Backscatter Protocols and Energy-Efficient Computing for RF-Powered Devices

Michael Buettner

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

Doctor of Philosophy

University of Washington

2012

Reading Committee:

David Wetherall, Chair

Joshua R. Smith, Chair

Matt Reynolds

Program Authorized to Offer Degree:
UW Computer Science and Engineering

University of Washington

Abstract

Backscatter Protocols and Energy-Efficient Computing for RF-Powered Devices

Michael Buettner

Chair of the Supervisory Committee:
Professor David Wetherall
Computer Science and Engineering

Associate Professor Joshua R. Smith
Computer Science and Engineering

RF-powered computers operate using energy they harvest from radio frequency (RF) signals. Compared to battery powered devices, they have the advantages of being small and long-lived as they do not need to carry an onboard energy-store, and they can be embedded inside objects, structures and even the human body.

In this dissertation, I explore how we can build networks of RF-powered computers that support rich functionality across a range of RF environments. Towards this goal, I identify the key challenges to running programs on RF-powered computers, and argue that devices must adapt their behavior to match their task energy consumption to the available power. To demonstrate this approach, I formulate the task scheduling problem for one class of RF-powered devices (computational-RFIDs), and implement Dewdrop, an energy-aware CRFID runtime. By waking tags at the right times, Dewdrop can complete tasks that previously could not complete, and at close to their maximum rate given the power that the RF environment provides.

The second challenge I tackle is how to build networks using backscatter communication. Backscatter signaling is an ultra-low power form of communication, but the simplicity needed to achieve such low power operation makes clients prone to interference. To under-

stand how this challenge can be overcome, I use measurement and simulation results to explore the mechanisms by which interference impacts network performance. I then use these insights to develop a network design that mitigates interference and enables backscatter networks to scale well. Experimental results show that this design improves both coverage and capacity in a building-scale network compared to existing designs.

This dissertation supports my thesis that RF-powered computers can support rich tasks in a variety of RF environments, and networks of such devices can scale to building-sized deployments. As technology advances, RF-powered devices will only decrease in size and increase in computational power and operational range. By demonstrating that they can also support rich functionality and be used to build building-scale networks, this dissertation demonstrates their potential to provide deeply embedded and long-lived sensing and computation.

TABLE OF CONTENTS

Page

List of Figures iii

Chapter 1: Introduction 1
1.1 RF-Powered Computing 3
1.2 Goals 4
1.3 Running Programs Using Harvested RF Power 6
1.4 Ultra-Low Power Networking 7
1.5 Thesis and Contributions 9
1.6 Organization 11

Chapter 2: Motivation and Background 12
2.1 Motivating Trends 12
2.2 Technologies for RF-powered Computers 19
2.3 Existing RF-Powered Devices 24

Chapter 3: Problem and Approach 28
3.1 Problem: Running Programs Using Harvested RF Power 29
3.2 Problem: Building Scalable Backscatter Networks 38
3.3 Summary 46

Chapter 4: Dewdrop: An Energy-Aware Runtime for Computational RFID 48
4.1 Task Model and Scheduling Goal 49
4.2 Dewdrop Design 50
4.3 Implementation 59
4.4 Dewdrop Evaluation 62
4.5 Dewdrop: Limitations and Lessons Learned 73

Chapter 5: Building Backscatter Networks That Scale Well 74
5.1 Study of Existing Technologies 74

5.2 Exploring the Trade-offs of Client Sensitivity 81
5.3 A Network Design for Scalable Backscatter Networks 89
5.4 Evaluation 100
5.5 Summary 112

Chapter 6: Related Work 113
6.1 Low Power Computing and Energy Management 113
6.2 Running Programs on RF-Powered Computers 114
6.3 Other Energy Harvesting Systems 116
6.4 Interference in Backscatter Networks 117

Chapter 7: Conclusions and Future Work 118
7.1 Thesis and Contributions 118
7.2 Future Work 120
7.3 Summary 123

Bibliography 124

LIST OF FIGURES

Figure Number — Page

1.1 Gen 2 tag (left), Intel/UW WISP (center), Telos mote (right). 3

2.1 Koomey et al. [59] showed that the energy efficiency of computing systems has doubled every 1.5 years. (Figure taken from [59]) 14

2.2 Comparison of the radiated power required for a 2.45 GHz signal to be received with a -95 dBm power, and the power consumption of a commercial low-power radio [102] and three microcontrollers [104, 94, 114] 16

2.3 Power consumption and data rates for transceiver technologies. Backscatter transceivers (circle) have three orders of magnitude better power consumption and two orders of magnitude better communication efficiency (nJ/Bit) than conventional radios. 18

2.4 Block diagram of an RFID tag showing power harvester, receiver and modulator circuits typical of RF-powered devices. 20

2.5 Gen 2 RFID reader message and backscattered tag response (as seen at the reader antenna) 23

3.1 Task execution rate versus distance to illustrate the three operating regimes of RF-powered devices. Note that the execution rate is zero when in the voltage limited regime. 30

3.2 Plot of voltage at time t and the derivative of energy with respect to time at time t for a capacitor being charged from a constant source with voltage Vs. 34

3.3 Diagram of controller-to-client interference as seen at the client. The dashed line indicates a 50% detection threshold. 41

4.1 Voltage drop for SenseTx (upper black items) and Sense (lower blue items). 52

4.2 WISP capacitor voltage over time 54

4.3 Response rates when using *Dewdrop* and the *HwFixed* runtimes. 64

4.4 Response rates for *Dewdrop* and *HwFixed* compared to an oracle. 65

4.5 Response rates for both tasks at 1.5 and 3m. X's indicate the operating point found by for *Dewdrop*. 66

4.6 Response rate and wasted time for Sense and SenseTx at 3m. 67

4.7 Charging time from 1.5V to 2V. 68

4.8 Effect of step size (β) on response rate for SenseTx at 3.5m. 69
4.9 Percent of tags that have an average response rate above 1/s and 5/s using the two runtimes. 70
4.10 CDF of response rates for the two runtimes as power is reduced. 72
4.11 Response rate for the two runtimes as tag population size increases. 73

5.1 Read rate of Gen 2 tag and controller in the presence of an interferer. 75
5.2 Network Performance for Always-On and Listen-Before-Talk 78
5.3 Network Performance for Always-On and Listen-Before-Talk with Rayleigh Fading 80
5.4 Sensitivity to interference for different ASK thresholds 82
5.5 Coverage areas of controllers when clients use different ASK thresholds. All controllers are transmitting 1 W using isotropic antennas. 84
5.6 Fraction of locations where a client would suffer from continuous wave and modulated interference for three network densities. 85
5.7 Illustration of regions where client-to-controller interference can occur. . . . 87
5.8 Impact of client sensitivity on uplink error rate. 88
5.9 Example packet exchanges for the contention and contention-free periods. In the contention case, there is a packet loss due to a collision. In the contention-free case, there is a packet loss due to noise. 95
5.10 Performance of the three approaches for a minimally covered area 103
5.11 Heatmaps of normalized throughput (Note: Colormap ranges are not equal.) 104
5.12 Performance of the three approaches for a densely covered area 105
5.13 Fairness vs throughput for the three approaches (+: Always-On, ◯: Listen-Before-Talk, ◊: Adaptive CDMA) across 4 network densities (Green: minimal coverage + 10 controllers, Red: min + 20, Black: min + 30, Yellow: min + 40) 106
5.14 Downlink loss rates for no CSMA, a fixed -36 dBm CSMA threshold, and our adaptive scheme. (All use a 10% ASK threshold) 108
5.15 Uplink loss rates for FM0, CDMA, and our adaptive scheme. 109
5.16 Performance for Always-On with different client thresholds and with or without CSMA 110
5.17 Performance for Always-On with sensitive clients and CSMA, and CDMA and Adaptive CDMA 111

Chapter 1

INTRODUCTION

Small, low-cost, low-power computers that sense the physical world have long been a goal for researchers. The vision of "smart-dust", articulated in the late 1990s, targeted millimeter-scale devices that could power themselves, sense the environment, perform computation, and communicate wirelessly [113]. Large-scale deployments of such devices would enable a wide range of applications such as dense environmental monitoring, sensor rich home automation and smart environments, and self-identification and context awareness for everyday objects.

The past 15 years have seen significant effort and progress towards the original motivating applications. Wireless sensor nodes, or "motes", are commercially available and combine a low-power microprocessor, radio, and a variety of sensors in a small form-factor. Wireless sensor networks (WSNs) based on "motes" have been applied to many real-world problems from sniper detection to structural monitoring [1]. There has also been good progress towards "motes" that are smaller and more energy efficient [94, 29]. Despite these successes, existing technologies have fallen short of the original vision of smart-dust. They have not led to ubiquitous sensing embedded in the fabric of everyday life, where buildings, clothing, consumer products, and even the human body are all equipped with networked sensors.

A key reason that existing technologies have failed to realize the vision of smart-dust is that they generally depend on batteries for power. The use of batteries increasingly limits how small devices can be; the transistor density of microprocessors doubles every 2 years according to Moore's Law while the energy density of batteries has increased much more slowly. Batteries also limit the number of devices that can be deployed and the places they can be embedded as batteries need to be replaced (or exhausted devices need to be

retrieved). Though solar cells can provide a perpetual power source for sensor motes, they impose similar limitations on the size of devices and where they can be embedded.

An attractive approach to overcoming these limitations is the use of RF power harvesting. RF-powered computers can be small as they harvest energy using a paper thin antenna rather than carrying an onboard battery. Moreover, their lifetime can be measured in decades as they have no exhaustable power source, and they can be embedded inside objects, structures and even the human body. Rudimentary RF-powered devices are already in wide use in the form of passive UHF RFID, which allows inexpensive tags to be remotely powered and interrogated for identifiers at a range of more than 10 meters. A key enabler of RFID is their use of ultra-low power backscatter communication, where they communicate by reflecting incident RF signals instead of radiating power. However, RFID technology is limited to simple object identification, similar to a barcode.

In this dissertation, I explore how combining the functionality of mote-based networks with the benefits of battery-free operation can bring us closer to the vision of smart-dust. I envision networks of devices that go far beyond RFID tags, and are capable of executing programs, sensing their surroundings, and communicating rich data to the Internet. Though RF-powered computing is in its infancy, prototype devices (most notably the Intel/UW WISP [91]) have recently become available that have allowed me to formulate key systems and networking challenges in the area and to experimentally validate solutions. To date, studies using these prototypes have generally been point demonstrations of applications that use at most a few devices located close to a single power source. In contrast, my work focuses on the challenges faced when many devices are distributed on a building-wide scale. Devices must be able to run programs efficiently using harvested power that varies widely with distance and over time, and they must be able to communicate via backscatter in the presence of many interferers.

In the following sections, I give an overview of RF-powered computing, and state the goals of this dissertation. I then describe two key challenges that limit RF-powered com-

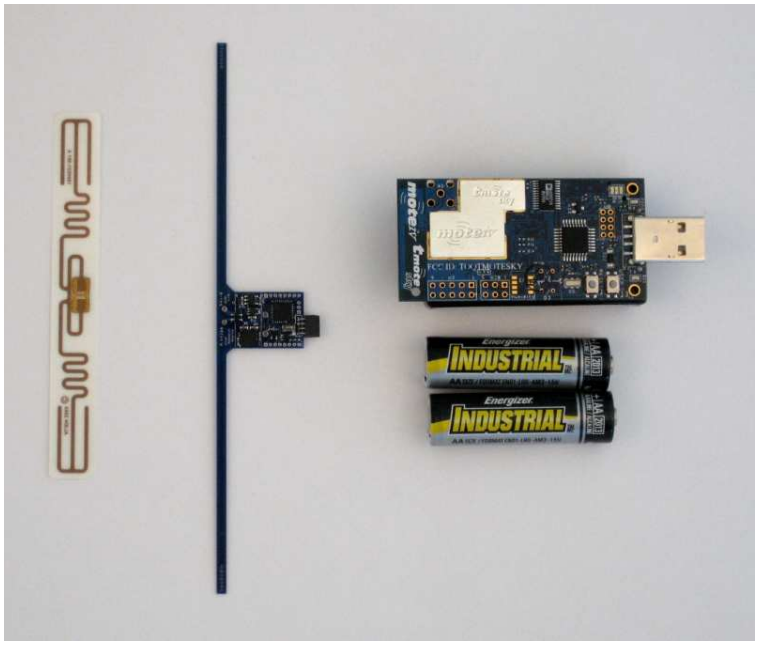


Figure 1.1: Gen 2 tag (left), Intel/UW WISP (center), Telos mote (right).

puting today – (1) running programs efficiently using harvested RF-power and (2) building ultra-low power networks that scale well – and outline my approaches to overcoming them. Lastly, I state my thesis and contributions.

1.1 RF-Powered Computing

It is well-known that Moore's law has led to an exponential increase in computational power for today's microprocessors, measured in terms of computations per second. However, the concomitant exponential increase in microprocessor performance in terms of *computations per joule* has seen less attention. Koomey et al. recently showed that the energy required to support a given computational workload has decreased by half every 1.5 years since 1946 [59]. Along with ushering in today's ubiquitous smart-phones, *this trend has made it feasible to build computer systems that are powered entirely by energy they harvest from RF signals.*

To see the potential for RF-powered computers, consider that today's commercially

available low-power microprocessors consume a few hundred μWs of power when operating at MHz frequencies, and 100 μW of RF-power is (theoretically) available 3 meters from an RFID reader (1 W), 40 meters from a GSM tower (40 W), or 5 km from a TV tower (1000 kW). RF-powered devices already exist which support modest workloads using energy harvested from RFID readers and TV towers [89]. For example, Figure 1.1 shows the Intel/UW WISP, a prototype RF-powered computer, in comparison to a Gen 2 RFID tag and a Telos sensor mote [79]. Like an RFID tag, the WISP is small, thin, and battery-free. Like a sensor mote, it is fully programmable, capable of running small programs, and equipped with an array of sensors. It runs when powered by energy harvested from a nearby RFID reader, and can operate at a range of up to 4 m. As the exponential improvements in energy efficiency are expected to continue for the foreseeable future, the range at which RF-computers can operate, and the workloads they can support, will only increase.

Because RF-powered computers can be very small, low-cost and long-lived, and embedded inside of objects and structures, they can realize applications which would be difficult using other technologies. In 2008, we described how RF-powered computers could enable applications such as tracking the temperature of blood during transport, sensing neural activity in the brain, and activity recognition for elder-care [11]. Since then, we and other researchers have used prototype RF-powered computers to implement applications including neural monitoring [46] and security for implantable devices [41], securing credit and access card transactions [27], tracking human movement for exercise and sleep monitoring [20, 47], and activity recognition for elder-care [12]. However, nearly all studies to date have used at most a few devices located very close to a power source. This is because current technology does not work well when energy is scarce or when many devices are present.

1.2 Goals

The goal of my work is to advance the state of RF-powered networks to the point that they can go beyond battery powered sensor networks to provide long-lived, deeply embedded sensing and computation. Sensor networks have proven useful in many problem domains,

but their use of batteries means they will never realize the vision of smart-dust. Looking forward, RF-powered devices have the potential to bring us close to this goal as they can be very small, long-lived, and deeply embedded in their environment. This dissertation targets deployments in indoor environments where powered infrastructure is feasible, and aims to achieve four subgoals:

- *Rich in Functionality:* We should not "dumb-down" RF-powered computers, or restrict their operation to simple, fixed tasks to make it easier to power them using harvested RF signals. They should support conventional computing and programming paradigms, particularly since they are long-lived and will need to be repurposed.

- *Adaptive to Varied Tasks and Available Power:* The power available to RF-powered devices will vary dramatically depending on their distance from a power source. Also, the energy requirements of different tasks will be different. Devices must adapt to both of these factors to make the best use of available power, which in turn extends operational range and increases responsiveness.

- *Easily Deployed:* Sensor networks and WiFi have succeeded in large part because they are easy to deploy. Developers should not need to explicitly account for energy usage or otherwise change how they write programs, and network deployment should not require detailed site-surveys or careful infrastructure placement.

- *Scalable:* For RF-powered networks to scale from point demonstrations to building-scale deployments, adding infrastructure nodes should increase coverage and network capacity.

No existing solutions satisfy these goals. Approaches to running tasks on RF-powered devices either restrict functionality to simple tasks [91], require hardware modifications to run different tasks [25], or do not account for the limitations of real hardware [85]. Existing protocols for backscatter networks require careful deployment to limit interference; naively adding infrastructure to RFID deployments can dramatically degrade performance.

Alternative approaches increase coverage at the cost of network capacity. *In contrast, this dissertation shows that RF-powered computers can execute a range of tasks in varied RF environments, and that networks of these devices can be easily deployed and can scale well in terms of both coverage and capacity.*

1.3 Running Programs Using Harvested RF Power

All networked computers realize two basic functions: they run programs and communicate with other nodes. Supporting either of these functionalities using harvested RF power requires that a unique set of challenges is overcome. Energy is the scarce resource that limits the amount of computation and sensing that can be performed. This is because it must be harvested at low rates from signals transmitted by distant power sources. However, where "motes" seek to minimize their power consumption to extend battery life, RF-powered devices seek to make the best use of an intermittent and unpredictable power source. When power is plentiful, devices can maximize their performance, whereas they should aim to extend range when power is scarce.

A key goal of this dissertation is to understand how the characteristics of RF-power harvesting change existing models of program execution. Towards this goal, I formulate the task scheduling problem and implement a runtime for one class of RF-powered devices: computational RFID tags (CRFID). CRFID tags, such as the WISP, extend the capabilities of conventional RFID tags by adding computing and sensing capabilities. They are powered by and communicate with commercial RFID readers, which enables them to make use of existing infrastructure. They have shown promise for a range of applications including interactive sensing applications such as activity inference [12]. Focusing on CRFIDs lets me identify key challenges for RF-powered computers in general, while grounding my solution in the particulars of a widely used platform.

Current CRFIDs tags adopt the operational model of RFID tags, where they turn on and run whenever they are powered by the reader. This approach works for conventional RFID tags because their functionality is very simple (a state machine with memory) and

can be run in the worst case at the limit of the energy harvesting range. However, CRFID tags consume far more energy running tasks that use sensors and computation, and the energy cost of tasks may vary from iteration to iteration. By adopting the model of running whenever there is power, naive CRFID designs limit the range at which a CRFID tag can operate and the kinds of tasks that it can run. In contrast, my approach is to match harvested energy to task consumption by viewing task execution as a scheduling problem. A tag should wake out of deep sleep only when it is likely to execute a task efficiently. Energy-aware scheduling will enable devices to run a wider variety of tasks in a range of operating environments.

While conceptually simple, a practical design is difficult to achieve for several reasons that are inherent to RF-powered computers. First, the energy needed to run a task and the input RF power both vary greatly over time due to factors such as non-deterministic protocols and reader frequency hopping. Second, the commonly-held intuition about energy storage as a simple reservoir is insufficient because the efficiency of both energy harvesting and energy consumption is state dependent. Finally, it is costly to gather the basic information needed to make scheduling decisions, e.g., the rate at which energy is harvested and the amount that is already stored, because CRFIDs are so energy impoverished.

As part of this dissertation, I present the design, implementation, and evaluation of Dewdrop, an energy-aware runtime for CRFID tags that matches harvested energy to task energy consumption. By waking tags at the right times, Dewdrop can complete tasks that previously could not complete, and at close to their maximum rate given the energy that the RF environment provides.

1.4 Ultra-Low Power Networking

The second basic function of any networked device is the ability to communicate with other nodes in the network. This is a challenge for RF-powered devices, because existing low-power radios consume orders of magnitude more power (even when idle) than is generally available for harvesting. Moreover, because the power consumption of analog radio hardware

will not improve as quickly as digital logic, conventional radio architectures will only become a worse fit for ultra-low power networks as time goes on. As I show in Section 2.1.2, the overall power consumption of computing and sensing platforms is already on par with the theoretical minimum for how much power must be radiated to communicate over useful ranges. As the power efficiency of computing platforms continues to decrease, this hard limit on transmit power will become a significant factor in overall energy consumption.

A second key goal of this dissertation is to rethink wireless communication in ways that can improve energy efficiency by orders of magnitude compared to conventional radios. The strategy that I explore in this dissertation is to build on backscatter signaling. In backscatter networks, clients do not radiate RF power to transmit data. Instead, they reflect (backscatter) incident RF signals transmitted by infrastructure nodes, referred to as controllers [1], and controllers decode messages by detecting changes in the reflected signal. This model of operation allows nearly all energy costs to be shifted from clients to the powered controller whose function is akin to an AP in an 802.11 network. The power needed for backscatter clients to communicate then approaches zero, and the transceiver can be powered using harvested RF signals. Existing backscatter devices can achieve ranges of 30 m or more and consume as little as 1 μW; this is four orders of magnitude less than existing low-power radios.

While backscatter signaling works well for an individual link, it has seen little use for full-fledged networks because the simplicity needed to power transceivers from RF signals makes clients susceptible to interference. To date, the only widespread use of backscatter communication is for inventorying passive RFID tags, a task that relies on carefully-placed controllers that are spatially separated to limit interference. In contrast, I aim to build backscatter networks that are easily deployed and provide good coverage and capacity on a building-wide scale.

---

[1] The more general term, "controller", is used throughout this dissertation instead of, "reader" or "interrogator", which are commonly used in the RFID literature. This reflects the function of infrastructure nodes which goes beyond simply reading identifiers from tags, and is instead to *control* the medium access behavior of the clients in the sense of a traffic controller.

To understand how we can extend backscatter technology to support RF-powered networks, I develop and evaluate an ultra-low power network design that is based on backscatter clients. I first use measurements of existing systems and simulation results to show how interference limits the scalability of backscatter networks, and to explore how the client transceiver design impacts network-wide interference patterns. I then use these insights to develop a network design that enables backscatter networks to scale well. Experimental results show that my design improves both coverage and capacity in a network setting compared to existing designs.

1.5 Thesis and Contributions

This dissertation supports my thesis that *RF-powered computers can support rich tasks in a variety of RF environments, and networks of such devices can scale well to building-sized deployments.* As technology advances, RF-powered devices will only decrease in size and increase in computational power and operational range. By demonstrating that they can also support rich functionality and be used to build scalable networks, this dissertation demonstrates their potential to approach the vision of smart-dust.

By identifying the key challenges to running programs on RF-powered computers, and formulating the task scheduling problem for one class of RF-powered devices (CRFIDs), I show how rich tasks and programs can be supported using harvested RF-power; the Dewdrop runtime is a proof-of-concept built on the WISP CRFID that effectively matches task behavior to available power across a range of tasks and RF environments. Through measurement of existing backscatter systems, I identify the root cause of interference in backscatter networks and show how a minimal client demodulator redesign coupled with coordinated controller behavior can largely mitigate the impact of this interference. Leveraging this insight, I design and evaluate a PHY/MAC protocol that enables network coverage and capacity to increase as infrastructure is added.

I make the following contributions in this dissertation:

Identifying key challenges to running programs on RF-powered computers. RF-powered computing is in its infancy, and its constraints and operating model are not yet well understood. I identify four characteristics I believe are fundamental to the technology: 1) RF-powered computers will duty-cycle to extend operating range, 2) The amount of energy that must be stored before executing a task will vary according to both the task itself and the amount of supplemental power harvested during execution, 3) Input power will vary greatly between devices based on distance, and over short time-scales due to motion and frequency selective fading, and 4) The use of capacitors to store energy means that the rate at which a joule of energy is harvested or stored is state-dependent. The last three factors make it difficult to match energy consumption to available power to achieve good performance.

Dewdrop, an energy-aware runtime for CRFIDs Taking into account the challenges above, I formulate the task scheduling problem for CRFIDs, a class of RF-powered devices. The goal is to run iterative tasks as often as possible given the available RF power. I then implement Dewdrop, a runtime for CRFIDs that adapts device behavior to match the requirements of the task with the available power. By waking tags at the right times, Dewdrop can complete tasks where they previously could not complete, and nearly as often as possible given the power that the RF environment provides.

A study of interference in backscatter networks Because backscatter requires infrastructure to transmit RF power both when receiving and transmitting, interference patterns in backscatter networks differ from those of conventional wireless networks. One key distinction, which has been overlooked by prior work, is that interference at clients is of two types: continuous wave interference and modulated interference. I show how these two types of interference limit network coverage and capacity for Gen 2 RFID systems. Then, via simulation, I show how the demodulator design of backscatter transceivers trades sensitivity to continuous interference against sensitivity to modulated interference, and how this

trade-off impacts network-scale performance.

A PHY/MAC protocol that enables scalable backscatter networks Based on the findings of the interference study, I propose a PHY/MAC protocol for backscatter networks that mitigates interference. My approach has three main components 1) A modified demodulator at the client that reduces *continuous wave* interference at clients, 2) a CSMA/CA approach that coordinates controller behavior to avoid *modulator* interference at clients, 3) CDMA uplink modulation overcomes interference between clients experienced at controllers. I evaluate my design via simulation, and show that it improves both coverage and capacity compared to other solutions.

1.6 Organization

The remainder of this dissertation is as follows. Chapter 2 presents necessary background material, and Chapter 3 builds on this background to describe the dual problems of running programs and communicating using harvested RF power. Chapter 4 presents the design, implementation, and evaluation of Dewdrop, and Chapter 5 presents the design, implementation, and evaluation of our PHY/MAC protocol for scalable backscatter networks. I cover related work in Chapter 6, and conclude in Chapter 7.

Chapter 2
MOTIVATION AND BACKGROUND

In this section, we outline the technology trends that motivate our approach of using RF power harvesting and backscatter communication to provide deeply embedded computation and sensing. Necessary background on three key aspects of RF-powered computers is then presented: RF power harvesting, energy storage, and backscatter communication. Lastly, we describe existing RF-powered computers.

2.1 Motivating Trends

There are two trends that motivate our work. First, the energy efficiency of computer systems is increasing exponentially. This has made it possible to power computers using harvested RF signals today, and moreover, the range at which this is feasible is also increasing exponentially [99]. The second trend is that the energy efficiency of conventional radio designs is *not* improving exponentially. This is because analog components do not benefit from Moore's Law, and the physics of RF propagation dictate how much power must be radiated to communicate a message over a given distance. This means we need to explore alternative methods of wireless communication, lest the power consumption of communication become the operating bottleneck. Backscatter communication is one such approach that is already orders of magnitude more energy efficient that conventional radios, and whose efficiency *will* scale with digital logic.

2.1.1 Range of RF-Powered Computers Doubles Every 3 Years

Moore's Law, the historical pattern of transistor density doubling approximately every two years, has driven the exponential increase in desktop and server performance over the past

few decades. However, a less well-known fact is that *the energy efficiency of computer systems has also increased exponentially, doubling every 1.5 years since 1946, and this trend is expected to continue even if Moore's Law fails.* [59]

Figure 2.1 shows the energy efficiency of selected computing systems from 1946 through 2010. Not only has there been a doubling in efficiency every 1.57 years, but this trend extends even to early vacuum tube computers. Moreover, this trend is expected to continue even if Moore's Law does not, as there is still large potential for efficiency improvements beyond just reduced transistor size. Physicist Richard Feynman famously estimated that, as of 1985, there was a factor of 10^{11} improvement that was theoretically possible for computers that use electrons for switching [33]. Since then, energy efficiency has improved by more than a factor of 10^4 [59], which suggests computer systems are likely to experience at least a few more decades of rapidly improving power efficiency. Already, commercial, millimeter-scale microprocessors operate at MHz clock rates while consuming only a few hundred picojoules per instruction (e.g., approximately 400 μW at 1 MHz [104]). Moreover, research platforms exist that improve efficiency even further to only a few picojoules per instruction [94].

As energy per instruction approaches 100s of picojoules, powering these devices at useful ranges using harvested RF signals becomes feasible. The physics of RF propagation mean that the power available for harvesting decreases as distance squared (in the best case) from the source. To give a sense of available power, consider the following: Friis transmission equation [35] predicts that (assuming isotropic antennas) 100 μW is available 3 meters from a RFID reader (1W), 40 meters from a GSM tower (40W), or 5 km from a TV tower (1000 kW). RF-powered platforms exist today which support modest workloads using power harvested from RFID readers and TV towers [89]. Moreover, by combining the d^2 fall-off of RF-power and the rate at which efficiency improves, we see that the range at which a modest workload can be powered is increasing exponentially at half the rate of energy efficiency [99]. In other words, the range at which RF-powered computers can operate is doubling every 3 years.

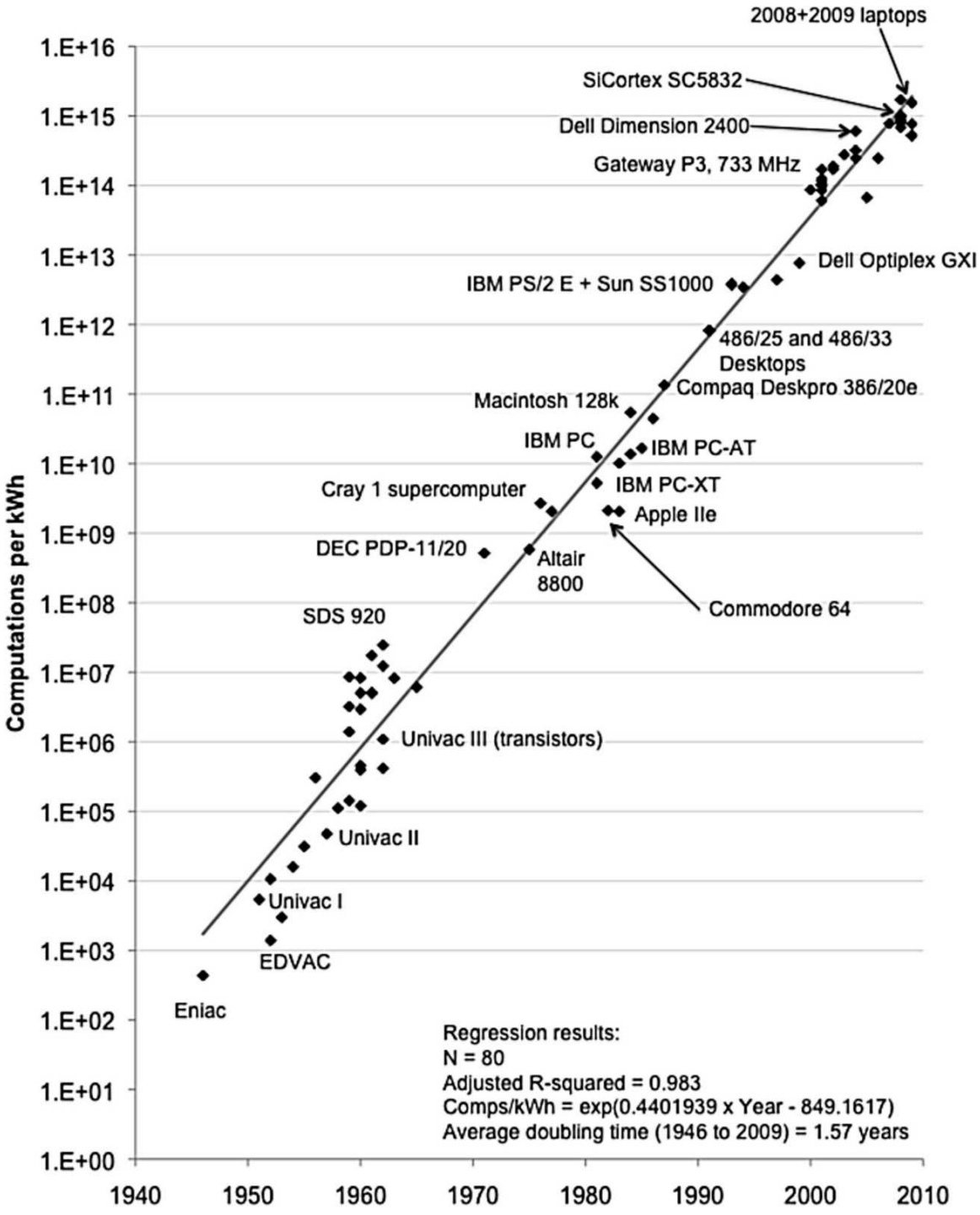


Figure 2.1: Koomey et al. [59] showed that the energy efficiency of computing systems has doubled every 1.5 years. (Figure taken from [59])

2.1.2 Radio Transceiver Efficiency Will Not Scale With Digital Logic

Though the energy efficiency of computer systems is doubling every 1.5 years, the efficiency of radio transceivers will increase much more slowly. For example, a study by Alcatel-Lucent found the energy efficiency of cellular networks is improving by less than 40% every two years [116]. There are two factors that limit the extent to which radio energy efficiency will improve. First, the efficiency of the analog components that make up conventional super-heterodyne type radios (low noise amplifiers, mixers, power amplifiers, etc.) does not benefit from Moore's Law. In fact, decreasing supply voltage, increasing transistor non-linearity, and the degradation of transistor intrinsic gain pose significant challenges to the development of high performance analog circuit design using nanoscale CMOS technologies [4, 71]. Consequently, compensation circuits have been developed which in turn result in higher power consumption. The power efficiency of fixed performance analog primitives, such as operational amplifiers, has also begun to plateau as a function of process technology [123, 122].

The second factor that limits the efficiency of conventional radios is that the physics of RF propagation dictate how much power must be radiated to communicate a message at a given distance. In Figure 2.2, the solid blue line shows the theoretical minimum for how much RF power must be radiated to be received at a given distance with a power of -95 dBm (the receive sensitivity of the Chipcon 2420 radio). The data is calculated using Friis equation for free-space propagation assuming isotropic antennas, and a 2.45 GHz signal. Along with this, the power consumption is shown for the Chipcon 2420 radio, the TI MSP43021xx series MCU running at 1 MHz, and 2 ultra-low power experimental MCUs seen in the literature [94, 114].

When transmitting at its maximum power of 0 dBm (i.e., 1 mW), the Chipcon 2420 consumes around 31 mW; two orders of magnitude more power than it radiates. Moreover, the transceiver consumes 34 mW even when it is in receive-mode waiting for packets to arrive. This is a result of the high power consumption and complexity of conventional

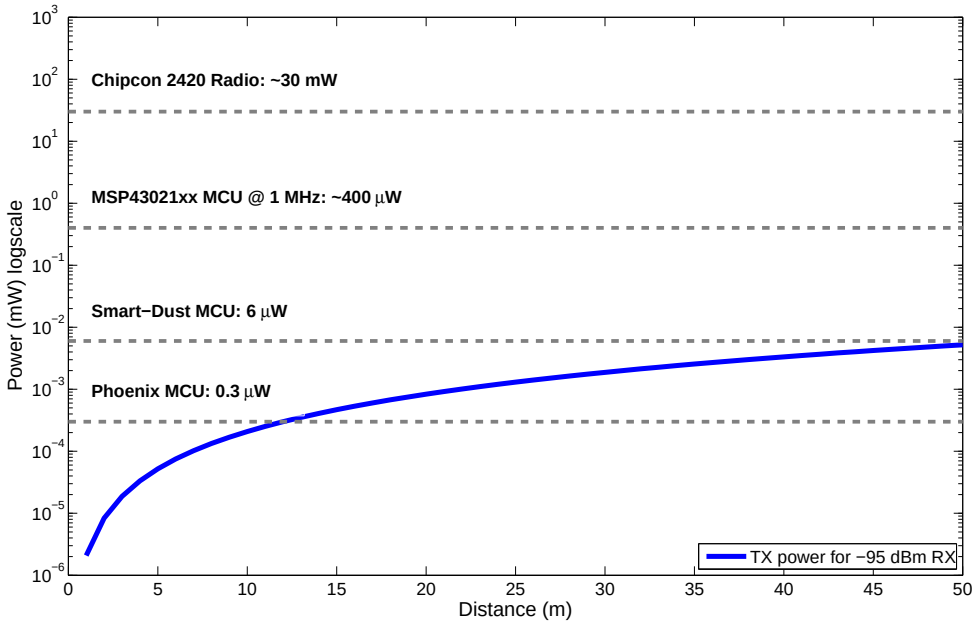


Figure 2.2: Comparison of the radiated power required for a 2.45 GHz signal to be received with a -95 dBm power, and the power consumption of a commercial low-power radio [102] and three microcontrollers [104, 94, 114]

radios. The figure also shows the power consumption of TI's popular MSP43021xx line of microprocessors: around 400 μW when operating at 1 MHz. This is two orders of magnitude less than the Chipcon radio. This shows that, even today, the radio is a dominant factor in the power consumption of low-power platforms.

The last two dashed lines in the figure indicate the power consumption of two experimental MCUs that push today's limits of energy-efficient computing. In the case of the Phoenix processor [94], its power consumption is equivalent to the absolute minimum amount of RF power that must be radiated by a transmitter for its message to be received at -95 dBm 12 meters away. While the power consumption of microprocessors will surely decrease over time (likely by 50% every two years), the rate at which RF power falls off with distance is a law of physics and will not change. Moreover, there are limits on how sensitive a receiver

can be, e.g., -113 dBm for 802.15.4 radios like the Chipcon 2420, though just reducing the noise of the LNA by 3 dB would increase the power consumption of the device by 33% [62].

To give some perspective to this data, consider the following. When sensing once every 10 minutes, the Phoenix sensing platform has an average power consumption of 3.9 x 10^{-11} W and can survive using a 1 mm^3 battery for over a year. If the platform was combined with a Chipcon 2420 radio that transmitted data for *only 1 ms every 10 minutes*, the average power consumption would increase to 5.2 x 10^{-8} W and the lifetime would be reduced to around 8 hours. In contrast, transmitting for 1 ms every 10 minutes using a backscatter transceiver like the one described in [78] would increase the power consumption to only 4.1 x 10^{-11} W, and the lifetime would still be over a year.

Backscatter Transceivers Already Orders of Magnitude More Energy Efficient

Because backscatter transceivers do not rely on complex analog components, they will largely track improvements in CMOS process scaling. Their "transmitter" consists of a single transistor which is used to toggle the impedance of the antenna, and the amplitude-shift-keying (ASK) demodulator consists of a diode, a resistor, a filter capacitor, and a comparator. The only other component aside from digital logic is a low frequency digital oscillator used for data recovery; this is in contrast to the oscillator of conventional radios that must run at the carrier frequency (100s - 1000s of MHz) to mix signals to baseband.

Even today, backscatter transceivers are orders of magnitude more efficient in terms of communication efficiency than other radios designs. Figure 2.3 plots power consumption versus data rate for four classes of RF transceivers, with the grey lines indicating constant communication efficiency (nJ/Bit). [1] The blue squares indicate consumer technologies such as 802.11a/g, Zigbee/802.15.4, and Bluetooth. Though the application domain for these technologies varies widely, from low-power sensor networks which use Zigbee/802.15.4 to high-speed wireless systems using 802.11, *the energy efficiency is nearly equivalent.*

[1] We calculate power consumption by averaging the power required for receiving and transmitting. In all cases, the two values were within an order of magnitude.

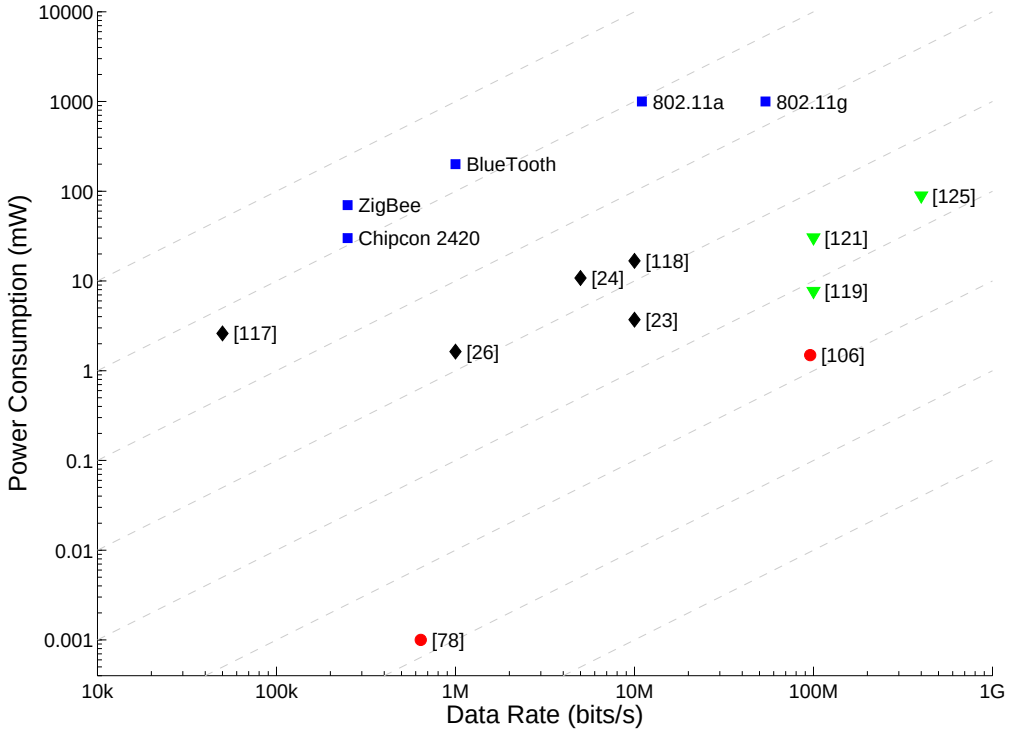


Figure 2.3: Power consumption and data rates for transceiver technologies. Backscatter transceivers (circle) have three orders of magnitude better power consumption and two orders of magnitude better communication efficiency (nJ/Bit) than conventional radios.

The black diamonds show experimental transceivers targeted for medical devices [118, 26] and wireless sensor networks [117, 23, 24]. These devices represent the state of the art in conventional transceiver architectures. They can achieve one or two orders of magnitude better efficiency than today's commercial technologies. Looking beyond conventional architectures, the green triangles indicate ultra-wide band transceivers. This is an emerging radio technology that uses very fast pulses to spread a signal over very large bandwidths, and has been shown to be more energy efficient than conventional radios by an order of magnitude [125, 119, 121].

In contrast, backscatter transceivers are shown as red circles. They consume as little as 1μW in the case of passive RFID, while supporting data rates up to 640 kbps [78], and can communicate at approximately 30 m with an FCC-compliant controller [72, 90]. This level of performance is comparable to the widely used TI Chipcon 2420 radio, but achieved using four orders of magnitude less power. More recent QAM based backscatter transceivers achieve up to 96 Mbps with an efficiency of 15.5 pJ/bit [106]. This is equivalent in rate to existing UWB transceivers but with an order of magnitude better communication efficiency.

The fact that most conventional radios duty cycle their transceivers between active and standby states to reduce their average power consumption only makes the comparison more favorable. Although not all the publications identified in Figure 2.3 reported idle power, [117] was clearly the lowest power of the non-backscatter transceivers (including UWB) with a reported standby power consumption of 1.2 μW. This means that even a sleeping radio uses 20% more power than an RFID tag that is continuously active.

2.2 Technologies for RF-powered Computers

Using RF signals to power computing and sensing is an attractive proposal as RF waves penetrate most non-metallic materials and are imperceptible to humans. In this section, we discuss the technologies that are fundamental to RF-powered computing: RF-power harvesting and energy storage. we then give background on backscatter communication, as it is the approach we take for building networks of RF-powered devices.

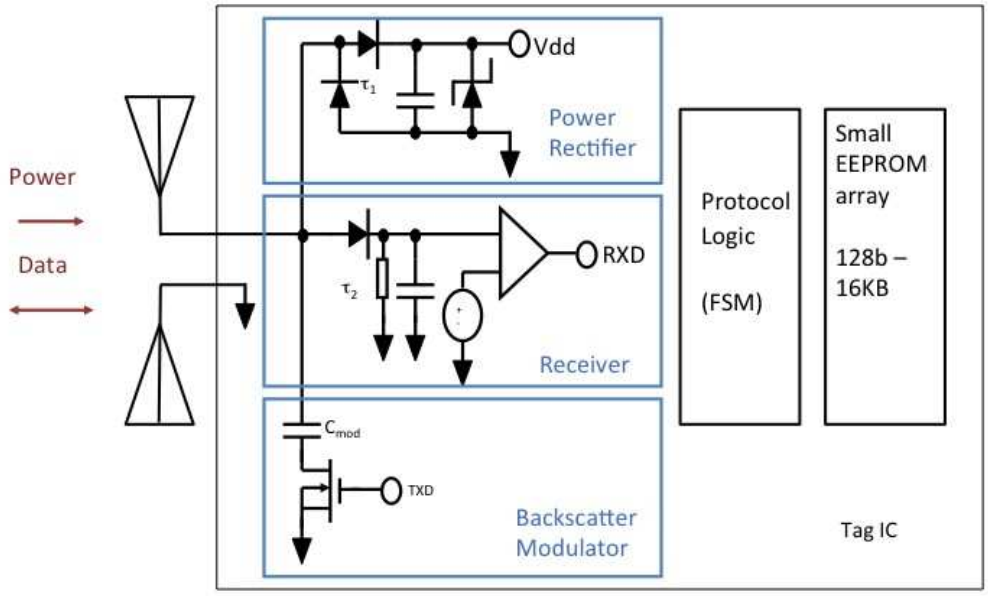


Figure 2.4: Block diagram of an RFID tag showing power harvester, receiver and modulator circuits typical of RF-powered devices.

2.2.1 RF Power Harvesting

RF-powered computers get the entirety of their operating energy from harvested RF power. The amount of RF power available for harvesting is a function of distance from the RF source as described by the Friis Equation given in Equation 2.1.

$$P_r = P_t G_r G_t (\frac{\lambda}{4\pi R})^2 \tag{2.1}$$

For a given transmit power (P_t), transmitter and receiver antenna gains (G_t, G_r), and wavelength (λ) the received signal power (P_r) falls off with the square of the distance $(\mathrm{R})^2$.

[2]The path-loss exponent of 2 describes free-space propagation. Indoor environments are modeled using exponents between 4 and 6.

In other words, available power falls off very quickly with distance. The Friis equation predicts that (assuming isotropic antennas) 100 μW is available 3 meters from a RFID reader (1 W), 40 meters from a GSM tower (40 W), or 5 km from a TV tower (1000 kW).

A second implication of Equation 2.1 is that directional antennas can be used to increase the received power. An example of this is the WARP platform [89], which uses a 5 dBi antenna to increase the power received from TV transmitters. This approach works well for statically deployed sensors, where the antenna can be carefully oriented; a misaligned high-gain antenna will reduce the received power. Directional antennas are also widely used by RFID readers to increase the power transferred to tags.

The last implication is that a longer wavelength can deliver more power at a given distance. However, this requires a larger antenna to capture the signal which increases the size of the device. Another consideration is that it is difficult to design wideband antennas, which means systems can harvest power only from a relatively narrow band, for instance the 900 MHz ISM band or a particular TV station, but not both.

The Friis equation describes how much power is available for harvesting, but real power harvesting circuits cannot capture all available power. Research into RF power harvesting circuits is still in its early stages, and a simple harvester (i.e., rectifier) circuit is shown in Figure 2.4. More complex designs exist, but they are still less than 20% efficient at ranges of more than a few meters [70]. However, even existing RF-power harvesters can drive a 1.2V DC output at a distance of around 20 meters [74], which is sufficient to power ultra-low power processors. Improvements in harvester efficiency is yet another way the operating range of RF-powered computers will increase in the near future.

2.2.2 Storing Harvested Energy

To support duty cycling, RF-powered devices must have some onboard energy store. There are two possibilities for this: batteries or capacitors. *Capacitors are the preferred choice for energy harvesting devices because they can be recharged indefinitely [51]; current battery*

technology can tolerate only a few thousand recharge cycles before losing the ability to hold charge. However, the use of capacitors has implications for how RF-powered computers operate. Equation 2.2 describes how the voltage across the capacitor (V_c) increases with time when a voltage source (V_s) is applied.

$$V_c = V_s(1 - e^{-t/RC}) \tag{2.2}$$

$$E_c = \frac{1}{2}CV_c^2 \tag{2.3}$$

The voltage across the capacitor approaches that of the source exponentially, and the rate at which the voltage rises is determined by the resistance of the circuit (R) and the size of the capacitor (C); a smaller capacitor means the RC term is smaller and the voltage rises more quickly. Equation 2.3 describes the amount of energy stored on the capacitor (E_c) for a given V_c. These two equations are key to understanding the operating model of RF-powered devices.

All computing devices have some fixed voltage below which they cannot operate (V_{thresh}), and standby or sleep modes generally affect only the current draw. For an RF-powered device to function, sufficient charge must be harvested and stored on the capacitor such that V_c is some small amount (ε) above V_{thresh}. Once the device switches to active-mode energy will be drained from the capacitor and V_c will drop. The amount of time the device can be in active-mode before the voltage drops below V_{thresh}, assuming a constant power consumption, depends on the capacitor size as described by Equation 2.3. There are three effects of increasing the capacitor size: 1) It increases the amount of time it takes to charge to V_{thresh} when the capacitor is empty (i.e., the cold-boot time), 2) It increases the amount of harvested energy that is unusable (equal to $\frac{1}{2}CV_{thresh}^2$), 3) It increases the amount of time the device can operate once $V_{thresh} + \varepsilon$ is reached (the usable energy is equal to $\frac{1}{2}CV_{\varepsilon}^2$ - $\frac{1}{2}CV_{thresh}^2$). The implications of these effects will be discussed in Section 3.1.4.

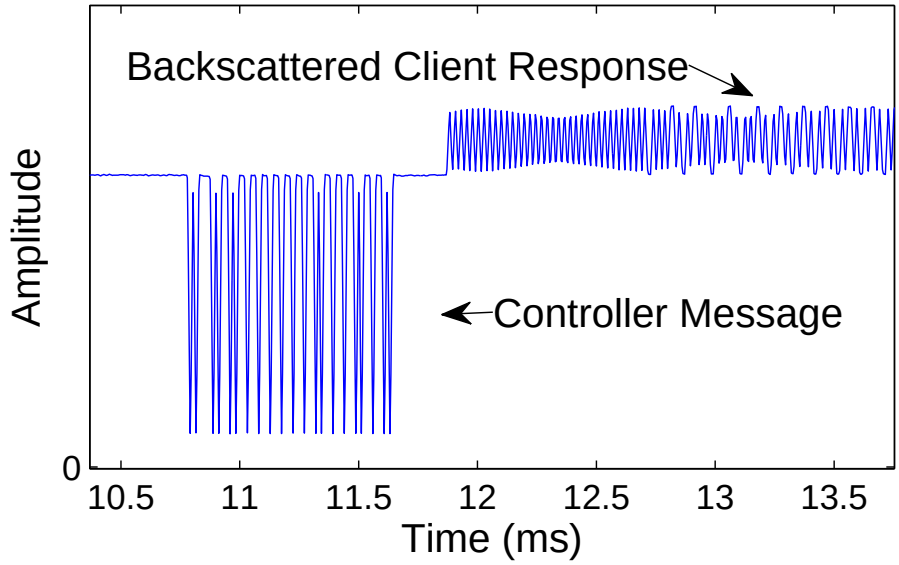


Figure 2.5: Gen 2 RFID reader message and backscattered tag response (as seen at the reader antenna)

2.2.3 Backscatter Communication

Backscatter signaling is a wireless communication technique that shifts nearly all energy costs to a *controller* which is generally powered and part of the infrastructure. The role of the controller is similar to that of an AP in 802.11 networks, and *clients* communicate only with the controller. One key feature of backscatter communication is that clients do not transmit RF energy to communicate with the controller. Instead, the controller transmits a continuous RF signal and the client modulates the reflection coefficient of its antenna. The controller detects the variation in the reflected signal to decode the uplink message; the client can modulate both the phase and amplitude of the reflected signal [108, 107].

Backscatter systems achieve very low operating power by keeping the client transciever simple. Figure 2.4 shows both the transmit and receive circuits of a typical backscatter client. The receiver consists of a resistor, a filter capacitor, and comparator which compares the rectified voltage to the filtered signal to demodulate the controller's amplitude-

modulated signal. Moreover, the backscatter modulator consists of a single transistor that is switched to change the reflectivity of the antenna.

Figure 2.5 shows communication between a controller and a client. The continuous RF signal results in the DC offset of the received waveform, with the series of low amplitude pulses being an amplitude-modulated transmission from a controller to a client. This downlink signal can be demodulated by the client using a simple envelope detector. The backscattered uplink transmission is seen as a combination of the incident continuous wave and the reflection from the client.

One limitation of backscatter is that the uplink path-loss is twice that of the forward link. This means there is a d^4 path-loss in the best case, which makes long range backscatter difficult. However, backscatter is a good fit for indoor scenarios where the distance to a controller is up to 30 meters [72].

2.2.4 Summary

In summary:

- Harvestable RF power drops off quickly with distance, but even existing, relatively inefficient, harvesters can provide 1.2V DC at close to 20 meters.
- Small capacitors will be used for short term energy storage as they charge quickly, recharge indefinitely, are small and inexpensive, and are non-toxic. However, their charging characteristics means there is a design trade-off between responsiveness and maximum task size.
- Backscatter communication enables low-cost, low-power client devices, with a communication range of 10s of meters when using an FCC compliant controller.

2.3 Existing RF-Powered Devices

In this section, we give an overview of existing RF-powered devices to understand the implemented design points in the space. we give particular focus to EPC Class-1 Generation-

2 (Gen 2) passive RFID technology as it is mature and widely deployed, and the Intel Wireless Identification and Sensing Platform (WISP) as it is the only available RF-powered research platform and hence supports the majority of our experimental work.

2.3.1 Passive RFID and EPC Gen 2 Tags

Passive Radio Frequency IDentification (RFID) is a wireless technology that allows small, inexpensive tags to be remotely powered and interrogated for identiers and other information. The tags are RF-powered but fixed function, and consist of a simple state machine and memory.[3] Because their functionality is simple, they work whenever harvested power is sufficient to drive their simple circuit, but do not accumulate appreciable amounts of energy or duty cycle between modes. Though they do not duty-cycle, the read rate of RFID tags does degrade with distance. Prior work has shown this to be a result of channel hopping and frequency selective fading, which causes tags to receive different amounts of power depending on what channel the reader is transmitting on [13].

Passive RFID technology was first developed at Los Alamos National Labs in the early 1970's, with the specific goal of developing an implantable temperature sensor and identifier for tracking lifestock [61]. Their original system transmitted 4 W of power, had a range of 5 m, and an identification time of around 100 ms [3]. Though modern systems have higher performance, e.g., current passive tags have a sticker form-factor and can be read hundreds of times per second at a range of more than 10 m, the reader and tag transceiver design look almost identical to modern passive RFID technology [58].

Passive RFID technology continued to develop through the 1980's and 1990's for applications such as vehicle tracking, electronic highway tolls, and supply chain management. In 1999 the Auto-ID Lab was formed to develop a standard for Electronic Product Codes (EPC) for supply chain management. This effort resulted in the Class-1 Generation-2 spec-

[3]The Gen 2 standard uses the term "reader" to refer to what we more generally call a *controller* and "tag" to refer to what we call a *client*. we use this language when referring to Gen 2 systems to be consistent with the RFID literature.

ification in 2004 [31] which defines the PHY and MAC layer interactions between powered RFID readers and RF-powered tags. Gen 2 technology has seen widespread use in the supply chain and is rapidly expanding to new applications such as large scale, item-level tracking of consumer goods [37], and identity documents such as the US Passport Card and Enhanced Drivers Licenses [60].

Gen 2 RFID is a mature RF-powered backscatter technology that is widely deployed. Consequently, we use it as a reference point for much of our research. However, the EPC standard is specifically designed for inventorying product codes. RFID tags only transmit identifiers and do not perform computation or sensing, and the protocol described by the Gen 2 standard is suited primarily for gathering identifiers.

2.3.2 Computational RFID and the Intel WISP

Our vision of RF-powered computers combines RFID technology for energy harvesting and backscatter communication with general purpose computation and sensing. The Intel Wireless Identification and Sensing Platform (WISP) [91] is an open-source prototype of such a device, and is available to the academic community. This class of device, which leverages RFID intrastructure but supports rich computation and sensing, is referred to as Computational RFID (CRFID). Thus far, WISPs are in use at nearly 50 universities and have been used for more than 50 publications.

The current WISP can harvest sufficient power to operate at up to 4 m. The WISP is fully programmable, capable of running small programs, and equipped with sensors. It runs programs written in C on an ultra-low power 16-bit MSP430 microcontroller and has 8 KB of flash memory, a 3D accelerometer, and a temperature sensors. Moreover, GPIO pins make it easy to interface the WISP to external sensors. Unlike an RFID tag, the WISP consumes considerably more power when computing, communicating and sensing than can normally be harvested from the RF signal. Consequently, the WISP must duty cycle between a low-power standby mode, in which the energy needed to run is gathered

into a short-term energy buffer, and an active mode in which stored energy is consumed.

The WISP provides a flexible platform for research into RF-powered computers, and we use it extensively in our work. Our experiences writing application software for the WISP and developing approaches to running programs and communication has given us insight into how future platforms could be designed. We are currently working with the WISP inventors on the next revision of the platform.

2.3.3 Other RF-powered Computers

Passive RFID tags with integrated sensors have been developed that sense temperature [63] and humidity [50] or even glucose and pressure levels in the human eye [21]. While useful for specific applications, these devices are fixed function; sensor data is simply written to memory and the memory location is read by the reader. We believe that device flexibility is needed to support rich applications, as has been suggested by prior work [113]. For example, with hundreds of sensors in the environment, intelligent filtering becomes a necessity, and their long life means they are likely to be retasked.

One exciting far-field technology is the Wireless Ambient Radio Project (WARP) which harvests power from TV stations to run a variety of applications, and communicates using a low-power 802.15.4 radio [89]. This has the benefit of not requiring a dedicated RF transmitter as part of the infrastructure, though initial implementations require a large, high gain antenna to be manually pointed at a TV transmitter to harvest sufficient power to operate.

Chapter 3

PROBLEM AND APPROACH

In the last chapter, we motivated the use of RF-powered computers and backscatter communication to achieve deeply embedded, long lived computation and sensing, and presented necessary background material. In this chapter, we describe the two key problems that are addressed in this dissertation: running programs efficiently using harvested RF signals, and building backscatter networks that perform well at scale. For each of the two problems, we detail the key challenges and outline our approach to overcoming them.

The first problem is how to run programs using harvested RF-energy. This is challenging because the rate at which energy is harvested and consumed varies widely and can be non-deterministic, and hardware constraints further complicate strategies for matching energy consumption to available power. Programs consist of some mix of computation, sensing and communication. RF-powered computers will duty cycle between a low-power mode where energy is stored in a capacitor, and higher power active modes where energy is consumed. This extends their operational range and enables them to support energy-intensive tasks such as sensing. Ideally, when a device is close to a power source it will be able to execute heavyweight tasks quickly, and when further from a power source it will still be able to run light-weight tasks less often. However, supporting efficient task execution across a range of tasks and operating environments is challenging. Existing solutions taken by RFID and CRFID devices simplify the problem by using fixed power or energy thresholds to determine when tasks should execute. However, these approaches mean that devices can only execute fixed tasks, and require hardware modification to target devices for a particular operating point. *My approach is for RF-powered computers to adapt their behavior in response to changing task needs and available energy to achieve good performance across a range of*

scenarios; no fixed design point will work well for different tasks or at different ranges.

The second problem is how to build scalable backscatter networks. This is challenging largely because backscatter clients are not frequency selective, which leads to them being highly susceptible to interference. A good example of a wireless technology that scales well is 802.11. Deploying 802.11 networks that provide good performance is straightforward, and even non-experts can easily add APs to their network to extend its coverage. In most cases, adding APs to a deployment increases network coverage, network capacity, or both. In contrast, it is difficult to achieve this type of behavior in backscatter networks. The ultra-low power client transceiver design means that client devices are not frequency selective, which makes them susceptible to interference when multiple controllers are transmitting. Moreover, clients cannot detect other client transmissions in order to avoid them, and infrastructure nodes must transmit a high power RF signal to power and communicate with clients, which further increases interference. Existing solutions to the problem involve careful spatial deployment, which reduces network coverage because dead-spots arise where coverage regions intersect, or careful temporal multiplexing which reduces capacity. *My approach is a network design that tackles interference directly to enable backscatter networks to scale well as infrastructure is added. That is to say, fewer clients are starved due to unmitigated interference, and capacity does not decrease due to overly restrictive time multiplexing.*

3.1 Problem: Running Programs Using Harvested RF Power

Running tasks using harvested RF power is difficult because both the energy costs of tasks and the rate of energy harvesting can vary widely and be non-deterministic. Hardware considerations further complicate the goal of using energy efficiently. In this section, we describe the key challenges to running programs using harvested RF power, describe prior work, and outline our approach of adapting device behavior to match available power.

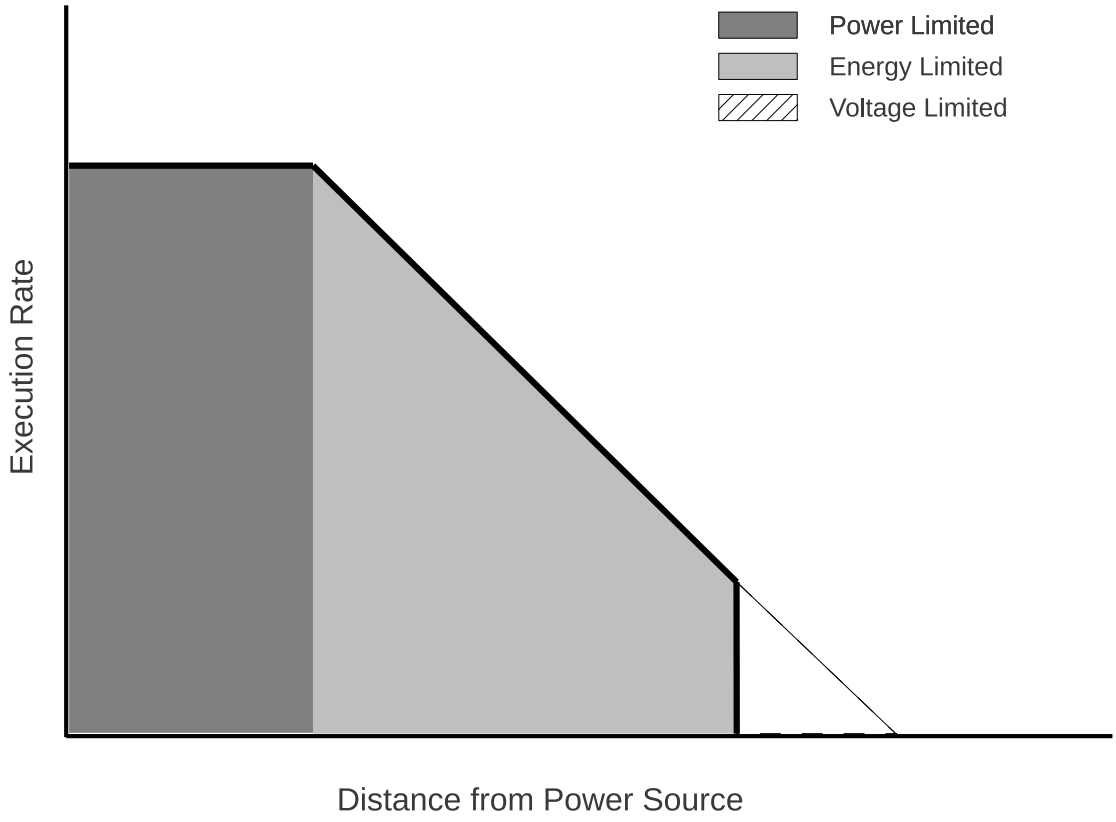


Figure 3.1: Task execution rate versus distance to illustrate the three operating regimes of RF-powered devices. Note that the execution rate is zero when in the voltage limited regime.

3.1.1 Device Operating Regimes and Duty-Cycling

In an RF-powered computer, the amount of power that is provided by the harvesting circuitry is largely determined by the distance from the source. A naive approach to using harvested power is to build devices that operate whenever the instantaneous power supplied by the harvester is greater than the power consumed by the device. Though easy to build, these devices operate at their maximum performance, or not all, which limits the range at which they can function. Another approach, which is necessary for RF-powered computers to work well across a range of RF environments, is for devices to manage their

power consumption by duty-cycling between the standby and active modes available in modern processors. As long as the harvester output power is greater than the standby power consumption, energy can be accumulated over time and then consumed when the device switches to active-mode. This enables devices to function whenever the power provided by the harvester exceeds their *average* power consumption. At longer ranges they are in active-mode less often, but can still achieve useful work.

Duty-cycling leads to three operating regimes for RF-powered computers, which are illustrated in Figure 3.1. The first regime is the *power limited* regime, where harvested RF power exceeds the active-mode power consumption of the device. In this regime, the devices can operate continuously and the task execution rate is constant; RFID tags operate only in this regime. The second regime is the *energy limited* regime, where harvested RF power is less than the active-mode power consumption but greater than the standby-mode power consumption. In this regime, the device duty-cycles to accumulate energy in standby-mode, and executes tasks and consumes stored energy in active-mode. As the input power decreases with distance, the device spends more time accumulating energy and the task execution rate decreases. The last regime is the *voltage limited* regime, where harvested RF power is not sufficient to power the device even in standby-mode. *As the* power limited *regime is similar to conventional computing, and the* voltage limited *regime can only be addressed by hardware improvements, this dissertation focuses on techniques that improve performance in the second regime.*

3.1.2 Challenge: Varying Task Needs

I define a *task* to mean a short program that is run to completion without pause. A task may be as simple as a few lines of code to sample a sensor, or may be a complex program that samples a sensor, encrypts the measurement, and communicates the result to the network. The key to the definition is that tasks have run-to-completion semantics, though tasks can be composed to achieve more complex application goals. An RF-powered device should not

start a task unless there is (likely) sufficient energy to complete it, as failing a task consumes energy without doing useful work. Yet, predicting whether a task will succeed is difficult because task energy requirements vary greatly due to two main factors.

Different size tasks. The energy consumption of a task depends on the sensors it uses, the computation it performs, and its communication pattern. Reading a sensor can consume very little energy, in the case of an on-chip temperature sensor, or a very large amount, for instance reading a CCD to capture an image. Programs need to checkpoint their progress because they may lose power at any time, and writing checkpoint data to persistent memory must run to completion. Moreover, the energy cost of writing to persistent memory can be prohibitive, so many tasks will require that sensing, computing, and communication all take place in one burst. This means that task size can vary from very small and lightweight, to very large and energy intensive. RF-powered computers need to support a wide range of tasks, so that they can be reprogrammed to support different applications. As an example, consider an application where devices sample a light meter, and when a light is turned on they begin capturing images and streaming frames back to the network. In this case, even a single program will have widely different energy needs depending on its state.

Non-deterministic tasks. Tasks may be non-deterministic, which causes their energy requirements to vary from execution to execution. Even sampling a sensor can consume different amounts of energy based on the state the sensor is in, the tolerance for inaccuracy, or even the temperature of the surrounding air. Also, communication protocols generally require that a series of messages be exchanged, e.g., the data-ACK sequence of 802.11, and the number of messages can change based on contention for the medium. As a consequence of the way these protocols work, a device that attempts to communicate must have sufficient energy stored to complete the transaction, or energy will be used but no useful work will be accomplished.

Supplemental power during execution. RF-powered computers harvest energy even when the task is being executed. Consequently, if a device is close to an RF source, less

energy needs to be stored before execution can begin. To achieve good performance, devices must consider both the amount of stored energy and the rate at which energy is currently being harvested when deciding if a task can begin execution. If supplemental power is not considered, tasks will not execute as often as they could.

3.1.3 Challenge: Varying Input Power

Even assuming that a device could accurately estimate the energy cost of executing a task, it is difficult to know how long to sleep to store sufficient energy because the rate at which a device harvests energy changes over time.

Widely varying input powers. RF power received at a device decreases at least as fast as the square of its distance from the controller. In practice, this means that the available energy varies by orders of magnitude over useful ranges. Hard-wiring devices to operate at the low end of the power scale wastes a significant opportunity at the high end of the scale, and restricting devices to operate at the high end of the scale limits operational range.

Multipath fading. Multipath fading occurs when multiple copies of a transmitted signal interfere at a receiving antenna. The copies of the signal can either constructively interfere, and the received power will increase, or destructively interfere which decreases the received power. This makes it difficult to predict the rate at which energy can be harvested for two reasons. First, due to regulatory and other reasons, many radio systems frequency hop, i.e., change the frequency at which they transmit, every few hundred milliseconds. Because the interference pattern for a given space depends on the wavelength of the transmitted signal, frequency hopping results in drastic changes in harvestable power over short time scales.

3.1.4 Challenge: Platform Inefficiencies

The variation in task energy requirements and harvestable power suggest that a good strategy might be to overestimate the task needs. For example, a device could harvest energy until its buffer is completely full before executing a task. In this way, it would run with “a

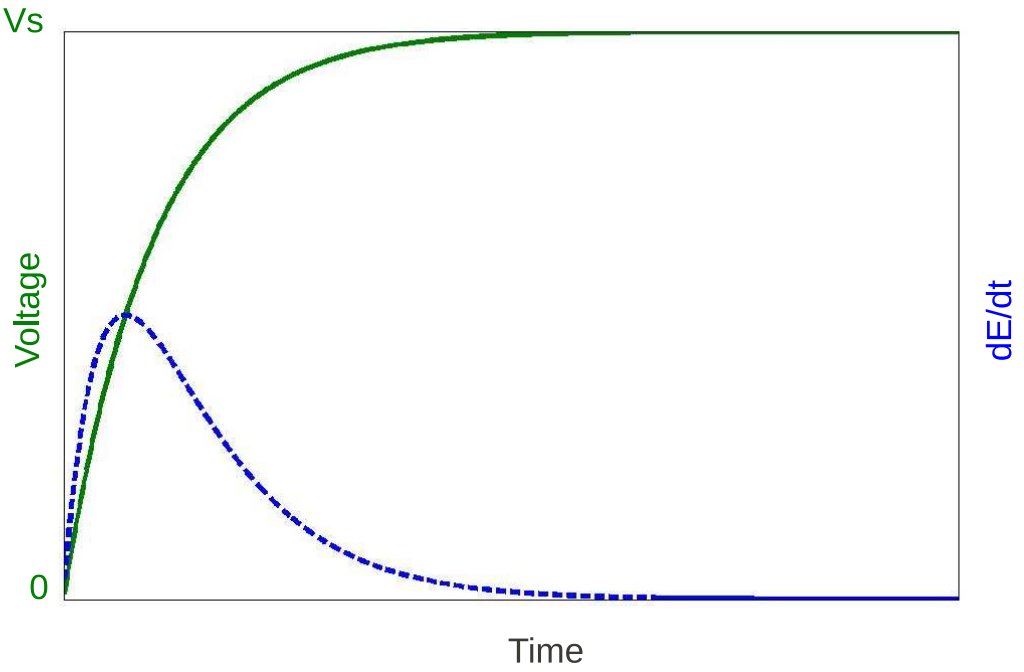


Figure 3.2: Plot of voltage at time t and the derivative of energy with respect to time at time t for a capacitor being charged from a constant source with voltage Vs.

full tank" to avoid preventable failures and top off between tasks. Unfortunately, storing excess energy is wasteful due to platform characteristics. The exact inefficiencies will vary with the device implementation, but we believe that all real platforms will have these kinds of inefficiencies. The implication is that a joule of energy may take longer to harvest, or be consumed more quickly, depending on the state of the energy store.

Sublinear charging.

RF-powered computers use capacitors for energy storage as they are well suited to energy harvesting devices [51]. They charge quickly, recharge indefinitely, are small and inexpensive, and are non-toxic. However, the rate at which capacitors store energy depends on how

much energy is already stored. This nonlinearity is fundamental to the way capacitors work when being charged by a constant voltage source such as a power harvester. Equation 3.1 shows the rate at which the voltage of a capacitor (V_c) increases with time, where V_s is the voltage of the source, C is the capacitance of the capacitor, and R is the resistance of the circuit.

$$V_c(t) = V_s(1 - e^{-t/RC}) \tag{3.1}$$

Figure 3.2 plots this function (solid green curve) for an arbitrary V_s, R and C; the exact values impact the axes labels but not the shape of the curve. As the capacitor voltage rises with increasing charge, V_c approaches V_s. However, the rate at which V_c approaches V_s decreases as it gets close to V_s. A corollary of Equation 3.1 is the rate at which energy is stored on a capacitor, where $E = \frac{1}{2}CV^2$. To see how this changes with time, we plot the derivative of energy with respect to time on Figure 3.2 (dashed blue curve) along side the voltage curve. As can be seen, the rate at which energy is stored on the capacitor peaks when V_c reaches $\frac{1}{2}V_s$, and decreases dramatically as V_c approaches V_s.

The implication is that it is inefficient for a device to operate at voltages above $\frac{1}{2}V_s$, as it takes longer to recover a fixed amount of energy after it is consumed. Though emerging power harvester designs aim to increase efficiency across voltages [124, 92] they do not eliminate the problem.

Superlinear discharging. RF-powered devices must regulate the supplied (input or stored) voltage to the operating voltage. Differences in voltage levels inevitably lead to some voltage-dependent conversion losses. For example, the WISP CRFID prototype uses a linear regulator that sheds the voltage difference by dissipating energy as heat. Other techniques are possible but come with their own tradeoffs (e.g., switching regulators are more efficient at stepping down the voltage, but have greater leakage current, don't work when the input voltage is near the target voltage, and are inefficient when they first start up). To minimize energy wasted while discharging, the device again should operate with

the capacitor as lightly charged as possible.

Trade-off between cold-boot time, and harvesting efficiency and range. Operating with a higher capacitor voltage is inefficient because energy is harvested more slowly and may be consumed more quickly. One way to operate more efficiently is to increase the size of the storage capacitor so that more energy is stored for a given voltage. One drawback to this approach is that increasing the capacitor size linearly increases the cold-boot time; e.g., doubling the size of the capacitor doubles the time it takes for the capacitor to charge to a useful voltage. This means that, aside from scenarios where a constant RF source can be guaranteed, the size of the capacitor is a key hardware design question. There are two trade-offs: 1) the cold-boot time which determines how quickly even the smallest task can be executed, versus the energy consumption efficiency for larger tasks 2) the cold-boot time time versus the range at which a given task can run (as input power decreases, the voltage provided by the harvester decreases as well).

3.1.5 Existing Solutions for Running Programs Using Harvested RF Power

UHF RFID is the only widely deployed, RF-power harvesting technology of which we are aware. The current UHF RFID standard, Gen 2, takes the approach that received power must always exceed active-mode power consumption (i.e., Gen 2 RFID tags do not duty cycle). If tags are receiving enough power to turn on, they can operate indefinitely until the received power falls below their operating threshold, in which case they do not operate at all. RFID tags do use a very small capacitor to store energy, but only to support the device during the few microsecond "off" periods of the ASK modulated reader data transmissions. The drawback to the RFID approach is that it cannot support rich tasks at range, there generally is not enough power available to, for example, drive power-intensive sensors. In other words, RFID tags are power limited, in that their range is limited by the amount of power they can rectify which must be greater than their active-mode power consumption.

The other widely available RF-powered device is the Wirelessly Identification and Sens-

ing Platform (WISP) developed at Intel Labs and the University of Washington. The WISP is a prototype CRFID device that is in use at dozens of universities, and it is the primary platform used for RF-powered computing research. WISPs extend the operating model of RFID tags by introducing duty-cycling as a way to support energy-intensive tasks. Power is harvested and stored in an onboard capacitor when the device is in low-power mode, and when a fixed threshold voltage is reached the device wakes up and consumes the stored energy. To support a more energy-intensive task, a WISP can be outfitted with a larger capacitor which will store more energy for a given voltage, and prior work has explored the problem of how to choose the right capacitor size for a given task [40]. Whereas RFID tags are power limited, WISPs are voltage limited in that their range is limited by the distance at which they can rectify sufficient voltage to reach their wake-up threshold. However, the wake-up threshold and capacitor size, which determine the maximum range and task size they can support, are fixed at the time of manufacture. Our prior work has found that this one size fits all approach limits the kind of tasks that can be run [16], so most experimental work has used WISPs around 1 meter away from an RFID reader.

3.1.6 Approach: Adapting Device Behavior to Match Available Power

I argue that RF-powered computers must adapt their behavior in response to changing task needs and available energy to achieve good performance across a range of scenarios; no fixed design point will work well for different tasks or at different ranges. This means devices must store different amounts of energy and run tasks at different rates depending on the task and the operating environment. To achieve this, we view the need to match harvested energy to task consumption as a scheduling problem. Devices must take into account not only available power, but also hardware constraints in order to run tasks when they are likely to execute efficiently.

By making the best use of available energy, this adaptive approach enables devices to run a range of tasks efficiently in a variety of operating environments. Specifically,

when harvested power exceeds active-mode consumption, the device operates continuously without storing any energy to maximize the execution rate. When harvested power exceeds the standby-mode consumption, the device automatically finds the duty cycle that best matches the available energy to the energy needs of the current task. At the limit of their range, devices find the minimum amount of energy that must be stored for a task to complete. Devices will only fail to execute tasks when there is insufficient voltage to operate.

In Section 4, we apply this approach in the context of CRFIDs. Using WISPs as an experimental platform, we implement Dewdrop, an energy-aware runtime for CRFIDs. By waking tags at the right times, Dewdrop can run tasks that previously could not run, and at close to their maximum rate given the energy that the RF environment provides.

3.2 Problem: Building Scalable Backscatter Networks

Building backscatter networks that scale is difficult because interference degrades performance when multiple controllers are present; this is largely because clients are not frequency selective. For backscatter to be a viable approach for RF-powered networks, it needs to provide general purpose communication that is easy to deploy and can scale gracefully to a large number of nodes. Extending the coverage of a network should be as easy as deploying an 802.11 AP, and deploying additional APs should also increase the throughput of clients; adding infrastructure should not result in clients *losing* connectivity. However, today's commercial backscatter technologies do not provide even this level of functionality. Moreover, it is inherently difficult to achieve these goals using existing designs due to the nature of interference in backscatter communication [30, 110]. The root cause of the interference problem is that client transceivers are broadband devices; this is a consequence of their low-power design. On the downlink, they cannot separate signals from controllers transmitting on different frequencies. On the uplink, they backscatter all RF signals in the environment. This means that backscatter networks cannot be partitioned into sub-networks that operate on difference channels, as is the case with 802.11 or cellular networks.

In this section, we first describe the operating model of the networks we envision. We next describe the challenges associated with interference in backscatter networks and prior approaches to solving these challenges. Finally, we give an overview of our network design that addresses the sources of interference on both the downlink and the uplink.

3.2.1 Operating Model

The networks we envision consist of large-numbers of RF-powered devices, which we refer to as clients, and some number of infrastructure nodes called controllers that power the devices and provide them with connectivity to the Internet. To maximize power transfer to nearby clients, controllers transmit a continuous carrier wave at the highest power and for as much of the time as possible. Along with providing power to the clients, this continuous wave is modulated to send messages to clients, and is backscattered by the clients to communicate with the controller. Controllers are deployed to provide power and communication to building-sized service areas, and consequently the transmission range of multiple controllers will often overlap. Dense deployment on the scale of a room may also be desirable as additional controllers increase the power available to clients. Lastly, controllers may be in different administrative domains, as is the case with 802.11 APs deployed by different tenants of an apartment building. This means that careful physical deployment cannot be assumed, nor can a backchannel for tight coordination between controllers.

Traffic in the network can flow in both the uplink and downlink direction. However, at least in the near-term, we expect most traffic to be in the uplink direction, as clients are more likely to provide sensor data than to view webpages. Clients can join or leave the network at any time, and we do not assume that traffic volume or timing can be known a priori. Unlike RFID applications, where only temporary identifiers are used by the MAC protocol, clients have persistent IP addresses and associate with a single controller much like 802.11 networks.

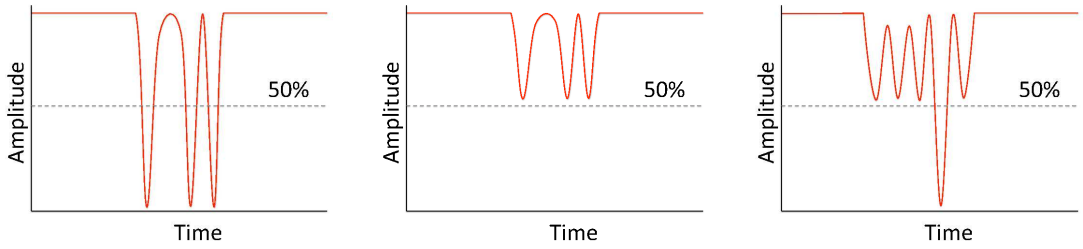
3.2.2 Challenge: Controller-Controller Interference

The key challenge to building scalable backscatter networks is overcoming interference. The nature of backscatter communication and the low-power client transceiver mean that conventional approaches to mitigating interference often cannot be applied. One approach that does apply is that controllers use analog and digital filters to suppress controller transmissions on different channels. However, controller-controller interference can still occur when a client is backscattering a message to a controller (Controller-A) and a second controller (Controller-B) begins transmitting on the same channel; i.e., two controllers are transmitting in the same frequency band. The backscattered client signal and the much higher power signal of Controller-B interfere at Controller-A and the client message will be not be received by Controller-A.

Interference caused by nodes transmitting on the same channel is a problem in any wireless network. However, the problem is compounded in backscatter networks for two reasons. First, controllers must transmit a continuous RF signal to power the clients, to transmit messages to them, and also to receive messages from them. Second, the controller-controller interference range can be more than a kilometer [56] while the range at which an RF-powered device can be powered and communicate is only 10s of meters. *This means that the the common case in dense networks is to have multiple high power transmissions that can interfere at long distances with the weak backscattered client signals.*

3.2.3 Challenge: Controller-Client Interference

Controller transmissions can also interfere at clients. *Because client demodulators are simple envelope detectors, controller signals interfere at clients even if the controllers are transmitting on different frequencies.* Figure 3.3 shows a diagram of three signal envelopes at a client. The interference-free case is shown in Figure 3.3(a); a bit is detected whenever the amplitude drops below the 50% threshold. The second case is when interference consists only of controllers transmitting continuous wave signals, and the latter case is when two



(a) No Interference (b) Continuous Wave Interference (c) Modulated Interference

Figure 3.3: Diagram of controller-to-client interference as seen at the client. The dashed line indicates a 50% detection threshold.

controllers are modulating simultaneously. To the best of our knowledge, no prior work distinguishes between interference caused when a interferer transmits only a continuous wave, and when the interferer is modulating data. However, the two cases are different and must be considered separately, as described below.

Continuous Wave Interference

Controllers transmit data by modulating the amplitude of the CW, usually approximating On/Off keying which means the modulation depth approaches 100%. If multiple controllers are transmitting simultaneously, the power of the signals sum at the client. This means that when one controller is modulating a message to a client and other controllers are transmitting CWs, the off period of the modulating client will only reduce the total power seen at the tag by some fraction; this is shown in Figure 3.3(b). If the modulator's signal accounts for 50% of the total power at the client, and it modulates messages with a 100% modulation depth, the depth seen at the client will be only 50%; this may not be deep enough for the client to detect. We refer to this as continuous wave interference. A key design question for client demodulators is the sensitivity, with more sensitive demodulators being able to decode shallower modulation depths.

Modulated Interference

The second type of controller-client interference is modulated interference, which occurs when multiple controllers modulate messages at the same time; this is shown in Figure 3.3(c). If two controllers account for equal power at the client, and they modulate simultaneously, there will be three amplitude levels seen by the client: the on periods of both combined (full amplitude), the on period of one and the off period of the other combined (50% modulation depth), and the off period of both clients (zero amplitude). Because the two signals are mixed together, the envelope detector of the client cannot separate the signals and both messages are lost. Again, the sensitivity of the client demodulator comes into play. If the client ignores all but very deeply modulated messages, a weaker modulation may be ignored. Imagine a case similar to that stated above, but with one client accounting for 9x the power of the interferer. The client would experience: the on periods of both (full amplitude), the on period of the stronger signal and the off period of the weaker (10% modulation depth), the on period of the weaker signal and the off period of the stronger signal (90% modulation depth), and the off period of both (zero amplitude). If the client only detects modulations deeper than 10%, the weaker modulator will be ignored and the stronger signal will be decoded error-free.

As mentioned previously, controllers transmit CWs to transfer power to tags, to transmit messages to them on the downlink, and also to let them backscatter messages on the uplink. *Consequently, continuous wave interference is the common case in large-scale networks, and its impact must be mitigated for these networks to scale well. Though increasing the sensitivity of the demodulator can mitigate the effect of continuous wave interference, it comes at the cost of increased sensitivity to modulated interference.*

3.2.4 Challenge: Client-Controller Interference

Client-controller interference occurs when multiple clients backscatter at the same time and their signals collide at the controller. Because each controller transmits its own CW,

a modulating client will backscatter all CWs present in the environment. Consider the case where a client (Client-A) is backscattering a message to Controller-A, and a second client (Client-B) begins backscattering a message to a second controller (Controller-B). Controller-A cannot filter out the transmission of Client-B because the client is modulating Controller-A's CW also. Another way to think about the problem is that if many controllers are transmitting CWs on different channels and one client is backscattering, the client transmissions will appear in every controller's channel. *Hence, clients will interfere at controllers even if controllers are operating on different frequencies.* Mitigating this type of interference is complicated by the fact that the ASK demodulator of the clients cannot detect the transmissions of other clients beyond a few centimeters [73], which precludes the use of CSMA-based approaches.

There has been significant work on anti-collision protocols by the RFID community [57]. These protocols are often based on Slotted-Aloha [86], which was designed for environments where clients cannot hear each other. These protocols work by having the controller arbitrate client access to the medium. *However, existing protocols only arbitrate access between clients that are transmitting to a single controller. They do not coordinate behavior between controllers to handle clients that are transmitting to different destinations.*

In summary, there are three characteristics of backscatter systems that make the interference problem different than that seen in conventional wireless networks:

- The ultra-low power transceiver of clients comes at the expense of frequency selectivity in both the uplink and downlink directions. Clients transmit uplink traffic by reflecting the CW of a controller, and any modulating client will scatter all incident CWs. In the downlink direction, the simple ASK envelope detector is not frequency selective, so all controller transmissions will interfere at all clients within range.

- Controllers transmitting a high power continuous wave to provide power to clients, to transmit messages to them, and also to receive messages from them. This means that continuous wave interference is the *common case* in RF-powered networks. How-

ever, even simply transmitting CWs can cause continuous wave interference which results in clients that are well-powered but that cannot demodulate messages from any controller.

- Clients cannot detect the transmissions of other clients beyond a few centimeters, so controllers must arbitrate client access to the channel. This is challenging when the multiple controllers are operating in the same environment under different administrative domains.

3.2.5 Existing Solutions to Overcoming Interference in Backscatter Networks

There are two approaches to managing interference seen in the literature: careful spatial deployment, and time multiplexing in the forms of time-division-multiplexing (TDM) or carrier-sense-multiple-access (CSMA). The first approach is to carefully deploy controllers to enforce non-overlapping coverage areas. In Gen 2 RFID systems, readers continuously query for tags. Though readers frequency hop to avoid controller-controller interference, and they arbitrate channel access for their tag population, they do not attempt to avoid interference between readers at tags, or between tags at multiple controllers. To mitigate interference, Gen 2 deployments use directional antennas and careful physical placement to avoid regions of overlap where interference can take place at tags. Dock door applications seen in warehouse environments, a key application space for RFID, have a single reader covering one dock door, and complete coverage of the warehouse floor is not desired.

Another way to avoid interference is for controllers to multiplex their transmissions in time. Time-division-multiplexing has been proposed where controllers are carefully scheduled to not transmit concurrently if they will interfere. Many commercial readers, such as those used by the University of Washington's RFID Ecosystem [115], the only building-wide Gen 2 deployment of which we are aware, use a rudimentary form of TDM where only one in four reader antennas transmits at a time and they alternate in a round-robin fashion. While this avoids interference and assures good coverage in the building, it comes at the cost of

reduced throughput and one fourth as much power available for harvesting. More complex TDM mechanisms have been proposed [10], but these are difficult to implement in practice and generally assume a high degree of coordination between controllers [43, 111], additional hardware [6], or knowledge of the location and traffic patterns of clients [112, 22]. Also, because prior work does not distinguish between continuous wave interference and modulated interference, they constrain the network such that only one controller can be transmitting RF energy of any kind in a given area which reduces network capacity.

A simpler approach to time multiplexing is CSMA, where devices listen to the channel before transmitting and defer their transmission if the channel is occupied. An example of this is the Listen-Before-Talk mechanism used by RFID systems operating in Europe [32]. However, the Listen-Before-Talk mechanism described in [32] permits concurrent transmissions on different channels, which means it does not avoid interference at the clients, or between clients at multiple controllers.

3.2.6 Approach: Tolerate Continuous Wave Interference and Enable Concurrent Client Transmissions via CDMA

My approach is to tolerate continuous wave interference, as it enables multiple controllers to power and receive messages from clients at the same time. To do this, we leverage the insight that the client demodulator design can trade off sensitivity to continuous wave interference against sensitivity to modulated interference. We propose that client demodulators be designed to tolerate continuous wave interference, and that controllers are responsible for avoiding modulated interference.

To arbitrate access between clients communicating with a single controller, our design uses a polling-based MAC protocol. This is more appropriate than Slotted-Aloha based approaches for the applications we envision, as clients have persistent addresses. However, client-controller interference can still occur when clients are transmitting to different controllers. To mitigate this problem, we propose code-division-multiple-access (CDMA) on

the uplink to enable controllers to decode their client's messages even in the presence of other client transmissions. By combining these techniques, our design allows backscatter networks to scale well as infrastructure is added. That is to say, fewer clients are starved due to unmitigated interference, and capacity does not decrease due to overly restrictive time multiplexing.

3.3 Summary

There are two problems we tackle in this dissertation. The first problem is how to run programs efficiently using harvested RF energy. To match their average power consumption to available power, RF-powered computers duty cycle between a sleep mode where energy is harvested, and active modes where stored energy is consumed. The problem then becomes how to manage the duty cycle to run programs efficiently, and to assure that devices do not run out of power while executing tasks. Achieving this is difficult because tasks can vary widely in size, and can be non-deterministic in their energy usage. Moreover, the amount of power available for harvesting can vary widely and be non-deterministic. To further complicate matters, fundamental hardware characteristics mean that energy is harvested and consumed less efficiently when excess energy is stored. We propose an approach where devices take these factors into account and adapt their behavior to run tasks efficiently given the available power. In the next section, we present *Dewdrop*, a runtime for CRFIDs that demonstrates the benefit of our approach in the context of CRFID. We show that *Dewdrop* is able to complete tasks where they could not previously complete, and at close to the best rate given the available RF power.

The second problem we tackle is how to build backscatter networks that scale as infrastructure is added. We want these networks to support conventional communication patterns, and to be easily extended by simply adding infrastructure nodes. However, this is difficult with backscatter communication, and existing technology can cause starvation of clients (Gen 2) or reduced capacity (TDM) as infrastructure is added. This is a result of the challenging interference problem inherent to backscatter communication. For example,

client transceivers are not frequency selective on either the up or downlink, and clients cannot hear the transmissions of other clients in order to avoid them. A key distinction, which has not yet been explored, is that continuous wave interference and modulated interference on the downlink are different phenomena. Leveraging this insight, we propose that client demodulators be designed to tolerate continuous wave interference as this allows multiple controllers to transfer power to and receive messages from their associated clients simultaneously. Controllers are then responsible for avoiding modulated interference. Client access to the channel is achieve using time multiplexing for clients transmitting to a single controller, and via CDMA for clients transmitting to different controllers. This network design enables backscatter networks to scale gracefully as infrastructure nodes are added.

Chapter 4

DEWDROP: AN ENERGY-AWARE RUNTIME FOR COMPUTATIONAL RFID

In Section 3.1, I described the challenges to running programs on RF-powered computers. Namely, the energy needs of tasks and the amount of available energy can very widely and be non-deterministic, and hardware constraints mean that the rate that a joule of energy is stored or consumed depends on how much energy is already stored. I then outlined my approach of having devices adapt their behavior so that they can efficiently run a range of tasks across a range of environments. In this chapter, I present the design, implementation, and evaluation of Dewdrop, an energy-aware runtime that applies this approach to Computational RFID (CRFID).

CRFIDs are a particular point in the larger design space of RF-powered devices, but they are the only design point that has a widely available research prototype: the Intel/UW WISP. As such, we use WISPs to demonstrate our adaptive approach to task execution on RF-powered devices. Prior work [16] has found that running programs on WISPs can be difficult, because they often start a task but run out of energy before completing it. For example, in our experience with an activity recognition system built using WISPs [12], we found that the WISPs would often take sensor readings but run out of energy before transmitting them to the controller, or would begin communicating with the controller but run out of energy before completing the transaction. This means the devices were consuming energy but not achieving useful work. The root cause we discovered was the WISP's simple approach to task execution: a fixed amount of energy was being stored before starting the task, independent of the task being executed. The program we installed on the devices often consumed more energy than this nominal amount, so the task often failed.

In contrast, Dewdrop adapts its duty cycle so that tasks run efficiently, no matter how

much energy they consume. If sufficient energy can be stored to run the task, the task is guaranteed to run. Moreover, the task will run about as often as possible given the RF environment. In the following section, I describe the task model and scheduling goal for CRFID. I then present the design of Dewdrop, and its evaluation which shows that Dewdrop generally achieves 90% of the best possible rate for a range of tasks, and results in both improved network coverage and higher response rates.

4.1 Task Model and Scheduling Goal

In keeping with other work seen in the literature, we assume that a CRFID repeatedly executes a single fixed task as often as possible (e.g., reporting a sensor value), but from time to time may be retasked to perform a different operation (e.g., switch from sampling the accelerometer to measuring the light level). Additionally, devices in a deployment may be executing different tasks. As a device considers only one type of task at a time, scheduling the order and execution of multiple tasks on a single device is both unnecessary and out of scope.

We define a *task* to mean a short program that is run to completion without pause. While it may be possible to break some tasks into phases, the timing requirements of the device hardware, the EPC Gen 2 protocol, and application requirements make it impractical to interrupt many tasks once they start.

4.1.1 Task Scheduling Goal

Given that devices repetitively execute a task whenever possible, maximizing energy efficiency is equivalent to maximizing the rate at which tasks successfully complete. We use task completion rate, in terms of how many task iterations succeed over a given time period, as a metric to evaluate the performance of *Dewdrop* in the steady state. Since energy falls off with distance (at least as quickly as distance squared), we expect the completion rate to fall with distance. But, it should not fall more quickly than the available energy.

RF-powered computers like CRFIDs collect the energy harvested from RF signals into

a capacitor. The default behavior for the CRFIDs, and other RF-powered computers we have seen in the literature, is to begin task execution whenever a fixed, hardware-defined power level is reached. Once a task iteration has started, it may either run to completion or fail if the device runs out of energy first [1]. We use this fixed, hardware approach as a baseline for comparison in our evaluation.

Dewdrop replaces the fixed, hardware approach with an adaptive software strategy. There is only one decision that a device can make to improve energy efficiency: to defer the start of a task it could otherwise begin, sleeping until the energy store becomes more full. This is useful because the larger store of energy increases the chance that the task will run to completion. However, it is wasteful in terms of time and energy if the task would have succeeded anyway. The runtime's job is to decide when to run and when to sleep depending on the task and RF environment.

4.2 Dewdrop Design

We now develop the design of our energy-aware runtime, *Dewdrop*. The main scheduling decision is when to start the next task iteration. Starting too soon wastes energy when the device runs out of power and the task fails. Starting too late collects excess energy, which is inefficient to both store and use. Our approach is to minimize both forms of waste.

4.2.1 Design Goals

From our problem formulation, the overarching goal of *Dewdrop* is to convert all available energy into completed task iterations. This goal is equivalent to two sub-goals that help to enable new applications:

Increased range. We want our runtime to execute a task at greater distances from the controller than the baseline WISP hardware. Each task should work from next to the

[1] The WISP effectively runs out of energy when the voltage across the capacitor drops below 1.5V, the minimum required to maintain state on the microprocessor.

controller out to the distance at which the device can no longer harvest enough energy for the task.

Improved responsiveness. At all distances, we want to increase responsiveness compared to the baseline WISP hardware. We never want to noticeably decrease responsiveness.

Both goals are met by maximizing the task completion rate for a given task and distance from the controller. In practice, achieving them implies that we must meet two other goals:

Low overhead. The implementation of *Dewdrop* must be extremely lightweight. Operations such as checking the level of the energy store or calculating sleep periods consume scarce energy. Even a modest amount of overhead can easily negate the benefits of scheduling tasks.

Adaptation. Devices must operate well across a range of deployment scenarios. For example, they may be configured to run either heavy or lightweight tasks, and they must run their task efficiently both when near and far from a controller. Our performance subgoals are stated across these factors, so *Dewdrop* must adapt to the environment at runtime.

4.2.2 Variation in Task Costs

To predict when to start a task, *Dewdrop* must estimate how much energy the task will need over and above the energy that will be harvested by the tag while it runs the task. This depends on the factors we previously identified: the task itself, other tags competing for the medium, the distance from the reader and the frequency on which the reader is transmitting, and the amount of energy already in the capacitor. All of these factors are fundamental. However, they may differ in magnitude with implications for system design. For example, if the energy needs depend mostly on the type of task, then each task could be profiled offline to characterize its fixed energy need.

To understand how much these factors matter in practice, we ran an experiment with two tasks running on the WISP: Sense (which just samples a sensor) and SenseTx (which

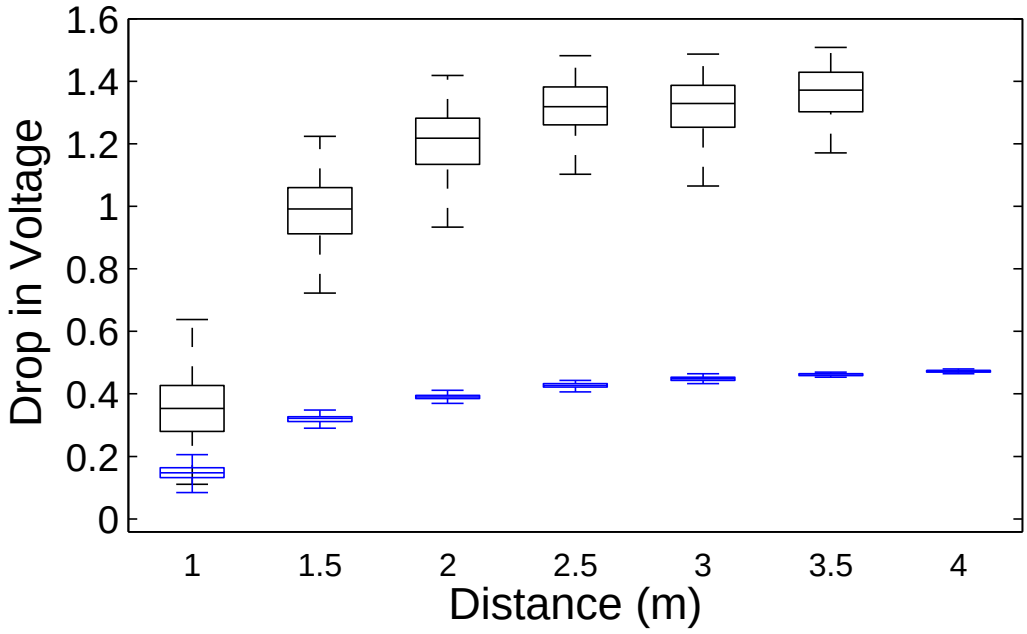


Figure 4.1: Voltage drop for SenseTx (upper black items) and Sense (lower blue items).

additionally transmits the sensor value to the controller). For the WISP, the energy consumption of a task can be measured by the drop in the voltage of the capacitor that acts as a short-term energy buffer[2]. Figure 4.1 shows this voltage drop as a function of distance for the two tasks. Box plots show the distributions over at least 300 task executions at each distance.

The Sense task is deterministic. However, we see that the voltage drop is significantly larger when the tag is far from the reader than when it is close to the reader; it more than triples. This is because the input power from the reader varies by more than an order of magnitude. A second effect is that the variance is larger when the task is run close to the reader because the input power supplements stored energy and varies with the reader transmit frequency. At 1m this variance is approximately 0.3V compared to 0.1V at 4m.

[2]The energy stored in a capacitor is calculated as $\frac{1}{2}CV^2$, where C is the capacitance and V is the measured voltage.

Looking at the SenseTx task, the drop in voltage is almost three times larger than for Sense. At 4m, the WISP cannot store sufficient energy to execute the task[3]. The variation is also higher at all distances because this task is non-deterministic. Its energy consumption depends on randomization in the Gen 2 MAC protocol, and the variation would be even greater if there were multiple WISPs (which we study as part of our evaluation).

These results support our argument that devices must adapt to both the task and the environment in which the device is operating. Any fixed energy target at which to start a task will be either too low, causing the tag to fail at a distance when it could still run, or too high, causing the device to run tasks more infrequently than it is capable of sustaining. A second implication is that it is likely not feasible to accurately estimate the energy needs of a particular task execution due to inherent variation. Consequently, *Dewdrop* must adapt an estimate of energy needs that captures the effects of the distribution.

4.2.3 Minimizing Wasted Energy

Sources of waste. Energy is wasted when the device starts too early and fails to complete the task, or waits too long and inefficiently collects excess energy. How much energy is wasted in these cases depends on how well the device converts RF power into harvested energy and how efficiently this energy is consumed.

To gain some insight, we performed a simple experiment by charging a WISP without running any task. Figure 4.2 shows the voltage of the WISP capacitor as it charges at different distances. (The RF source powers on at approximately 200 ms.) This is the expected behavior. A capacitor's charging rate decreases by a factor of e every RC seconds, where R and C are the resistance and capacitance of the RC circuit and e is the base of natural logarithms, and asymptotically approaches zero as the capacitor charges to the voltage of the power source.

This charging behavior has two implications. First, it shows the effects of distance. Far

[3]To even run the task over a range of distances we needed to modify the baseline WISP behavior.

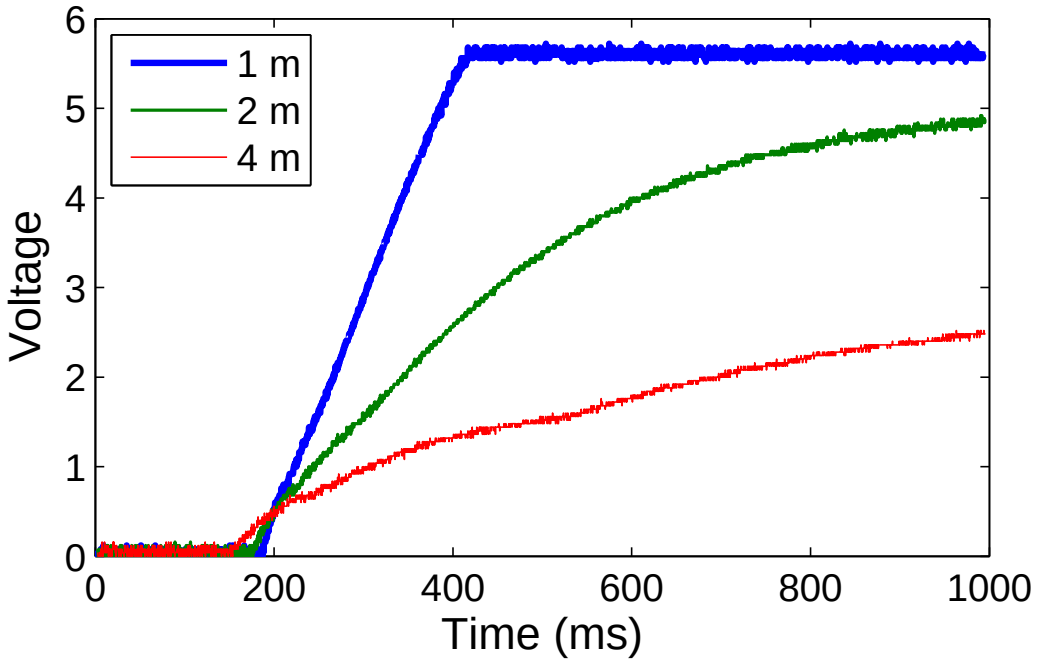


Figure 4.2: WISP capacitor voltage over time

from the controller, the low received power limits the maximum energy that can be stored [4]. At 4m the capacitor approaches only 2.75V, while at 1m it rises quickly to 5.8V (at which point an over-voltage protection circuit kicks in). This means that heavy tasks will not run as far from the controller as lightweight tasks no matter how long the device sleeps.

The second implication is that, even for a fixed input power, it is inefficient to charge to a higher voltage than necessary. Because the rate at which energy accumulates in a capacitor decreases exponentially as it charges, storing excess energy wastes *time*. There is a penalty for charging too high and leaving spare energy in the capacitor. In a sense, that leftover energy was "cheaper" to store. This effect is magnified by the linear regulator of the WISP, which consumes more power when there is a higher charge on the capacitor.

To capture these factors, *Dewdrop* estimates waste in terms of time, and measures ef-

[4]The energy stored in a capacitor is calculated as $\frac{1}{2}CV^2$, where C is the capacitance and V is the measured voltage.

ficiency in terms of how often a task can execute. This accounts not only for the energy consumed by a task, even if it fails, but also accounts for how long it took to store that energy. While the constraints of the hardware will differ, all platforms are likely to have nonlinearities with respect to storing and consuming energy that make it useful to measure waste in terms of time. For instance, capacitors are the natural choice for short-term energy storage, and all RF-powered computers that use capacitors will have this kind of inefficiency.

4.2.4 Balancing sources of waste

Intuitively, starting tasks later, at a higher energy level, will decrease the time wasted due to tasks failing but increase the time wasted due to excess charging. Our goal is to minimize the total wasted time due to both causes. Since the energy cost of executing a task cannot be estimated precisely, *Dewdrop* aims to reduce the expected wasted time in the following manner. Let $P(fail|V_s)$ be the probability that the task will fail given a starting voltage level V_s. The runtime's job is to choose a V_s in the range $[V_0, V_{max}]$ that minimizes the wasted time :

$$\begin{aligned} t_{wasted}(V_s) &= P(fail|V_s)t_{under} \\ &+ (1 - P(fail|V_s))t_{over} \end{aligned}$$

where t_{under} is the time to charge back to V_s after a failure (plus the time the task executed before failing) and t_{over} is the time spent overcharging, i.e., the time spent charging beyond the energy level that would have been sufficient. Note that this implies that some rate of failures may be desirable as charging high enough to assure success incurs a penalty that accumulates on every execution. This is particularly the case near the limit of the operating range when voltage and stored energy rise slowly.

A naive approach to finding the V_s that minimizes wasted time would be to try every value of V_s. This is impractical, as the device would need to examine a sufficiently long series of task execution attempts at each V_s to determine which had the best performance,

or whether V_s needed to be increased or decreased. Furthermore, this search would need to be repeated periodically as the RF environment and other factors change.

To avoid this search, we use our intuition that the two kinds of wasted time tradeoff against each other to find an approximate solution. Devices can adjust V_s in the correct direction at every task iteration. Let P_f be the current task failure rate at a fixed starting voltage V_s and $T_{under} = P_f * t_{under}$ and $T_{over} = (1 - P_f) * t_{over}$. If $T_{over} >> T_{under}$, then the runtime is too conservative; it could have chosen a lower V_s. If $T_{under} >> T_{over}$ then it is being too aggressive; V_s is too low and tasks are failing too often.

Dewdrop uses the heuristic that balancing the two sources of waste tends to minimize overall wasted time; this at least finds a reasonable operating point by ensuring that neither factor is a major source of inefficiency. Additionally, tracking and comparing the two sources of wasted time requires minimal computation which is key for any viable solution. The balance point can be found by slowly updating V_s to trade T_{under} against T_{over}. To do this, *Dewdrop* maintains separate estimates of T_{under} and T_{over} that are updated with an exponentially weighted moving average (with parameter α) each time a task executes depending on its success or failure. The two estimates are then compared, and the energy level V_s is adjusted by β in the direction that will balance the averages. That is, it is increased if more time is being wasted on failures than on charging too high.

More precisely, let V_e be the voltage at the end of running a task, and V_0 be the voltage at which the device ceases to operate, and ϵ be a small voltage. A task succeeds if and only if $V_e \geq V_0 + \epsilon$. *Dewdrop* computes estimates and uses them to adjust the target energy level, V_s as follows:

$$T_{over} = \begin{cases} (1-\alpha)T_{over} + \alpha t_{over}, & \text{if } V_e \geq V_0 + \epsilon \\ (1-\alpha)T_{over}, & \text{if } V_e < V_0 + \epsilon \end{cases}$$

$$T_{under} = \begin{cases} (1-\alpha)T_{under}, & \text{if } V_e \geq V_0 + \epsilon \\ (1-\alpha)T_{under} + \alpha t_{under}, & \text{if } V_e < V_0 + \epsilon \end{cases}$$

$$V_s = \begin{cases} V_s - \beta, & \text{if } T_{over} > T_{under} \\ V_s + \beta, & \text{if } T_{under} > T_{over} \end{cases}$$

Of course, there are degenerate cases where this heuristic will fail, e.g., tasks that exhibit bimodal energy consumption where some executions consume a large amount of energy and some executions consume very little. An example of this is a program that samples a very low power sensor on some iterations, and a more energy-intensive sensor on other iterations. But, based on applications we have seen in the literature, our approach is a good fit and has the benefit of being both simple and efficient.

4.2.5 Charging to a Target Energy Level

Given a target energy level, the device runtime must arrange for the task to begin execution when stored energy reaches that target. The baseline WISP uses hardware support in the form of a voltage supervisor to start execution when the capacitor voltage reaches a fixed level of 2V. Unfortunately, there are no designs for variable voltage supervisors that can be used in RF-powered computers to the best of our knowledge.

Instead, *Dewdrop* uses a software polling approach to determine when the target energy level has been reached and execution should begin. It sleeps while energy is being harvested, and occasionally wakes up to sample the capacitor voltage using an analog to digital converter (ADC). This is a general strategy that can be used on most platforms regardless of how the target energy level is determined.

However, polling is difficult to achieve at low cost because charge times can vary over orders of magnitude and waking up and sampling the capacitor consumes precious energy. In our experiments with the WISP, we found that reaching a given threshold can take less

than 10ms or 100s of ms depending on the input power. This variation, combined with the non-trivial cost of waking up to take a sample, means that polling at any fixed interval is problematic. If the device is close to the controller, a long interval means that the device will store excess energy and miss opportunities to execute tasks. Conversely, if the device is far from the controller, it will accumulate energy very gradually and pay a disproportionately greater overhead if the interval is short.

To gather energy over a large range of input powers and target voltages, *Dewdrop* uses an exponentially adapted polling interval. Specifically, let V_r be the voltage a device has gained since it last woke up, and t be the current sleep interval. Then,

$$t_{next} = \begin{cases} 2t, & \text{if } V_s - V > 2V_r \\ t/2, & \text{if } V_s - V < V_r/2 \\ t, & \text{otherwise.} \end{cases}$$

This mechanism is very lightweight because it only involves shift operations to scale the polling interval, not multiply, divide, or floating point operations (which are not likely to be available in hardware as they are power-intensive). In our evaluation we find it to be responsive, sleeping for short amounts of time at high input power, and to have low overhead, gathering energy out to low input power levels.

4.2.6 Maintaining State Information

If the voltage across the capacitor drops below V_0, the device will lose the contents of RAM; WISPs do not have non-volatile memory. This includes the values for T_{over} and T_{under}. Consequently, *Dewdrop* must create the illusion of persistent memory. If a task is going to fail, it must stop executing before the voltage drops below $V_0 + \epsilon$, and ϵ must be high enough to give the device time to transition into standby-mode where energy consumption is minimized and harvested power can recharge the capacitor. To achieve this, *Dewdrop* uses a low energy polling mechanism to track the current voltage level during task execution, and

stops a task before running out of energy. Future hardware is likely to use a fixed voltage supervisor to fire an interrupt when $V_0 + \epsilon$ is reached, and the interrupt handler will put the device to sleep.

4.3 Implementation

The WISP firmware is written in a mix of C and assembly, for timing sensitive operations. The code can be broken down into two main components: the *Dewdrop* runtime and task support. The *Dewdrop* runtime code must execute quickly and infrequently to reduce overhead. Task support includes the Gen 2 RFID communication protocol, which requires tags to respond to reader commands quickly, generally within 10s of microseconds. This section describes our implementation of a functioning prototype as it relates to these challenges, and introduces a monitoring board we developed to gather experimental data from WISPs in-situ without impacting their behavior.

4.3.1 WISP Hardware

The WISP draws approximately 600μA when the CPU is in active mode and 1.5μA when in a state-preserving sleep mode. By default, the WISP wakes up at a fixed power level; a voltage supervisor waits for sufficient power to operate (defined by its capacitor reaching 2V) and then triggers a hardware interrupt to wake the device. We use the term *HwFixed* to refer to this hardware method of waking up at a fixed voltage. *Dewdrop* disables this mechanism and instead uses a timer interrupt to wake the device.

The WISP stores energy in a 10μF capacitor and the voltage of the capacitor can be sampled via its analog to digital converter.[5] If the voltage of the capacitor drops below 1.5V, the WISP will black out and lose all state. We found that the time to fully charge the capacitor varied from 10s to 100s of milliseconds, depending on distance. Discharging a full capacitor to below 1.5V in the absence of a reader signal takes 10s of ms when active,

[5] A 10 μF capacitor is a reasonable trade-off between charge time (a smaller capacitor charges faster) and charge capacity.

but more than 8s when in sleep mode. Thus, the WISP can carry state across relatively long periods of reader inactivity by sleeping.

4.3.2 Dewdrop

Low power wake-up. *Dewdrop* puts the WISP into a deep sleep state for a specified period to gather energy, and the CPU is woken up by the timer interrupt. The process is repeated until the target wake-up voltage, V_s, is reached. This approximates the behavior of a hardware voltage supervisor, which wakes a device when a specified voltage is reached, but allows us to vary V_s. A potential drawback to this approach is an increased current draw due to keeping the crystal oscillator active to drive the timer, but in practice this increase is acceptably small (2 μA vs 1.5 μA with the crystal off).

Low cost voltage sampling. *Dewdrop* checks the capacitor voltage to see if enough energy has been stored to warrant starting a task, and goes back to sleep if not. The energy overhead of this polling approach is determined by the polling interval and how long the WISP must be awake for each sample. The per sample cost is directly proportional to how long the WISP must stay in active mode. Sampling the capacitor voltage should take 90μs according to the MSP430 data sheet instructions for using the ADC. However, we found that ADC values stabilized much faster—20μs including setup time—with sufficient accuracy (10mV). This shorter awake time drastically reduced the cost of voltage sampling.

Calculating the energy storage rate. *Dewdrop* also tracks how quickly energy is being stored, as it uses this information to adapt the sleep period and to calculate how much time is wasted overcharging. Our adaptive sleep function generally results in a series of sleep periods, where the WISP wakes up and checks its voltage, adjusts the sleep period, and returns to sleep. When a task completes, $V_e - V_o$ tells us how much energy is leftover. We use the last period's charging rate and the average charging rate over all periods to estimate how much time was wasted overcharging. When a task fails, $V_s - V_o$ tells us how much energy was wasted. We use the average charging rate to calculate the time wasted

undercharging.

4.3.3 Task Support

Order of operations. The computation and sensing components of tasks must take place before or after communicating with the reader; the deadlines imposed by the Gen 2 protocol are too tight to interleave task processing and message handling. Therefore, in the SenseTx task, for example, the WISP samples the sensor immediately after waking up and then begins decoding reader commands and waiting for the next Query.

Detecting task failures. To avoid blacking out and losing state, the WISP needs to detect when task failures are imminent and then quickly enter sleep mode, i.e., if the voltage drops below $V_o + \epsilon$ the task must be aborted. We found that an ϵ of 0.15V was sufficient to protect against blackout. That is, if any voltage sample measures below 1.65V, the WISP will sleep and record a task failure.

Sampling the voltage during the communication phase proved difficult, but it was necessary because message processing is a major factor in energy consumption. The Gen 2 message timing constraints are such that the WISP does not have time to take a sample between messages without losing synchronization with the reader, even with a sampling time of only 20μs. However, we found that we could carefully schedule a voltage sample during the preamble of every reader command, so long as the inspection of the sample was deferred until after the command was decoded. As the WISP must be in active mode to accurately track the preamble, this approach amortizes the cost of keeping the CPU active for decoding. This strategy makes it possible for us to closely track the voltage of the capacitor at every reader command with essentially zero overhead.

Randomness. The Gen 2 MAC protocol requires that tags choose slots randomly. As a source of randomness, we sample the voltage in the capacitor once immediately when the WISP first powers up, and use this value as a seed for a pseudo-random number generator. The variance in this voltage sample, due to input power and noise in the ADC, gives us

sufficient randomness. Alternatively, we could have used SRAM state as a random source, with similar efficiency [45].

4.3.4 Monitoring Support

Monitoring WISP state and operation for debugging and experimentation is difficult. Traditional methods for debugging embedded systems, such as a JTAG connection, would supply power to the WISP and change its behavior. Instead, we use a custom monitoring board we developed for debugging WISPs [80]. The board communicates with a PC via USB, attaches to the debug and other output pins of the WISP, but does not add to or consume energy harvested by the WISP. The monitor board can also sample the voltage in the WISP's capacitor. For our study, we instrument the WISP to toggle debug pins at key points in its operation, and the monitor board records what event happened and immediately samples the WISP capacitor to determine its voltage. This results in a trace of WISP operations from which we can determine task costs, and response rates even for tasks that do not communicate with the reader.

4.4 Dewdrop Evaluation

In this section, we evaluate *Dewdrop* experimentally. We show that our approach of balancing sources of waste generally achieves 90% of the best possible response rate for the SenseTx and Sense tasks and across a wide range of RF environments. *Dewdrop* improves performance over the default WISP runtime, providing applications a benefit in terms of both improved coverage and higher response rates.

4.4.1 Experimental Setup

Our experiments were conducted using an Impinj Speedway RFID reader that continuously transmits energy and commands. This is the normal reader behavior. For experiments involving a single tag, the WISP was placed on a poster board 1m from the reader antenna

and the output power was variably attenuated from 30dBm (1 Watt), the maximum allowed for "Gen 2" readers, to 18dBm. This method increases repeatability by limiting the multipath effects that would occur if we moved the WISPs. We present results in terms of an equivalent distance that is calculated using free-space propagation, as we find them to be more intuitive than results in terms of transmit power.

In all experiments, we ran *Dewdrop* and the default WISP hardware, which we call *HwFixed*, that starts tasks at a fixed energy level of 2.0V. *HwFixed* provides a baseline for comparison. When possible, we also report results for *Oracle* as the best result found from an exhaustive offline search of starting energy levels (at which the WISP wakes-up and starts a task) using 0.03V steps. We report results for both the Sense and SenseTx tasks described in Section 4.2.

To evaluate our approach in a realistic deployment, complete with multipath effects, we deployed 11 WISPs with accelerometers on a 1.2m x .75m table of a model apartment at Intel Labs Seattle. This deployment is similar to that seen in [12], though we only consider a single workspace instead of the complete apartment. An RFID reader was installed in the ceiling and equipped with one antenna approximately 2m above the table pointing downwards. We configured the reader to run the SenseTx task to gather samples continuously for one minute. We performed three separate trials for each configuration to allow for variability from both the RF environment and communication protocol.

4.4.2 Single Tag Evaluation

We first assess how well *Dewdrop* performs compared to *HwFixed* for a single WISP.

Figure 4.3 compares the response rate of Sense and SenseTx when using the two runtimes. *We find that the performance of* Dewdrop *consistently matches or exceeds that of* HwFixed. For the light Sense task, the performance of *Dewdrop* closely matches that of *HwFixed* and actually performs better at 1m. This is because, at close range, the received power supplements stored energy enough to allow an energy level 0.2V below *HwFixed*'s

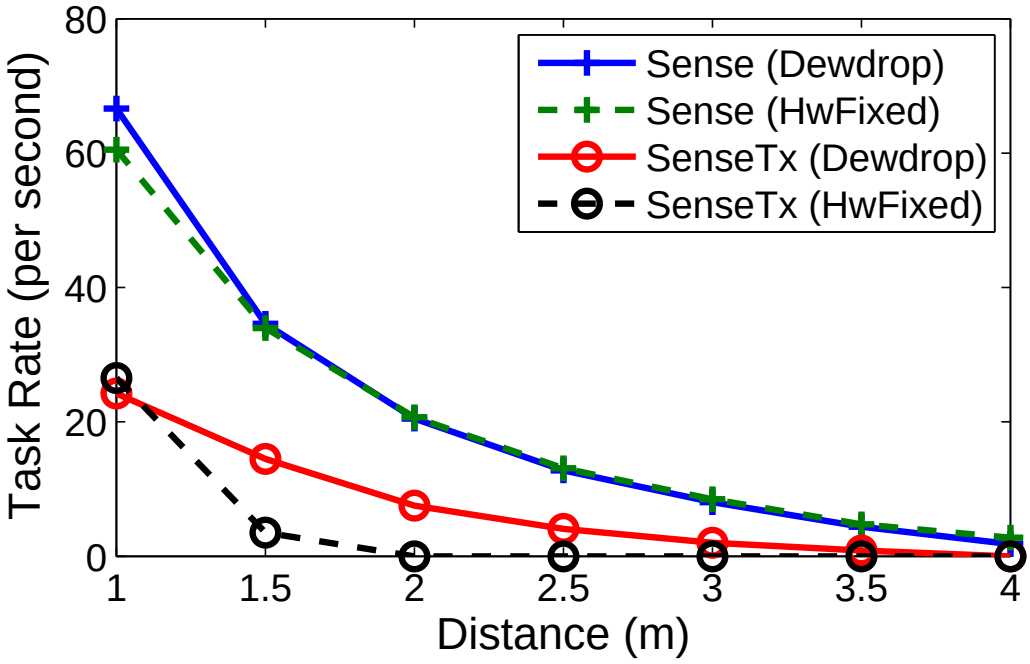


Figure 4.3: Response rates when using *Dewdrop* and the *HwFixed* runtimes.

fixed value.

In the case of the heavier SenseTx task, *Dewdrop*'s response rate decreases smoothly as reader power falls to 3.5m. *HwFixed* fails to execute the task beyond 1.5m. *Dewdrop* adapts to the higher energy requirements of this task, and stores more energy before beginning execution, whereas *HwFixed* does not. This improvement more than doubles the operating range of the tag.

To find an upper bound on how well *Dewdrop* could work, we compare to the *Oracle* results. Gathering the *Oracle* test data takes hours and is thus not a candidate for a practical CRFID runtime. Figure 4.4 again shows the response rates for the two tasks when using *HwFixed* and *Dewdrop*, but the rates are normalized by the best rates found using the *Oracle*. *We find that* Dewdrop *generally achieves better than 90% of the maximum rate seen by* Oracle *for both tasks.* Interestingly, *Oracle* always beat *HwFixed*. This means that the fixed 2 V energy level was never the best choice.

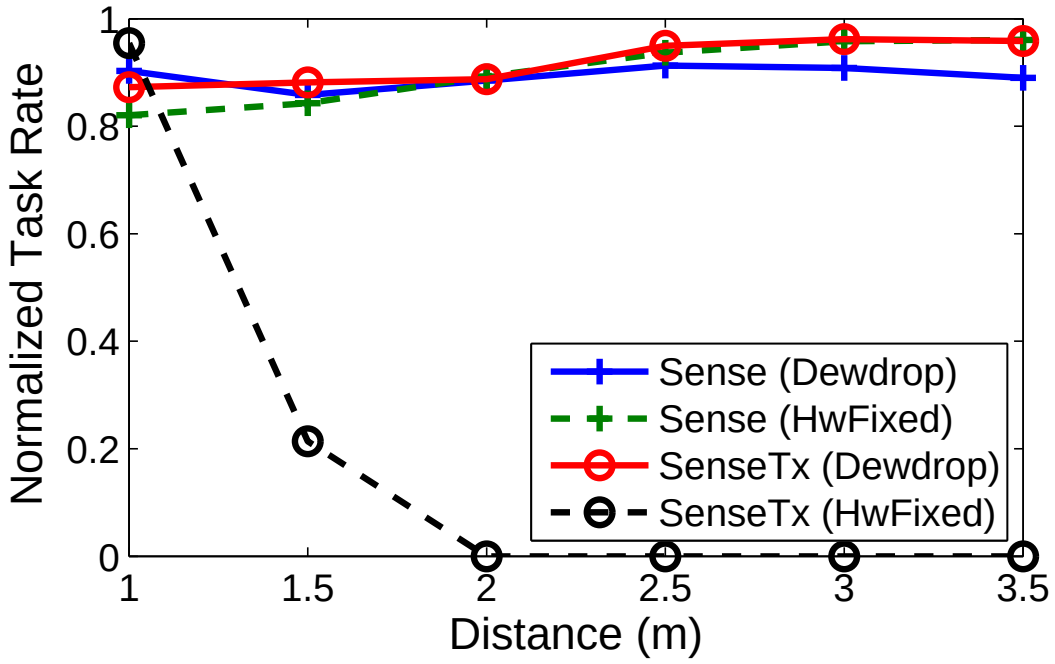


Figure 4.4: Response rates for *Dewdrop* and *HwFixed* compared to an oracle.

Evaluating the choices of *Dewdrop*. To understand why *Dewdrop* performs well, we looked at the starting energy levels it selects. *Dewdrop* must choose starting energy levels that are close to the best level found by the *Oracle* if it is to be efficient. To show that this is a non-trivial task, Figure 4.5 shows examples of response rate versus energy level curves. The figure is based on data from the *Oracle* for both tasks at 1.5 and 3m.

We see that the best starting energy level varies widely for different tasks and at different distances. For Sense, the best energy level is 1.9V at 1.5m, when input power close to the reader supplements stored power, and 2.1V at 3m. Similarly, for SenseTx the best level varies from 2.5 to 3V over the same distance. These results emphasize that no fixed threshold will work either for all tasks or for all distances. For example, the best energy level for SenseTx at 3m is 3V. This level achieves only 50% of the maximum response rate for Sense at the same distance. It is even worse if the best level for Sense at 3m is chosen, as SenseTx cannot execute the task even once at 3m with an energy level of 2.1V.

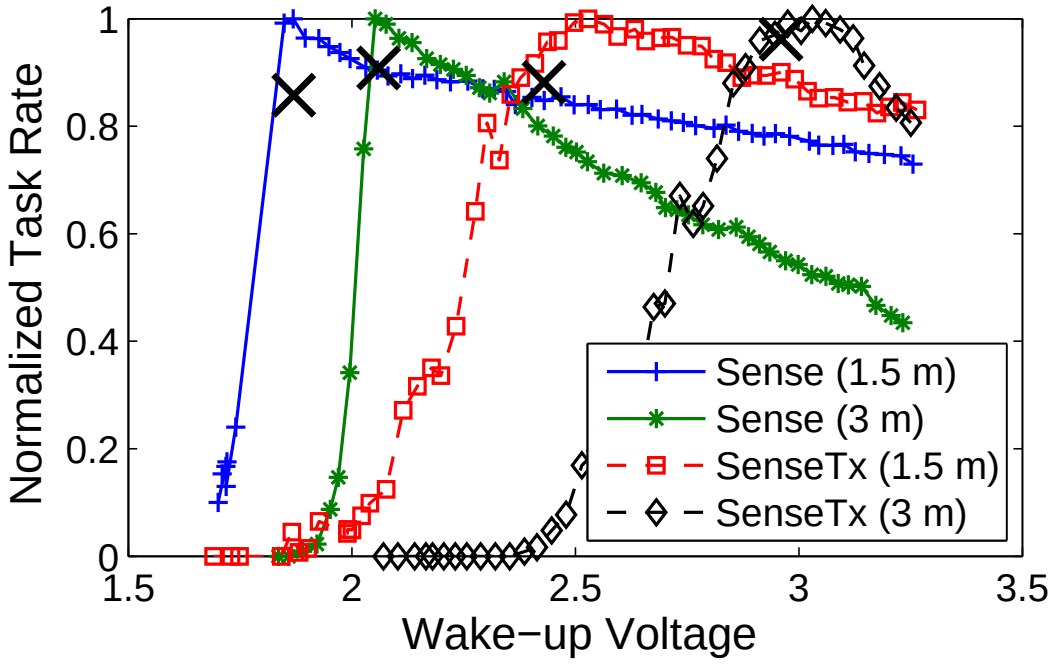


Figure 4.5: Response rates for both tasks at 1.5 and 3m. X's indicate the operating point found by for *Dewdrop*.

The figure also shows the operating points found by *Dewdrop* marked with Xs. *We see that our runtime finds points very close to the best energy level despite the differences between response curves.* Across all of our data the energy levels found by *Dewdrop* were within 0.1V of the best level found by *Oracle*.

To see how *Dewdrop* selects a good starting energy level, we looked at how it minimizes wasted time. We calculated the average wasted time per task due to failing and due to charging too high. Figure 4.6 shows this data, along with response rate, for an illustrative case of Sense and SenseTx at 3m. The data are normalized by their maximum values. We see that as the starting energy level increases, the average wasted time due to failing generally decreases. (The waste is low at low wake-up thresholds despite tasks failing a greater fraction of attempts. This is because waste is computed in terms of time spent charging, and at low wake-up thresholds, very little time is spent charging.) Beyond 2.6V,

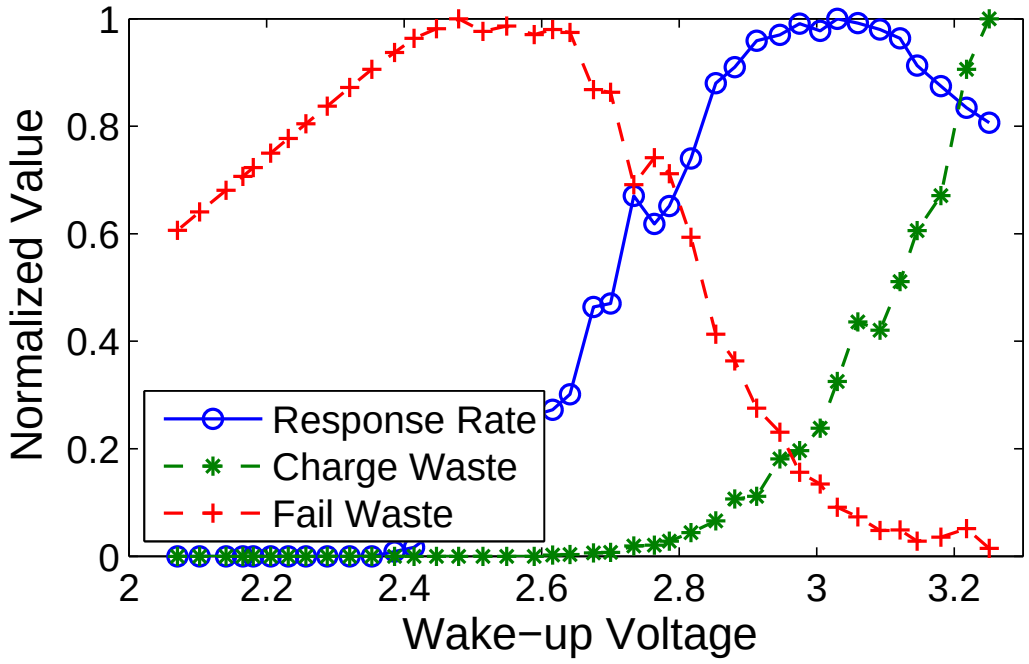


Figure 4.6: Response rate and wasted time for SENSE and SENSETX at 3m.

waste from failed tasks decreases, as the task fails less often. Conversely, the wasted time from overcharging increases with the starting energy level because the energy is stored less efficiently at higher voltages.

Dewdrop seeks the intersection of the two waste curves, and uses the corresponding energy level. This appears to be a good strategy as the maximum response rate in the figure occurs near the intersection. Moreover, since the rates plateau around the maximum, *Dewdrop* can miss its mark by a fairly wide margin (±0.1V), without affecting performance significantly. Though the figure shows only a single example, *we found the energy level that equalized the two sources of waste generally achieved better than 95% of the maximum rate for both tasks at all distances.*

Evaluating the overhead of *Dewdrop*.

This section investigates two possible inefficiencies in *Dewdrop*: the overhead of our

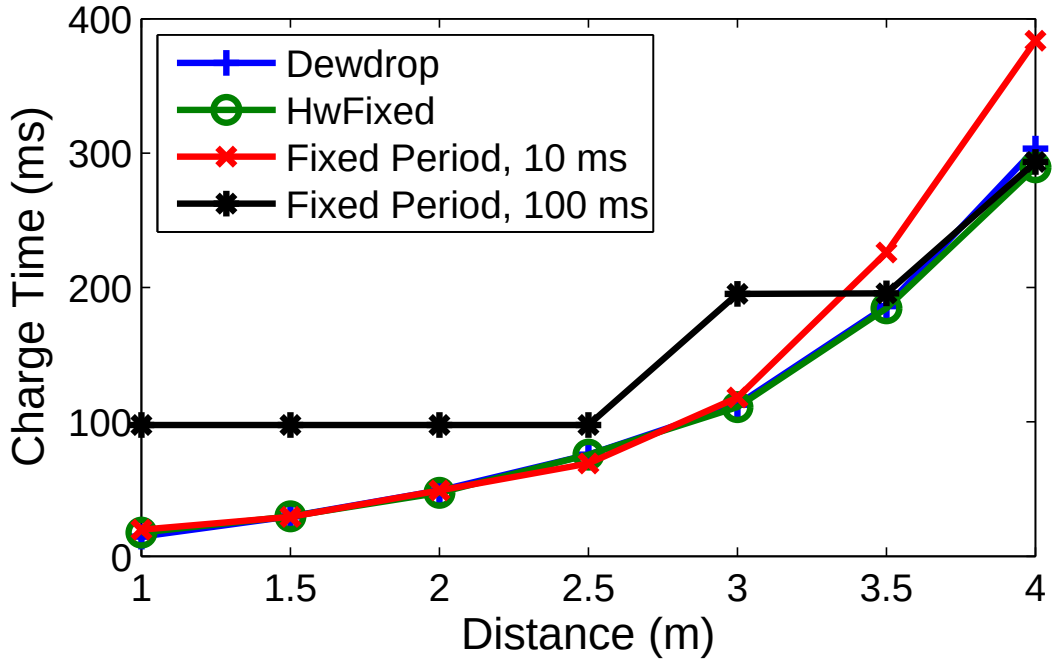


Figure 4.7: Charging time from 1.5V to 2V.

timer-based adaptive sleep scheme, and the effect of our choice of step size for maintaining the starting energy level. We show that both are efficient, which is in keeping with our runtime performing almost as well as the *Oracle*.

To be effective, our runtime must not appreciably increase charging time. Figure 4.7 shows the median charging time from 1.5V to 2V for *Dewdrop*'s adaptive sleep mechanism, the hardware wake-up of *HwFixed*, and two strawman versions of our software controlled sleep mechanism that use fixed sleep periods.

We find that, at all distances, our adaptive scheme achieves a charge time within 5% of the charge time of the hardware mechanism. Moreover, as expected, its performance is good over a wider range of distances than schemes that do not adapt their sleep periods. For example, the fixed period of 100ms does well at 4m (1.3% longer than *HwFixed*), but performs poorly at close range (600% longer than *HwFixed* at 1m). Likewise, fixing the period at 10ms works well at close range, but incurs significant overhead farther away (32%

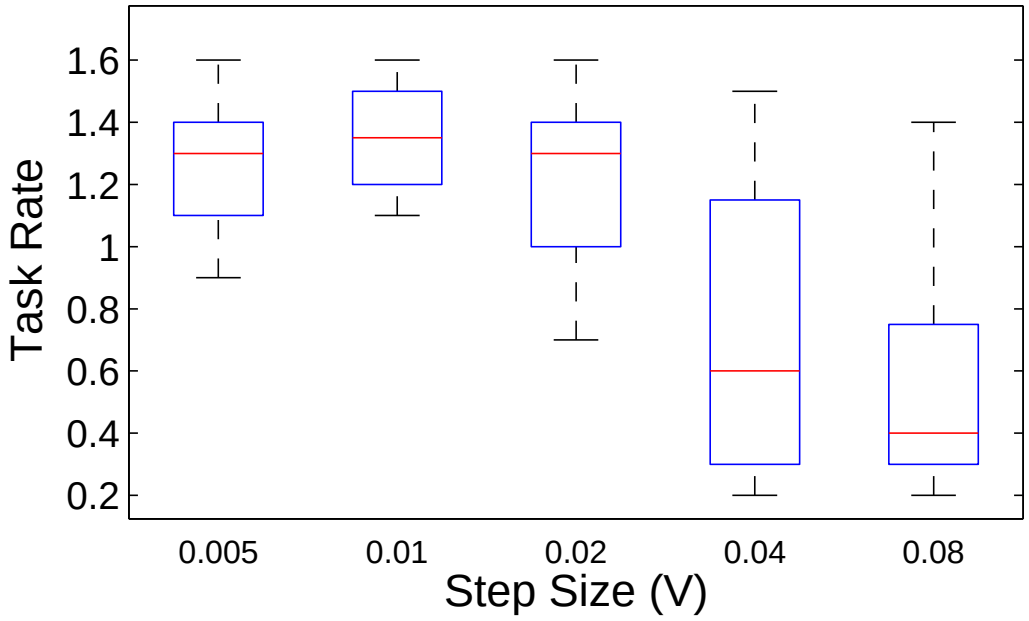


Figure 4.8: Effect of step size (β) on response rate for SenseTx at 3.5m.

at 4m).

The second potential source of inefficiency in our system comes from our choice of step size (β) when seeking the best starting energy level. In *Dewdrop*, upward pressure on the level is only exerted after it drops fairly low and tasks begin to fail; after failures, the starting energy level rises until the cost of overcharging outweighs the cost of failing. A small β increases the time it takes to adapt to environmental changes, while a larger β can result in large oscillations around the ideal wake-up threshold.

Figure 4.8 shows the effect of different step sizes on task rate for SenseTx at 3.5m. The average task rate per second is calculated over a 10 second sliding window. As step size increases, the task rates generally decrease and vary more widely. A larger step size means that *Dewdrop* increases/decreases its starting energy level too quickly, resulting in significant over/undercharging. The reverse then happens and the voltage is reduced by too much and more tasks fail. We found that a step size of 0.01V gave a good balance between

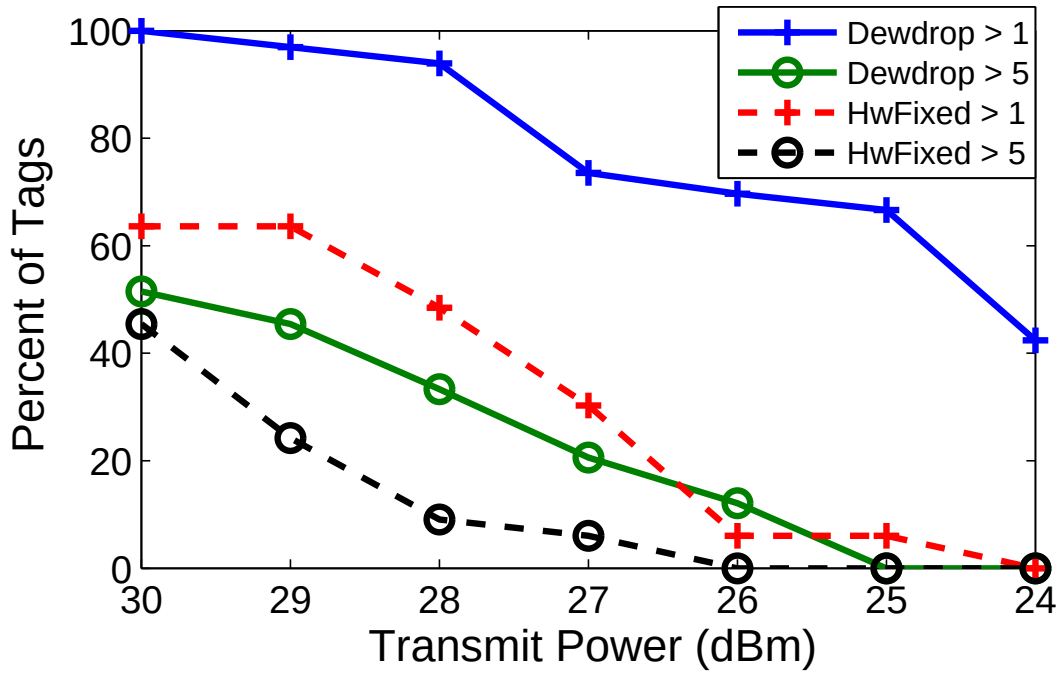


Figure 4.9: Percent of tags that have an average response rate above 1/s and 5/s using the two runtimes.

damping oscillations in energy level and quickly adapting to environmental changes.

4.4.3 Multiple Tag Evaluation

Next, we evaluate *Dewdrop* in a realistic deployment consisting of multiple tags. To support CRFID applications such as activity recognition, our runtime should both increase the coverage region of the reader (e.g., so that distant devices respond) and also increase the response rates of the devices (e.g., so that object motion can more accurately be tracked). We consider both of these metrics for the 11 WISPs deployed in the model apartment.

Coverage. The coverage goal is to have as many devices as possible responding at a useful rate. Based on prior experience, we define two useful rates: a rate of 1/s, as is useful for low-rate object use detection; and a rate of 5/s, as is useful for higher-rate gestural recognition. To characterize the coverage of the deployment, the transmit power of the

reader is reduced gradually to determine the "headroom" (in dBm) tags have for a given level of performance.[6]

We find that Dewdrop *has much better coverage than* HwFixed *because it enables tags to operate when much less incoming power is available.* Figure 4.9 shows the percentage of tags with average response rates above 1/s and 5/s when using the two runtimes. At 30dBm, all tags with *Dewdrop* respond at least once per second as compared to 64% with *HwFixed*. Coverage is better even when tags with *Dewdrop* receive one third the power of tags with *HwFixed* (viz., 67% for *Dewdrop* at 25dBm vs 64% for *HwFixed* at 30dBm). Moreover, at a four-fold reduction in power (24dBm), 42% respond with *Dewdrop* while none respond with *HwFixed*.

For a response rate of more than 5/s, the two runtimes perform equally well at 30dBm. This is because *HwFixed* works well when a tag receives good power from the reader. However, *HwFixed*'s coverage decays much more quickly with power than does *Dewdrop*'s coverage, e.g., at 27dBm *Dewdrop* has three times the coverage of *HwFixed*.

Response Rates. Figure 4.10 shows the distribution of the response rates of the tags when the reader is transmitting at 30 and 24dBm. The rates are computed over one second windows for both runtimes. *We find that* Dewdrop *consistently achieves higher rates, especially for the tags receiving less energy;* 30% of the data points are zero for *HwFixed* versus 5% for *Dewdrop*. *Dewdrop*'s ability to achieve useful rates is even more apparent when the reader transmits at 24dBm and tags are receiving one fourth as much power. *Dewdrop* obtains response rates greater than once per second 30% of the time, as compared to 2% with *HwFixed*. At 30dBm, *Dewdrop* and *HwFixed* achieve nearly the same rates for those tags that receive the most energy; 25% of the data points are above 9/s, and median rates are 5/s and 3/s respectively.

When more tags are present, the energy cost of communicating with the reader increases.

[6]This "attenuation thresholding" technique [44], has been shown to be more appropriate for characterizing RFID deployments than varying distance due to the high sensitivity of RFID to multipath.

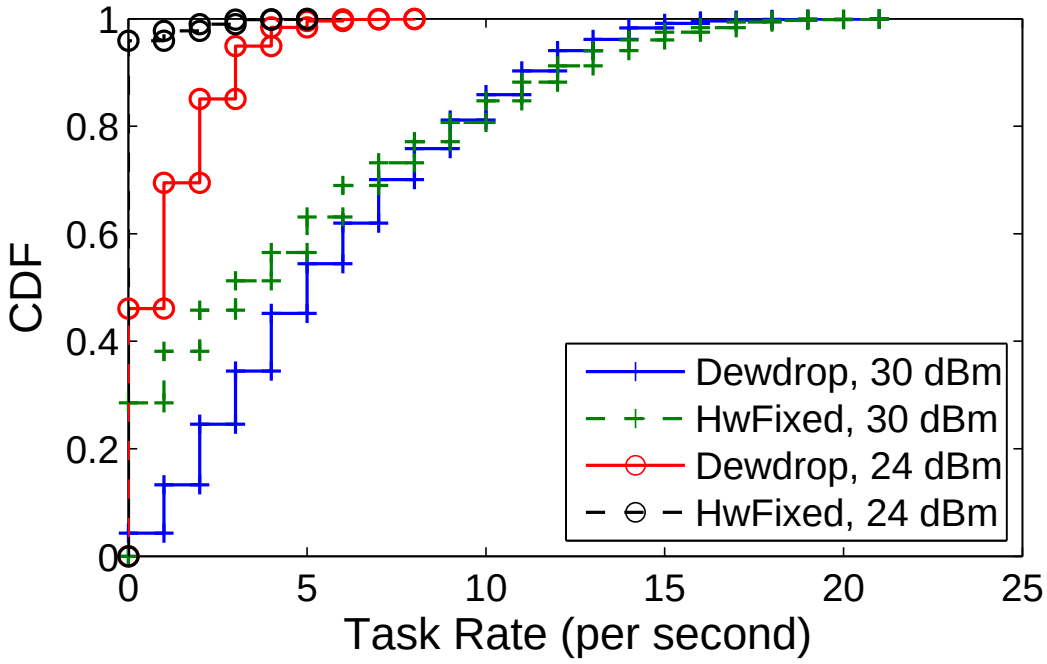


Figure 4.10: CDF of response rates for the two runtimes as power is reduced.

This is because the reader increases the number of slots it uses to limit the likelihood of tag collisions, so CRFID tags must process more messages before transmitting to the reader.

Figure 4.11 gives the performance for a single tag when the reader transmits at 30dBm as additional tags are added to the deployment. The performance of *HwFixed* rapidly decreases with the number of tags. This is because the number of slots is increasing, and a tag cannot remain powered when it chooses a later slot. In contrast, *Dewdrop* simply increases its starting energy level to accommodate the additional communication overhead. With one tag, it wakes up around 2.5V whereas with 25 tags it wakes up closer to 3V. The result is that *Dewdrop* provides nearly three times the response rate as *HwFixed* when 25 tags are present.

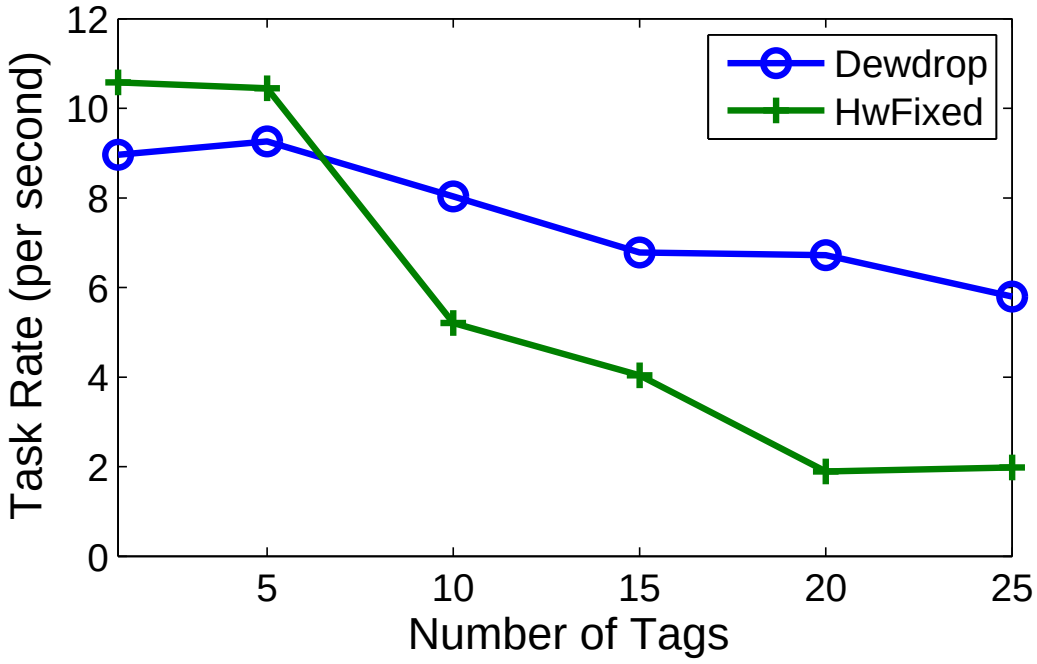


Figure 4.11: Response rate for the two runtimes as tag population size increases.

4.5 Dewdrop: Limitations and Lessons Learned

Dewdrop proved to work well in our scenario compared to the state-of-the-art. Moreover, the project helped us identify the operating characteristics that are fundamental to RF-powered computers and those that are short-term artifacts. For example, variations in received RF power, the fact that many tasks must run-to-completion (such as sensors), and the need to use capacitors for short term energy storage seem fundamental. However, hardware improvements will make the job of the runtime easier by enabling devices to operate more efficiently across a range of scenarios, and by helping to simplify the task execution model. In Section 7.2, we discuss these future directions in detail.

Chapter 5

BUILDING BACKSCATTER NETWORKS THAT SCALE WELL

In Section 3.2, we described how interference is the key limitation that makes building scalable backscatter networks challenging. In this chapter, we use simulation and measurement to show that existing technologies fail to perform well as the network scales. We then explore the role that the client demodulator plays in both controller-to-client and client-to-controller interference, and find that adjusting the client sensitivity can be used to trade robustness to continuous wave interference for susceptibility to modulated interference. Leveraging this insight, we propose a network design that mitigates downlink interference using a redesigned client demodulator and CSMA-based interference avoidance by controllers, and mitigates uplink interference using code-division-multiple-access for client transmissions. We evaluate this design and show that it performs far better than previous approaches, and that it is able to provide good performance to all clients even in very dense deployments.

5.1 Study of Existing Technologies

EPCGlobal's Gen 2 RFID is the only widely adopted backscatter protocol of which we are aware. It is a mature standard used for many applications, and we initially hoped to leverage Gen 2 technology to build RF-powered networks. However, we found that it is not well-suited to this scenario. This is partly a result of Gen 2 being designed exclusively for enumerating a set of unknown tags in a well-defined area, in contrast to supporting good performance for a set of known clients on a building-wide scale. This led to a design where tags only respond to a reader if its signal power is far above all other readers in the area. The result is that, when multiple readers are present, continuous wave interference often

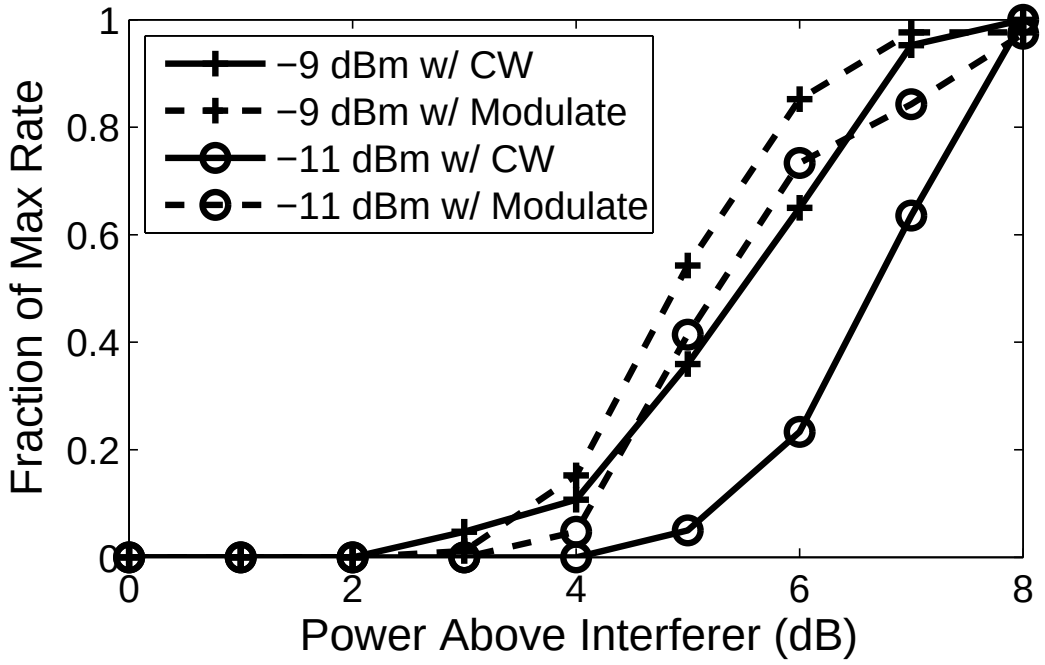


Figure 5.1: Read rate of Gen 2 tag and controller in the presence of an interferer.

results in tags that are receiving sufficient power to operate, but that are unable to decode messages from any RFID reader. As infrastructure is added, for example to extend the coverage area, some locations can *lose* connectivity.

In this section, we use measurements to illustrate this problem using Gen 2 tags and readers. Next, we use a network simulator to show that existing approaches fail to perform well even in sparsely deployed networks, as they result either in many clients having no connectivity, or all clients having very low throughput.

5.1.1 Performance of Gen 2 Systems

In 802.11 networks, both capacity and coverage of a network can be increased by adding APs on different channels. In contrast, adding controllers to backscatter networks may reduce coverage and capacity for clients. To understand the extent of the problem in practice, we performed an experiment using two Impinj RFID controllers and a Gen 2 tag. We combined

the signals from the controllers using a power combiner, and connected them to a single antenna. The tag was placed 1 meter from the controller, and we varied the relative powers of the controllers while measuring the response rate of the tag.

The dashed lines in Figure 5.1 show the results of this experiment when the tag was receiving around -9 and -11 dBm as measured using a power meter, which approximates a range of 5-8 meters. We compared the response rate of the tag with the interferer present to the maximum rate seen when only the controller was transmitting. When the controller transmit power is less than 3 dB above the interferer power, the tag does not respond at all. [1] The controller needed to transmit at 6 dB above the interferer to achieve 75% of the maximum rate. To communicate at the maximum rate, the controller transmit power needed to be 8 dB or more above the interferer.

We next experimented with a continuous wave interferer. This models the situation of a controller transmitting a message to a tag while a second controller transmits a continuous wave so a different tag can harvest power or backscatter a response. We ran the experiment by replacing the interferer with a signal generator that transmitted a continuous carrier. As shown by the solid lines in Figure 5.1, even having a continuous carrier in the environment dramatically degrades network performance. That is, controllers cause interference even when "receiving" messages from clients.

To better understand these results, we used a software-radio based RFID reader [15] to experiment with two passive RFID tags from different manufacturers to determine what modulation depth they could detect. We found that the tags required modulation depths of greater than 50% to decode individual bits (which agrees with existing work [73]). This is a reasonable threshold as it eliminates the possibility of modulated interference; no more than one controller can ever account for more than 50% of the total power. However, we found that tags required a depth of more than 80% to detect the brief power down periods that are required to operate correctly according to the protocol. Consequently, even one relatively

[1] Using a software radio to capture signal near the tag, we verified that the tag was not responding, and when the tag did respond the controller could hear it.

weak interferer can degrade performance, which agrees with our experimental results.

These results tell us that if a Gen 2 tag is located between two readers, and their signal powers are approximately equal, the tag will not have any connectivity. This is the starvation problem caused by continuous wave interference. The problem is exacerbated as readers are added to the area, because the signal power of one reader must be sufficiently greater than the sum of all other reader powers received by the tag.

5.1.2 Trade-off Between Coverage and Capacity

Because even weak interferers can cause a client to lose connectivity, and controllers interfere at clients even when transmitting on different channels, one approach to ensuring coverage is to time-multiplex controllers. In this section, we use a simulator to better understand the network-wide impact of Gen 2's approach of being always-on, and Listen-Before-Talk's approach of deferring communication if nearby controllers are transmitting.

The simulator is described in depth in section 5.4.2, and is intended to model the impact of interference in large-scale backscatter networks. Clients are associated with the strongest controller, and controllers poll their associated clients for messages in a round-robin fashion. Clients respond with packets of various lengths. In this experiment, we use Friis transmission equation to calculate signal powers at all clients and controllers given a 1 W transmit power, and we assume that controllers do not frequency hop.

For this experiment, we set the client sensitivity to 50% and controller messages are received error-free if the controller accounts for greater than 50% of the total power at the client. The bit-error rate of client transmissions is calculated using standard techniques based on the power of the background noise and other transmitting clients [98]. We simulate controllers with receive sensitivities of -80 dBm and a noise figure of 25 dB, equivalent to existing RFID readers [49]. Controllers are randomly distributed in a 50 x 50 meter area, and additional controllers are added until all locations are within range of at least one controller; this tends to require around 10 controllers. Ten trials are performed with the

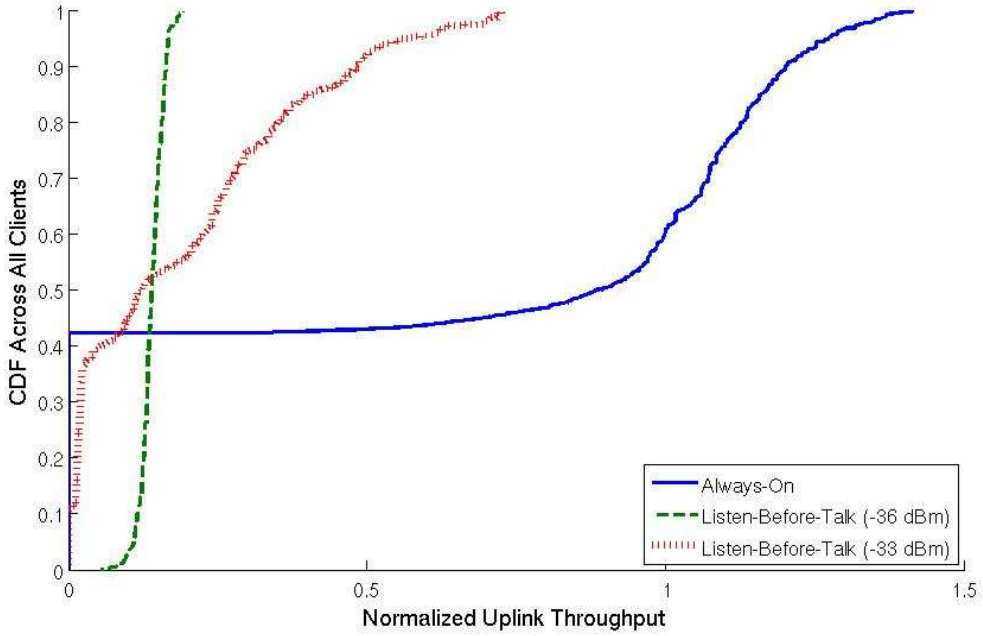


Figure 5.2: Network Performance for Always-On and Listen-Before-Talk

controllers and 250 clients being redistributed between trials.

Figure 5.2 shows the throughput results for the Always-on approach taken by Gen 2, and a Listen-Before-Talk approach where controllers listen to the channel and if they sense other transmitters above some threshold, they will not transmit. Prior work suggests that controllers should time-multiplex if they are within twice the controller-to-tag range [10]. In our simulation, this translates to a carrier-sense threshold of -36 dBm, and we use this threshold in our experiments. For simplicity, we assume there are no collisions caused by controllers sensing a clear channel and then beginning to transmit at the same time. Because controllers can have different numbers of associated clients, we normalize the throughput of each client by the rate it would achieve if there were no interfering controllers.

If additional controllers did not cause interference, all clients would achieve a normalized uplink throughput of 1, and the distribution would be a vertical line at 1 on the figure. However, in the case of always-on (blue curve) we see close to a step-function, where more

than 40% of clients do not have coverage and their throughput is zero, and the rest see close to or greater than 100% of the expected throughput. Values greater than one are possible because when a downlink message is lost due to errors, the client will not transmit an uplink packet, and the other clients can transmit more often. This results in a network that is not fair, where some clients are starved completely and others achieve very high throughput.

In contrast, the Listen-Before-Talk approach (using a -36 dBm threshold) achieves nearly perfect fairness, but at the cost of very low throughput. All clients achieve approximately the same throughput, but with the median value being only 14% of the expected rate. The red curve is an example of when the controllers use intermediate values for their carrier-sense threshold that allow for some simultaneous transmissions. By adjusting the carrier-sense threshold, controllers can trade-off fairness among all clients for higher throughput for a few clients. However, even with a -33 dBm threshold, allowing concurrent transmissions results in more than 10% of the clients having no connectivity.

The prior experiment did not take into account controllers that frequency hop. As backscatter systems are narrow-band transmitters, they are prone to frequency selective multipath fading. Moreover, as controllers frequency hop according to FCC regulations, the pattern of fades changes accordingly. This results in time and frequency dependent attenuation, and the magnitude of attenuation can be approximated as a Rayleigh distribution. Though generally considered an impediment to network performance, multipath fading can actually help reduce the impact of continuous wave interference. The variation in signal power across channels means there is often at least one channel on which a client can communicate. Figure 5.3 shows the results for the same experiment as above, but when controllers frequency hop every 400 ms. For the Listen-Before-Talk case, there is a widening of the distribution but the median value does not change significantly. For the Always-On case, fewer clients have a throughput of zero because, over time, clients are likely to experience at least some "good" fading profiles where they can decode controller messages. However, 7% of the clients still have no connectivity in the Always-On case, and the bottom

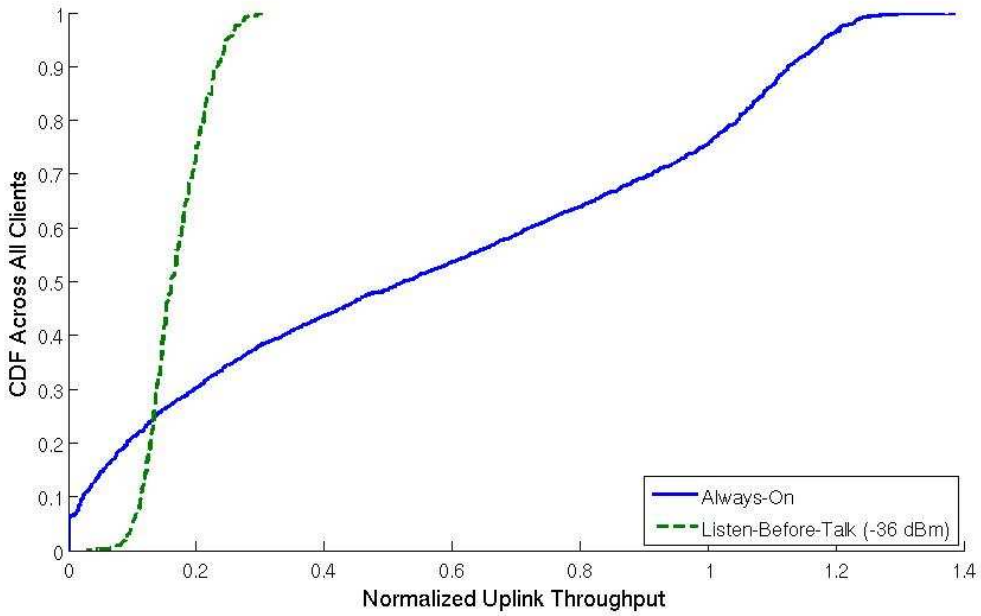


Figure 5.3: Network Performance for Always-On and Listen-Before-Talk with Rayleigh Fading

25% of clients have normalized throughputs of less than 14%.

5.1.3 Summary

Gen 2 tags use a demodulator design that requires a modulation depth of 50% (or more) to operate. This eliminates the possibility of modulated interference, because no more than one controller can ever account for more than 50% of the total power at a client, but it means that clients are prone to continuous wave interference. This limits network performance and fairness when more than one controller is present, as no controller accounts for more than half the power at some tags and they cannot decode any controller messages. One approach to mitigating continuous wave interference is to multiplex controllers in time. While this avoids starvation and improves fairness, it comes at the cost of dramatically reduced network capacity because controllers are not transmitting much of the time. Though more complex TDM-based approaches may achieve better performance than Listen-Before-Talk, turning

controllers off will always reduce capacity as clients cannot transmit on the uplink if their controller is powered down. Additionally, turning off controllers reduces the amount of power available to clients for harvesting. *Our goal is to develop a network design that enables controllers to be "always-on", while still avoiding interference between controllers and clients.*

5.2 Exploring the Trade-offs of Client Sensitivity

Gen 2 tags suffer from continuous wave interference, which means they often cannot operate even though they have sufficient power. This is because their demodulators are designed to eliminate modulated interference at the cost of being prone to continuous wave interference. In this section, we use a custom-built software radio-based backscatter demodulator to show how the client sensitivity can be manipulated to trade robustness to continuous wave interference for susceptibility to modulated interference. We then use simulation to demonstrate how the client sensitivity impacts network-wide performance.

5.2.1 Reducing the ASK Threshold of the Client Demodulator

Gen 2 tags are prone to continuous wave interference because their ASK demodulator can only detect messages if the modulation depth is greater than 50%. To understand how the demodulator would behave if it used a different ASK threshold, we built a simple ASK demodulator (based on the demodulator of the WISP [91], and very similar to conventional RFID tag designs [78]), and interfaced it with a USRP software radio via a Low-Frequency Receive (LFRX) daughterboard. This front-end acted as an envelope detector with a low-pass frequency of 500 kHz. To enable experimentation, we did not include a comparator in the circuit. Instead, the USRP captured the raw output of the envelope detector, and we experimented with different detection thresholds in software. To let us experiment with commercial RFID readers, we modified a USRP-based Gen 2 monitor we built previously [14] to process the signal captured by the front-end.

Our experiments were conducted by connecting two ThingMagic Mercury 5E RFID

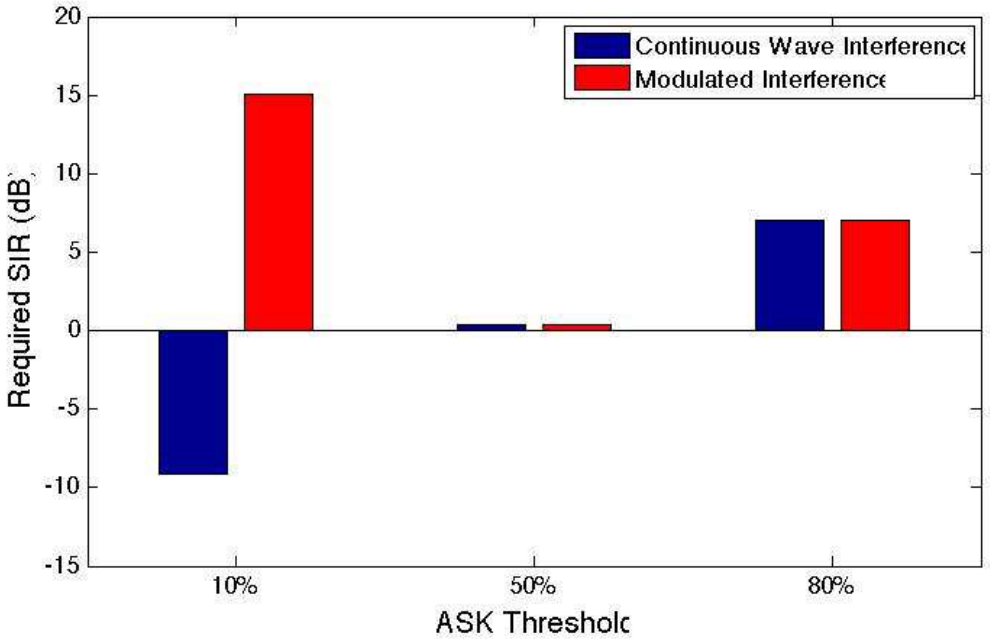


Figure 5.4: Sensitivity to interference for different ASK thresholds

reader development kits to our custom-built front end via a power combiner. The development kits let us experiment with two scenarios: 1) One reader is querying for RFID tags, while the second reader (the interferer) is transmitting a continuous wave, and 2) One reader is querying for RFID tags, while the second reader (the interferer) is transmitting a pseudo-random series of Gen 2 modulated ones and zeros. We varied the power of the interferer using a variable attenuator, and recorded the signal-to-interference ratio (SIR)that was required to successfully decode the reader messages.

Figure 5.4 shows the SIR in the two scenarios when different ASK thresholds are used. In the presence of continuous wave interference, a 10% modulation depth can decode reader messages even when the signal power is 9.3 dB less than the interferer. However, in the presence of modulated interference, the signal must be more than 15 dB greater than the interferer. In the case of a 50% ASK threshold (as is common in Gen 2 tag), if the signal power is even 0.35 dB greater than the interferer, the client can decode the message in the

presence of both continuous wave and modulated interference. Using thresholds above 50% does not perform well in either scenario, as this only increases the SIR required to decoded messages in the presence of both continuous wave and modulated interference; in the case of an 80% threshold, an SNR of nearly 7 dB is required.

The results when using a 50% and 80% threshold support the results shown in the previous section, and show why Gen 2 tags perform poorly in the presence of multiple readers. However, the 10% case shows that thresholds below 50% provide a mechanism to change the balance between continuous wave and modulated interference. In the next section, we use the simulator to explore the impact on overall network performance when using different thresholds.

5.2.2 Network-wide Effect on Controller-to-Client Interference

To build understanding of how the ASK threshold of the client demodulator impacts the balance between continuous wave interference and modulated interference, consider the diagram in Figure 5.5. The diagram shows 7 controllers (black dots) arranged in a 80 by 70 meter grid, with all controllers transmitting continuously at 1W using an omnidirectional antenna. The colored regions indicate where a given controller's signal accounts for some fraction of the received power, and clients with a particular demodulator sensitivity can decode its messages. To the left of the figure, the blue and red regions illustrate the case where clients can detect messages with only a 10% ASK modulation depth (i.e., a controller must only account for 10% of the power received at the client). As can be seen, the regions of the upper left and lower left controllers overlap considerably, and clients in the overlapping region can detect transmissions from both controllers. If the controllers modulate simultaneously, clients in that region will suffer from modulated interference and the messages will be lost.

In contrast, the grey and green regions to the right of the figure illustrate the case where clients use an ASK threshold of 50%. Here, there is a region between the two controllers

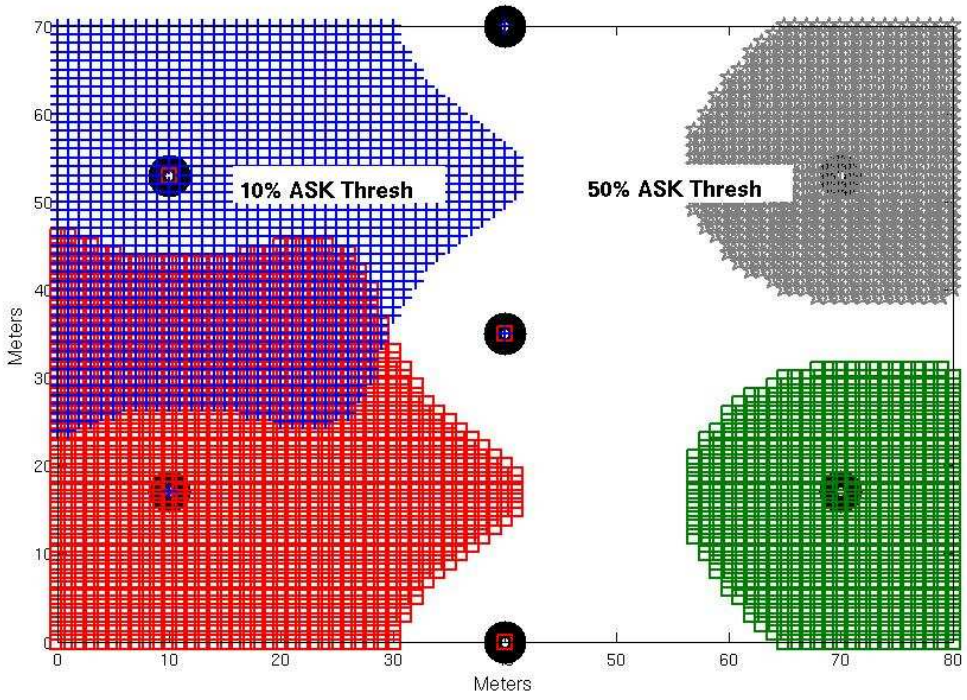


Figure 5.5: Coverage areas of controllers when clients use different ASK thresholds. All controllers are transmitting 1 W using isotropic antennas.

where neither signal accounts for greater than 50% of the received power, and clients cannot detect transmissions from either controller. Clients in this regions suffer from continuous wave interference and cannot receive messages from any controller.

In this ideal scenario, a threshold can be chosen such that both forms of interference can be minimized. However, the choice of threshold must be decided at the time of manufacture as it is a function of the hardware, and there is no ideal threshold for real scenarios. This is because controllers will not be perfectly deployed in a grid, propagation patterns will not be perfectly circular, and clients may not want to associate with the controller with the highest power.

Figure 5.6 shows the likelihood of interference in a more realistic scenario where 5-25 controllers are randomly placed in the 70 by 80 meter grid. The data is generated by

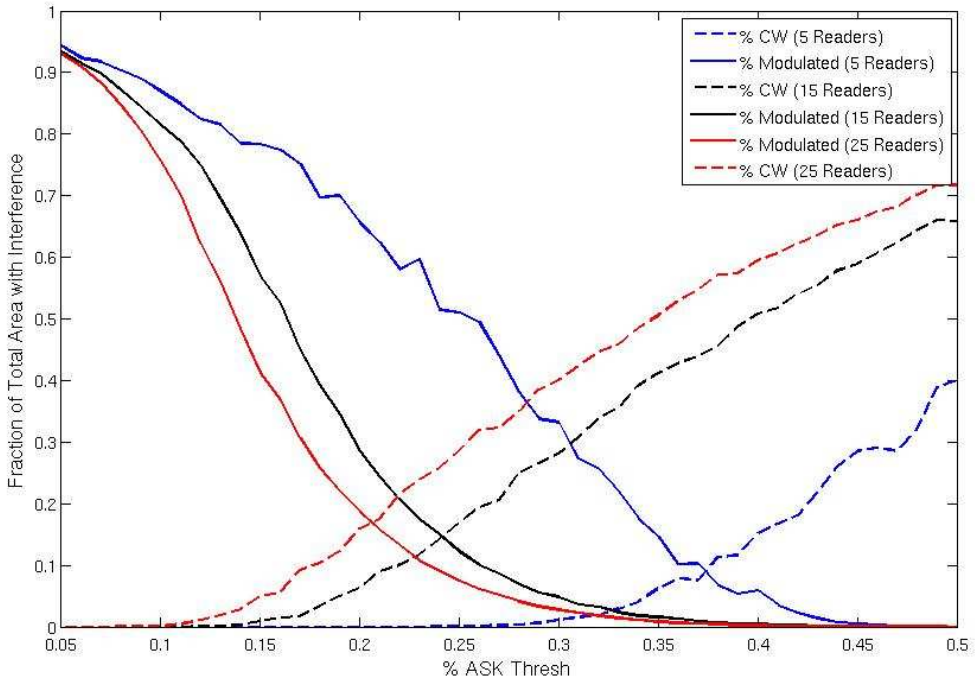


Figure 5.6: Fraction of locations where a client would suffer from continuous wave and modulated interference for three network densities.

running 50 trials each with a different arrangement of controllers and clients. Controllers transmit RF power at all times, and detection threshold of the clients is varied from 5% modulation depth (very sensitive) to 50% (similar to Gen 2 tags we measured). Two metrics are measured: 1) the fraction of locations where messages will be lost due to continuous wave interference, and 2) the fraction of locations where messages may be lost due to modulated interference.

To the left of the graph, where the client demodulator can detect low modulation depth transmissions, there are no positions where messages will be lost due to continuous wave interference. With only 5 controllers present, there are no cases of continuous wave interference even with a detection threshold of around 20%. This makes sense, as with only 5 controllers, the signal of at least one must account for 20% of the power at a given client.

As the density of the network increases to 25 controllers, even a threshold of 5% cannot eliminate continuous wave interference completely, though the probability is still low. The maximum threshold that still eliminates continuous wave interference depends on the density of the network, which cannot be known when the client is manufactured.

Though CW interference can be reduced by increasing the sensitivity of the client, this increases loss due to modulated interference. A loss due to modulated interference only happens when multiple controllers happen to transmit polling messages at the same time, and the likelihood of loss will vary with downlink traffic patterns. However, it is clear that low thresholds increase the likelihood that a message will be lost from modulated interference, particularly as controller density increases and clients often are in range of multiple controllers. Hence, there is a fundamental trade-off between the two forms of interference, and no threshold will minimize both forms for all controller densities. Another consideration is that losses due to modulated interference will decrease the throughput for a client as some messages will be lost, whereas continuous wave interference means that a client cannot receive any messages from any controller unless some readers stop transmitting RF power; this type of starvation is likely unacceptable for many applications.

5.2.3 Network-wide Effect on Client-to-Controller Interference

Increasing the sensitivity of clients essentially increases the downlink communication range of controllers (in the presence of continuous wave transmissions by other controllers). A side effect of this increased range is that it increases the likelihood of client-to-controller interference. Consider the diagram in Figure 5.7: the black dots are controllers transmitting at 1W, and the blue X is the location of a client that is backscattering a message to controller-A. If a second client located anywhere in the green region begins to modulate a message, it will also backscatter controller-A's continuous wave and that signal will be within 3 dB of the signal coming from the blue X. This will result in the message from the blue X being lost due to client-to-controller interference.

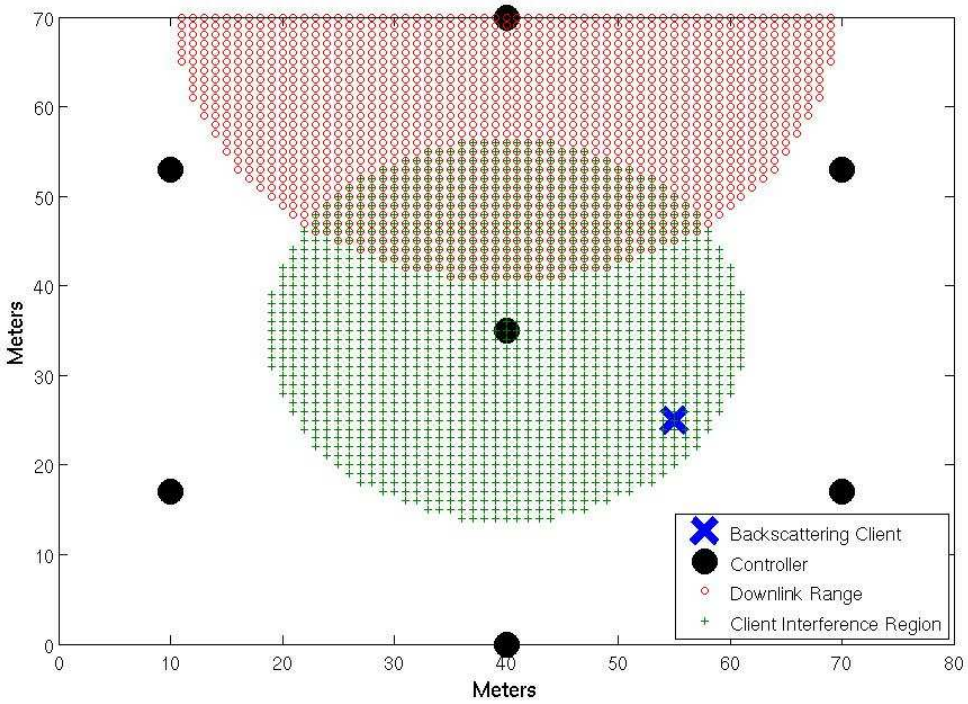


Figure 5.7: Illustration of regions where client-to-controller interference can occur.

The red region in the diagram indicates the downlink range of controller-B if clients can detect messages with a 10% modulation depth. Because these two regions overlap considerably, it is likely that controller-B will poll a client in the region of overlap, and the client will begin backscattering and interfere at controller-A. If the demodulator of the client was less sensitive, the region of overlap would be reduced and client-to-controller interference become less likely.

To show the extent of this effect, we simulated the case of 10 controllers randomly placed in the 70 by 80 meter grid along with 600 clients. Ten trials were performed with the locations of all nodes changing between trials. The controllers continuously polled their associated clients for data, and the number of uplink messages that were lost due to client-to-controller interference was recorded. To take into account the capture effect, if a client's

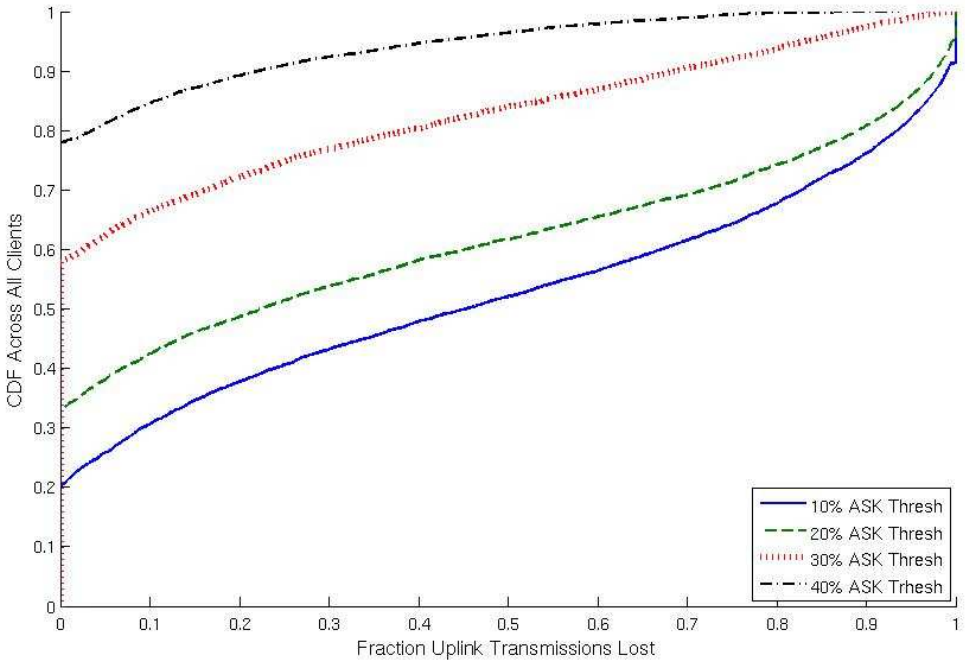


Figure 5.8: Impact of client sensitivity on uplink error rate.

message was 6 dB above the combined power of all interferers, then we determine that it was received error free.

Figure 5.8 shows the CDF of the fraction of uplink messages that were lost across all clients as the client ASK threshold is varied from 10% to 40%. When the threshold is 40%, nearly 80% of the clients experience no losses on the uplink, and only around 5% of the clients experience a loss rate greater than 50%. Of course, many clients did not transmit any messages because they suffer from continuous wave interference and were unable to detect polling messages from any controller.

In contrast, with a 10% ASK threshold, clients can always receive polling messages from at least one controller. However, only 20% of the clients could transmit uplink messages error-free, and around 10% saw *all* of the uplink messages being lost. This shows that the client sensitivity impacts network performance not only because it changes the balance

between continuous wave interference and modulated interference, but also the prevalence of client-to-controller interference. Any solution that addresses the interference problem in backscatter networks must account for this fundamental trade-off.

5.2.4 Summary

Increasing the client sensitivity reduces the likelihood of a client being starved because of continuous wave interference. Reducing this likelihood is desirable, because clients that cannot decode messages from any controller due to continuous wave interference will not able to communicate unless one or more controllers stops transmitting. Requiring that controllers often power down should be avoided because it reduces the amount of power available for harvesting, and it means that clients cannot transmit uplink data as they have no CW to backscatter.

However, increasing the sensitivity of controllers increases the prevalence of modulated interference, because clients are able to hear messages from more than one controller. In addition, because clients are now "in range" of multiple controllers, it becomes more common that two clients will backscatter to different controllers and their messages will interfere at the controllers. In the next section, we will show how these insights can be exploited to build a design for backscatter networks that will scale well as infrastructure is added.

5.3 A Network Design for Scalable Backscatter Networks

In this section, we use the insights of the previous section to develop our design for scalable backscatter networks. We first describe the goals of our design, and the operating assumptions our design targets. We then describe the PHY and MAC layer techniques that enable the network to scale well as infrastructure nodes are added. Our design consists of five main components. The first two components address the challenge of controller-to-client interference by (1) increasing the sensitivity of the client demodulator, thereby making clients more robust to continuous wave interference, and (2) having controllers use a CSMA-based approach to avoid modulated interference. In practice, mitigating these

two forms of interference provides the greatest benefit as they are the primary factors that limit network scalability. The next two components use techniques seen in the literature, (3) code-division-multiple-access for uplink client transmissions and (4) a controller-driven polling MAC protocol, to mitigate client-to-controller interference. The last component of our design (5) consists of two techniques that reduce the overhead associated with using CDMA for uplink transmissions. *This design allows controllers to operate in an always-on distributed manner, while still avoiding client starvation and without dramatically reducing capacity.*

5.3.1 Design Goals and Assumptions

In a nutshell, we aim to build networks that operate and can be deployed similarly to technologies like 802.11 or 802.15.4, but where clients consume orders of magnitude less power. We envision backscatter networks that provide communication coverage to large areas such as a home or office building, and can achieve link rates of 10-100s of kbps; this is on par with other low power wireless technologies such as 802.15.4 (i.e., ZigBee). A network consists of controllers that provide coverage to an area, and clients should be able to communicate with at least one controller to access the Internet or other infrastructure. As with 802.11, it should not be necessary to carefully manage deployment or have centralized coordination between controllers in a region to obtain good performance. The coverage and capacity of the network should increase when a reasonable number of controllers are added to the network.

We assume that, like Gen 2 readers, controllers operate under FCC regulations for the 900 MHz ISM band. This means that they frequency-hop across 50, 500 kHz channels and have a maximum transmit power of 1 W. To eliminate controller-to-controller interference, controllers do not transmit on the same channel at the same time. If more than 50 controllers are operating in a location, they will need to time-multiplex their transmission. We do not consider this extreme case. We also do not explicitly consider the cost, complexity,

or power consumption of controllers, though all techniques must be implementable using existing approaches. Controllers can sense the power of other controllers on other bands, and can determine if they are modulating. Additionally, out-of-band transmissions from other controllers can be supressed when receiving in-band client transmissions. We believe these assumptions are reasonable, as controllers are powered infrastructure.

Lastly, we assume that clients use an ultra-low power transceiver that operates in a power limited regime. In other words, if a client can communicate at all, it can communicate continuously. Ultra-low power backscatter transcievers already exist, with a power consumption of 1-2 μW which is is on par with the sleep-mode of the WISP [78]. In other words, if the WISP used one of these transceivers for communication, the range at which it could harvest and store energy would be equivalent the range at which it could communicate continuously.

5.3.2 Clients Use Sensitive Demodulators to Mitigate Continuous Wave Interference

Allowing controllers to be always-on increases power in the environment and allows clients to transmit concurrently on the uplink. To enable this while avoiding continuous wave interference, client demodulators should be more sensitive than Gen 2 tags. However, making them too sensitive means that modulated interference can be caused even from controllers that are very far away. *In our design, client transceivers are designed so that they can detect controller messages if they have a modulation depth of greater than 10%; Figure 5.6 shows that this avoids continuous wave interference even in dense networks.* This achieves a balance that eliminates nearly all cases of continuous wave interference, but does not make clients overly susceptible to distant modulators.

5.3.3 Controllers Use CSMA to Avoid Modulated Interference

To limit modulated interference, controllers should not transmit messages if they will cause interference at a client. *To avoid modulated interference, controllers use a CSMA/CA-*

based approach where they listen on all channels for modulating controllers, and hold off transmitting if the power of any modulator is above a certain threshold. A threshold that is too high will result in a greater likelihood of modulated interference, whereas threshold that is too low will reduce spatial reuse. Selecting an appropriate carrier-sense threshold is challenging and has been studied in the context of 802.11 [8].

One complicating factor in choosing a CSMA strategy is that, unlike conventional wireless networks, modulated interference is a matter of the relative signal strengths of the controllers as compared to the total power seen at the clients; if two controllers modulate at the same time and each accounts for greater than 10% of the power at a client, modulated interference will occur. However, controllers only interfere at clients located in regions where their signal powers are similar, and controllers transmitting continuous wave effectively "drown out" more distant modulators. *As such, we take the approach that controllers only defer transmission if the few nearest controllers are modulating, regardless of their absolute power.* The intuition is that more distant controllers are unlikely to have regions where their coverage areas overlap, particularly when many controllers are present. Through experimentation, we found that deferring to the 5-10 strongest controllers worked well across the network densities we considered, and we settled on deferring to the top 8 as a good compromise between likelihood of interference and good spatial reuse.

5.3.4 Clients Use CDMA to Mitigate Client-to-Controller Interference

Client-to-controller interference is challenging because clients backscatter all controller CWs, clients cannot hear each other to avoid transmitting at the same time, and it is difficult to coordinate controllers such that they do not simultaneously give channel access to clients that will cause interference. Code division multiple-access (CDMA) is set of techniques widely used for cellular and satellite communication that enable devices to transmit concurrently over the same frequency band. This makes it a good fit for mitigating client-to-controller interference, where all clients essentially share a single frequency band. Concurrent trans-

mission is achieved by assigning unique pseudo-random codes to transmitters, and having them XOR their data stream with the code before transmitting. Generally, the coding rate (or chipping rate) is much higher than the underlying bit rate, and the transmitted signal is consequently "spread" across a wider frequency band with a lower power spectral density. However, in our design, the chipping rate is equal to the original bit rate and transmissions are spread in time. The reason for this is that if a client transmission is spread across more than one 500 kHz channel, controllers on adjacent channels will occupy the same frequency as the client message and this will cause controller-to-controller interference.

To demodulate transmissions, the receiver correlates the received signal with the code of the transmitter. When many coded transmissions are concurrent, the combination of their signals may appear as random noise. However, by correlating with a known code the bit-stream of interest can be extracted. Codes are chosen such that their cross-correlation is low, which means that one code cannot easily be mistaken for another. Additionally, codes must have good auto-correlation properties such that there is one clear peak at a time lag of zero to allow timing information to be recovered.

In our design, CDMA codes are assigned to controllers. The particular code (or codes) used by a reader are indicated in the contention-period beacon. Clients then code their data transmissions using the appropriate code, using both the code directly or the inverse of the code to transmit one bit. Controllers use the known code to recover the transmitted data, and a preamble is used to indicate the starting polarity so that the controller can determine if the code was inverted.

Code Selection and Demodulation

CDMA systems can be synchronous or asynchronous. In synchronous systems, transmitters begin their transmissions at the same time and can use orthogonal codes to encode their transmissions, e.g., when a cellular basestation combines messages intended for multiple clients. Orthogonality between codes means that they do not interfere with each other at the

receiver at all. Asynchronous systems, on the other hand, do not require that transmitters are synchronized. This is the appropriate choice for our system so that controllers do not have to be tightly synchronized. The challenge with asynchronous-CDMA systems, however, is that it is not possible to generate codes that are strictly orthogonal at all offsets. Instead, codes are used that are statistically uncorrelated, which result in some interference between codes.

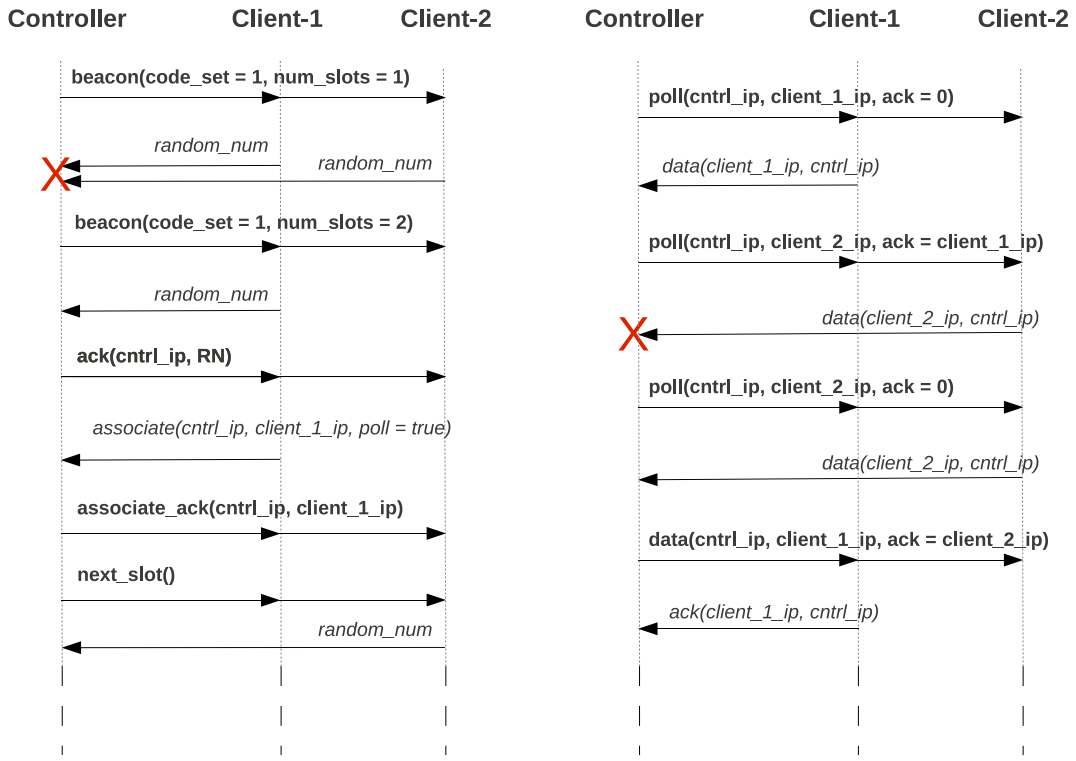
We use Gold codes [28], which are commonly used in asynchronous CDMA systems because they have good cross-correlation properties, with bounded cross-correlation between any two codes at all offsets. The degree of cross-correlation between codes is a function of the code length, with longer codes having better characteristics. However, while longer codes provide better detection performance they do so at the cost of a lower bit rate for a given link frequency. To maximize throughput, we use the shortest code length that is effective for our operating scenario. When using a code length of N, there exists $N + 2$ codes with low bounded cross-correlation. We select 31 chip codes as they are the shortest codes that have reasonably low cross-correlation values. [2]

Controllers use a matched filter to decode client transmissions. The benefit of correlating with a matched filter is that it is simple and works well when the signal of interest is similar in power to the interferers. However, matched filter correlation is not near-far resistant, so a client message cannot be detected in the presence of a much stronger interferer. We find in our evaluation that our simple approach reduces client-to-controller interference significantly and achieves good performance. However, more advanced near-far resistant detectors are well-studied and could be directly applied [81]. We leave this to future work.

5.3.5 A Polling MAC Protocol to Arbitrate Client Channel Access

The MAC design is inspired by the Gen 2 protocol [31] and the point coordination function (PCF) of 802.11 to provide general purpose communication for backscatter networks. A

[2] Gold codes are generated using a "preferred pair" of m-sequences, and no pair exists to generate codes of length 15.



(a) Contention Period

(b) Contention-Free Period

Figure 5.9: Example packet exchanges for the contention and contention-free periods. In the contention case, there is a packet loss due to a collision. In the contention-free case, there is a packet loss due to noise.

contention period is used to learn about new clients and to associate them with a given controller, and a contention free period is used to efficiently transmit downlink traffic to clients and to poll them for uplink traffic. Prior work has shown that contention free approaches are well suited to backscatter networks [82], as CSMA-based protocols are precluded and Aloha-based protocols are inefficient; in the best case, only around 40% [93] of transmissions will be collision-free when using Slotted-Aloha.

The Contention Period and Client Association

During the contention period, clients can gain asynchronous access to the medium. Slotted Aloha protocols work well in this context, and our contention period is modeled after the Gen 2 anti-collision mechanism[3]*.* The purpose of the contention period is primarily to allow association from new clients. An example message exchange is shown in Figure 5.9(a). Controllers start a contention period by transmitting a beacon that includes the code-set used by the controller (explained in Section 5.3.4), and the number of slots in the round. Clients randomly choose a slot in which to reply, and if a single client responds in a slot, the controller will acknowledge it and it can begin transmitting messages. If clients collide in a slot (as shown by the red X in the figure), they wait for another beacon and try again. Controllers transmit beacons, adjusting the number of slots accomodate the number of clients, until all clients have had a change to transmit

During association, the client can request to be polled during the contention-free period. If they do not expect to have significant uplink traffic, they can forego polling and transmit data to the controller during the contention period. This may be desirable to reduce overhead from polling, particularly as the contention period should have low contention as clients with significant traffic will transmit during the contention free period, and new clients will likely be rare.

The Contention-free Period

During the contention free period (CFP), the controller can transmit downlink traffic at any time, and poll associated clients for uplink traffic. An example message exchange for the contention-free period is shown in Figure 5.9(b). Controllers transmit *poll* messages to their associated clients, and clients respond with data or a short NACK message to let the controller know it is still in the network. Acknowledgements are piggy-backed on *poll* messages using an "ACK" field which contains the address of the client whose data is being

[3] Any of the many Aloha-based protocols would be sufficient for our purposes.

acknowledged. Along with polling clients for data, controllers can transmit data to clients during the contention-free period.

We see three open questions concerning the CFP. The first question is how to reduce overhead, for example by using short pseudo-handles in polling messages. The second question is how to decide, and who decides, if a client should transmit data during the contention period or the CFP. The third question is how to integrate QoS requirements into the protocol. We leave these MAC-related questions to future work, and focus instead on the design aspects pertaining to interference in the network.

5.3.6 Increasing Uplink Throughput

Backscatter transmissions have an underlying link frequency, the frequency at which they modulate their transistor to backscatter the CW, and this frequency determines the bandwidth consumed by the transmission. Client transmissions must remain in-band, i.e., consume less than 500 kHz worth of spectrum. Otherwise, the client transmissions will extend into adjacent bands where other controllers may be transmitting. Because the client transmission will occupy the same frequency band as adjacent controllers, controller-to-controller interference will occur. Because our design has clients code their messages using Gold codes with 31 chips and the link frequency should not be increased to remain in-band, it takes nearly 16 cycles of the link frequency to transmit one bit.

We use two techniques to offset this reduction in bit rate. First, multiple codes are assigned to each controller, and each code then communicates more than one bit of information. The 33 codes are broken out into some smaller number of code sets, and different sets are assigned to different controllers. Clients use different codes from the set to transmit multi-bit symbols, with the bits per symbol being equal to $log_2(\#\text{ codes in set} \times 2)$. For example, if the coverage area of a given controller will overlap with no more than 8 other controller's coverage areas, the 33 codes can be broken into 8 sets of 4 codes each. Controllers that have overlapping coverage areas cannot use the same code set. Clients can then

transmit any of the 4 codes or their inversions, which results in 8 distinct symbols carrying 3 bits per transmitted code. This means that the uplink bit rate of clients will be 5.33 times lower than the rate of a modulation that transmitted one bit per cycle of the link frequency (such as FM0 encoding used in Gen 2).

The second technique we use is to only have clients code their transmissions when they are likely to be interfered with. For instance, if a client is very close to a controller, there is likely no other client whose signal power would be great enough to cause interference, but that is associated with a different controller. Such clients do not need to code their transmissions. Our approach is similar to rate adaptation seen in 802.11 networks. In our design, controllers use historical data to determine if a client should code its messages or not. Clients always associate using coded messages, and then controllers specify in the polling message if a client response should be coded or uncoded. Controllers keep track of the packet-error rate of clients, and if the historical throughput is below the expected throughput using coded messages, the controller tells the client to code its packets in the polling messages. To revert to the higher rate if conditions change, once 10 coded messages are received without error the controller tells the client to attempt an uncoded message, and uncoded packets are sent until the throughput again drops below the expected rate. In the worst case, when interference is high and all uncoded messages are lost, probing once every ten packets adds only 3% overhead.

5.3.7 Summary

There are five main aspects to our design; the first two mitigate downlink interference and the last three mitigate uplink interference. A novel contribution of this dissertation is showing the distinction between continuous wave and modulated interference on the downlink. As we will show in the next section, first two aspects which mitigate these two forms of interference provide the largest benefit in practice. The last three aspects which address uplink interference have been seen in prior work, and function here to complete the

network design.

- To mitigate continuous wave interference, client demodulators are tuned to detect modulation depths greater than 10%. This makes them resistent to continuous wave interference, but prone to modulated interference.

- To mitigate modulated interference, controllers use a CSMA/CA-based approach to avoid modulating when it is likely to cause interference at clients. To achieve a good balance between interference avoidance and spatial reuse across network densities, controllers defer their transmissions only if the strongest controllers (from their vantage point) are modulating.

- To mitigate client-to-controller interference caused by clients associated with different controllers, asynchronous CDMA is used for uplink communication. Each controller has a unique set of Gold codes, and clients code their uplink transmissions. Controllers use a matched filter to decode messages from their clients.

- To arbitrate channel access among clients associated with a single controller, the controllers use a polling MAC protocol. A contention-period (similar to the anti-collision protocol of Gen 2 RFID) is used to associate new clients with a controller. Then, a contention-free period is used where the controller transmits downlink data and polls clients for uplink data.

- To reduce the CDMA overhead, we use two techniques. 1) Controllers use one of eight codes sets, with each consisting of four codes. Clients use different codes (and their inversions) to transmit 3 bits per symbol. 2) Controllers use historical packet loss information to determine if a client should be coding their transmissions or not. For instance, clients very close to a controller can use a higher rate modulation as other clients are unlikely to interfere with their transmissions.

5.4 Evaluation

In this section, we evaluate the network design developed in the previous section. We first describe the experimental methodology and the network simulator that we use for the evaluation. We then show how well our design performs compared to the Always-On and Listen-Before-Talk approaches. We find that our design avoids the problem of unfairness and client starvation seen with the Always-On approach, and significantly improves the network capacity compared to Listen-Before-Talk. Most importantly, the network performance is stable as the network scales, unlike Always-On or Listen-Before-Talk where performance degrades dramatically as the network grows. Lastly, we present supplemental results to quantify the impact of each aspect of our design.

5.4.1 Experimental Methodology

The goal of our network design is to enable backscatter networks that scale well. A network should provide good performance not only when it consists of a single controller and a few clients, but also when it grows to hundreds or thousands of devices on a building-wide scale. First and foremost, a network should be able to provide complete coverage to an area, with all clients having connectivity. In addition, this condition should hold even when the density of infrastructure nodes is high, and achieving it should not dramatically reduce the capacity of the network.

As we am primarily concerned with how the design will work for large-scale networks, the primary tool we use in this evaluation is a wireless network simulator custom-built to model networks of hundreds (or more) devices. Care was taken to make the simulation as realistic as possible, including frequency hopping and a Rayleigh fading channel. This simulator allows me to demonstrate the soundness of the design, and we leave real-world deployment using custom-built hardware to future work. Moreover, a strength of the design is that the physical layer components, i.e., CSMA and ASK modulation on the downlink and CDMA for the backscatter uplink, are well-known and have been applied in the context

of RFID or other technologies. As such, we forgo table-top demonstrations that show the feasibility of these physical layer techniques.

One key metric we use throughout the evaluation is the "normalized uplink throughput" of the clients. In the topologies we evaluate, controllers have differing numbers of associated clients, and clients are polled for data in a round-robin fashion. This means that the greater the number of clients that are associated with a given controller, the lower their throughput will be in terms of packets per second. To account for this effect, the throughput of each client is normalized by the throughput that it would achieve if the controller were operating in isolation. This normalizing factor is solely a function of the uplink and downlink data rates, the packet sizes, and the number of clients associated with the controller. In the ideal (though unrealistic) case, all clients would have a normalized throughput of 1 because other devices in the network would have no effect on their performance. In practice, normalized throughput will be well below 1, as interference does exist and must be avoided in some manner.

5.4.2 Simulator Implementation

Propagation Model The simulator models a network consisting of controllers and clients in a 2D plane. Controllers transmit at 1W using omnidirectional antennas, and we assume tags are power limited with a threshold of 2 μW. Friis equation is used to calculate the power of all controller transmissions at all clients, and the power of all clients at all controllers. Because backscatter controllers transmit their continuous wave on a single frequency, frequency-selective multipath-fading is a dominant factor in how much power is received on both the up and downlink. We model multipath-fading by attenuating the received power according to the Rayleigh distribution, which models fading in indoor environments with a large number of reflectors. According to FCC regulations, controllers frequency hop every 400 ms, and the power received from all devices to all devices is recalculated.

The Physical Layer Model Controller receiver performance is modeled after an existing RFID reader [49], which has a sensitivity of -80 dBm and a noise figure of 25 dB. The SNR (in dB) of a signal for a receiver is given by:

$$SNR = sensitivity - (NoiseFigure + 10log_{10}(BW) - 174dBm)$$

where BW is the bandwidth of the signal, which is set to 500 kHz to model FCC rules for the 900 MHz ISM band. This means that a signal received at -80 dBm has an SNR of 12 dB. We calculate SNR to BER mappings for two modulations, FM0 [98] and BPSK CDMA [52]. To calculate the signal-to-interference-and-noise-ratio (SINR), we sum the signal powers in the case of FM0. For the CDMA case we assume worst case correlation properties of the interferers, and we sum the power of their signals scaled by this correlation factor according to [52]. Keeping with FCC bandwidth requirements, the bit rate of the downlink is 135 kHz (the maximum rate for a pulse-interval encoded, PR-ASK signal causing -60 dB adjacent-channel interference), and the uplink has a link rate of 250 kHz. This means that an FM0 encoded messages will have a bit-rate of 250 kbps. To allow for dense reader deployments, we use 8 CDMA code-sets of 4 codes each. This allows for up to 8 controllers to have coverage areas that overlap, and results in a coded uplink bit-rate penalty of 5.33x; this translates to a bit-rate of around 47 kbps.

MAC Layer and Traffic Model As we are mainly interested in the interference effects in the network, only the contention-free period is simulated. Clients are associated with the strongest controller, and communicate only with that one controller. Controllers continuously transmit 32-byte polling messages to their associated clients in a round-robin manner, and clients respond with data packets of size 32, 64, 96, or 128 bytes. We borrow the minimum and maximum packets sizes from 6LoWPAN [95], as it is designed to integrate low-power networks with the Internet so the packet formats are good candidates for our

use. Though relatively simple, the simulator is sufficient to give insight into the interference problem and to explore the benefits of our design.

5.4.3 Providing Good Network Coverage

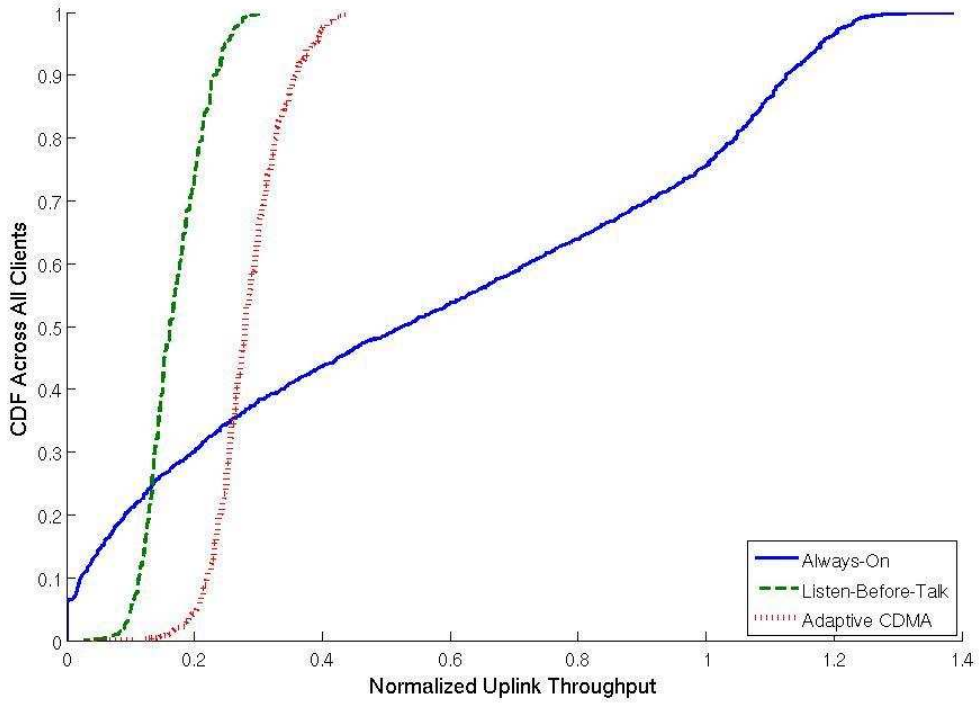
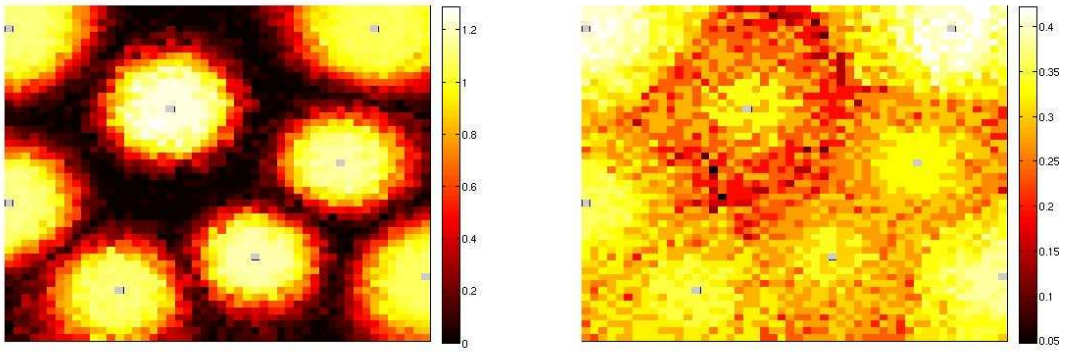


Figure 5.10: Performance of the three approaches for a minimally covered area

Figure 5.10 shows the normalized throughput for Always-On, Listen-Before-Talk (LBT), and our Adaptive CDMA approach for a minimally covered 50 x 50 m area. Minimal coverage was achieved by adding controllers to random locations until a client at any location could backscatter to at least one controller with a received power greater than -80 dBm. Then, 250 clients were randomly distributed, and ten trials were simulated with controllers and clients being redistributed between each trial.

The data for Always-On and Listen-Before-Talk (LBT) are the same as in Figure 5.3,



(a) Always-On (b) Adaptive CDMA

Figure 5.11: Heatmaps of normalized throughput (Note: Colormap ranges are not equal.)

and show how these existing approaches result in high performance for some clients and starvation for others (Always-On), or more uniform but low rates for all clients (Listen-Before-Talk). In contrast, our adaptive CDMA-based approach provides a balance between the two extremes. No clients experience starvation (as compared to 7% starvation for Always-On), and only the bottom 1% of clients have a normalized throughput of less than 16% (in contrast, only 48% achieve this rate using LBT, and only 27% in the case of Always-On).

In the case of Always-On, some clients have a normalized throughput greater than 1. This is because clients that are starved do not transmit uplink packets, which gives other clients more opportunities to transmit than they would get if the controller was operating in isolation. Hence, the high throughput for some clients is a direct result of other clients having very poor connectivity. In contrast, the Adaptive CDMA-based approach provides good connectivity to all clients in the network.

To better understand the behavior of the Always-On and the Adaptive CDMA approach, we reran the experiment with a tag at every location in the grid. Figure 5.11 shows heatmaps when 8 controllers are present (which is sufficient to provide coverage everywhere), with brighter colors indicating higher normalized uplink throughputs. In the case of Always-On,

it is clear that clients close to controllers see very high rates, while clients located between controllers see low rates or are starved completely due to continuous wave interference. In contrast, the Adaptive CDMA approach provides more uniform performance, though clients in intermediate regions still have lower throughputs. This is because modulated interference can still occur, and our simple matched-filter detector cannot eliminate all uplink interference.

5.4.4 Scaling Well with Network Density

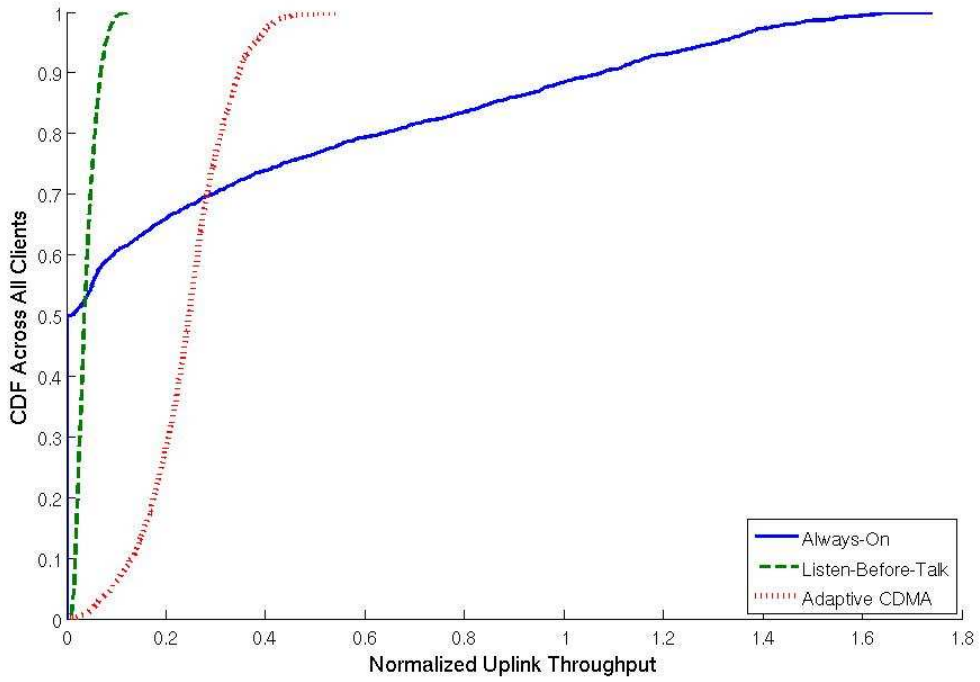


Figure 5.12: Performance of the three approaches for a densely covered area

The prior data was for a minimally covered area. However, more densely deployed infrastructure provides more power to clients, and should also increase the overall capacity of the network as each controller has fewer associated clients. Figure 5.12 shows data for a network where 40 additional controllers were added beyond those needed to minimally cover

the area. In this case, the drawbacks to the prior approaches become more severe. Always-On results in nearly 50% of the clients having no connectivity, and the median normalized throughput for both Listen-Before-Talk and Always-On is less than 4%. In contrast, with our approach no clients starve, the median normalized throughput is 25%, and less than 1% of clients achieve a normalized throughput of 4% or less. This shows that our approach works well even in densely deployed networks.

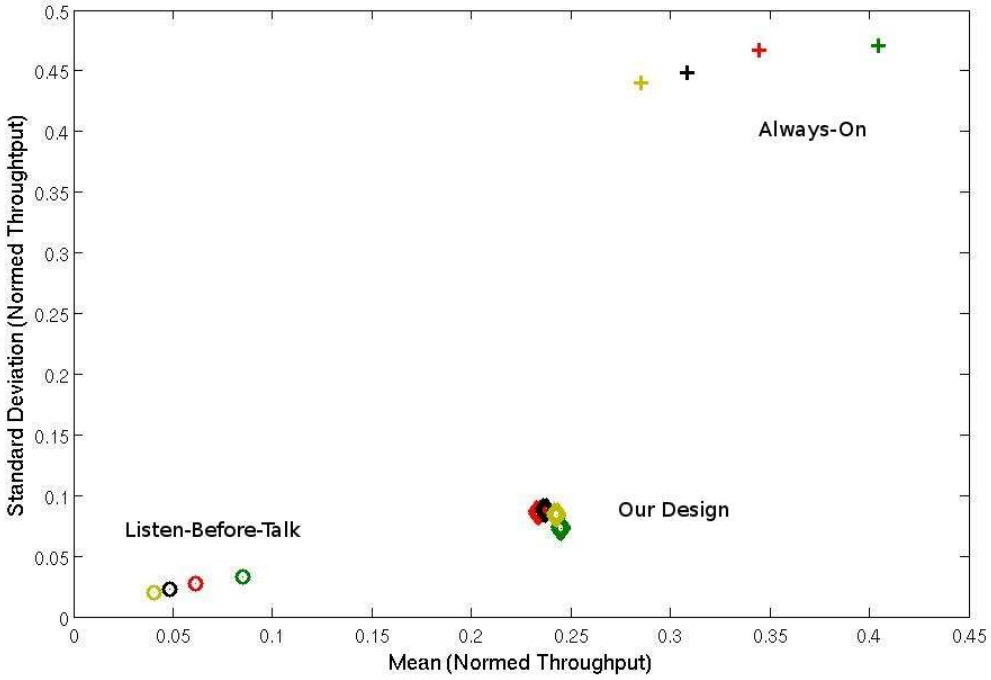


Figure 5.13: Fairness vs throughput for the three approaches (+: Always-On, ◯: Listen-Before-Talk, ◊: Adaptive CDMA) across 4 network densities (Green: minimal coverage + 10 controllers, Red: min + 20, Black: min + 30, Yellow: min + 40)

To illustrate how the three approaches behave in terms of fairness and client throughput as network density increases, we plot the standard deviation of the normalized client throughputs versus the mean of their normalized throughputs in Figure 5.13. The figure shows data for four network densities in a 50 x 50 m area: the first is the minimal level of

coverage plus 10 additional controllers, and the last three densities have an additional 10 controllers added in each case.

Across all network densities, the Always-On approach has a standard deviation around 0.45. This further illustrates that the network is unfair, because some clients have normalized throughputs greater than 1 while others have a normalized throughput of 0. Additionally, the mean normalized throughput of clients degrades by nearly 30% as the density increases from having 10 additional controllers to having 40. This is because the increased density means more clients suffer from continuous wave interference and are unable to communicate with any controller, as seen in Figure 5.12.

In the case of Listen-Before-Talk, the standard deviation is less than 0.05, which means all clients see approximately the same level of performance. Unfortunately, the mean normalized throughput is poor in all cases, and degrades by more than 50% as the density increases. This is because with higher densities, there are more controllers contending for the medium.

In contrast, the Adaptive CDMA approach has a standard deviation of less than 0.1 in all cases, the minimum mean throughput is 24% of the rate clients would achieve if the controller were operating in isolation, and this value changes by less than 1% across the 4 densities. This means that our approach results in a network that is both fair, and provides good performance across network densities. In fact, the network performance is largely independent of network density. Hence, our approach enables backscatter networks to scale well as infrastructure is added.

5.4.5 Reducing Interference

Our network design improved network performance by reducing both uplink and downlink interference. However, interference is not eliminated entirely. Figure 5.14 shows the loss rate on the downlink when using no CSMA, a conservative -36 dBm threshold, and our adaptive CSMA scheme. The data is from experiments run with a minimally covered area with 40

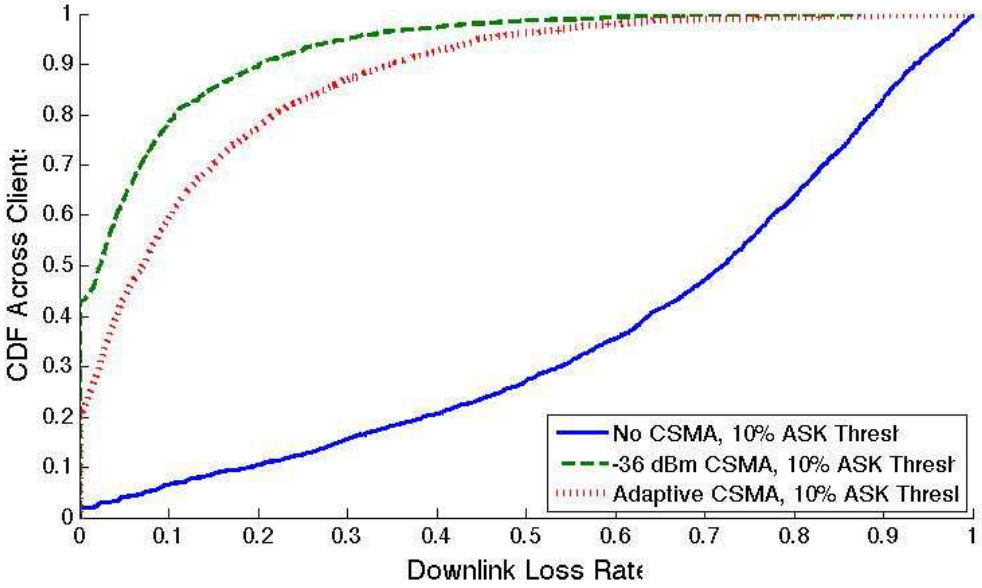


Figure 5.14: Downlink loss rates for no CSMA, a fixed -36 dBm CSMA threshold, and our adaptive scheme. (All use a 10% ASK threshold)

additional controllers. In all cases, clients have a 10% ASK threshold because continuous wave interference otherwise dominates.

In the no-CSMA case, only 2% of the clients have a loss rate of 0% (i.e., all downlink packets were received without errors). The median value is 72%, which means that for half the clients, at least 72% of the packets transmitted to them were lost. In contrast, when using the CSMA threshold of -36 dBm, more than 40% of the clients have no downlink errors, and 90% of the clients suffer loss rates of 20% or less. However, as shown previously, this comes at the cost of poor network performance because many opportunities for concurrency are not exploited; in this configuration, the median normalized uplink throughput is just 7%. In contrast, our adaptive CSMA approach greatly reduces the error-rate compared to the no-CSMA approach, but accepts slightly more errors than Listen-Before-Talk in the interest of higher network performance. This shows that our approach of setting the CSMA threshold to avoid only the top 8 interferers finds a good operating point.

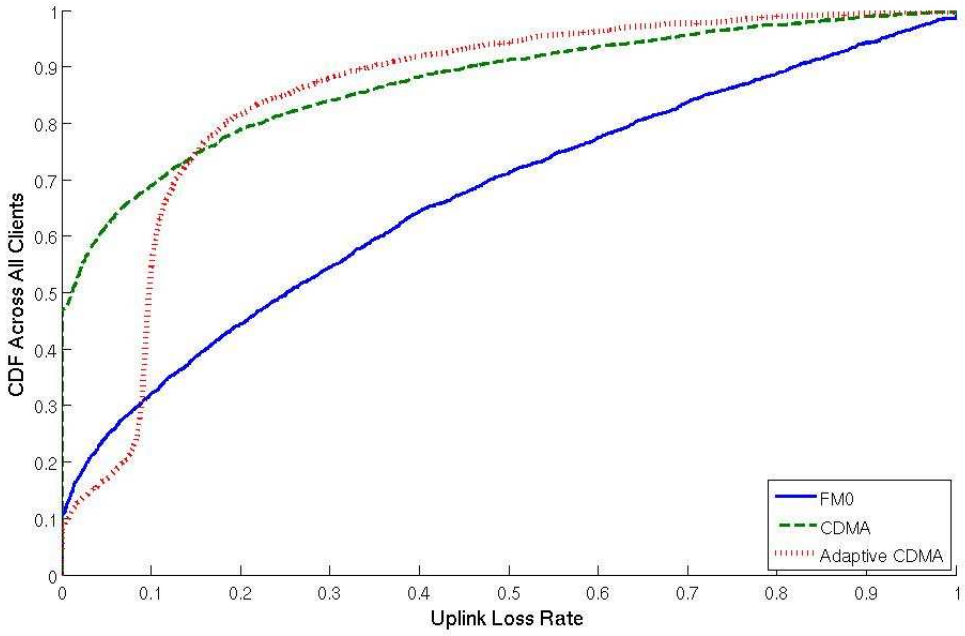


Figure 5.15: Uplink loss rates for FM0, CDMA, and our adaptive scheme.

Similarly, using CDMA on the uplink does not completely eliminate uplink interference. Figure 5.15 shows the uplink loss rates when always using an FM0 uplink, always using the CDMA uplink, or using our adaptive approach. With FM0, only 10% of the clients have error-free communication (these are the clients close to their controller), and the median is 25%. In contrast, when using CDMA for all uplink packets, nearly 50% of the clients have error-free communication. However, 10% of the clients still have error rates above 45%, which means that CDMA is not a panacea. This is due to the near far problem, where a strong interferer will still cause errors. This is a limitation of our simple matched-filter approach to decoding, and a more complex signal processing would further reduce the loss rate. For the adaptive approach, the median loss rate is 10%, with most clients having loss rates between 8 and 15%; this is an artifact of the probing mechanism where an uncoded messages is transmitted after 10 successful coded packets. Though the rate adaptation results in more packet errors than the CDMA approach, it only incurs a 3%

overhead in the worst case, and we will show in the next section how it improves overall network performance.

5.4.6 Benefit of Individual Techniques

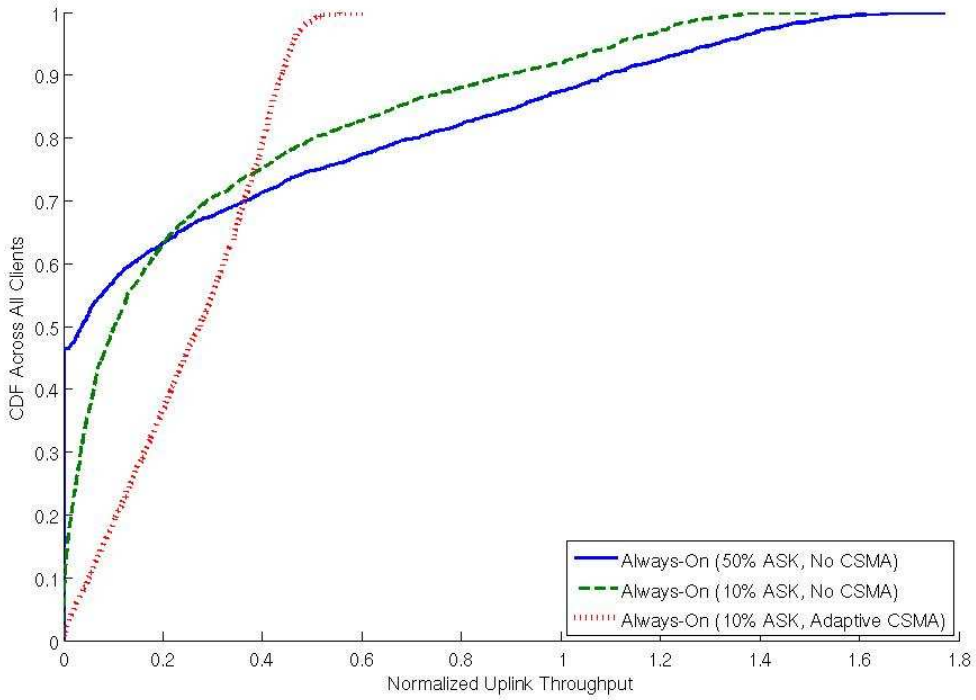


Figure 5.16: Performance for Always-On with different client thresholds and with or without CSMA

In this section, we investigate the benefits of the individual aspects of the network design: a more sensitive client demodulator to reduce CW interference, carrier-sense to avoid modulator interference, and CDMA to reduce uplink interference. Figure 5.16 shows the uplink throughput for three network configurations in the scenario where the area is minimally covered, and then 30 additional controllers are added. The blue curve is the Always-On approach seen previously, where the client sensitivity is set to 50% modulation depth and many clients suffer from continuous wave interference. The green curve shows

the throughput when clients use a 10% threshold, but controllers do not use CSMA to avoid modulated interference. Simply increasing the receiver sensitivity reduces the fraction of clients with no connectivity from 47% to only 8%. However, 25% of the clients still achieve normalized throughputs of less than 2.5%. The red curve shows the benefit of using the adaptive CSMA mechanism to avoid modulated interference. Here, only 1% of clients have no connectivity, and only 6% of the clients have normalized throughputs of less than 2.5%. However, this comes at the cost of no client having a normalized throughput greater than 60%.

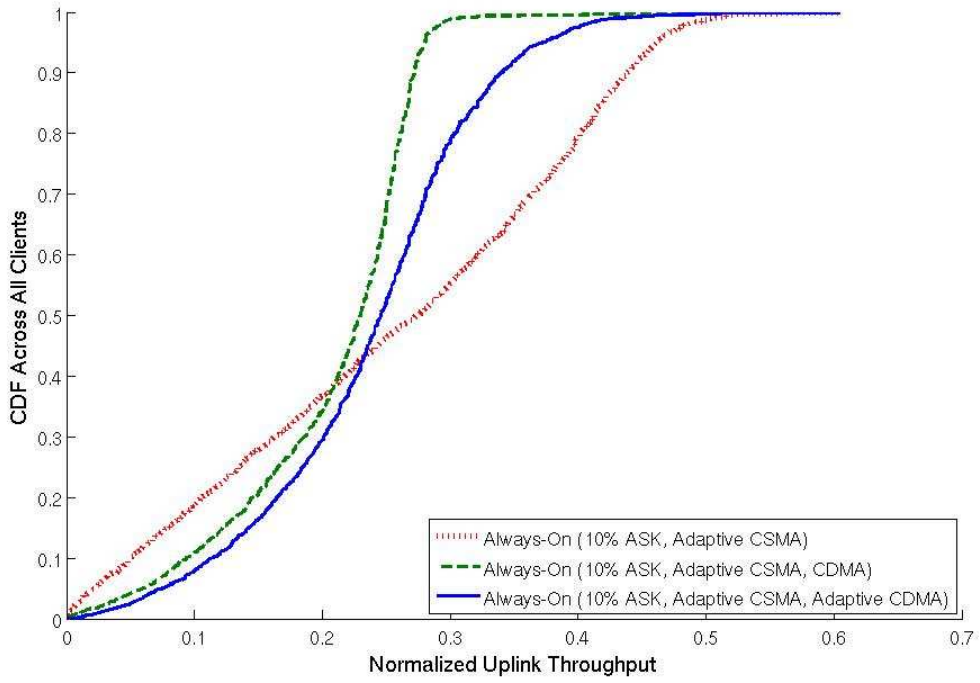


Figure 5.17: Performance for Always-On with sensitive clients and CSMA, and CDMA and Adaptive CDMA

Next, we show the benefit of a CDMA uplink, and the approach of adapting to a higher rate if interference is unlikely for a given client. Figure 5.17 shows the normalized uplink throughput for three more network configurations. The red dotted line is the same data

as the red dotted line in the prior graph, where clients have a 10% ASK threshold and controllers adapt their CSMA threshold. The green dashed line shows the impact of using CDMA for all uplink transmissions. This has the effect that all clients have connectivitity, and the bottom 38% of clients achieve higher throughput than if CDMA is not used; even though using CDMA entails a 5.3x uplink penalty. This means that uplink interference was the limiting factor for those clients. Lastly, the solid blue line shows the benefit of rate adaptation on the uplink. If clients do not suffer from interference, for example because they are very close to a controller, they will adapt their modulation and use the higher bit-rate FM0 modulation. This has the effect of improving performance in all cases, because frequency-selective fading and channel hopping results in clients experiencing constructive interference on some frequency and their SINR improves to the point that FM0 works well. Moreover, the top 25% of clients improve by 20% or more because they can can use FM0 most of the time.

5.5 Summary

In this chapter, we first use measurement and simulation results to show that networks built using existing backscatter technology do not scale well, largely due to downlink interference. We next introduce the distinction between *continuous wave* and *modulated* interference on the downlink, and show how a more sensitive client demodulator that trades resilience to continous wave interference for susceptibility to modulated interference can improve network performance.

We then present a network design that combines a sensitive client demodulator (to mitigate continuous wave interference) and a CSMA-based scheme for controllers (to avoid modulated interference), with CDMA client transmissions and a polling MAC protocol (to mitigate uplink interference). Experimental results show that this design outperforms existing approaches. Moreover, it achieves nearly identical levels of performance for both sparse networks and networks that are densely deployed, which shows that the design achieves our goal of scalability.

Chapter 6

RELATED WORK

In this chapter, I present work related to this dissertation. I first highlight important work in low-power computing and communication outside of the power harvesting literature. I next look at systems that harvest power from sources other than RF, and then move on to prior work in RF-powered computing. Last, I give an overview of other work on interference mitigation in backscatter networks.

6.1 Low Power Computing and Energy Management

Computer systems researchers have long strived for techniques to increase the energy efficiency of server, desktop, mobile, and embedded computers. Hardware trends have been a driving force in improving the energy efficiency of computing systems in general [67, 59], and there has been significant work that targets ultra-low power computers [19]. These forces will continue to increase the energy efficiency of low power computers, which will extend the range and capabilities of RF-powered computers.

A recent trend in hardware design is the development of platforms that are more flexible in terms of their energy consumption. For example, energy-proportional computing has been proposed for data center scenarios [5]. Advances in server technology have improved energy efficiency at peak load, or when idle, but have not addressed efficiency at intermediate loads. This is a poor fit for modern data centers where servers are rarely operating near their maximum utilization or completely idle. For battery-powered devices such as mobile phones, there has been considerable work towards more flexible platforms. For example, dynamic voltage scaling has been a key technique whereby the supply voltage and operating frequency can be scaled to support both good peak load performance and efficient

background activity [38, 77]. The furthest extent of this trend can be seen in microsensor design, where it is argued that the operation of all aspects of the hardware (including the memory subsystem, radio hardware, and ADC) should be scalable to achieve good energy efficiency across a range of sensing tasks and environments [17].

Given new hardware capabilities that allow systems to more closely manage their energy consumption, there is the question of how software can take advantage of those capabilities. In the case of duty-cycling between an idle mode and an active mode, there has been considerable work in the sensor network literature. For example, TinyOS, a widely-used operating system for sensor platforms, automatically puts the processor to sleep when the task queue is empty [42]. Moreover, the dominant factor that determines the power consumption of sensor nodes is whether the radio is active and able to transmit and received messages. This has led to an astonishing number of proposals for duty-cycled MAC protocols [2, 120] that can provide good network performance while minimizing energy consumption. Unfortunately, it is difficult to apply these techniques in the context of RF-powered computers, because they seek to keep long-term energy expenditures below long-term harvesting or to maximize node lifetimes measured in days [64], whereas RF-powered computers aim to match their average power consumption to the available power, which can vary by orders of magnitude over short time-scales. Additionally, how to manage fine-grained mechanisms that go beyond simple duty-cycling is still an open question. It will be interesting to see if these mechanisms make their way into RF-powered platforms, and how they can be leveraged to better match device behavior to available energy.

6.2 Running Programs on RF-Powered Computers

Aside from Dewdrop, runtimes for RF-powered computers have used a simple model for duty-cycling, where devices begin program execution when a fixed voltage threshold is reached, and stop executing when the program finishes or the device runs out of stored energy. As a consequence, work related to running programs on RF-powered computers has mainly tackled the problem of maintaining state across power losses. In [84], the authors

use offline profiling to estimate when state should be saved on the WISP, or transmitted to the infrastructure [87], due to impending depletion of the energy store. This approach requires that tasks are fixed and consume a known amount of energy per execution, and their offline profiler does not consider nonlinearities inherent to real devices.

Other work has focused on reducing the energy cost of persistent memory so that saving state locally consumes less energy and can be used more often. To do this, they use the insight that FLASH memory can be written reliably at a low voltage by writing the same data multiple times [88], and that this can consume less energy overall than writing once at a high voltage. Looking forward, the use of ferro-electric random-access memory (FRAM) [34], which consumes very little power to read and write and retains data even after it is powered off, has the potential to greatly simplify program execution for many tasks on RF-powered computers. For example, if a program can pause at an arbitrary point in its execution, it can simply execute until the device runs out of energy, and restart when more has been harvested. Microprocessors that use FRAM are already available [103], and the technology continues to improve. However, scheduling run-to-completion tasks will still require special consideration, as sufficient energy for them to complete must be stored before they begin execution.

Lastly, to avoid programs running out of energy before completing their execution, offline profiling has been used to determine the appropriate capacitor size for a device designed to execute a particular task [40]. For example, if a task will consume a large amount of energy, the device will be outfitted with a large capacitor. When the device wakes at its fixed voltage threshold, there will sufficient energy to run the program. The drawback to this approach is that it requires hardware modifications to support different tasks. Dewdrop enables a given platform to run a wide range of tasks, though hardware tuning may be required for applications with vastly different energy requirements.

6.3 Other Energy Harvesting Systems

In the context of sensor networks, the design and system trade-offs of solar harvesting systems have been widely studied [51, 101, 9, 83]. This body of work looks closely at how hardware, such as the size of the solar cell and energy store, should be provisioned based on the target tasks and operating environment. Many design insights apply to RF-powered platforms; e.g., [51] finds that capacitors should be used as the primary buffer to tolerate rapid charge/discharge cycles.

Given hardware that can harvest solar energy and store it locally, the scheduling problem for energy harvesting devices has also been considered [48, 54, 68, 53]. A key idea is that device operation should be "energy neutral", i.e., energy consumption should not be more than the energy provided by the environment. This is achieved by predicting the amount of energy that will be harvested in the future, and modeling the amount of energy that is currently stored. However, the scheduling problem for these systems differs significantly from the RF-powered devices we consider as they manage tasks and harvested power on the order of days, energy consumption and harvesting patterns are predictable, and there is no penalty for storing excess energy.

Beyond solar, other harvesting techniques have been proposed for mobile and embedded computers [75]. Examples are thermoelectric conversion that exploits temperature gradients to generate power [100], piezoelectric materials [97] and vibrational excitation [66] that harvest mechanical energy, and even systems that harvest energy by breaking down glucose inside the body [55]. Though the constraints of these technologies differ from those of RF power harvesting, devices based on them still need to match their energy consumption to available power. As such, they will likely benefit from the techniques developed for RF-powered devices.

6.4 Interference in Backscatter Networks

Managing interference in backscatter networks has been a major focus of the RFID community. The first type of interference is when backscattered messages from multiple clients interfere at the controller. Because clients cannot hear each other's transmissions, mitigating collisions between clients is challenging and has been the focus of most protocol work seen in the RFID literature; a recent survey gives an overview of 42 distinct proposals [57] targeted at RFID networks, and our prior work introduces a protocol to gather high rate sensor data from backscatter sensors [15]. While more efficient MAC layer protocols have been proposed to increase performance for a single controller [39, 76], our approach makes it feasible to build large-scale networks consisting of many controllers. Beyond MAC layer approaches, successive interference cancellation [96] and code-division multiplexing [65, 69] have been used to enable concurrent client transmissions.

The second and third type of interference are controller-controller interference, and controller-client interference [30]. Solutions [10] to the first of these have ranged from listen-before-talk [32], to frequency-hopping [31], to distributed channel assignment [36]; our approach assumes such a technique is used to avoid controller-controller interference. Proposals that explicitly consider controller-client interference have focused exclusively on time-multiplexing readers that are in range of each other [111, 43, 18]. This work has been purely theoretical, and has not explored the mechanism of interference or measured it in practice.

While the majority of the work related to backscatter networking has focused on RFID, using backscatter for ultra-low power sensing has been explored [109, 7]. However, clients in these system are transmit only, have a data rate of 10 bps, and have no receive capability. Interference is avoided on the uplink by having every sensor transmit on a different subcarrier, and networks of multiple controllers are not considered.

Chapter 7
CONCLUSIONS AND FUTURE WORK

RF-powered computers have the potential to provide deeply embedded, long lived computation and sensing. They can be small and operate indefinitely as they harvest energy using a paper thin antenna rather than carry an onboard battery, and they can be embedded inside objects, structures and the human body. Though the technology is still in its early stages, technology trends suggest that the capabilities and operational range of RF-powered devices will increase exponentially in the coming decades. In this dissertation, I have identified and proposed solutions for two problems that limit the environments in which RF-powered devices can operate and the applications they can support. Specifically, I address the problems of how varied programs can run efficiently in a range of environments using harvested RF power, and how backscatter networks can be built so that they scale well. In this section, I summarize my work and the thesis it supports. I then point towards avenues for future work in the area.

7.1 Thesis and Contributions

This dissertation supports my thesis that *RF-powered computers can support rich tasks in a variety of RF environments, and networks of such devices can scale well to building-wide deployments.*

I make the following contributions in this dissertation:

Identifying key challenges to running programs on RF-powered computers. RF-powered computing is in its infancy, and its constraints and operating model are not yet well understood. I identify four characteristics I believe are fundamental to the technology: 1) RF-powered computers will duty-cycle to extend operating range, 2) The energy consumed

by a task will vary according to both the task itself and the amount of supplemental power harvested during execution, 3) Input power will vary greatly between devices based on distance, and over short time-scales due to frequency selective fading, and 4) The use of capacitors to store energy means that the rate at which a joule of energy is harvested or consumed depends on the state of the capacitor. The last three factors make it difficult to match power consumption to available power to achieve good performance.

Dewdrop, an energy-aware runtime for CRFIDs Taking into account the challenges above, I formulate the task scheduling problem for CRFIDs, a class of RF-powered devices. The goal is to run iterative tasks as often as possible given the available RF power. I then implement Dewdrop, a runtime for CRFIDs that adapts device behavior to match the requirements of the task with the available power. Dewdrop was implemented using the Wireless Identification and Sensing Platform (WISP), and evaluated using a deployment of WISPs in an apartment setting. By waking tags at the right times, I show that Dewdrop can complete tasks where they previously could not complete, and about as often as possible given the energy that the RF environment provides.

A study of interference in backscatter networks Because backscatter requires infrastructure to transmit RF power both when receiving and transmitting, interference patterns in backscatter networks are different than for conventional wireless networks. One key distinction, which has been overlooked by prior work, is that interference at clients is of two types: continuous wave interference and modulated interference. I show how these two types of interference limit network coverage and capacity for Gen 2 RFID systems. Then, via simulation, I show how the demodulator design of backscatter transceivers trades sensitivity to continuous interference against sensitivity to modulated interference, and how this trade-off impacts network-scale performance.

A PHY/MAC protocol that enables scalable backscatter networks Based on the findings of the interference study, I propose a PHY/MAC protocol for backscatter networks that mitigates interference. My approach has three main components 1) A modified demodulator at the client that reduces *continuous wave* interference at clients, 2) a CSMA/CA approach that coordinates infrastructure behavior to avoid *modulator* interference at clients, 3) CDMA uplink modulation overcomes interference between clients experienced at infrastructure nodes. I evaluate my design via simulation, and show that it improves both coverage and capacity compared to existing solutions such as "Gen 2" RFID and Listen-Before-Talk.

7.2 Future Work

In this section, I suggest three avenues for future research based on this dissertation.

7.2.1 Hardware

This dissertation shows a few ways in which hardware mechanisms could improve the performance of RF-powered devices. When implementing Dewdrop, there were two mechanisms built in software that could be provided more effectively by hardware. First, Dewdrop approximates a programmable voltage supervisor by polling the voltage of the capacitor and waking up when a target threshold is reached. A hardware-based programmable voltage supervisor could be more energy efficient, as the device would not need to wake up to check the voltage. However, existing designs support only a few fixed voltages, so the energy savings may come at the cost of limited wake-up threshold granularity [105].

Second, Dewdrop had to sample the voltage across the capacitor throughout program execution, so that it could abort execution before the device ran out of stored energy; this was to maintain state across power failures. This has the drawback that programmers need to insert these energy checks in their programs, and that these checks consume energy. A better approach would be to use a hardware voltage supervisor to trigger an interrupt when stored energy drops below some threshold, and the interrupt handler would put the device to sleep. Better yet, platforms could use microcontrollers that utilize FRAM so that

aborting programs to maintain state is unnecessary; state is maintained automatically as FRAM is non-volatile.

The next hardware improvement would be a means by which to dynamically change the capacitance of the energy store. There are two ways to store more energy in a capacitor: charge the capacitor to a higher voltage, or increase the capacitance of the capacitor. As described in Section 3.1.4, RF-powered platforms operate more efficiently when there is a lower voltage across the capacitor. Unfortunately, increasing the capacitance usually means physically replacing the capacitor. However, techniques do exists where the capacitance scales with the amount of stored energy [92]. Instead of a single capacitor, an array of capacitors is used and they are progressively connected in parallel to increase the capacitance. Combining this mechanism with Dewdrop's mechanism of increasing the wake-up threshold would enable devices to efficiently run a much wider range of tasks.

The last hardware improvement would be to have the communication protocol implemented in a low-power, application-specific IC. Communication protocols are generally have run-to-completion semantics and are non-deterministic, as their behavior depends on the number of clients that are present and their traffic patterns. Unfortunately, this type of task is difficult to support using harvested RF power because it is hard to predict how much energy must be stored before execution begins. A better approach is to have the communications task operate in a "power limited" regime, in that it can operate continuously whenever the IC can be powered via harvested RF. Unfortunately, implementing the protocol in an IC has the drawback that the protocol cannot be easily modified, and protocols appropriate for RF-powered computers is an area of active research.

7.2.2 Physical Layer

A limitation of the network design proposed in this dissertation is that, even if only one controller is present, clients confine their transmissions to a 500 kHz channel. In the US, there is a 25 MHz band of spectrum in the 900 MHz ISM band. Though controller transmissions

are regulated, and always must remain within the 500 kHz channel, backscatter transmitters are not currently regulated. Hence, in networks with only a few controllers, clients could increase their modulation rates significantly. The limiting factor is that controller transmission (including continuous wave) and client transmissions cannot be in the same frequency band, so controllers will need to coordinate they transmissions. For example, if clients backscatter at 1 MHz, their bit-rate will increase by 4x, but controller transmission must remain 2 MHz apart lest they cause controller-controller interference.

In the latter half of this dissertation, I aimed to demonstrate techniques whereby interference could be mitigated in backscatter networks. The design could benefit from a more careful treatment of how the CSMA threshold of controllers can be set, and how the controllers determine whether a client should code its transmissions. Additionally, the feasibility of using a decorrelating detector to eliminate the near-far problem should be considered, as this would nearly further reduce uplink interference.

7.2.3 Higher Layer Protocols

Further work is needed to develop a MAC protocol for RF-powered computers that takes into account both the difficult interference environment of backscatter networks and the rich task model of emerging RF-powered devices. In particular, managing contention between new clients and clients with little data to transmit, and polling for known clients will be challenging as traffic patterns will depend on the tasks the clients are running and how much power they are receiving. Moreover, protocols should take propagation characteristics into account to help predict when is a good time to communicate with a given client, e.g., only when transmitting on a channel that is received well by the target client.

An avenue for research that goes well beyond this dissertation is looking at how RF-powered devices can be connected to the larger Internet. Prior work, culminating in the 6LoWPAN RFC, has considered the protocol challenges, such as mismatched packet sizes and the need to reduce packet overhead, to integrating low power devices over IPv6 net-

works [95]. A similar effort will be required to understand how RF-powered devices can be connected over the "narrow-waist" which is IP networking. Along with 6LowPAN, techniques borrowed from delay-tolerant (or alternately, disruption-tolerant) networks will likely apply, as RF-powered nodes will be intermittently powered.

7.3 Summary

RF-powered computers show great potential for deeply embedded and long-lived sensing and computation. The technology is in its infancy, and thus far RF-powered computers have generally been limited to running fixed tasks when in the presence of a single controller. I set out to advance the technology so that it can run a variety of tasks in a range of scenarios, and in building-scale deployments. By identifying the key challenges to running programs on RF-powered computers, and formulating the task scheduling problem for one class of RF-powered devices (CRFIDs), I show how RF-powered computers can support rich tasks and programs in a range of operating scenarios. I then show how a minimal client demodulator redesign coupled with coordinated controller behavior and a CDMA uplink can enable building-wide networks that scale well. I hope my work will inspire others to rethink the applications that RF-powered computers can be support.

BIBLIOGRAPHY

[1] I. Akyildiz and M.C. Vuran. Wireless sensor networks. 2010.

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

[3] D.M. Anderson, J.A. Landt, and P.H. Salazar. Electronic weighing, identification and subdermal body temperature sensing of range livestock. *Los Alamos Scientific Laboratory Technical Report*, January 1980.

[4] A.J. Annema, B. Nauta, R. van Langevelde, and H. Tuinhout. Analog circuits in ultra-deep-submicron CMOS. *Solid-State Circuits, IEEE Journal of*, 40(1):132–143, 2005.

[5] L.A. Barroso and U. Holzle. The case for energy-proportional computing. *IEEE Computer*, 40(12):33–37, 2007.

[6] S. Birari and S. Iyer. PULSE: a MAC protocol for RFID networks. In *Embedded and Ubiquitous Computing–EUC 2005 Workshops*, pages 1036–1046. Springer, 2005.

[7] A Bletsas, S Siachalou, and J Sahalos. Anti-collision backscatter sensor networks. *IEEE Transactions on Wireless Communications*, 8(10):5018–5029.

[8] Micah Z. Brodsky and Robert T. Morris. In defense of wireless carrier sense. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM '09, pages 147–158, New York, NY, USA, 2009. ACM.

[9] D. Brunelli, L. Benini, C. Moser, and L. Thiele. An efficient solar energy harvester for wireless sensor nodes. In *Proceedings of the conference on Design, automation and test in Europe*, pages 104–109. ACM, 2008.

[10] M.V Bueno-Delgado, J Vales-Alonso, C Angerer, and M Rupp. A comparative study of RFID schedulers in dense reader environments. In *IEEE Conference on Industrial Technology (ICIT)*, 2010.

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

[12] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall. Recognizing daily activities with RFID-based sensors. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 51–60. ACM, 2009.

[13] M. Buettner and D. Wetherall. An empirical study of UHF RFID performance. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 223–234. ACM, 2008.

[14] M. Buettner and D. Wetherall. A Gen 2 RFID monitor based on the USRP. *ACM SIGCOMM Computer Communication Review*, 40(3):41–47, 2010.

[15] M. Buettner and D. Wetherall. A software radio-based UHF RFID reader for PHY/MAC experimentation. In *RFID (RFID), 2011 IEEE International Conference on*, pages 134–141. IEEE, 2011.

[16] Michael Buettner, Ben Greenstein, and David Wetherall. Dewdrop: an energy-aware runtime for computational RFID. In *Conference on Networked Systems Design and Implementation (NSDI)*, 2011.

[17] Benton H. Calhoun, Denis C. Daly, Naveen Verma, Daniel F. Finchelstein, David D. Wentzloff, Alice Wang, Seong-Hwan Cho, and Anantha P. Chandrakasan. Design Considerations for Ultra-Low Energy Wireless Microsensor Nodes. *IEEE Trans. Comput.*, 54(6):727–740, June 2005.

[18] B. Carbunar et al. Redundant reader elimination in RFID systems. In *IEEE Sensor and Ad Hoc Communications and Networks*, 2005.

[19] A.P. Chandrakasan, S. Sheng, and R.W. Brodersen. Low-power CMOS digital design. *Solid-State Circuits, IEEE Journal of*, 27(4):473 –484, apr 1992.

[20] Rohit Chaudhri, Jonathan Lester, and Gaetano Borriello. An RFID based system for monitoring free weight exercises. In *Conference on Embedded Network Sensor Systems (SenSys)*, 2008.

[21] Gregory Chen et al. A cubic-millimeter energy-autonomous wireless intraocular pressure monitor. In *International Solid- State Circuits Conference - (ISSCC)*, 2011.

[22] N.K. Chen, J.L. Chen, and C.C. Lee. Array-based reader anti-collision scheme for highly efficient RFID network applications. *Wireless Communications and Mobile Computing*, 9(7):976–987, 2009.

[23] Namjun Cho, Joonsung Bae, and Hoi-Jun Yoo. A 10.8 mW Body Channel Communication/MICS Dual-Band Transceiver for a Unified Body Sensor Network Controller. *IEEE Journal of Solid-State Circuits*, December 2009.

[24] Namjun Cho, Long Yan, Joonsung Bae, and Hoi-Jun Yoo. A 60 kb/s - 10 Mb/s Adaptive Frequency Hopping Transceiver for Interference-Resilient Body Channel Communication. *IEEE Journal of Solid-State Circuits*, March 2009.

[25] S.S. Clark, J. Gummeson, K. Fu, and D. Ganesan. Towards autonomously-powered CRFIDs. In *ACM Workshop on Power Aware Computing and Systems*, 2009.

[26] B.W. Cook et al. Low-Power 2.4-GHz Transceiver With Passive RX Front-End and 400-mV Supply. *IEEE Journal of Solid-State Circuits*, Dec. 2006.

[27] Alexei Czeskis, Karl Koscher, Joshua R Smith, and Tadayoshi Kohno. RFIDs and secret handshakes: defending against ghost-and-leech attacks and unauthorized reads with context-aware communications. In *Conference on Computer and Communications Security (CCS)*, 2008.

[28] E.H. Dinan and B. Jabbari. Spreading codes for direct sequence CDMA and wideband CDMA cellular networks. *Communications Magazine, IEEE*, September 1998.

[29] P Dutta and D Culler. Epic: An open mote platform for application-driven design. In *Conference on Information Processing in Sensor Networks (IPSN)*, 2008.

[30] D.W Engels and S.E Sarma. The reader collision problem. In *Conference on Systems, Man and Cybernetics*, 2002.

[31] EPCglobal. Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz, 2008.

[32] European Telecommunications Standards Institute. EN ETSI 302 208-2 v1. 1.1.

[33] Richard P. Feynman. *The Pleasure of Finding Things Out: The Best Short Works of Richard P. Feynman*. Penguin Books, 2001.

[34] GR Fox, F. Chu, and T. Davenport. Current and future ferroelectric nonvolatile memory technology. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures*, 19(5):1967–1971, 2001.

[35] H.T. Friis. A note on a simple transmission formula. *proc. IRE*, 34(5):254–256, 1946.

[36] Filippo Gandino, Renato Ferrero, Bartolomeo Montrucchio, and Maurizio Rebaudengo. Probabilistic DCS: An RFID reader-to-reader anti-collision protocol. *Journal of Network and Computer Applications*, 34(3):821–832, May 2011.

[37] G.M Gaukler. Item-level RFID in a retail supply chain with stock-out-based substitution. *IEEE Transactions on Industrial Informatics*, 2011.

[38] Dirk Grunwald, Charles B. Morrey, III, Philip Levis, Michael Neufeld, and Keith I. Farkas. Policies for dynamic clock scheduling. In *Proceedings of the 4th conference on Symposium on Operating System Design & Implementation - Volume 4*, OSDI'00, pages 6–6, Berkeley, CA, USA, 2000. USENIX Association.

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

[40] Jeremy Gummeson, Shane S Clark, Kevin Fu, and Deepak Ganesan. On the limits of effective hybrid micro-energy harvesting on mobile CRFID sensors. In *Conference on Mobile systems, Applications, and Services (MobiSys)*, 2010.

[41] Daniel Halperin et al. Pacemakers and implantable cardiac defibrillators: software radio attacks and zero-power defenses. In *Symposium on Security and Privacy*, 2008.

[42] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. *SIGPLAN Not.*, 35(11):93–104, November 2000.

[43] Junius Ho, D.W Engels, and S.E Sarma. HiQ: a hierarchical Q-learning algorithm to solve the reader collision problem. In *Symposium on Applications and the Internet Workshops (SAINTW'06)*, 2006.

[44] Steve Hodges et al. Assessing and Optimizing the Range of UHF RFID to Enable Real-World Pervasive Computing Applications. In *Pervasive Computing*. Springer-Verlag, 2007.

[45] D.E. Holcomb, W.P. Burleson, and K. Fu. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In *Proceedings of the Conference on RFID Security*, volume 7, 2007.

[46] J Holleman, D Yeager, and R Prasad. Neuralwisp: An energy-harvesting wireless neural interface with 1-m range. *Biomedical Circuits and Systems Conference (BioCAS)*, 2008.

[47] Enamul Hoque, Robert F Dickerson, and John A Stankovic. Monitoring body positions and movements during sleep using WISPs. *Wireless Health*, 2010.

[48] Jason Hsu, Sadaf Zahedi, Aman Kansal, Mani Srivastava, and Vijay Raghunathan. Adaptive duty cycling for energy harvesting systems. In *Symposium on Low Power Electronics and Design (ISLPED)*, 2006.

[49] Impinj. Indy R2000 Reader Chip Electrical, Mechanical, and Thermal Specification, 2012.

[50] Yi Jia, M Heiss, Qiuyun Fu, and N.A Gay. A prototype RFID humidity sensor for built environment monitoring. *Workshop on Geoscience and Remote Sensing (GRS)*, 2008.

[51] Xiaofan Jiang, J Polastre, and D Culler. Perpetual environmentally powered sensor networks. *Symposium on Information Processing in Sensor Networks (IPSN)*, 2005.

[52] Sujit Jos, Preetam Kumar, and Saswat Chakrabarti. Performance Comparison of Orthogonal Gold and Walsh Hadamard Codes for Quasi-Synchronous CDMA Communication. In *Proceedings of the 10th International Conference on Distributed Computing and Networking*, ICDCN '09, pages 395–399, Berlin, Heidelberg, 2009. Springer-Verlag.

[53] A. Kansal, J. Hsu, M. Srivastava, and V. Raghunathan. Harvesting aware power management for sensor networks. In *Proceedings of the 43rd annual Design Automation Conference*, pages 651–656. ACM, 2006.

[54] Aman Kansal et al. Power management in energy harvesting sensor networks. In *ACM Transactions on Embedded Computing Systems*, 2006.

[55] S. Kerzenmacher, J. Ducre, R. Zengerle, and F. von Stetten. Energy harvesting by implantable abiotically catalyzed glucose fuel cells. *Journal of Power Sources*, 182(1):1 – 17, 2008.

[56] Do-Yun Kim, Byung-Jun Jang, Hyun-Goo Yoon, Jun-Seok Park, and Jong-Gwan Yook. Effects of reader interference on the RFID interrogation range. In *European Microwave Conference*, 2007.

[57] D.K Klair, Kwan-Wu Chin, and R Raad. A survey and tutorial of RFID anti-collision protocols. *Communications Surveys & Tutorials, IEEE*, 2010.

[58] A R Koelle, S W Depp, and R W Freyman. Short-range radio-telemetry for electronic identification, using modulated RF backscatter. *Proceedings of the IEEE*, 1975.

[59] J.G. Koomey et al. Implications of Historical Trends in the Electrical Efficiency of Computing. *Annals of the History of Computing, IEEE*, March 2011.

[60] Karl Koscher, Ari Juels, Vjekoslav Brajkovic, and Tadayoshi Kohno. EPC RFID tag security weaknesses and defenses: passport cards, enhanced drivers licenses, and beyond. In *Conference on Computer and communications security (CCS)*, 2009.

[61] J Landt. The history of RFID. *IEEE Potentials*, 2005.

[62] S. Lanzisera and K.S.J. Pister. Theoretical and practical limits to sensitivity in ieee 802.15. 4 receivers. In *Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on*, pages 1344–1347. IEEE, 2007.

[63] M.K Law, A Bermak, and H.C Luong. A sub-uW embedded CMOS temperature sensor for RFID food monitoring application. *Journal of Solid-State Circuits*, 2010.

[64] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *WSNA*, 2002.

[65] Gustaw Mazurek. Design of RFID system with DS-CDMA transmission. In *Conference on Automation Science and Engineering*, 2008.

[66] S. Meninger, J.O. Mur-Miranda, R. Amirtharajah, A. Chandrakasan, and J.H. Lang. Vibration-to-electric energy conversion. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 9(1):64 –76, feb. 2001.

[67] G.E. Moore. Cramming More Components Onto Integrated Circuits. *Proceedings of the IEEE*, 86(1):82 –85, jan. 1998.

[68] Clemens Moser, Davide Brunelli, Lothar Thiele, and Luca Benini. Real-time scheduling for energy harvesting sensor nodes. *Real-Time Systems.*, 2007.

[69] Carlo Mutti and Christian Floerkemeier. CDMA-based RFID Systems in dense scenarios: concepts and challenges. In *Conference on RFID*, 2008.

[70] H Nakamoto et al. A passive UHF RFID tag LSI with 36.6% efficiency CMOS-only rectifier and current-mode demodulator in 0.35/spl mu/m FeRAM technology. In *Solid-State Circuits Conference (ISSCC)*, 2006.

[71] B. Nauta and A.-J. Annema. Analog/RF circuit design techniques for nanometerscale IC technologies. In *IEEE Solid-State Circuits Conference*, September 2005.

[72] P V Nikitin and K V S Rao. Performance limitations of passive UHF RFID systems. In *Symposium on Antennas and Propagation*, 2006.

[73] R. Martinez P. V. Nikitin, S. Ramamurthy and K. V. S. Rao. Passive Tag-to-Tag Communication. *IEEE Conference on RFID*, 2012.

[74] G. Papotto, F. Carrara, and G. Palmisano. A 90-nm CMOS Threshold-Compensated RF Energy Harvester. *Solid-State Circuits, IEEE Journal of*, 46(9):1985 –1997, sept. 2011.

[75] Joseph A. Paradiso and Thad Starner. Energy Scavenging for Mobile and Wireless Electronics. *IEEE Pervasive Computing*, 2005.

[76] Deepak Ganesan Pengyu Zhang, Jeremy Gummeson. BLINK: A High Throughput Link Layer for Backscatter Communication. In *Proceedings of the 10th International Conference on Mobile Systems, Applications and Services (MobiSys 2012)*, 2012.

[77] Padmanabhan Pillai and Kang G. Shin. Real-time dynamic voltage scaling for low-power embedded operating systems. In *Proceedings of the eighteenth ACM symposium on Operating systems principles*, SOSP '01, pages 89–102, New York, NY, USA, 2001. ACM.

[78] Vijay Pillai et al. An ultra-low-power long range battery/passive RFID tag for UHF and microwave bands with a current consumption of 700 nA at 1.5 V. *Transactions on Circuits and Systems*, 2007.

[79] J Polastre, R Szewczyk, and D Culler. Telos: enabling ultra-low power wireless research. *Symposium on Information Processing in Sensor Networks (IPSN)*, 2005.

[80] Richa Prasad, Michael Buettner, Ben Greenstein, and David Wetherall. WISP monitoring and debugging. In *Wirelessly Powered Sensor Networks and Computational RFID*. Springer, 2011.

[81] John Proakis. *Digital Communications*. McGraw-Hill, fourth edition, 2001.

[82] Yan Qiao, Shigang Chen, Tao Li, and Shiping Chen. Energy-efficient polling protocols in RFID systems. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '11, pages 25:1–25:9, New York, NY, USA, 2011. ACM.

[83] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava. Design considerations for solar energy harvesting wireless embedded systems. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 64. IEEE Press, 2005.

[84] Benjamin Ransford et al. Getting Things Done on Computational RFIDs with Energy-Aware Checkpointing and Voltage-Aware Scheduling. In *HotPower*, 2008.

[85] Benjamin Ransford, Jacob Sorber, and Kevin Fu. Mementos: system support for long-running computation on RFID-scale devices. In *Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2011.

[86] Lawrence G. Roberts. ALOHA packet system with and without slots and capture. *SIGCOMM Comput. Commun. Rev.*, 1975.

[87] Mastooreh Salajegheh et al. CCCP: Secure remote storage for computational RFIDs. In *USENIX Security*, 2009.

[88] Mastooreh Salajegheh, Yue Wang, Kevin Fu, Anxiao (Andrew) Jiang, and Erik Learned-Miller. Exploiting half-wits: Smarter storage for low-power devices. In *Proceedings of the 9th USENIX Conference on File and Storage Technologies (FAST 2011)*, 2011.

[89] Alanson Sample and Joshua R Smith. Experimental results with two wireless power transfer systems. *IEEE Radio and Wireless Symposium (RWS)*, 2009.

[90] A.P. Sample, J. Braun, A. Parks, and J.R. Smith. Photovoltaic enhanced UHF RFID tag antennas for dual purpose energy harvesting. *IEEE RFID*, 2011.

[91] A.P. Sample, D.J. Yeager, P.S. Powledge, A.V. Mamishev, and J.R. Smith. Design of an RFID-Based Battery-Free Programmable Sensing Platform. *Instrumentation and Measurement, IEEE Transactions on*, 57(11):2608–2615, Nov. 2008.

[92] W. Sanchez, C. Sodini, and J.L. Dawson. An energy management IC for bio-implants using ultracapacitors for energy storage. In *VLSI Circuits (VLSIC), 2010 IEEE Symposium on*, pages 63–64. IEEE, 2010.

[93] F Schoute. Dynamic Frame Length ALOHA. *Communications, IEEE Transactions on*, 1983.

[94] Mingoo Seok et al. The Phoenix processor: A 30pW platform for sensor applications. In *Symposium on VLSI Circuits*, 2008.

[95] Zach Shelby and Carsten Bormann. 6LoWPAN: The Wireless Embedded Internet. Wiley, 2010.

[96] Dawei Shen et al. Separation of multiple passive RFID signals using Software Defined Radio. *IEEE RFID*, 2009.

[97] N.S. Shenck and J.A. Paradiso. Energy scavenging with shoe-mounted piezoelectrics. *Micro, IEEE*, 21(3):30 –42, may/jun 2001.

[98] M. Simon and D. Divsalar. Some interesting observations for certain line codes with application to RFID. *Communications, IEEE Transactions on*, 54(4):583 – 586, april 2006.

[99] Joshua R. Smith. Range Scaling of Wirelessly Powered Sensor Systems. In *Wirelessly powered sensor networks and computational RFID*, 2012.

[100] James Stevens. Optimized Thermal Design of Small ΔT Thermoelectric Generators. *Proceedings of the 34th Intersociety Energy Conversion Engineering Conference*, 1999.

[101] J. Taneja, J. Jeong, and D. Culler. Design, modeling, and capacity planning for micro-solar power sensor networks. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 407–418. IEEE Computer Society, 2008.

[102] Texas Instruments. Chipcon 2420 Datasheet.

[103] Texas Instruments. MSP430FR573x Mixed Signal Microcontroller.

[104] Texas Instruments. MSP430x2xx Family User's Guide.

[105] Texas Instruments. Quad Supply Voltage Supervisor with Adjustable Delay and Watchdog Timer ,.

[106] S.J. Thomas and M.S. Reynolds. A 96 Mbit/sec, 15.5 pJ/bit 16-QAM modulator for UHF backscatter communication. In *RFID (RFID), 2012 IEEE International Conference on*, pages 185–190. IEEE, 2012.

[107] S.J. Thomas, E. Wheeler, J. Teizer, and M.S. Reynolds. Quadrature Amplitude Modulated Backscatter in Passive and Semipassive UHF RFID Systems. *Microwave Theory and Techniques, IEEE Transactions on*, 60(4):1175 –1182, april 2012.

[108] Stewart Thomas and Matthew S Reynolds. QAM backscatter for passive UHF RFID tags. In *Conference on RFID*, 2010.

[109] Giovanni Vannucci, Aggelos Bletsas, and Darren Leigh. A Software-Defined Radio System for Backscatter Sensor Networks. *IEEE Transactions on Wireless Communications*, 7(6):2170–2179.

[110] H Vogt. Efficient object identification with passive RFID tags. *Pervasive Computing*, 2002.

[111] J Waldrop, D.W Engels, and S.E Sarma. Colorwave: a MAC for RFID reader networks. In *Wireless Communications and Networking Conference (WCNC)*, 2003.

[112] Wang J. Wang, D. and Zhao. A novel solution to the reader collision problem in RFID system. *IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WICOM)*, 2006.

[113] B Warneke, M Last, B Liebowitz, and K S J Pister. Smart Dust: communicating with a cubic-millimeter computer. *IEEE Computer*, 2001.

[114] B.A. Warneke and K.S.J. Pister. An ultra-low energy microcontroller for Smart Dust wireless sensor networks. *IEEE International Solid-State Circuits Conference*, 2004.

[115] Evan Welbourne, Leilani Battle, Garret Cole, Kayla Gould, Kyle Rector, Samuel Raymer, Magdalena Balazinska, and Gaetano Borriello. Building the Internet of Things Using RFID: The RFID Ecosystem Experience. *IEEE Internet Computing*, 13(3):48–55, 2009.

[116] Stephen Wilkus. Power Consumption Trends in Wireless Networks. In *2nd International Conference on Energy-Efficient Computing and Networking*, 2011.

[117] A.C.W. Wong et al. A 1 V Wireless Transceiver for an Ultra-Low-Power SoC for Biotelemetry Applications. *IEEE Journal of Solid-State Circuits*, July 2008.

[118] Wanghua Wu, M.A.T. Sanduleanu, Xia Li, and J.R. Long. 17 GHz RF Front-Ends for Low-Power Wireless Sensor Networks. *IEEE Journal of Solid-State Circuits*, September 2008.

[119] Lingli Xia et al. 0.15-nJ/b 3-5 GHz IR-UWB system with spectrum tunable transmitter and merged-correlator noncoherent receiver. *IEEE Transactions on Microwave Theory and Techniques*, April 2011.

[120] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330, 2008.

[121] Changwook Yoon et al. Design of a Low-Noise UWB Transceiver SiP. *IEEE Design Test of Computers*, Jan - Feb. 2008.

[122] T. Ytterdal. Design of energy efficient analog circuits in nanoscale CMOS technologies. In *IEEE International Conference on Solid-State and Integrated Circuit Technology*, November 2010.

[123] T. Ytterdal and C. Wulff. On the energy efficiency of analog circuits in nanoscale CMOS technologies. In *International Conference on Microelectronics*, December 2008.

[124] P. Zhao, Y. Zheng, and M. Glesner. Automatic impedance matching in microwave power harvesters. In *Ph. D. Research in Microelectronics and Electronics (PRIME), 2010 Conference on*, pages 1–4. IEEE, 2010.

[125] Yuanjin Zheng et al. A CMOS carrier-less UWB transceiver for WPAN applications. In *IEEE Solid-State Circuits Conference*, February 2006.