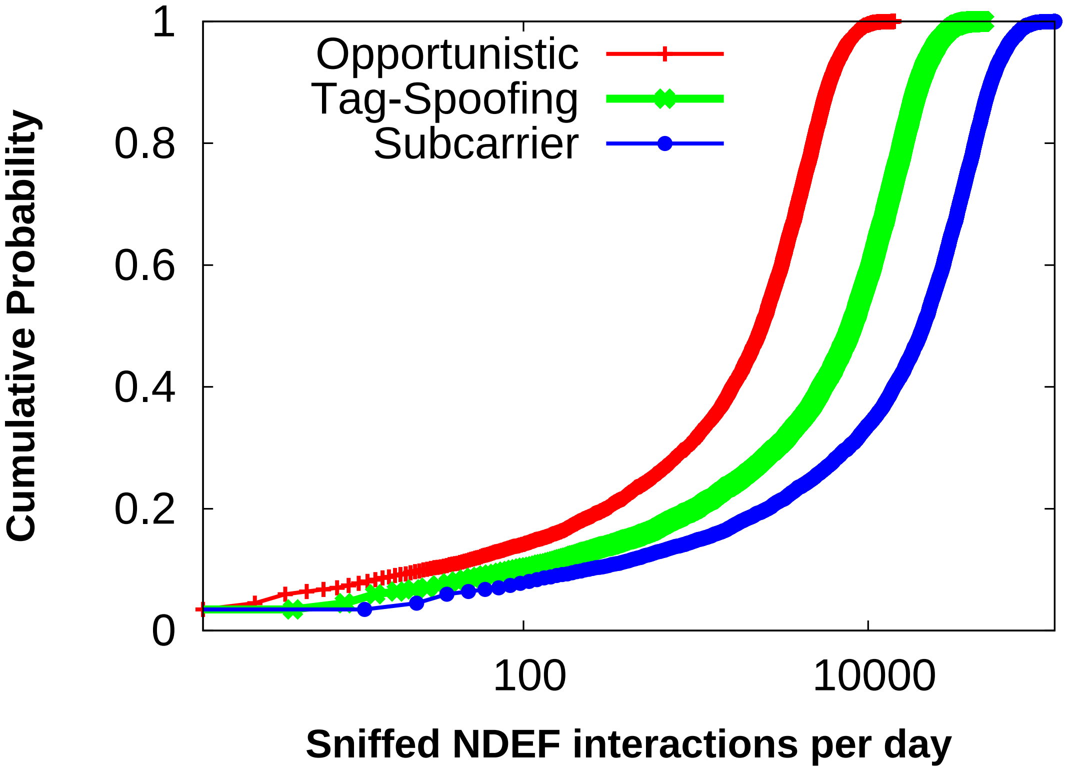

Figure 5.12. *EnGarde* harvests sufficient energy to sniff a large number of NDEF interactions between a mobile phone and passive transponder card.

signed to be a reasonable approximation for a user that frequently interacts with NFC devices throughout the day.

We summarize the results of this study in Table 5.6 and show performance results in terms of the average % of simulated NFC interactions that *EnGarde* could not sniff given trace dynamics across all days and users; we also show the corresponding impact on the phone's daily % battery capacity. We note that the demand driven harvesting algorithm achieves a miss rate of only 0.42%, which is very close to the 0.37% miss rate achieved by the most aggressive, subcarrier-only strategy. We also note that the demand driven harvesting algorithm uses only 0.145% of the host phone's battery, which nearly matches the 0.096% consumed by the most efficient, opportunistic only strategy. These two metrics together show that our adaptive harvesting algorithm is

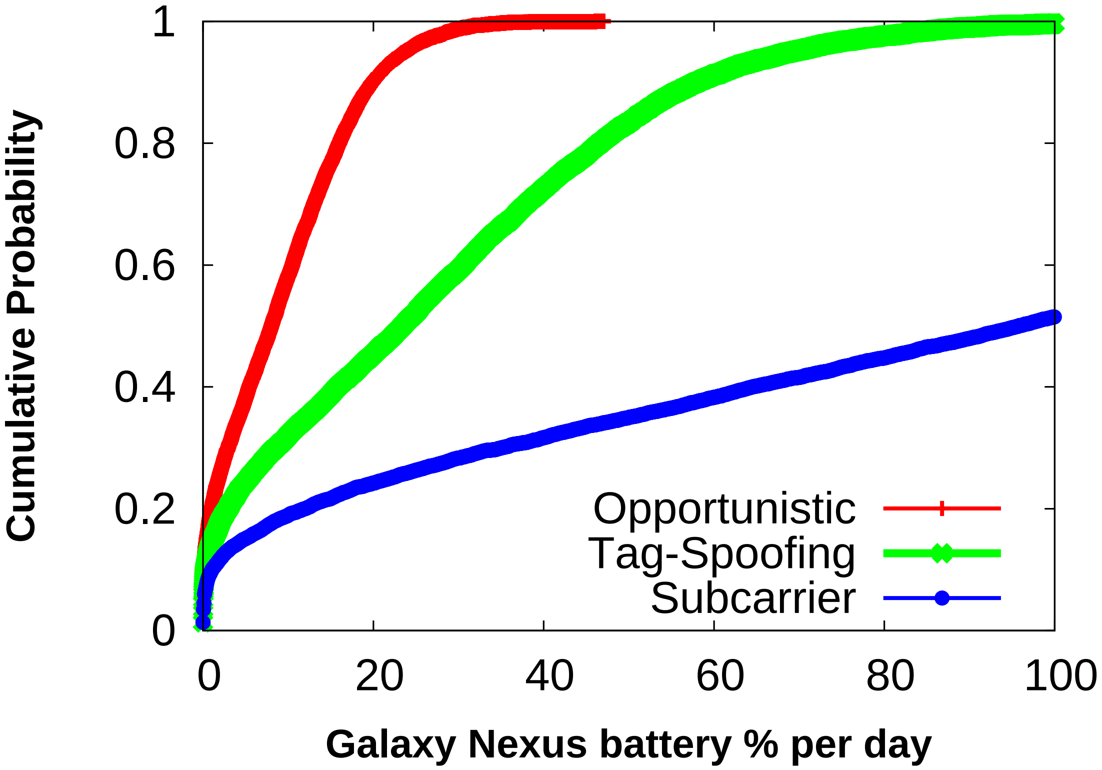

Figure 5.13. The energy harvesting strategies available on *EnGarde* use considerably different amounts of a phone's battery. The opportunistic harvesting mode consumes an average of 10% of the phone's battery, tag-spoofing consumes an average of 24%, while subcarrier mode would consume the phone's entire battery.

effective at achieving low miss-rates while having minimal impact on a mobile phone's battery for a reasonably approximate workload.

5.10 Related Work

Most similar to our work is the RFID Guardian [66]. The RFID Guardian monitors and jams specific NFC communication sessions in its vicinity. This longer range performance comes at a cost in form factor and power consumption. While useful for monitoring and protecting arbitrary sets of readers and tags, *EnGarde* is considerably more targeted and is designed to protect an individual mobile phone.

The Proxmark RFID tool [27] has been used extensively in NFC security research. It has the capability of decoding arbitrary protocols with an FPGA and additionally, it can emulate a tag. Since it uses an FPGA to decode and emulate tag responses, it can be programmed to decode any potential protocol. Two major drawbacks are its size and power consumption – while a valuable tool for debugging and security analysis, it is not suitable for continuous use on a mobile phone.

Another way to harvest energy from a mobile phone is through the audio interface [49]. Much like our NFC energy scavenger, the audio jack is universal across different phones. While a wired connection can harvest energy more efficiently, we instead opt to power *EnGarde* from the same power source as the attack surface.

Selective jamming devices are of particular interest in the area of implanted medical devices (IMD). One application is the implementation of zero power defenses [42]. *EnGarde* behaves much like a zero-power defense in that it generates a jamming signal completely passively. More recently, a non-invasive approach towards IMDs was proposed [35]. While the proposed IMDShield has parallels to our approach towards non-invasive jamming, they use a power-hungry software radio while ours operates entirely passively.

Other hardware and software systems that perform the task of packet filtering share much in common with our approach towards selective blocking of malicious NFC messages [21, 31]. While our approach towards filtering messages based on their contents is similar, we can relax our hardware requirements, since current NFC data rates are significantly slower than Gigabit Ethernet.

Security is also of critical importance to mobile health. One approach towards providing security for on-body sensors is to provide a security proxy with which they communicate through [79]. While not currently implemented as such, *EnGarde* could potentially be used as a similar security proxy.

Since NFC is a relatively new technology, new vulnerabilities are constantly being exposed [50, 58, 57]. *EnGarde* addresses these issues by providing a flexible set of security features that protect a mobile phone while remaining decoupled from platform vulnerabilities.

Finally, there have been significant efforts in securing RFID technology at the software level [11], and to place secure hardware elements in mobile phones [81]. We view our work as complementary. *EnGarde* serves as a hardware firewall that can augment software protection mechanisms to protect a mobile phone from potentially devastating attacks via NFC.

5.11 Conclusion

In this chapter, we have outlined a practical, fully functional hardware shield, *EnGarde*, for phones that can intelligently protect phones from malicious NFC interactions while letting benign ones pass through. Our design is entirely passive thereby making our form-factor small enough to be placed as a patch on a phone or even integrated within a phones case. Perhaps the most compelling aspect of *EnGarde* is that it is widely deployable as-is on NFC mobile phones that are emerging in the market, thereby making our system market-ready.

While this paper focuses on jamming, *EnGarde* has immense potential in forensic analysis of NFC interactions. There is currently limited understanding of how NFC interactions work in practice — what information is sent in the clear? how do different phones implement mobile payments? and so on. *EnGarde* is a powerful tool that can log any NFC interaction that it decodes (including those in the vicinity of the phone), which can be used to perform such analysis. We defer such analysis to future work.

More information regarding *EnGarde* is available at: `http://sensors.cs.umass.edu/projects/engarde`

CHAPTER 6

WIRELESSLY POWERED BISTABLE DISPLAY TAGS

Another application scenario for passive devices that harvest energy from HF RFID equipped mobile phones is where the passive device is intended to be largely independent from the phone and only brought into the vicinity of the device for occasional interactions. In the prior chapter, we described a security peripheral that is opportunistically charged while the phone's display is unlocked. When considering applications that do not offer such opportunities for harvesting, we must reconsider the design of such a device. While it is true that this problem could be partially addressed by elongating interactions between the phone and a passive device to provide more harvested power, this approach would be unacceptable in terms of usability.

This chapter describes a bistable display tag that, from an energy standpoint, is capable of perpetual operation. A commercial off-the-shelf NFC-enabled phone generates RF signals carrying both the information and energy necessary to update the display. After the update is complete, the display continues to present the information with no further power input. We present one example implementation, a companion display for a mobile phone that can be used to capture and preserve a screenshot. We also discuss other potential applications of energy neutral bistable display tags.

6.1 Introduction

Despite significant reductions in the power consumption of sensing, computation, communication, and storage over the past decade [48, 77], one system component is currently dominating the overall energy budget of mobile phones – the display. A mobile phone can efficiently handle many tasks in the background while in a user's pocket, but every time a user interacts with the mobile phone, this results in a large amount of energy expenditure used to present the user data. The charactestics of these power hungry displays result in devices with large batteries that need to be recharged at least daily and more frequently during periods of heavy useage.

One new technology that has addressed this limitation is the bistable display [24]; this type of display has the unique property of perpetual operation with zero power required to maintain a rendered image. While no energy expenditure is required to maintain an image, each display update requires energy to change pixels from one state to another. In this chaper, we present an elegant solution to this limitation by providing a pixelated e-paper display that is both powered and updated wirelessly. To provide power and communications, we use near field communications (NFC) to power and communicate with the propose display.

The electronic paper display tag lies dormant in a low-power sleep state, waiting for updates; when an NFC-equipped mobile phone approaches the display tag, it provides the energy needed to flip display states, as well as the indivdual pixel values that define the new image. After the energy and data transfer completes, the display tag continues to display the new information, even after the phone's display is deactivated. These characterstics enable an application we call the "mobile phone companion display", shown in Figure 6.1. When used in this way, the display tag can lift the energy burden of the phone for display maintenance by allowing the phone's display to remain off or increase the total screen real-estate available while the phone's display remains on.

NFC encompasses multiple 13.56 MHz radio frequency identification (RFID) standards including ISO 14443, used in this work. The NFC reader-to-tag link makes use of inductive coupling for both data and energy transfer to tags which often have no

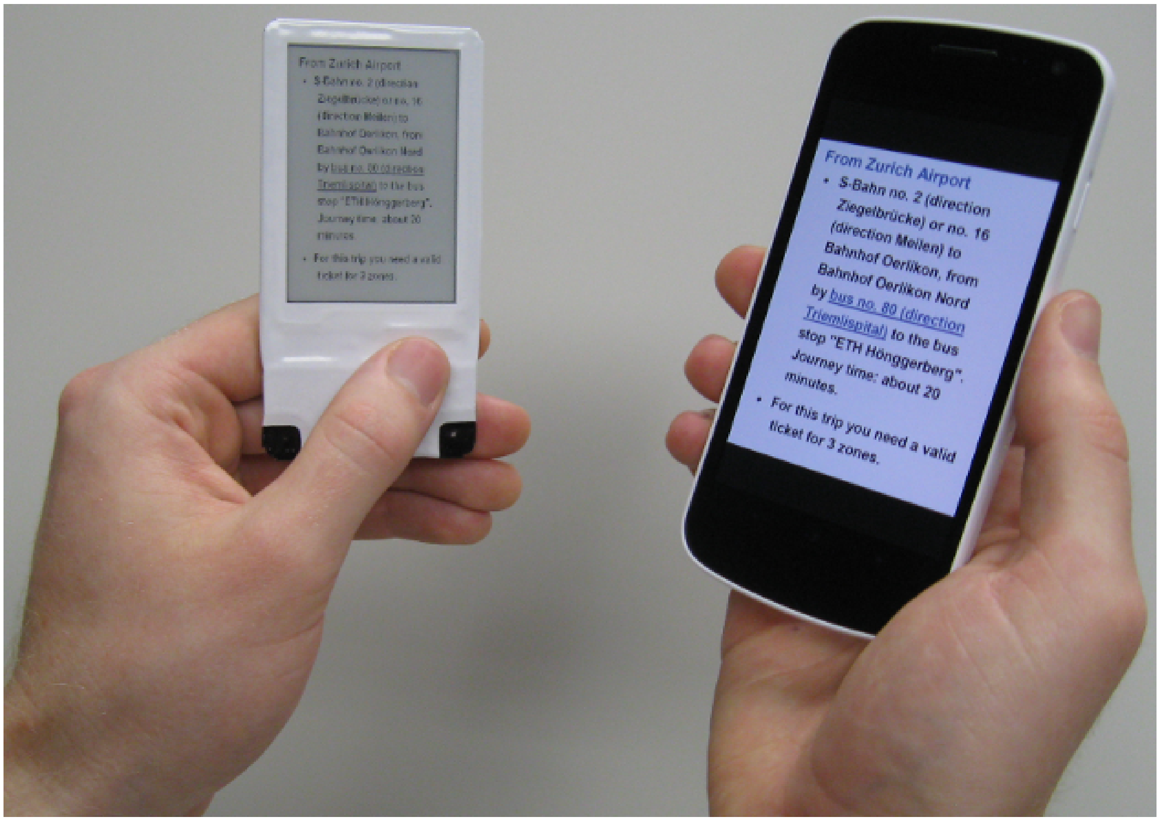
From Zurich Airport
• S-Bahn no. 2 (direction Ziegelbrücke) or no. 16 (direction Meilen) to Bahnhof Oerlikon, from Bahnhof Oerlikon Nord by bus no. 80 (direction Triemlispital) to the bus stop "ETH Hönggerberg". Journey time: about 20 minutes.
• For this trip you need a valid ticket for 3 zones.

Figure 6.1. One application of the display tag is a companion display for a mobile phone. The pixelated NFC display tag on the left mirrors the travel directions shown on the screen of the phone on the right. The power and data required to update the display are provided by an NFC connection between the tag and phone.

onboard energy source (passive tags). Inductive coupling is implemented by what is essentially an air-core transformer between the reader and the tag.

There are a number of commercial products involving display tags making use of RFID for communications, mainly targeting industrial and logistics applications. Omni-ID's Visual Tagging System [6] consists of an ultra-high frequency (UHF) RFID tag with an LCD display. The Smart Tag, by AIOI Systems [5], is an RFID tag with an e-paper display that uses NFC for communications. Neither of these products are wirelessly powered, resulting in a bulky form factor and a limited lifetime due to battery requirements. While RFID has been extensively investigated for use in user interfaces and smart object applications [56], the number of academic publications which explore RFID-based displays is small. One research prototype demonstrated an RFID-powered e-paper display for a smart card applications [53], but the segmented e-paper display used could only display few numbers or letters.

Our contributions are as follows: In this chapter we present the first pixelated e-paper display tag that uses NFC for power and communications. We evaluate how the e-paper performs in the energy constrained wirelessly-powered scenario. We present a mobile phone companion display application. Finally, we introduce the NFC Wireless Identification and Sensing Platform (NFC-WISP), on which the display tag is based.

The NFC-WISP is a software defined passive 13.56 MHz RFID tag. The NFC-WISP is based on a low-power MSP430 microcontroller (Texas Instruments), making the display tag easily re-programmable for future research and development efforts. The NFC-WISP was inspired by the WISP, a passive, software-defined 915 MHz tag for sensing applications [68].

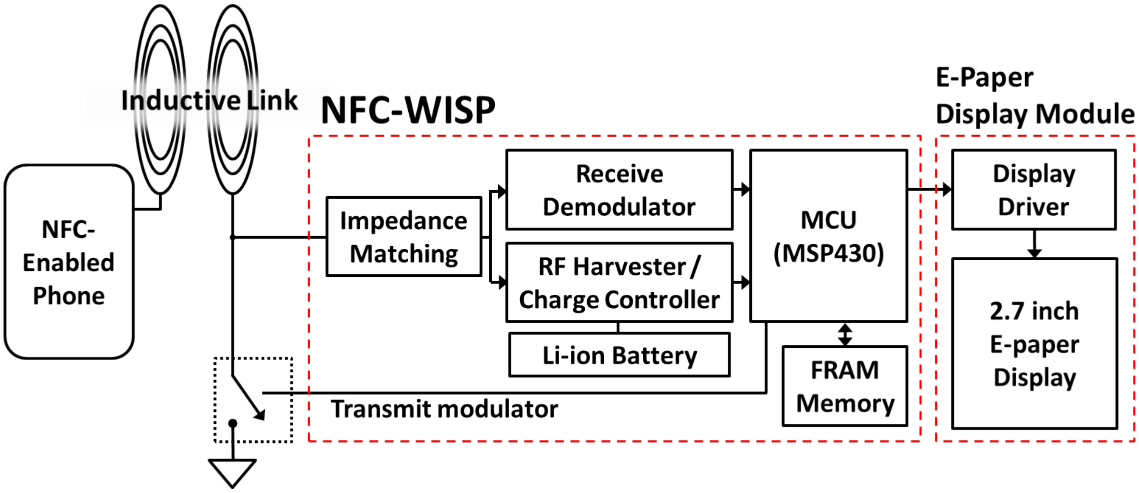

Figure 6.2. Display tag hardware block diagram.

6.2 System Design

A hardware block diagram of the display tag is shown in Figure 6.2. An MSP430 microcontroller is used to implement the software defined NFC communications, and is used to control the e-paper driver chip embedded in the 2.7 inch e-paper display (Pervasive Displays). The e-paper driver IC is used to generate the 15V needed to flip the display state and address the indivudal pixels of the display. An onboard 2 Mbit nonvolatile ferroelectric (FRAM) memory (Ramtron FM25V20) allows for up to 20 images to be stored for future retrieval. Nexus S and Galaxy Nexus (Samsung) phones with Android 4.2.2 (Jelly Bean) operating system are used as NFC readers.

6.2.1 Physical Design

The display tag integrates the NFC-WISP and an e-paper display on a compact 4-layer FR4 circuit board, shown in Figure 6.3. Display tag dimensions are: 100 mm by 50 mm, with a maximum thickness of 3 mm. All components are placed on one side to reduce thickness. Two tactile buttons provide a simple user interface.

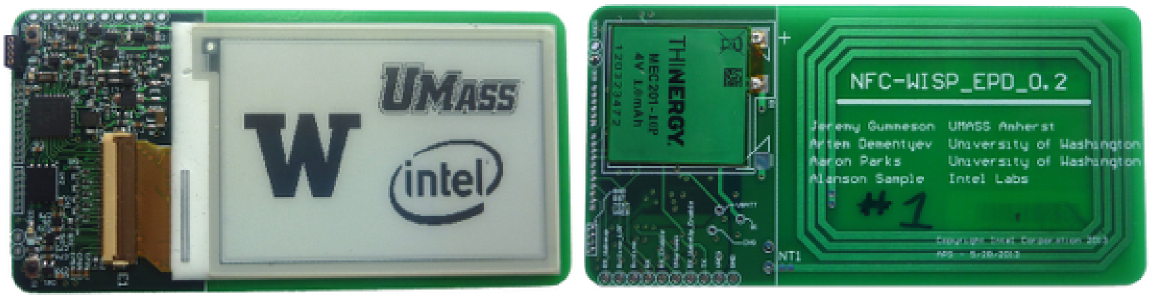
UMASS
W
intel
THINERGY.
MEC201-10P
4V 1.0mAh
120323472
NFC-WISP_EPD_0.2
Jeremy Gummeson UMASS Amherst
Artem Dementyev University of Washington
Aaron Parks University of Washington
Alanson Sample Intel Labs
#1
Copyright Intel Corporation 2013
APS - 5/28/2013
NT1

Figure 6.3. Display tag without the plastic case: front and back

6.2.2 Wireless Power Harvesting

To utilize the incoming radio frequency (RF) energy, the MAX17710 (Maxim) harvester IC is employed. The primary purpose of this IC is to boost the incoming RF voltage to the 3.3 V necessary for powering the display and the microcontroller. Furthermore, the harvester provides over-voltage protection and a battery charge controller. A rechargeable Lithium-ion battery (Thinergy MEC201) with 1 mAh capacity stores the harvested energy. The thickness of the battery is 0.17 mm. The battery buffers the energy to update the e-paper display, since more power is required to start up the display's boost converter than the RF harvester can deliver instantaneously. Furthermore, the battery stores the excess energy harvested during the NFC transaction, allowing for display updates away from the phone. The system does not fundamentally require a battery; a capacitor could instead be used for energy storage.

6.2.3 NFC Communications Hardware

Communication from the reader to the NFC-WISP is implemented through amplitude-shift keying (ASK) of the 13.56 MHz RF carrier field, as per the ISO 14443 specification. The tag's demodulator consists of a low-power analog comparator circuit that detects the ASK fluctuation in the carrier, converting it to a rail-to-rail logic level signal which then enters the MSP430 microcontroller for decoding.

The tag-to-reader uplink is accomplished by modulating the RF field produced by the reader. This technique, known as load modulation, is commonly used in RFID systems to eliminate the power consumption and complexity of implementing an active radio on the tag side.This modulation is accomplished through controlled shorting of the NFC-WISP's receive coil, which in turn produces detectable changes in the reader's transmit coil voltage.

6.2.4 Software Defined NFC Communications

The communication software for the NFC-WISP is designed around the 13.56 MHz ISO 14443-4 type B RFID protocol. ISO 14443 B was chosen because of its high reader-to-tag data rate of 106 kbps, and because it is supported by Android NFC-enabled phones. The NFC-WISP also supports ISO 15693 with 1-out-of-256 coding, but this is not used due to its slower data rate of 1.6 kbps.

During reader-to-tag communications, bits are encoded with non-return-to-zero (NRZ) line coding, where logical ones are represented by the unmodulated 13.56 MHz carrier and zeros by the modulated carrier. The NFC-WISP sends data back to the phone by modulating its receive coil load with an 847 KHz subcarrier. Bits are coded using binary phase-shift keying (BPSK), where a change in logical state is determined by 180 degree phase shifts in the subcarrier. Because each bit is represented with eight periods of the 847 KHz subcarrier, the resulting data rate is 106 kbps.

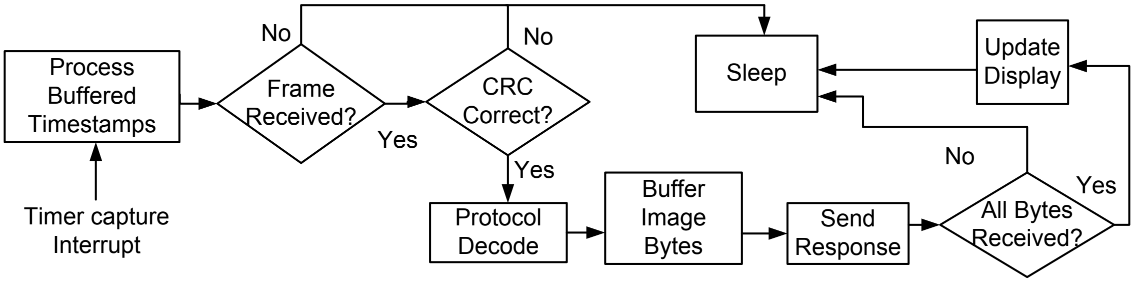

Figure 6.4. The NFC-WISP firmware flow diagram. While waiting for data from a phone, the NFC-WISP remains in a low power sleep state

The NFC-WISP uses the MSP430's hardware timers and serial communication modules to adhere to strict protocol timing requirements. A 13.56 MHz crystal oscillator is used as a stable and accurate timing reference for decoding phone-to-tag communications. To decode byte values sent by the phone, the NFC-WISP uses its UART (Universal Asynchronous Receiver/Transmitter) unit and an interrupting input pin. All messages begin with a start of frame (SOF) delimiter which is captured by the interrupting input. Once a valid SOF is found, the UART begins capturing ISO 14443 B messages to a receive buffer. Data is transmitted back to the phone using the pulse width modulation (PWM) capability of a hardware timer module. To guarantee the integrity of image data received by the display tag, the NFC-WISP validates a 16-bit cyclic redundancy check (CRC) across each frame sent by the phone. When sending a response back to the phone, the MSP430's 16-bit CRC module is used to generate and append a CRC to the response.

To ensure that all bytes of an image are successfully received, we use a simple stop-and-wait transmission protocol; the phone uses ISO 14443 B's higher layer protocol field (INF) to embed our NFC-WISP protocol's "write block" command, which can send 8 blocks (32 bytes) of image data and will only send another chunk of data once the previous block has been acknowledged by the NFC-WISP. Figure 6.4 illustrates the software procedure implemented for transferring an image to the display tag.

6.2.5 Power Optimization

Since the display tag needs to be powered wirelessly, unique design constraints are introduced. Display tag operation is divided into two tasks: first, the display tag harvests energy as it receives the image data from the phone. Second, when enough energy is stored or all the data is received, the e-paper display is updated. This scheme is energy efficient because it updates the power-hungry display all at once, thereby minimizing the time during which the display is powered on.

The display tag presents an interesting tradeoff between computation and communication. For example, implementing data compression will decrease communication energy and time but will increase the computational load imposed on the display tag to decompress the image. Preferably, any significant computation should be done on the phone, as it has a relatively abundant energy supply.

6.2.5.1 E-paper Power Consumption

Since the display has a large impact on the overall energy consumption, its energy behavior was optimized. For each image update, the display consumed on average 5.2 mJ to initialize, 2.8 mJ for each frame update, and 5.9 mJ to finalize the update operation. Although the energy consumed to initialize and finalize is fixed, the number of frame updates performed can vary, and this dictates the total energy required. Unique to e-paper, the display needs to be updated multiple times with the same image to avoid ghosting (the overlapping of new and previous images). Figure 6.5 illustrates the contrast difference between 1, 2, 3, and 4-frame updates, and the total measured energy required in each case. In our tests, the old image was not visible to the eye after four frames, so four frames is used in the final implementation.

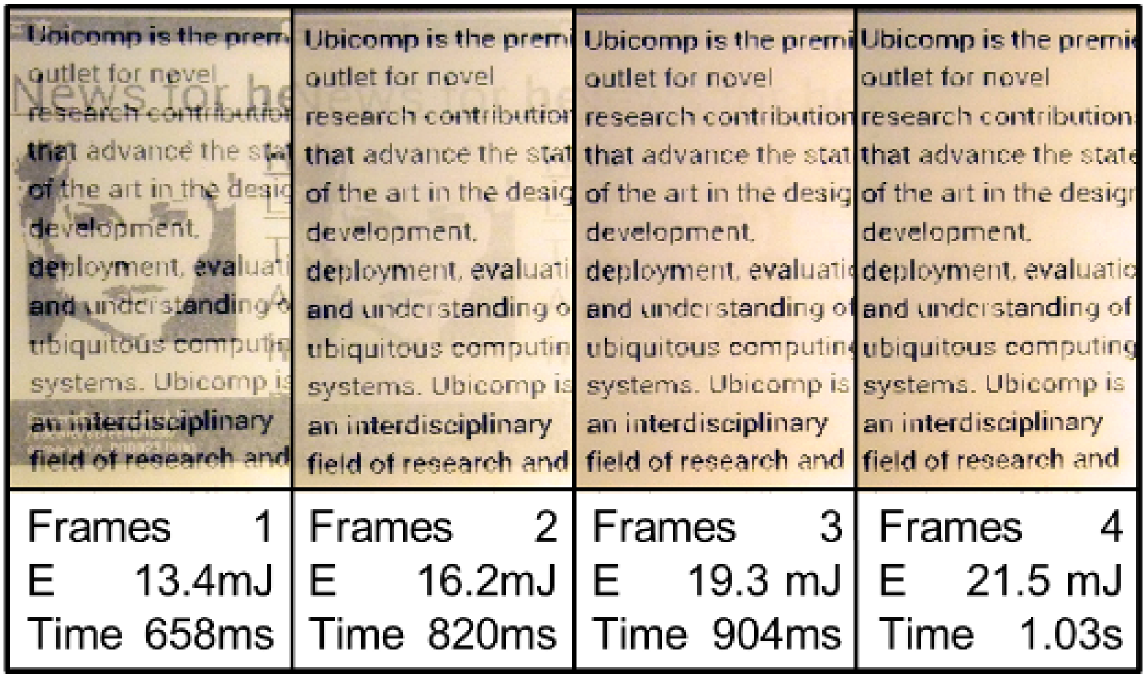
Ubicomp is the premier outlet for novel research contributions that advance the state of the art in the design, development, deployment, evaluation and understanding of ubiquitous computing systems. Ubicomp is an interdisciplinary field of research and
Frames 1
E 13.4mJ
Time 658ms
Frames 2
E 16.2mJ
Time 820ms
Frames 3
E 19.3 mJ
Time 904ms
Frames 4
E 21.5 mJ
Time 1.03s

Figure 6.5. When overwriting an old image (face) with the new (text), there is a tradeoff between number of display rewrites and image ghosting effects. The figure shows 1, 2, 3 and 4 repeated rewrites and the total energy and time required in each case.

6.3 Performance Evaluation

The time the display tag needs to be in contact with the phone to update its image is an important aspect of the user experience. This update time depends on two factors: Communication data rate and harvested power level.

6.3.1 Wireless communications

In the current prototype the time to transfer the 5.67 kilobyte image frame from the phone to the tag is measured to be 3.4 seconds, achieving an average data rate of 13.3 kbps. This data rate is slower than the theoretical maximum of 106 kbps because of protocol overhead (control bits, CRC, etc) and occasional data errors. The data rate for an NFC communication system can get as high as 424 kbps using the ISO 18000 protocol, implying a theoretical 100 ms to transmit an image frame to the tag. To further decrease transmission time, the image file size could be reduced with compression; simple run-length encoding (RLE) will work well for black and white images with extensive white space.

6.3.2 RF Energy Harvesting

Fundamentally, the time needed to update the display tag is determined by how much RF energy can be harvested. Fortunately, NFC provides a significant amount of RF power in comparison to other wireless protocols due to the use of inductive coupling at close range. The energy that is stored on the battery is largely determined by the difference between the harvested RF power and the power consumed to run the MSP430 microcontroller and other peripherals (including leakage current of these devices).

The phone radiates about 200 mW, of which 17.7 mW (12.3 mA at 1.44 V) is harvested by the display tag. We measure that 8.25 mW (2.5 mA at 3.3 V) is consumed by the display tag during communications, mostly by the microcontroller. The remaining 9.4 mW is stored on the battery. The surplus of energy required to update the display is 21.5 mJ, which could be accumulated in 2.3 seconds if the harvester and battery efficiencies are not considered. Taking into account the practical inefficiencies of the battery and harvester, the amount of time required to collect the 21.5 mJ needed for a display update will still remain within the length of one 3.4 sec communication transaction. Although it is not practical in a standard phone reader to increase the transmitted RF power, the microcontroller firmware can be optimized to make use of low-power modes at opportune moments during communications, to improve the total power surplus.

6.4 Mobile phone companion display

To demonstrate the capabilities of the display tag, we developed an Android application which takes a screenshot of the phone's display when the phone is shaken, and sends it to the display tag automatically once the tag is detected. Up to 20 images can be stored on the display tag's nonvolatile memory, and can then be cycled through using tactile buttons even when the tag is away from the reader.

The screenshot needs to be processed by the phone to be presented on the e-paper display. The 32-bit screenshot bitmap is resized, rotated, restructured, converted to gray scale and finally to black and white by simple thresholding. The black and white thresholding produces good quality text images on the companion display and can be reliably used for mirroring directions, barcodes, or maps. Since the e-paper display is black and white it inherently doesn't show colored images well, but quality could be significantly improved by half-toning or adaptive thresholding instead of the simple thresholding used here. All these operations could be done quickly and easily on the mobile phone using existing image processing tools. The Android OS doesn't generally allow applications to take screenshots, so root access was gained on the phone to enable full control of the system.

6.5 Conclusion

In this chapter we presented a wirelessly powered pixelated e-paper display tag. We showed that the display tag can receive data as well as power from an NFC-enabled mobile phone. The display tag is based on the NFC-WISP, a software defined passive NFC tag. We demonstrated an application example using an e-paper display as a mobile phone companion display.

The display tag makes use of software defined communications, and therefore is flexible with respect to the various 13.56 MHz NFC RFID protocol standards. Two standards were implemented, ISO 14443B and ISO 15693. Currently the display update time is 3.4 seconds, and we have described how this time can be reduced to 2.3 seconds.

The major technical restriction of the NFC display tag is the limited operational range of a few centimeters between the reader and the tag. Furthermore, e-paper products currently have a slow update rate, and lack in visual quality and versatility compared with traditional LCD displays, limiting the type of information that can be displayed to text or static, simple black and white images. However, it is foreseeable that future e-paper products may rival LCDs in quality and versatility.

With further optimization, the display tag shown can become an inexpensive, ubiquitous display platform. We envision many useful applications for the display tags, such as enhanced security identification badges and shareable information tiles. Furthermore, since the display tags contain a low-power microcontroller, display tags can interface with sensors and other devices. In future work, the display tags will be tested with different phone models, to understand the issues around compatibility and performance. We also plan to conduct user studies in order to better understand the value of the display tag. Finally, the NFC-WISP and display tag hardware and software will be open-sourced.

CHAPTER 7
SUMMARY AND FUTURE WORK

7.1 Sumary

This thesis has shown several hardware and software techniques that enhance the capbilities of passive embedded systems with respect to power and communications throughput. We have also implemented and proposed two novel applications that make use of these systems unique propoerties.

Hybrid Energy Harvesting. First, we described a design exploration study where we extensively explored the benefits of equipping a passive CRFID device with a small solar harvester. We showed that we can improve the effective read range of these devices by more than 2x. We also demonstrated the tradeoffs in responsiveness and survivability by looking at the performance and system-level behavior when using several different-sized capacitors. We also presented a simulator that allows a system designer to swap out virtual solar panels and capacitors and experiment with several harvesting traces we collected.

Flit. Next, we described a burst communication protocol for UHF CRFID devices, *Flit*, that is built upon existing primitives provided by the EPC Gen 2 protocol currently used for tag inventorying applications. We demonstrated that we can achieve 60% higher throughput than a standard Gen 2 implementation and save up to 6.0x of Gen 2's power consumption by duty-cycling.

EnGarde. Next, we described *EnGarde*, a passively powered peripheral for mobile phones that protects the host phone's NFC interface from malicious interactions. We showed that by using a low-power subcarrier generator, we can achieve 8.6x higher harvesting rates than what is available from a phone by default; we use this harvested power to sniff and selectively block malicious messages. We also provide several circuit designs that are optimized for low-power tag and protocol detection, as well as a dual-coil sniffing design that minimizes our impact on non-malicious interactions.

Bistable Display Tags. Finally, we presented the design of a passive, bistable display tag that is completely energy neutral in operation. By leveraging NFC power and communications commonly found in mobile phones we implement a compact companion device that displays information while the phone is switched off.

7.2 Future Work

In this section we discuss some potential directions of future research that have emerged from the work presented in this dissertation.

Multi-modal Harvesting for RFIDs. Energy harvesting opportunities for RFID-like platforms go beyond the example of solar we explored. Over the past several years, thermoelectric and vibrational harvesters have become commercially available for integration in future RFID platforms. Our simulation platform could be extended to look at the relationships between these harvesting modalities to learn harvested energy availability and application in real, mobile deployments of RFID-scale devices.

In-Situ NFC Forensics. Our embedded security peripheral, *EnGarde*, provides unique opportunities for forensic analysis of NFC on mobile phones. Previous hardware tools that analyze low level protocol message exchanges have been too large for realistic deployment during normal mobile phone use. To better understand how NFC is being used, as well as identify potential vulnerabilities, *EnGarde* could be deployed long-term to gain insight towards its use and provide hardware vendors the knowledge necessary to better secure mobile operating systems.

Dynamic Bistable Displays. One aspect of the bistable display device we have not yet explored, is looking at applications where displayed data is dynamically updated. Since the tag actually harvest more energy than the cost of a single display update, the additional energy can be used for additional updates while decoupled from the mobile phone – this is especially useful when considering tags that generate sensor data. In appliications such as statically deployed weather stations, or supply chain applications such as cold chain monitoring, buffered sensor data can be shown on the tag's e-paper display. This would implement a low-power sensor that can be inspected for useful data visually, before retrieving higher-fidelity data over an NFC communications link.

BIBLIOGRAPHY

[1] Green high peformance computing center. `http://citi.umass.edu/ghpc/`.

[2] Ic-tag solutions 13.56 mhz rfid sensor tags. `http://www.ictagsolutions.com/sensortag.html`.

[3] Powercast website. `http://www.powercastco.com`.

[4] xda-developers forum post. `http://forum.xda-developers.com/showthread.php?t=1980356`.

[5] AIOI-Systems "Smart Tag," Specifications. `http://aioismarttag.com/specification.php`.

[6] Omni-ID, ProVIEW Visual Tagging System. `http://www.omni-id.com/pdfs/Omni-ID_View_Tags_datasheet.pdf`.

[7] The Near Field Communication Forum. `http://www.nfc-forum.org`.

[8] Aho, Alfred V, and Corasick, Margaret J. Efficient string matching: an aid to bibliographic search. *Communications of the ACM 18*, 6 (1975), 333–340.

[9] Allen, M.S., Alexander, M., Wright, C., and Chang, J. Designing the powerpc 60x bus. *Micro, IEEE 14*, 5 (oct. 1994), 42.

[10] Banerjee, Nilanjan, Sorber, Jacob, Corner, Mark D., Rollins, Sami, and Ganesan, Deepak. Triage: Balancing Energy Consumption and Quality of Service in a Microserver. In *Proceedings of The Fifth International ACM/USENIX Conference on Mobile Systems, Applications, and Services (MobiSys '07)* (2007).

[11] Bhole, V.A., More, R.R., and Khadke, N.C. Security in near field communication (nfc) strengths and weaknesses. In *In proc. of Eit-2007)* (2007), IK International Pvt Ltd, p. 71.

[12] Brunelli, Davide, Benini, Luca, Moser, Clemens, and Thiele, Lothar. An Efficient Solar Energy Harvester for Wireless Sensor Nodes. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)* (2008).

[13] Buettner, M., and Wetherall, D. A software radio-based uhf rfid reader for phy/mac experimentation. In *RFID (RFID), 2011 IEEE International Conference on* (2011), IEEE, pp. 134–141.

[14] Buettner, Michael, Greenstein, Ben, Sample, Alanson, Smith, Joshua R., and Wetherall, David. Revisiting Smart Dust with RFID Sensor Networks. In *HotNets* (October 2008).

[15] Buettner, Michael, Greenstein, Ben, and Wetherall, David. Dewdrop: An energy-aware runtime for computational rfid. In *NSDI* (2011).

[16] Buettner, Michael, Prasad, Richa, Philipose, Matthai, and Wetherall, David. Recognizing daily activities with rfid-based sensors. In *Proceedings of the 11th international conference on Ubiquitous computing* (2009), ACM, pp. 51–60.

[17] Buettner, Michael, Prasad, Richa, Philipose, Matthai, and Wetherall, David. Recognizing Daily Activities with RFID-based Sensors. In *UbiComp* (Orlando, Florida, USA, October 2009).

[18] Buettner, Michael, and Wetherall, David. An empirical study of uhf rfid performance. In *MobiCom* (2008), pp. 223–234.

[19] Burleson, Wayne, Clark, Shane S, Ransford, Benjamin, and Fu, Kevin. Design challenges for secure implantable medical devices. In *Proceedings of the 49th Annual Design Automation Conference* (2012), ACM, pp. 12–17.

[20] Chen, Haisheng, Cong, Thang Ngoc, Yang, Wei, Tan, Chunqing, Li, Yongliang, and Ding, Yulong. Progress in electrical energy storage system: A critical review. *Progress in Natural Science 19*, 3 (2009), 291–312.

[21] Cho, Young H, and Mangione-Smith, William H. Deep packet filter with dedicated logic and read only memories. In *Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on* (2004), IEEE, pp. 125–134.

[22] Clark, Sarah. http://www.nfcworld.com/2012/02/09/313079/researcher-hacks-google-wallet-pin-on-rooted-android-phone/, Feb. 2012.

[23] Clark, Shane S., Gummeson, Jeremy, Fu, Kevin, and Ganesan, Deepak. Towards Autonomously-Powered CRFIDs. In *HotPower* (October 2009).

[24] Comiskey, Barrett, Albert, JD, Yoshizawa, Hidekazu, and Jacobson, Joseph. An electrophoretic ink for all-printed reflective electronic displays. *Nature 394*, 6690 (1998), 253–255.

[25] Corke, Peter, Valencia, Philip, Sikka, Pavan, Wark, Tim, and Overs, Les. Long-duration Solar-powered Wireless Sensor Networks. In *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets)* (2007), pp. 33–37.

[26] D. Halperin et al. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Proceedings of the 29th Annual IEEE Symposium on Security and Privacy* (May 2008).

[27] de Koning Gans, G., Hoepman, J.H., and Garcia, F. A practical attack on the mifare classic. *Smart Card Research and Advanced Applications* (2008), 267–282.

[28] Dementyev, A., Gummeson, J., Parks, A., Thrasher, D., Sample, A., Ganesan, D., and Smith, J. R. Wirelessly Powered Bistable Display Tags. In *UbiComp 2013* (Sept 2013).

[29] Dementyev, Artem, and Smith, Joshua R. A wearable uhf rfid-based eeg system.

[30] Desai, Neil. Increasing performance in high speed NIDs. *A look at Snort's Internals* (2002).

[31] Dharmapurikar, Sarang, Krishnamurthy, Praveen, Sproull, Todd S, and Lockwood, John W. Deep packet inspection using parallel bloom filters. *Micro, IEEE 24*, 1 (2004), 52–61.

[32] Estrin, D., Culler, D., Pister, K., and Sukhatme, G. Connecting the physical world with pervasive networks. *Pervasive Computing, IEEE 1*, 1 (2002), 59–69.

[33] Farhangi, Hassan. The path of the smart grid. *Power and Energy Magazine, IEEE 8*, 1 (2010), 18–28.

[34] Finkenzeller, K. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication.* Wiley, 2010.

[35] Gollakota, S., Hassanieh, H., Ransford, B., Katabi, D., and Fu, K. They can hear your heartbeats: non-invasive security for implantable medical devices.

[36] Gorlatova, Maria, Kinget, Peter, Kymissis, Ioannis, Rubenstein, Dan, Wang, Xiaodong, and Zussman, Gil. Challenge: Ultra-low-power Energy-harvesting Active Networked Tags (EnHANTs). In *MobiCom* (2009).

[37] Gummeson, J., Clark, S.S., Fu, K., and Ganesan, D. On the limits of effective hybrid micro-energy harvesting on mobile crfid sensors. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (2010), ACM, pp. 195–208.

[38] Gummeson, J., Priyantha, B., D., Ganesan, D., Thrasher, and P., Zhang. Engarde: Protecting the mobile phone from malicious nfc interactions. In *Proceedings of the 11th international conference on Mobile systems, applications, and services* (2013), ACM, pp. 71–84.

[39] Gummeson, J., Zhang, P., and Ganesan, D. Flit: A bulk transmission protocol for rfid-scale sensors. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (2012), ACM, pp. 71–84.

[40] Gummeson, Jeremy, Clark, Shane S., Fu, Kevin, and Ganesan, Deepak. On the limits of effective hybrid micro-energy harvesting on mobile crfid sensors. In *MobiSys* (2010), pp. 195–208.

[41] Hackmann, Gregory, Sun, Fei, Castaneda, Nestor, Lu, Chenyang, and Dyke, Shirley. A holistic approach to decentralized structural damage localization using wireless sensor networks. *Computer Communications* (2012).

[42] Halperin, D., Heydt-Benjamin, T.S., Ransford, B., Clark, S.S., Defend, B., Morgan, W., Fu, K., Kohno, T., and Maisel, W.H. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *In Proc. of SSP* (2008), IEEE, pp. 129–142.

[43] Hicks, J., Paek, J., Coe, S., Govindan, R., and Estrin, D. An Easily Deployable Wireless Imaging System. In *Workshop on Applications, Systems, and Algorithms for Image Sensing (ImageSense)* (2008).

[44] Jiang, Xiaofan, Polastre, Joseph, and Culler, David. Perpetual Environmentally Powered Sensor Networks. In *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)* (April 2005).

[45] Kansal, A., and Srivastava, M.B. An environmental energy harvesting framework for sensor networks. In *ISLPED* (August 2003).

[46] Kansal, Aman, Hsu, Jason, Zahedi, Sadaf, and Srivastava, Mani B. Power Management in Energy Harvesting Sensor Networks. *ACM Transactions Embedded Computing Systems 6*, 4 (2007), 32.

[47] Kim, Sukun, Fonseca, Rodrigo, Dutta, Prabal, Tavakoli, Arsalan, Culler, David E., Levis, Philip, Shenker, Scott, and Stoica, Ion. Flush: a reliable bulk transport protocol for multihop wireless networks. In *SenSys* (2007), pp. 351–365.

[48] Koomey, Jonathan G, Berard, Stephen, Sanchez, Marla, and Wong, Henry. Implications of historical trends in the electrical efficiency of computing. *Annals of the History of Computing, IEEE 33*, 3 (2011), 46–54.

[49] Kuo, Y.S., Verma, S., Schmid, T., and Dutta, P. Hijacking power and bandwidth from the mobile phone's audio interface. In *In Proc. of Dev 2010* (2010), ACM, p. 24.

[50] Lemos, Robert. http://www.eweek.com/c/a/security/android-phone-hacked-by-researchers-via-nfc-843123/, June 2012.

[51] Li, Ming, Agrawal, Devesh, Ganesan, Deepak, and Venkataramani, Arun. Block-switched networks: A new paradigm for wireless transport. In *NSDI* (2009), pp. 423–436.

[52] Liang, Chieh-Jan Mike, Liu, Jie, Luo, Liqian, Terzis, Andreas, and Zhao, Feng. Racnet: a high-fidelity data center sensing network. In *Proceedings of the 7th*

ACM Conference on Embedded Networked Sensor Systems (2009), ACM, pp. 15–28.

[53] Lin, Liang-Han, and Lee, Da-Sheng. Ubiquitous display. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on* (2010), IEEE, pp. 4008–4011.

[54] Liu, H., Saroiu, S., Wolman, A., and Raj, H. Software abstractions for trusted sensors. In *In Proc. of MobiSys* (2012), ACM, pp. 365–378.

[55] Liu, Vincent, Parks, Aaron, Talla, Vamsi, Gollakota, Shyamnath, Wetherall, David, and Smith, Joshua R. Ambient backscatter: wireless communication out of thin air. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM* (2013), ACM, pp. 39–50.

[56] Martinussen, Einar Sneve, and Arnall, Timo. Designing with rfid. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction* (2009), ACM, pp. 343–350.

[57] McAfee Labs. http://www.mcafee.com/us/resources/reports/rp-threat-predictions-2013.pdf, Dec. 2012.

[58] Miller, C. Exploring the nfc attack surface. In *Proceedings of Blackhat* (2012).

[59] Moser, Clemens, Brunelli, Davide, Thiele, Lothar, and Benini, Luca. Real-time Scheduling for Energy Harvesting Sensor Nodes. *Real-Time Systems 37*, 3 (2007), 233–260.

[60] Nemmaluri, Aditya, Corner, Mark D, and Shenoy, Prashant. Sherlock: automatically locating objects for humans. In *Proceedings of the 6th international conference on Mobile systems, applications, and services* (2008), ACM, pp. 187–198.

[61] Otis, Brian, and Yeager, Dan. SoCWISP: Ultra-low Power Wireless Sensing RFID Chip. WISP Summit Workshop, 2009. Presentation.

[62] Paradiso, J., and Starner, T. Energy Scavenging for Mobile and Wireless Electronics. *IEEE Pervasive Computing 4*, 1 (2005), 18–27.

[63] Ransford, Benjamin, Clark, Shane, Salajegheh, Mastooreh, and Fu, Kevin. Getting Things Done on Computational RFIDs with Energy-Aware Checkpointing and Voltage-Aware Scheduling. In *Proceedings of USENIX HotPower Workshop* (December 2008).

[64] Ransford, Benjamin, Sorber, Jacob, and Fu, Kevin. Mementos: system support for long-running computation on rfid-scale devices. In *ASPLOS* (2011), pp. 159–170.

[65] Reynolds, Matt, and Thomas, Stewart. The Blue Devil WISP: Expanding the Frontiers of the Passive RFID Physical Layer. WISP Summit Workshop, 2009. Presentation.

[66] Rieback, M.R., Gaydadjiev, G., Crispo, B., Hofman, R.F.H., and Tanenbaum, A.S. A platform for rfid security and privacy administration. In *USENIX LISA* (2006), pp. 89–102.

[67] Sample, Alanson, and Smith, Joshua R. Experimental Results with two Wireless Power Transfer Systems. In *IEEE Radio and Wireless Symposium* (2009).

[68] Sample, Alanson P, Yeager, Daniel J, Powledge, Pauline S, Mamishev, Alexander V, and Smith, Joshua R. Design of an rfid-based battery-free programmable sensing platform. *Instrumentation and Measurement, IEEE Transactions on 57*, 11 (2008), 2608–2615.

[69] Schaller, Robert R. Moore's law: past, present and future. *Spectrum, IEEE 34*, 6 (1997), 52–59.

[70] 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)* (2007).

[71] Sharma, Navin, Barker, Sean, Irwin, David, and Shenoy, Prashant. Blink: managing server clusters on intermittent power. In *ACM SIGARCH Computer Architecture News* (2011), vol. 39, ACM, pp. 185–198.

[72] Sharma, Navin, Gummeson, Jeremy, Irwin, David, and Shenoy, Prashant. Cloudy computing: Leveraging weather forecasts in energy harvesting sensor systems. In *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on* (2010), IEEE, pp. 1–9.

[73] Sharma, Navin, Gummeson, Jeremy, Irwin, David, and Shenoy, Prashant J. Srcp: Simple remote control for perpetual high-power sensor networks. In *EWSN* (2009).

[74] Sharma, Navin, Irwin, David, Zink, Michael, and Shenoy, Prashant. Multisense: proportional-share for mechanically steerable sensor networks. *Multimedia systems 18*, 5 (2012), 425–444.

[75] Shen, Dawei, Woo, Grace, Reed, David P, Lippman, Andrew B, and Wang, Junyu. Separation of multiple passive rfid signals using software defined radio. In *RFID, 2009 IEEE International Conference on* (2009), IEEE, pp. 139–146.

[76] Shepard, C., Rahmati, A., Tossell, C., Zhong, L., and Kortum, P. Livelab: measuring wireless networks and smartphone users in the field. *ACM SIGMETRICS Performance Evaluation Review 38*, 3 (2011), 15–20.

[77] Smith, Joshua R. *Wirelessly Powered Sensor Networks and Computational RFID*. Springer, 2010.

[78] Smith, Joshua R., Sample, Alanson P., Powledge, Pauline S., Roy, Sumit, and Mamishev, Alexander. A Wirelessly-Powered Platform for Sensing and Computation. In *UbiComp* (2006).

[79] Sorber, J., Shin, M., Peterson, R., Cornelius, C., Mare, S., Prasad, A., Marois, Z., Smithayer, E., and Kotz, D. An amulet for trustworthy wearable mhealth. In *In Proc. of Hotmobile* (2012), ACM, p. 7.

[80] Sorber, Jacob, Kostadinov, Alexander, Garber, Matthew, Brennan, Matthew, Corner, Mark D., and Berger, Emery D. Eon: A Language and Runtime System for Perpetual Systems. In *SenSys* (November 2007).

[81] Sorber, J.M., Shin, M., Peterson, R., and Kotz, D. Plug-n-trust: practical trusted sensing for mhealth. In *In Proc. of MobiSys* (2012), ACM, pp. 309–322.

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

[83] Taneja, Jay, Jeong, Jaein, and Culler, David. Design, Modeling, and Capacity Planning for Micro-solar Power Sensor Networks. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN)* (2008).

[84] Texas Instruments TPS61097-33 Boost Converter Datasheet. `http://focus.ti.com/docs/prod/folders/print/tps61097-33.html`.

[85] viaForensics. https://viaforensics.com/mobile-security/forensics-security-analysis-google-wallet.html, Dec. 2011.

[86] Vigorito, C., Ganesan, D., and Barto, A. Adaptive Control for Duty-cycling in Energy Harvesting-based Wireless Sensor Networks. In *Fourth IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON)* (June 2007).

[87] Want, Roy. Enabling ubiquitous sensing with rfid. *Computer 37*, 4 (2004), 84–86.

[88] Want, Roy. RFID Explained. In *Synthesis Lectures on Mobile and Pervasive Computing* (2006), Morgan & Claypool Publishers.

[89] Welbourne, Evan, Koscher, Karl, Soroush, Emad, Balazinska, Magdalena, and Borriello, Gaetano. Longitudinal Study of a Building-Scale RFID Ecosystem. In *MobiSys* (2009), pp. 69–82.

[90] Yang, Yong, Wang, Lili, Noh, Dong Kun, Le, Hieu Khac, and Abdelzaher, Tarek F. SolarStore: Enhancing Data Reliability in Solar-powered Storage-centric Sensor Networks. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)* (2009), pp. 333–346.

[91] Yeager, Daniel J., Powledge, Pauline S., Prasad, Richa, Wetherall, David, and Smith, Joshua R. Wirelessly-Charged UHF Tags for Sensor Data Collection. In *IEEE RFID* (2008).

[92] Zhang, Pengyu, Ganesan, Deepak, and Lu, Boyan. Quarkos: Pushing the operating limits of micro-powered sensors. In *Proceedings of the 14th USENIX conference on Hot topics in operating systems* (2013), USENIX Association.

[93] Zhang, Pengyu, Gummeson, Jeremy, and Ganesan, Deepak. Blink: A high throughput link layer for backscatter communication. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (2012), ACM, pp. 99–112.