Title
Investigating Low Energy Wireless Networks for the Internet of Things

Author
Ghena, Branden

Publication Date
2020

Peer reviewed|Thesis/dissertation

Investigating Low Energy Wireless Networks for the Internet of Things

by

Branden R. Ghena

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Associate Professor Prabal Dutta, Chair
Professor David Culler
Professor Scott Shenker
Associate Professor Philip Levis

Fall 2020

Investigating Low Energy Wireless Networks for the Internet of Things

Abstract

Investigating Low Energy Wireless Networks for the Internet of Things

by

Branden R. Ghena

Doctor of Philosophy in Electrical Engineering and Computer Science

University of California, Berkeley

Associate Professor Prabal Dutta, Chair

The Internet of Things (IoT) encompasses a broad array of technologies that connect the physical world with large-scale data processing and storage. Today, we can build ultra-low power devices that last for decades on ambient energy and we can deploy highly scalable internet services to process streams of IoT data. The dominant challenge of IoT lies in connecting these two domains. To meet this challenge, many new networks have been developed with low-energy, low-throughput, and predominantly uplink use cases in mind.

In this work, we explore the capabilities and limitations of several recent IoT-focused networks. While these novel networks are not yet deployed at scale, modeling aspects of them enables us to predict the success or failure of particular applications and explore the ramifications of potential protocol improvements.

In the local-area domain, Bluetooth Low Energy (BLE) has arisen as an ubiquitous method of connecting deployed devices directly to users' smartphones. One communication mechanism that BLE provides is the advertisement—a simple, periodic, broadcast message intended for device discovery. With advertisements, we can create a single-hop, star-topology network in full compliance with the BLE specification that allows any number of devices to send data to any number of gateways. By modeling advertisement transmissions, we can accurately predict congestion in advertisement networks, which enables an a priori understanding of performance and an in-situ adaptation mechanism to meet reliability expectations.

In the wide-area domain, multiple low-power, wide-area networks (LPWANs) have arisen to enable city-scale deployments. Their ability to transmit at ranges over a kilometer while drawing only a few hundred milliwatts could enable exciting new applications. To understand LPWAN capabilities, we propose a new metric, bit flux, which describes communication in terms of throughput over a coverage area. By using bit flux to model the performance of networks and the needs of applications, we demonstrate problems with the existing designs of several LPWANs that make them unsuitable for many real-world deployments.

In both domains, we explore the strengths and weaknesses of communication technologies and their potential to serve real-world applications. The potential for these networks to promote a new generation of easy-to-deploy sensors is high. However, through the application of communication models, we demonstrate both potential use cases and also very real concerns that may limit them. By uncovering concerns that future deployments will face, we hope to guide improvements to these protocols that will improve their use and support the growth of the Internet of Things.

To my family and my friends.
I have been blessed by your love.

Acknowledgments

I was surprised to discover in high school that engineering was a field that I would be interested in. I was surprised in college when I ended up deciding that after five years of undergraduate studies, going on to more school would be the best path for me. I was surprised when teaching, not research, ended up being the part of grad school I loved the most. I was surprised every step of the way. No one else was. My family and friends had already seen and long encouraged in me things that took me years to realize for myself. I am so very lucky to have them in my life.

My parents, Robert and Denise Ghena, have always supported me in my endeavors. They were excited no matter if it was a spelling bee, a FIRST robotics competition, or deciding to stay in school even longer. They have helped me move, driven hours through snowstorms to pick me up, and always been there when I needed to talk. I have been so lucky to have them with me, knowing that they will support me however I need. Nate and Chelsea Ghena, my brother and sister-in-law, have been my allies as well. They are always there to share sarcastic jokes with, and have understood my struggles. We have more degrees between us than is reasonable. My extended family have been insanely loving, and spending holidays with them is my favorite thing. That my grandparents could be online for my dissertation talk was a wonderful gift.

In school I was supported by Ben Naveaux and Andy Felder, who kept me social and shared many laughs with me over D&D. Getting to grow up with them has been an honor, as has been watching our group grow with Jessica Felder and the kids. In college, I was lucky enough to stumble into the most amazing hall. The perfect mix of scholarship and strangeness that was exactly what I needed. I miss the Laurium House, icicles and all, and the weekends spent in the living room laughing and complaining as Cameron Kardel or Sam Dietrick played games. We got work done occasionally as well, and many a long night was spent on campus working with Andy Mauragis, Andrew Maine, and Kevin Nelson. As I navigated the academic job market and graduating, Charlotte Rosenfield was there to support and love me through my anxieties and my celebrations.

At Michigan Tech, I have Dr. Brad King and Dr. Roger Kieckhafer to thank for leading me down the path to graduate school. Working on Oculus-ASR made me the engineer I am today. Even back then I worked on the radios, computer hardware, and software systems. At Michigan, Dr. Mark Brehob demonstrated the combination of compassion and capability that I hope to approach in my own teaching. In my research life, Dr. Phil Levis taught me thoughtful design practices and that even as a professor, you are still an engineer as well.

Dr. Prabal Dutta gave me the opportunity to take my knowledge and build upon it as part of Lab11. He reminded me to always focus on why the research we do is important and what its impacts are. He taught me that being able to explain research through writing and speech is the most valuable skill that grad school provides. He also made it clear that there are many paths towards being successful in academia, and when I realized that teaching was my path, he supported me through it without fail.

Finally, I want to thank Lab11 and friends. I felt like the dumbest person in the room for years, but that meant I was able to learn from all of you. Ye-Sheng Kuo spent years teaching me how to fix my soldering mistakes. Sam Debruin brought me in to help on PowerBlade, which led me down the path of this dissertation, and helped me to become a better writer. Brad Campbell and Pat Pannuto were my collaborators on Tock, along with Amit Levy, and taught me to fight for the projects I believed in. They were my inspirations on how to be a grad student. Meghan Clark, Noah Klugman, and Will Huang provided much needed emotional support. We were always allies as the second wave of lab members. Josh Adkins and Neal Jackson were undergraduates when I met them and my colleagues when I left. We spent many productive afternoons together discussing research possibilities. The many master's students, undergrads, and high schoolers who I got to work with along the way inspired me to stay on the teaching path. The rest of Lab11 and friends: Ben Kempke, Thomas Zachariah, Rohit Ramesh, David Adrian, Matt Podolsky, Jean-Luc Watson, Shishir Patil, Andreas Biri, and others were there as parts of my daily life and helped with countless practice talks and paper revisions. My time in grad school was made perfect for having spent it with all of you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There is enormous potential for computation and connectivity to assist and improve our lives in ways we cannot today predict. One aspect of this is the dream of ubiquitous computing—computers interwoven into our homes, workplaces, and cities. Today we are realizing this dream through the Internet of Things (IoT), which promises intelligent devices that will enable new, impactful application domains.

A great example of the Internet of Things is the Nest thermostat [1]. It is not that thermostats are a new technology, and prior thermostats have included simple computers to manage schedules. What makes the Nest thermostat intellectually interesting is the combination of sensing, local compute, communication, and cloud computation. Now the thermostat can not only sense temperature in the home, but also can check local weather reports and predictions. Now smartphones can remotely monitor and control the thermostat from anywhere around the world. Local networks allow the thermostat to incorporate measurements and commands from nearby devices. The fusion of compute, communication, and sensing has transformed a simple device into something more intelligent and, hopefully, more useful.

Maybe the Nest thermostat is also a useful example because it is still a work in progress. The pricing model and consumer friendliness of these devices is still being continuously developed, and the usefulness is very much in question. The Internet of Things has not yet been solved. The processes for making intelligent devices and the mechanisms to link them to each other and to their users are still uncertain. As a researcher, this is what makes the domain so exciting. Ubiquitous computing holds much promise, but still has many hurdles to be cleared along the way. Each challenge is an opportunity for engineers and scientists to shape our future world.

One commonality for the diverse world of IoT devices is the need for communication. It is the connections to each other and to the internet at large that allow them to expand their capabilities. Machine-to-machine communication has different requirements than human-centric networks. Uplink is dominant for sending sensed data. Downlink is used for command, configuration, and infrequent device updates. Throughput requirements are reduced by orders of magnitude. Energy concerns become a first-order consideration for battery-operated devices.

To support the Internet of Things, numerous new networks have been created over the last decade, each emphasizing different tradeoffs and capabilities. For typical indoor settings Bluetooth Low Energy (BLE) [2] and 802.15.4 Zigbee [3] and Thread [4] networks balance capability with low-energy operation. BLE enables direct communication with people through smartphone BLE radios. Thread enables traditional IP-based networking. WiFi has also reached down to embedded devices. The outdoor wide-area space has had even more growth. Unlicensed-band technologies like LoRaWAN [5] and Sigfox [6] enable 1-100 kbps communication over kilometers of range. Cellular, having previously focused on human-based communication for smartphones, is reaching into the machine-to-machine space as well with LTE-M and NB-IoT. The successes and failures in this space are still being determined.

The focus of this body of work is on wireless communication in particular, at both the local and wide-area levels. Novel IoT networks are not yet deployed at scale and not yet well understood. We approach this problem through a combination of modeling and deployment. Modeling aspects of these networks can allow us to predict success or failure for specific applications prior to deployment. We demonstrate the capability of novel IoT networks to service application needs and suggest possible modifications for protocols.

1.1 Bluetooth Low-Energy Local Communication

In the domain of household communication, this work focuses on the use of Bluetooth Low-Energy. Today, almost all smartphones have Bluetooth Low Energy (BLE) radios, as do laptops, desktops, and wearables. BLE beacons (short-range broadcast transmitters) are widely used as a method of enabling these consumer devices to detect the presence of interactable objects and locations [7]. Academic projects are exploring the use of BLE as well through applications such as long-term health tracking [8], environmental monitoring [9], and indoor localization [10].

One communication mechanism that BLE provides is the advertisement—a simple, periodic, broadcast message intended for device discovery. Advertisements reduce or eliminate listening costs for energy-constrained devices, avoid interference via channel diversity, and are simple to implement and use in software. With advertisements, we can create a single-hop, star-topology network in full compliance with the BLE specification that allows any number of devices to send data to any number of gateways.

While they only provide unidirectional communication, advertisements are useful for several applications. They facilitate the creation of beacons that notify the existence of some resource. The short-range nature of BLE and lightweight nature of advertisements lend themselves well towards location-based applications such as tracking and proximity-based communication. They also fit the use case of periodic sensing. Placing sensor measurements in advertisement payloads allows nearby gateways and smartphones to simultaneously receive and interpret the data. Despite these use cases, we find that communication over BLE advertisements has not been rigorously explored in literature. Existing deployments are often ad hoc in nature. In this work, we explore the BLE advertisement primitive to

understand how well it performs under various conditions and how it can best be applied towards emerging use cases.

To predict expected performance before deployment, models are needed that explain the impacts of network configurations, such as transmission frequency and number of deployed devices, on BLE advertisement networks. While advertisements are ALOHA transmissions at heart [11], they are not identical to them. BLE advertisements are periodic, which means that the probability of repeat collisions is greater than the normal ALOHA expectation. We extend the efforts of prior work [12, 13] to analytically describe reception rates for periodic transmissions in terms of BLE advertisement parameters. We borrow from literature [14] to understand the energy costs of these configurations. We also experimentally validate our models, demonstrating that they accurately represent reality through controlled studies.

While these models do not fully describe deployed networks, they are useful for determining expected performance. The simple access control mechanism of BLE leads to significant packet loss as the number of devices in a deployment increases, but we find that through the addition of redundancy, data reception rates can remain high. Deployment-specific factors, such as external interference or distance from nodes to the gateway, may additionally hinder the connectivity of the network, but descriptions of network capacity alone allow a baseline performance expectation to be determined. Furthermore, the models can be used to explore tradeoffs for new applications, such as contact tracing, before deployment begins.

Another application of the models is the identification of when real-world deployments are underperforming. We analyze a dataset collected from a previous deployment of sensors using a BLE advertisement network [15] with 335 power meters [16] installed across nine locations for an average of 68 days in each location. Given the deployment configuration, we anticipate reception of up to 99% of measurements, but instead we find that network performance falls far short. An investigation reveals the problem was not in the network, but rather in shortcomings of the BLE receiver hardware and software used to create the gateway. We explore the gateway issues, and in new deployments we show that with improved gateways we can accurately predict average network performance, demonstrating the efficacy of our models for planning successful sensor network deployments.

However, the communication environment of a deployed BLE device is difficult to predict in advance. It may change over months as additional devices are deployed or over minutes as people move between locations. We demonstrate that armed with the ability to model advertisement collisions, devices are capable of automatically adjusting to the density of nearby transmissions, remaining reliable even under order-of-magnitude changes in traffic.

Building on the success of the original Bluetooth protocol, the Bluetooth Low Energy standard has blossomed to a point of stability, reliability, and ubiquity. This work takes a step towards deeply understanding BLE for sensor network applications, demonstrating that BLE advertisements can be used as a reliable transport for real-world deployments.

1.2 Low-Power Wide-Area Networks

With the growth in urban population, there is growing interest in making cities safer, cleaner, healthier, more sustainable, more responsive, and more efficient—in a word, smarter. Supporting this interest, there are numerous funding opportunities [17–19] and active research projects [20–24], all targeting new technology to enable smarter cities. The hope is that city-scale application can improve quality of life for a city's inhabitants.

A key problem of city-scale sensing is communication. Particularly when deployed over large areas, normal solutions no longer suffice. Manual data collection is too time-intensive. Local WiFi networks likely are available in city-based deployments, but gaining access to each administrative domain is an overwhelming complication. While cellular networks solve wide-area communication needs for personal smartphones, the costs in terms of energy and money are too high for numerous deployed devices. Novel network solutions are necessary targeting the machine-to-machine communication over wide-area deployments.

Over the past few years, a number of low-power wide-area networks (LPWANs) have emerged, focusing on the area of low-throughput, long-range communication that has long been underserved. Their use of simple protocols and unlicensed bands allowed them to take a first-to-market approach, and their ability to transmit at ranges over a kilometer while drawing only a few hundred milliwatts enables exciting new applications. This work explores several new Low-Power Wide-Area Networks (LPWANs) to understand their capabilities and limitations. While they do not yet have a legacy of real-world deployments to draw experience from, by modeling their throughput, range, and energy use we can gain insights into their ability to serve application needs.

From our investigations, we argue that connectivity for the Internet of Things remains an unsolved problem. Unlicensed-band, LPWAN technologies as they exist today can serve only a narrow class of IoT applications. Furthermore, it is unclear if even the improvements provided by recent research will be enough to expand these use cases. We find that two technical challenges remain for LPWAN protocols to be broadly useful: capacity and coexistence.

The first problem, *capacity*, is the available throughput shared by all devices on the network. LoRaWAN, one unlicensed-band LPWAN, provides 60 kbps of total throughput shared by devices over the range of several kilometers that a single base station can cover. Individually, low-bandwidth and long-range are not a problem; together, however, they prohibitively restrict the utility of the technology. To explore this, we define a metric called *bit flux*, which measures the bit rate a protocol can provide over a unit area. Comparing the bit flux requirements of applications and the bit flux that LPWANs provide, we find that unlicensed LPWANs are only suitable for low-rate, sparse sensing applications.

The second problem faced by unlicensed LPWANs is *coexistence*. Even the limited capacity that LPWANs provide assumes that there is only a single network operating in a given area. The use of the unlicensed bands means this is unlikely to be true as the number of IoT applications and stakeholders using these applications grows over time, especially in urban areas. Unless coexistence between networks is addressed or the capacity of networks

operating in the unlicensed band is increased to well above existing and future application needs, contention is likely to lead to poor performance and ultimately a lack of use by future deployments. In a competitive market, this could also result in a winner take-all density arms race where the first widely deployed network de facto controls the band.

Recognizing both the problems and potential for solutions in this space, 3GPP has been developing cellular standards targeting LPWAN applications, with the most notable protocols, NB-IoT and LTE-M, now operational in the US and abroad. While higher in cost, complexity, and power, our evaluation shows that these technologies meet many of the capacity needs that unlicensed LPWANs currently do not, and they avoid problems of coexistence by operating in licensed spectrum. They also provide coverage without requiring gateway deployments, a valuable consideration for many applications.

Rather than wait until deployments make these concerns obvious, we demonstrate that by modeling LPWANs, we can recognize these issues today and use them to inform and motivate future research in wide-area, unlicensed-band communications. With the release of these cellular technologies we are at a critical turning point in this space. We could drive to improve the protocols of unlicensed LPWANs so they are sufficient for application needs, and we could push for coexistence strategies in both protocol and regulation to ensure graceful degradation of applications. Or we could watch as dense and critical applications shift away from unlicensed LPWANs to cellular networks, taking with them the rich opportunity for future innovation and research that has traditionally followed the ubiquitous use of unlicensed bands.

1.3 Thesis Statement

We show that simple yet accurate models of local and wide area communications for wireless Internet of Things networks that incorporate reception rate, energy use, and data throughput enable us to accurately predict expected network performance for a given real world deployment, and that we can use performance predictions to both inform real time adaptations to network conditions and determine the potential impact of protocol modifications.

1.4 Contributions of this Dissertation

This dissertation presents investigations into recent IoT wireless networks using models to explore network capabilities and limitations.

In the local-area domain we focus on Bluetooth Low Energy advertisements, exploring their capability for reliable communication. Reception rates and energy use for BLE advertisements are modeled to enable prediction of network behavior. Deployment results are included to show capabilities and limitations of collision modeling and demonstrate the use of models for real-time adaptation. This includes Chapters 2 to 4.

Chapter 2 provides an overview of the Bluetooth Low Energy protocol including specific details of parameters and configurations. While it focuses on the widely-deployed version 4.2, it also describes some considerations and new capabilities in BLE version 5.0. As we focus on the BLE advertisement primitive, this chapter also explores existing use cases for communication over advertisements. We define four classes of applications: beacons, tracking, proximal communication, and periodic sensing.

Chapter 3 builds models for advertisement collisions and energy use. We first build a model for packet collisions in terms of BLE parameters and then extend that to model data reception rate. We demonstrate the validity of our models with empirical testing on deployments of up to fifty devices. Finally, using the models we explore the effects of possible modifications to the BLE protocol and demonstrate an introspective look into configuration for an emerging application: contact tracing.

Chapter 4 applies these models to real-world deployments. It first explores the success of a real-world deployment of plug-load power meters using BLE advertisements for data transfer. This sensor was created in collaboration with Samuel DeBruin, Ye-Sheng Kuo, and Prabal Dutta and presented at SenSys'15 [16]. Identifying gateway problems with the deployment that limited packet reception, we statically plan and execute a new deployment that meets expected success rates. Finally, we demonstrate the ability to use collision models to automatically adapt to the transmission environment.

In the wide-area domain we focus on the unlicensed LPWANs Sigfox and LoRaWAN as well as the cellular IoT networks LTE-M and NB-IoT. We present a new metric, bit flux, that allows the throughput and range capabilities of a network to be compared to the data rate and area requirements of a deployment. Applying this metric, we investigate problems with unlicensed communication and demonstrate the possible impact of protocol modifications. This includes Chapters 5 to 7. This work was developed through a collaboration with Joshua Adkins, Longfei Shangguan, Kyle Jamieson, Philip Levis, and Prabal Dutta and presented at MobiCom'19 [25].

Chapter 5 surveys popular LPWAN technologies. It describes two categories of networks: unlicensed LPWANs and cellular networks. For each, it discusses details of the protocol including throughput and power considerations.

Chapter 6 develops models for network capabilities and application requirements. First it develops theoretical upper limits for throughput and range. Then it presents our bit flux metric that describes bit rate per unit area for networks and applications. Bit flux is particularly useful in that it accounts for pervasive wide-area deployments that require multiple gateways to service them. We wrap up with a discussion of pervasive applications taken from literature and their bit flux.

Finally, Chapter 7 uses the bit flux model to explore the suitability of LPWANs, both unlicensed and cellular. It demonstrates two problems facing unlicensed LPWANs: capacity and coexistence, and explores possible solutions from literature that could be applied to unlicensed protocols to improve them. A key takeaway is that unlicensed LPWANs are not yet sufficient for the needs of city-scale applications. Sharing of unlicensed bands among the many stakeholders of an urban region will require new research solutions.

Chapter 2

BLE Background

In exploring the capability of communication with BLE advertisements, we first turn our attention to the Bluetooth Low Energy protocol. The protocol defines multiple communication methods, each of which have capabilities and tradeoffs. We then explore and classify use cases of BLE advertisements to inform communication requirements.

2.1 Bluetooth Low Energy Overview

Bluetooth Low Energy (BLE) is defined by the Bluetooth Special Interest Group. While it shares a name and has several similarities to classical Bluetooth networking, it is a distinct protocol. This overview and analysis covers the Core 4.2 version of the specification, which is in common use [2] and finishes with a brief discussion of potential impact of the 5.0 version of the specification once it becomes widely adopted. We discuss the primitives that make up BLE networking and explore some of the prior ideas for how BLE could be adapted to sensor networking applications. Figure 2.1 depicts the major protocol elements of BLE advertising discussed throughout this section.

2.1.1 Advertising

Advertisements are brief, periodic broadcasts sent by a device that often include some information identifying it. An advertisement event is made up of three advertisement packets sent in rapid succession at periodic intervals. Each advertisement packet is followed by a brief window for listening. This limited listening for end devices is the "low energy" part of BLE, allowing devices to primarily sleep with their radios entirely off. Advertising in BLE is nominally the method for device discovery. Reasonably frequent advertisement events enable quick device discovery and interaction.

Advertisement packets include a maximum of 47 bytes: a fixed 16 bytes of preamble, address, CRC, and other headers, and up to 31 bytes of payload. Sent over the 1 Mbps physical layer, advertisement packets have an on-air time of 128–376 µs. While some payload

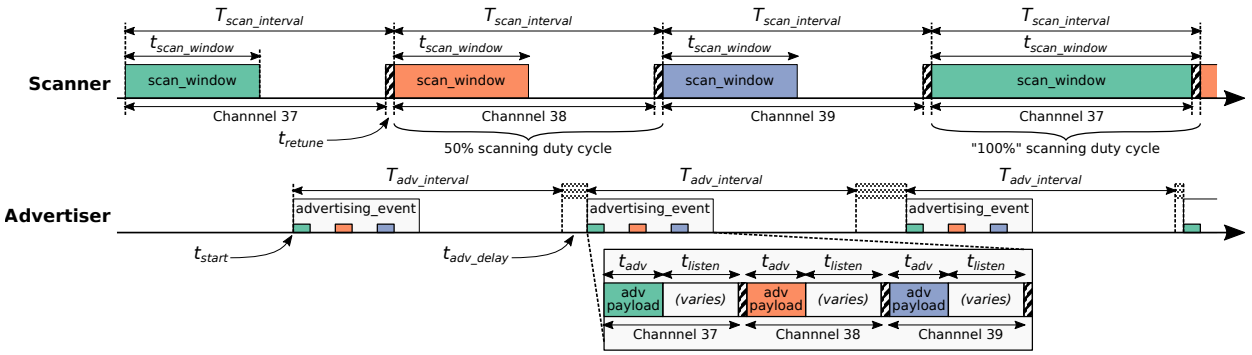

Figure 2.1: BLE advertisement overview. During BLE advertising, the advertiser and scanner are unsynchronized and operate independently. The scanner rotates listening across the three advertising channels, changing channels every $T_{scan\_interval}$. Energy-conscious scanners can separately configure t_{scan_window} which controls the duty-cycle of listening on each channel. At any point t_{start} an advertiser may begin sending advertisement events. Inside an advertisement event, advertisers send an identical payload on all three channels. The delay between advertisement events is controlled by $T_{adv\_interval}$, which dictates the minimum interval as a random 0–10 ms t_{adv_delay} which is added to each interval. A scanner will successfully receive an advertisement when the advertisement payload transmission and scan window align, which occurs once on channel 39 in this example.

bytes have specialized purposes, such as a leading flags field or a company identifier, in practice advertisers can send up to 31 arbitrary bytes.[1]

In an advertising event, identical advertisement packets are sent redundantly on three channels to mitigate interference, with a listening period before changing channels. The BLE specification reserves three channels for these advertisements, placed in the spectrum to avoid WiFi channels 1, 6, and 11.[2] Devices emit advertisement events at a set advertisement interval, $T_{adv\_interval}$, (configurable from 20 ms to 10.24 s) plus a stochastic jitter, t_{adv_delay} (selected uniformly from 0–10 ms). The injected jitter allows transmissions to be randomly distributed in time and prevents collisions from repeating indefinitely.

Broadcasts are sent without any collision avoidance or channel sensing. As a network of unsynchronized blind transmitters, the aggregate throughput of BLE advertisements can thus be modeled as an ALOHA network [26]. The use of three advertisement channels provides no collision avoidance in the common case as they are iterated in the same order by all transmitters and there is no entropy in t_{listen}. In the absence of significant jitter or fading, a collision on one channel will result in a collision on the others.

[1] This relies on the underlying platform permitting scanning applications access to the raw advertisement, today accessible on all major desktop operating system and Android. iOS requires that the first four bytes (flags and company identifier) be well-formed for applications to recover an advertisement.

[2] The advertisement channels are named 37, 38, and 39, but are not actually adjacent in the spectrum. They are at 2402, 2426, and 2480 MHz respectively.

2.1.2 Scanning

BLE scanners are devices listening for advertisements. Smartphones and gateways act as BLE scanners to discover nearby devices. They listen on one advertisement channel at a time, periodically rotating to the next channel.

The BLE specification allows scanners to set a configuration for the receiver duty cycle on each channel, which can be scaled from 0 to 100%. In practice, gateways running on wall power select 100% duty cycle to receive all advertisements. Power-constrained devices, such as smartphones, can select lower duty cycles to conserve energy during long-term background listening. Android, for instance, defines multiple scanning modes that applications can select, which correspond to 10%, 25%, and 100% duty cycle scanning [27].

Reducing the scanner from 100% duty cycle can have a large impact on the success of receiving data. Particular choices of scanning interval and advertising interval can result in particularly long discovery latency [12, 28]. Some platforms, such as iOS, provide no application control over the scan window [29] which can require advertisers to set aggressive advertisement intervals to realize responsive designs, e.g. Apple recommends a 20 ms advertising interval for discovery [30].

In practice, even scanning hardware set to 100% duty cycle does not achieve a perfect reception rate. Perez-Diaz et al. study the performance of the major BLE chipsets, finding that in practice receivers fail to receive packets for brief periods after switching channels, while decoding received packets, and periodically throughout the duration of the scan [31]. The losses from these "blind spots" can be as high as 10% depending on packet size, advertising interval, and scanning interval.[3]

2.1.3 Scan Requests

When a scanner receives an advertisement, it may choose to request additional data from the advertiser by sending a scan request. During the advertiser's listening period following a transmission, if it receives a scan request, it responds by sending an additional advertisement payload of up to 31 bytes, termed a scan response. Scan responses are generally used to provide additional data, such as device name or service descriptions that did not fit in the original advertisement payload.

Hernandez et al. propose that the presence of a scan request could be used as a form of acknowledgment, allowing transmitters to reduce or cease transmitting for some duration after receiving one, reducing overall contention in the network [35]. However, Harris and Kravets explain how the BLE backoff protocol works against this idea [36, 37]. If a scanner makes a request and does not receive a response, it assumes there was a collision with another scanner and adds a random delay before requesting again. This backoff mechanism cannot

[3] For example, while the nRF52832 hardware is capable of tuning frequencies in 40 µs [32], in practice the Nordic softdevice takes approximately 800 µs to switch scanning channels [33]. The popular Noble BLE library sets a default scan interval of 10 ms [34]. Were one to scan with Noble atop a Nordic softdevice, the scanner would have an effective duty cycle of only 92%.

distinguish the case where the request (or response) collided with another advertiser, which becomes increasingly likely as network density grows. This is further confounded as scanners back off exponentially, thus even a modest number of collisions will result in an artificially low number of scan request "acknowledgments", incorrectly underestimating link quality and necessitating yet more advertisements, exacerbating the problem. Rather than send additional data in scan responses, Kravets et al. recommend splitting data across successive advertisement payloads [37].

2.1.4 Connections

Connections are the method for high throughput, bi-directional communication in BLE. After receiving an advertisement, a scanner can send a connection request to the advertiser. Both devices then move into a hopping pattern across the 37 channels reserved for connections. The scanner, which initiated the connection, becomes the master in charge of scheduling connection events—when packets are actually exchanged. A master connected to multiple peripherals schedules them with both time division and channel division multiplexing.

Theoretically the only limit to the number of connected devices is the ability to schedule them. At connection time the master adds an offset, specified in 1.25 ms steps, to the start time of the first communication event. As this offset is longer than the 80 µs minimum link layer interaction, it dominates scheduling ability. The specification allows connection periods from 7.5 ms to 4000 ms, which translates to a maximum number of 6 to 3200 devices that can be connected to at one time without overlap.[4]

Real-world BLE chipsets are significantly more constrained than this theoretical limit. The firmware on many BLE radios limits the number of simultaneous connections to less than ten [38]. The open source MyNewt BLE stack supports the most simultaneous connections of any we survey at 32 [39].

2.1.5 BLE 5.0

While this study focuses on the BLE 4.2 specification, BLE 5.0 has been released and is beginning to see adoption [40]. BLE 5.0 does not fundamentally change any of the traditional advertisement mechanisms. It does, however, add a new data transmission option, termed periodic advertising, that enables devices to send multiple, large payload packets (up to 255 bytes), with the large payload possibly leveraging a faster or more robust physical layer. Periodic advertisements are still initiated via the original BLE 4.2 advertisement mechanism, however, and then jump to the connection channels to exploit the new features. Explorations into periodic advertisements applicability for communication use cases seems well warranted.

Other improvements in the BLE protocol include randomized channel hopping during advertising, which could greatly reduce collisions as discussed in Section 3.3.1, and power

[4] Interestingly, connections also permit devices to miss several events. In principle, masters could trade reliability (and latency) for more connections by scheduling overlapping devices on separate frequencies and rotating through them.

control, which could reduce energy used while communicating. Another exciting improvement is the "SyncInfo" field which allows a device to inform scanners of its advertisement schedule. This could conceptually allow the creation of reliable, energy-efficient scanners. These features could greatly improve future generations of IoT devices.

2.2 Advertisement Use Cases

There are several advantages to using BLE advertisements for communication. Advertisements, and BLE in general, enable communication directly with people through the BLE radios present in all smartphones. Anyone with a phone can easily discover and collect transmissions from BLE devices, something other low-power networks have never accomplished. Second, advertisements are simple. For radios implementing BLE, the interface for sending data over advertisements can be as straightforward as providing a payload and interval. Even when used on top of a raw radio interface, BLE advertisements are fairly straightforward to implement, not requiring tight timing or complex access control. Due in part to this simplicity, BLE advertisements are very low energy. This makes them a good choice for power-constrained devices, for example the emerging intermittent computing class, which cannot reliably participate in scheduled networks as devices may not have energy when needed [41, 42].

Finally, advertisements scale to many devices in a way that connections do not in practice. While a single gateway can theoretically connect to many devices, BLE firmware has much lower limits. Users report that common chipsets used in USB dongles allow less than ten connections, regardless of connection parameters [38]. In contrast, there is no limit to the number of devices from which a scanner can receive advertisements apart from channel utilization. Future chipsets may allow for more simultaneous connections since the limitation is not due to the protocol.

Using advertisements for data transport does not preclude the use of connections. Indeed, connections may be very useful for infrequent and complex operations such as updates to device configuration or firmware. In this study, we focus on the steady-state operation of networks during data collection, assuming such bi-directional interactions are rare.

Several classes of BLE devices use advertisements as their primary method of communication. Here, we discuss the classes: beacons, tracking, proximal communication, and periodic sensing. For each we explain how the class uses BLE advertisements and specific concerns for that class.

2.2.1 Beacons

Beacons use advertisements to broadcast a message to nearby smartphones. A retailer might install beacons promoting the existence of the store to nearby devices. A beacon installed in a conference room could transmit the room's number and installed equipment. A bus stop could use a beacon to broadcast a website with estimated arrival times for various buses.

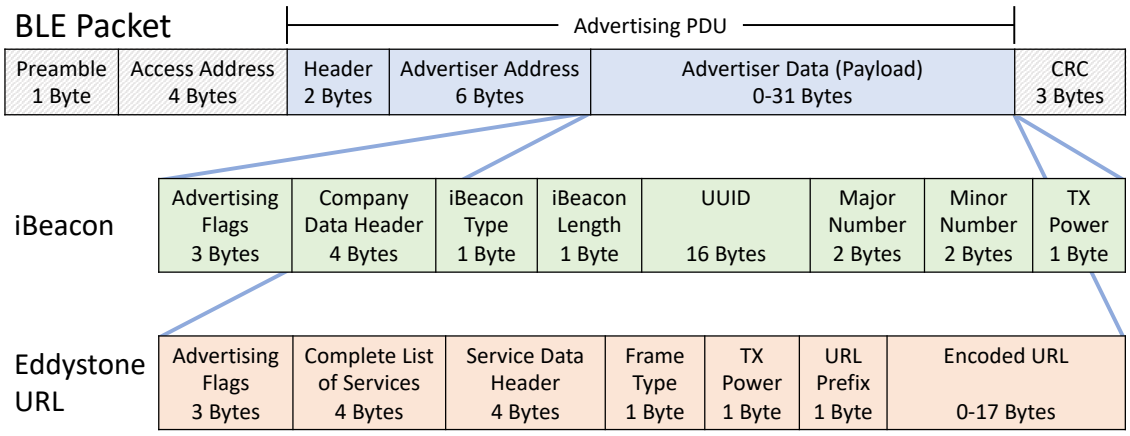

Figure 2.2: Protocol format for common BLE beacon types. The default BLE PDU is shown in gray, with advertising protocol data unit in blue. Both iBeacon and Eddystone use BLE advertisements to broadcast information. The iBeacon format uses a 16-byte UUID along with major and minor numbers to identify devices and locations. The Eddystone URL format allows a location for a web resource to be broadcast. Both beacon formats enable the connection of physical objects and places to virtual identities and resources.

Beacons could be used as literal marketing advertisements as well, installed along sidewalks like virtual billboards.

Beacons are frequently used to provide a virtual presence for something that exists in the physical world. The Google Physical Web project uses beacons with URLs to connect real-world objects and locations to web-based resources [43]. BLE advertisements are used to get the URL to nearby smartphones, and all further communication happens over HTTP (or other protocols further bootstrapped from HTTP).

While BLE advertisements do have specified formats for sending various types of data, several beacon-specific formats have been created by third parties that more concretely specify how to transmit unique IDs and URLs. The iBeacon protocol [44] was released by Apple in 2013. It allows for the broadcast of a 16-byte UUID. Eddystone [45], a protocol by Google in 2016, similarly enables 16-byte UUIDs. Eddystone can also be used to send ephemeral IDs and to directly specify URLs up to 17 bytes in length. Figure 2.2 visualizes the packet formats for Eddystone URLs and iBeacon UUIDs. Both of these specifications include a "transmission power level" byte, which can used along with RSSI to estimate distance from the receiving smartphone to the beacon.

2.2.2 Tracking

BLE advertisements are also used for tracking and localization systems. Tile trackers [46] are a combination of battery, radio, and plastic enclosure that are intended to be connected

to possessions like keys or backpacks. They emit periodic advertisements for the purposes of finding those possessions, if lost nearby, with the use of a smartphone application. Tile also crowdsources detection through its community of users to determine where an object was last seen if lost outside of the home.

Apple "Find My" system uses BLE advertisements as a part of their device discovery system [47]. When a device is reported missing, its ephemeral ID pattern is added to a list on Apple's servers. Whenever a new Apple device is detected with a BLE scan, the ephemeral ID is checked against that list. Matches are tagged with GPS coordinates and anonymously sent to Apple, which can forward that location to the device's owner.

Localization systems want to not just detect a nearby object, but be able to precisely locate it. Fingerprinting localization systems do so by first mapping signal strength from a wireless network throughout a building. After the mapping is complete a device that wishes to know its location can take a signal strength measurement and look up the location. Projects like Redpin [48] and Ariel [49] use this method to achieve high-accuracy room-level localization based on existing WiFi deployments. BlueSentinel [50] utilizes BLE advertisements for fingerprinting by intentionally deploying beacons throughout a building and mapping received signal strength from those advertisements. The system is shown to have an accuracy of 84% at determining the room-level location of a user.

2.2.3 Proximal Communication

Proximal communication methods allow for interaction only between devices in close proximity with each other. Ultrasonic, vibratory, and visible-light communication methods dominate this space. However, the relatively short range of BLE, less than 50 m in most indoor environments, makes it a possible medium for this type of communication. In practice, we find that most BLE devices can only reliably communicate over distances of two to three rooms in typical indoor environments. This means that two devices communicating over BLE are likely to be near to each other.

Apple Continuity [51] leverages BLE advertisements for event notification between Apple devices. For example, when text is copied on a Mac, a BLE advertisement is sent signalling that a copy operation has occurred. A nearby iPhone that, has been configured to share clipboard with that Mac, upon receiving the advertisement checks Apple's servers for the updated clipboard contents. The end result is that a user can seamlessly copy text on one device and paste it on another. Apple Continuity signals several types of events, including WiFi hotspots, image capture, and cellular calls. Primary data transfer still occurs over a normal internet connection, but BLE advertisements serve as notifications that an event has occurred. This service has been the target of several security investigations, reverse engineering the protocol and demonstrating privacy problems inherent to publicly broadcasting device activities [52, 53].

The "Exposure Notification" system developed by Google and Apple in response to COVID-19 uses a combination of proximal communication and tracking techniques to determine when two people have been in "contact" [54]. In the system, all participating

Figure 2.3: BLEES and PowerBlade. At left is BLEES: an environmental sensor capable of sensing temperature, humidity, pressure, light, and acceleration. At right is PowerBlade: a low-profile, plug-load power meter. Both sensors broadcast periodic measurements using BLE advertisements.

smartphones send periodic BLE advertisements with ephemeral IDs. The smartphones also perform a scan of their environment at least once every five minutes in order to collect advertisements from nearby devices. In the case of an infection, the ephemeral IDs used by a smartphone over the last several days can be uploaded to the cloud, allowing other users to determine if they have been close enough to receive packets with that ID and therefore may need to be tested for infection. The system relies on BLE advertisements only being detected when two devices are relatively close to each other, but also includes a transmission power indication to potentially perform more accurate distance measurements.

2.2.4 Periodic Sensing

Finally, BLE advertisements also fit the goals of periodic sensing, a primary focus in this work. This particularly fits the goals of consumer, indoor sensing devices for temperature, light, or energy consumption. Many devices perform periodic sensing actions, and the period of their BLE advertisements could be matched to that of the sensor. Rather than repeating the same data, each message sent can contain a unique sensor reading, although to improve reliability at the expense of latency and throughput, an advertiser could repeat the same data for several packets in a row before updating the data. The frequency of advertising packets is controlled by the advertisement interval, $T_{adv\_interval} \in [20\,\text{ms}, 10.24\,\text{s}]$, plus a random additional delay, t_{adv_delay} drawn uniformly from $[0, 10]\,\text{ms}$ at each period. As a reference point, the maximum theoretical goodput via BLE advertisement for one node is thus $9.92\,\text{kb/s}$. The sensor data can be received by nearby smartphones in real time, or can be collected by deployed BLE gateways, which would receive the advertisements, process the data, and send measurements to the cloud.

One particular issue with transmission of data over BLE advertisements is that it is an inherently unreliable channel given the lack of acknowledgements. Through the use of redundancy, data transmission can become probabilistically reliable, but there is still no guarantee. This makes BLE advertisements particularly suited for low-priority, real-time data where the loss of any particular measurement is not harmful.

Two particular sensors making use of BLE advertisements for communication are BLEES and PowerBlade, shown in Figure 2.3. BLEES is a device from the University of Michigan that collects various environmental sensor data: specifically temperature, humidity, pressure, light, and acceleration [55]. In its default configuration, it takes measurements and sends them over BLE advertisements once per second. PowerBlade is a plug-load power meter meant to be deployed at scale in homes to study energy usage patterns across the long-tail of household loads [16]. It produces new power measurements once per second and transmits that value in three advertisements for redundancy.

2.3 Summary

In this chapter we have reviewed the BLE protocol, particularly focusing on capabilities and limitations of the primitives it provides. Narrowing our focus towards BLE advertisements, we explored four classes of use cases which are useful in guiding our thinking. Chapter 3 takes the next step in the investigation by creating the models of communication that will be used to predict performance.

Chapter 3

Modeling BLE Advertisements

To understand the capability of BLE to provide for application needs, we must understand how the protocol behaves under varied conditions. As collisions are a dominant factor of packet loss for BLE advertisements, we focus on modeling them. Predicting collisions allows for packet and data reliability to be predicted in turn. We demonstrate the accuracy of these models and later show that they are useful for informing real-world deployments in Chapter 4.

When thinking about low-power devices, this ability to predict is particularly useful for providing reliability while not wasting energy unnecessarily. PowerBlade, for example, has a 6 mW average power budget. Any of that spent on transmissions cannot be directed toward sensing or computation. For devices, with limited battery supplies, each additional packet transmitted limits total device lifetime. To that end, we also explore models for advertisement energy use. The specifics of these will inevitably be hardware-specific, but the general themes should apply broadly.

Finally, we investigate what models alone can tell us. We demonstrate lessons learned for the BLE protocol as a whole and for emerging "contact tracing" applications.

3.1 Collisions

For BLE advertisers within range of a receiver, packet collisions are the primary reason for communication failures. This is due to the uncoordinated, broadcast nature of advertisements, leading to a probability of collision that is proportional to the number of devices and duration of transmission. More devices in a single area means more packet collisions.

Modeling packet collisions is a straightforward step we can take to determine communication reliability. Here, we start with probabilistic model of packet collision from literature, which is used to calculate packet reception rate. We then extend the model to determine data reception rate in the presence of redundant transmissions. At each step along the way, we use the models to explore ramifications for BLE advertisement communication. Finally, we perform experiments to empirically validate the packet collision models.

The theoretical models for advertisement networks are informed by several prior works. The ALOHA system first describes the access control method that will later be used by BLE [11]. Liu et al. first study the probability of collisions for BLE advertisements using the Poisson distribution, much like ALOHA [26]. They calculate the delay before device discovery rather than packet reception rates. Similarly motivated by neighbor discovery, Jeon et al. create an iterative model for determining discovery latency [12]. Harris et al. adopt a probabilistic model for packet collisions, similar to our own models but missing the effect of the delay added to each interval [56]. Perez-Diaz et al. [31] include that random delay, coming to the same result we do in Equation (3.1). Our models go beyond this prior work by accounting for heterogeneous node configurations and observing the increased probability of repeat collisions in BLE. Rather than just presenting models for collisions, we also use BLE parameters to describe network reception rates.

3.1.1 Modeling Packet Reception

To begin creating a probabilistic collision model, we must state several simplifying assumptions. First, let us assume that distance and channel do not affect packet reception—that all sensor nodes in the network are within range of the gateway receiving their data and that the received signal strength from each is identical on all advertising channels. This is the worst-case for collisions; in practice the capture effect will mitigate some collisions [57].

Next, consider a wall-powered gateway that is always listening and always able to receive packets (such a gateway is quite reasonable, effectively $t_{scan\_window} = T_{scan\_interval}$ and $t_{scan\_window} >> t_{retune}$). As explained in Section 2.1.1, because a collision on any advertising channel will collide on all advertising channels and we are treating the propagation of each channel as identical, we can therefore ignore the channel of the scanner altogether.

Literature describes the basic model for BLE advertisement collisions [31]. The probability of collision, P_c, defined using BLE advertisement parameters is:

$$P_c = 1 - \left(1 - \frac{2 \times t_{adv}}{T_{adv\_interval} + \mathbb{E}(t_{adv\_delay})}\right)^{N-1} \tag{3.1}$$

The notation defined in Figure 2.1, except for N which denotes the number of devices. Note that, because the delay is chosen in a uniform random fashion, its expected value is 5 ms in practice.

The explanation for this model requires an understanding of collision and transmission windows. We define a collision as any time a transmission occurs on the same channel that overlaps any portion of another transmission. During an advertising event, an advertiser requires t_{adv} seconds to transmit one packet. If another device begins sending any time during this transmission, or up to t_{adv} earlier, this will result in a collision.

The next step towards a collision model is the probability that another transmission occurs during this $2 \times t_{adv}$ window. If all devices were activated arbitrarily, then their transmission times can be assumed to be independent and identically distributed. Even if

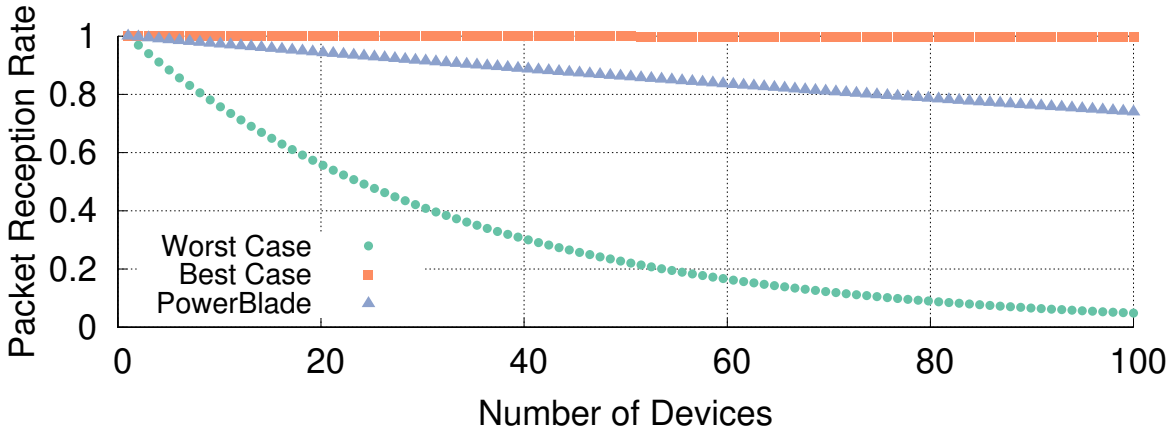

Figure 3.1: Packet reception rates as deployment size increases. Both the worst case (31 byte payload broadcast every 20 ms) and best case (0 byte payload broadcast every 10240 ms) are displayed. The configuration for PowerBlade [16], a research BLE sensor node that transmits a 23 byte payload every 200 ms, is also included as a real-world example. For small deployments (fewer than 10 nodes), any configurations will likely result in acceptable PRR. As deployments scale, however, they must balance desired throughput and reception rate.

devices begin broadcasting simultaneously, the addition of the random delay, t_{adv_delay}, means that over time, node transmissions will become randomly distributed and independent. This means that a transmission is equally likely at any point, and the probability of collision is therefore the ratio of the collision window and the duration between advertisements (which is increased by the expected value of t_{adv_delay}). This results in Equation (3.1).

Additional losses can be accounted for in this model. For example, we can consider losses that come from interference with other technologies, denoted as P_i. We can express the packet reception rate, PRR, more generally then as the probability of neither colliding nor being interfered with:

$$PRR = (1 - P_c)(1 - P_i) \tag{3.2}$$

In general, the BLE advertisement channels are positioned so they fall outside of the bands of the normal WiFi channels (1, 6, and 11). Narendra et al. study BLE interference from a single WiFi access point and find little to no packet loss when using BLE channels that avoid the main WiFi channels [58]. However, empirical studies find that many real-world WiFi deployments use every channel [59] and further interference may still be caused by 802.15.4 traffic or other transmissions in the 2.4 GHz ISM band. As interference is an independent factor, we simplify by assuming that P_i is zero for the remainder of this analysis and focus on refining the estimate of collision probability, P_c. In general, our experience is that this assumption seems acceptable in household environments, but should be checked in areas where 2.4 GHz wireless technologies other than WiFi and Bluetooth are in common use.

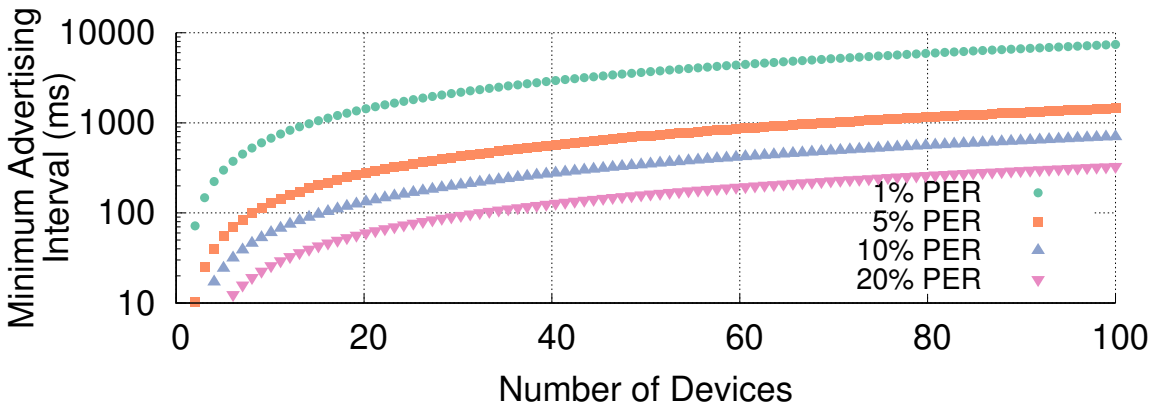

Figure 3.2: Minimum advertising intervals to realize target packet error rates. Given a fixed payload (here 31 bytes), to realize a target packet error rate the minimum advertising interval must grow with the number of devices. Even small deployments require several hundred milliseconds between transmissions to achieve a 1% packet error rate. Accepting 10% error rates allows sub-second intervals even as deployments expand to one hundred devices.

In Figure 3.1 we use this model to explore the effects of number of devices, payload length, and advertising interval on packet reception rate. Configuring devices for the highest throughput—full payloads transmitted at the highest frequency—results in the highest probability of collision and therefore the lowest packet reception rate.

Certain applications may have acceptable packet error rates, PER, to meet their requirements. We can solve Equation (3.1) to determine the minimum advertising interval that satisfies a given packet error rate and number of devices:

$$T_{adv\_interval} = \frac{2 \times t_{adv}}{1 - (1 - PER)^{\frac{1}{N-1}}} - \mathbb{E}(t_{adv\_delay}) \tag{3.3}$$

Figure 3.2 plots the impact on latency as network density scales for various target error rates. As expected for an ALOHA-style network, high throughput for many devices can only be achieved by sacrificing reliability for any given packet.

One assumption made previously was that all devices on the network act identically. But we can extend the model from Equation (3.1) to remove this requirement. Assume a primary device with an advertisement transmission time of t_{adv_0} and a second possibly colliding device with a transmission time of t_{adv_i}. A collision occurs if the second device begins transmitting any time during t_{adv_0} or up to t_{adv_i} before the primary device begins transmitting. If the transmission from the second device is uniformly distributed in time, it is equally likely to occur during any point in its advertising interval. Generalizing this second device to all nodes in the network, we can express the probability of collision with the primary device, P_{c_0}, as:

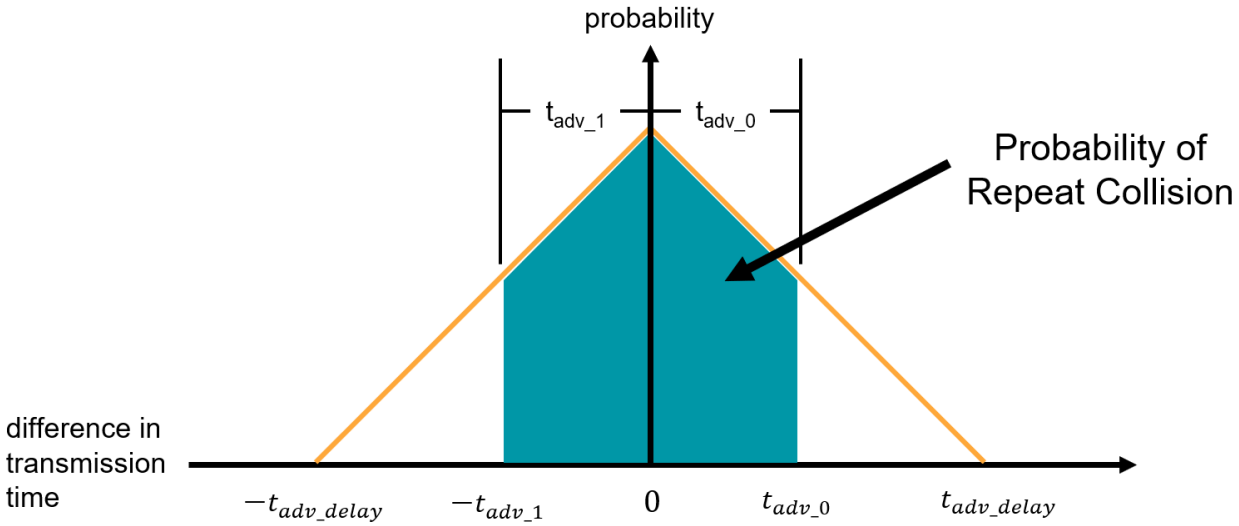

Figure 3.3: Probability of a repeat packet collision. A repeat collision occurs if the difference in delays applied to each previous colliding transmission is less than the size of the collision window. The difference in uniform random variables (the delays) creates a triangular distribution, which can be integrated across the collision window to determine the probability of a repeat collision. The resulting repeat collision probability is significantly higher than the original probability of an uncorrelated collision.

$$P_{c_0} = 1 - \prod_{i=1}^{N-1} \left(1 - \frac{t_{adv_0} + t_{adv_i}}{T_{adv\_int_i} + \mathbb{E}(t_{adv\_delay})}\right) \tag{3.4}$$

In practice, the variation on packet on-air duration (from 128 µs to 376 µs) is less impactful than the variation on advertisement intervals (from 20 ms and up).

3.1.2 Modeling Data Reception

In real-world deployments, we are not interested in the reception of individual packets, but rather the eventual recovery of their payload. If we want to increase the probability that any particular payload is received, we can repeat it. Then, for data to be lost all redundant packets sent must be lost.

A naïve model for this would use P_c as the probability for each failure. However, repeat collisions in BLE are *not* independent. Given that the first packet collided, the probability of a second collision is greater than a random collision. Indeed, absent t_{adv_delay} (or clock drift), once a single packet suffered a collision all future packets would collide too, as long as each device is using the same $T_{adv\_interval}$ (i.e. a network of homogeneous devices). This is a change from the traditional ALOHA analysis, which does not have an assumption of periodicity, and results in an increased probability of a repeat collision.

After a collision, a repeat collision occurs if the sum of the differences in initial transmission times, advertising intervals, and selected random delays for two devices are less than the duration of the advertisement. For the homogeneous deployment case (advertising interval difference and average difference in initial transmission time are both zero), a repeat collision occurs if the difference in random delays has not moved one advertisement outside of the transmission time of the other. The difference of random delays creates a triangular distribution which we can integrate to determine the probability of collision, as demonstrated in Figure 3.3.

The full equation for the probability of a repeat collision is demonstrated in Equation (3.5). To simplify the math, we return to the assumption of a homogeneous deployment of advertisers.

$$P_{rc} = 1 - \left[\left(1 - 2 \int_0^{t_{adv}} \frac{1}{t_{adv\_delay}} - \frac{x}{t_{adv\_delay}^2} \mathrm{d}x \right) \times \left(1 - \frac{2 \times t_{adv}}{T_{adv\_interval} + \mathbb{E}(t_{adv\_delay})} \right)^{N-2} \right] \tag{3.5}$$

Note that this does not account for the case where more than one node collides with the primary transmission, which means it slightly underestimates the number of repeat collisions that occur.

Figure 3.4 visualizes the impact of repeat collisions for the two-node case and finds that the odds of a repeat collision range from 2% to 7%, more than twice as much as the naïve ALOHA assumption. This increased probability of repeat collisions has a particular impact on device discovery, possibly leading to extended periods when all packets from a device collide.

To extend this to model the probability that data is received, note that the probability of a second collision is the same as the probability of a third collision, and so on. The probability of the first collision is the normal probability of collision from Equation (3.4), while the second and onward are the probability of a repeated collision from Equation (3.5). For a series of M total packets with the same data payload, we can express the data reception rate, DRR, as the odds that the data is received at least once:

$$DRR = 1 - (P_c)(P_{rc})^{M-1} \tag{3.6}$$

Figure 3.5 applies this model to demonstrate the impact of redundant advertising on data reception rate. Even at the fastest advertising frequency, redundant transmissions allow devices to overcome poor packet reception rates.

Redundancy is not always beneficial, however. To preserve the same data update rate, devices sending redundant advertisements must advertise more frequently. This leads to increased transmission contention, increasing the probability of a collision, and possibly defeating the goal of more reliable receptions.

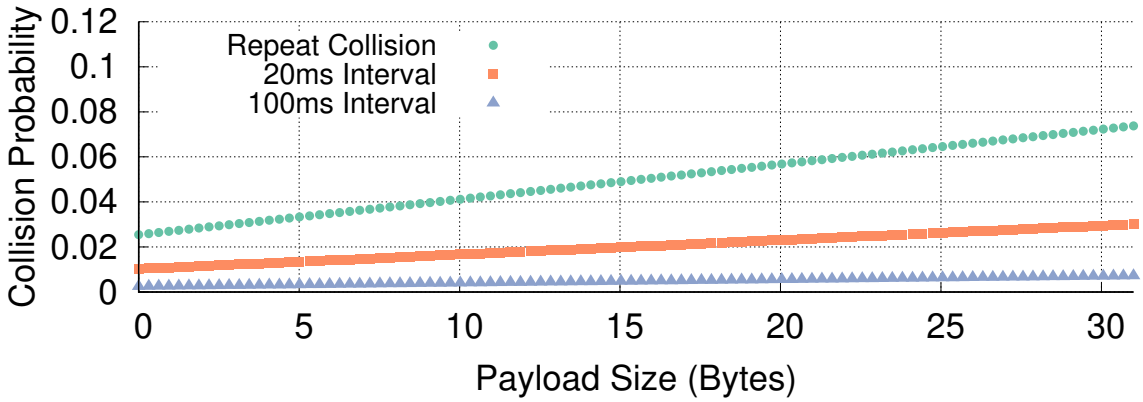

Figure 3.4: Collision probability for two devices. The probability of a collision between two BLE transmitters grows as the payload size of the packet increases. The general probability of collision is plotted for several advertising intervals. The probability of a repeat collision is increased due to the periodicity of BLE. Given a collision on the previous packet, the probability of collision for the current transmission in BLE is twice as high as the normal probability for even the fastest advertisement interval.

Consider an application that wishes to reliably transmit data each second. Is it more reliable for each node to send five copies (once every 200 ms) or to minimize potential contention and send only one copy (once every 1000 ms)? Figure 3.6 shows how the expected data reception rate for these scenarios responds as network density grows. Advertising redundantly is initially more successful than advertising slowly but at 432 devices in a deployment there is a crossover. With more devices, the added contention from redundant packets actually reduces data reliability. Since we are considering deployments where all devices are within range of each other, more than 432 devices is highly unlikely. For practical network scenarios, redundant transmissions are the right choice for reliability.

The worst case for BLE collisions occurs when an event triggers data transmission, a common architecture in sensing systems. Per the BLE specification, no random delay is added before the first packet. This means all transmitting devices triggered from the same event will collide on their first transmission. The probability of collision decreases each interval through the addition of the 10 ms random delay. For such triggered systems, a random delay should be added before beginning advertising to avoid this failure mode.

When calculating data reception rates, it is often acceptable to use the naïve model, with P_c to the power of the number of redundant packets. Even though the odds of a repeat collision are increased, that they only occur after a first collision has already taken place makes then negligible in many real world scenarios. For example, for a deployment of 50 devices transmitting maximum-length advertisements every 200 ms, the largest absolute error between the naïve model and the full model in Equation (3.6) is about 1%. For scenarios where we are exploring very large deployments or modifications to the BLE protocol, as discussed in Section 3.3.1, we will continue to use the full model. For scenarios where efficient

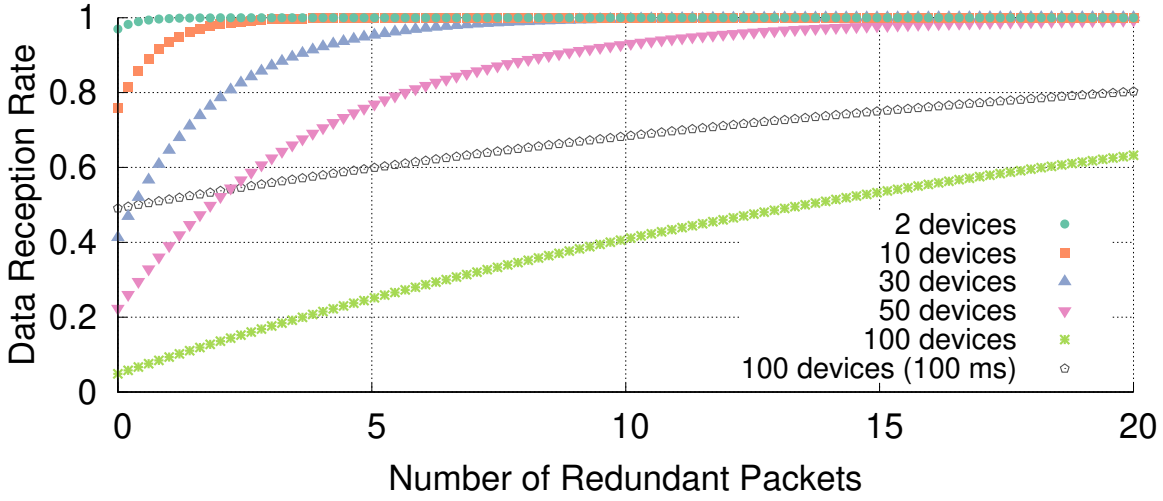

Figure 3.5: Data reception rate for redundant transmissions. The number of redundant packets transmitted is varied for multiple deployment sizes, each transmitting at a 20 ms interval except the 100 device scenario which considers both 20 ms and 100 ms intervals. As the density of a deployment grows, advertisers need to send a greater number of redundant packets to maintain data reception reliability. Latency and throughput also affect reception rate, slowing advertising from 20 to 100 ms improves reception at the expense of responsiveness and throughput. At zero redundant packets, the reception rate is identical to packet reception rate. For higher values, reception of any one packet is enough to receive the data. Sending redundant packets can significantly improve data reception when there is contention in the network.

math is advantageous, such as calculating DRR onboard a microcontroller as discussed in Section 4.3, we will revert to the naïve model.

3.1.3 Advertisement Network Takeaways

We find that advertisement-based sensor networks should be capable of achieving data reception rates over 99% given the right parameter selection. Such network benefit from the ease of implementation afforded by the advertisement primitive, the ubiquity of BLE radios, and the extensibility afforded by connections for infrequent maintenance tasks. To maximize the performance of advertisement-based networks, advertisers should add packet redundancy. Applications demanding very high density (100 or more devices in the same broadcast domain) and sub-second latency constraints may not be well served by advertisement-based networks.

Additionally, the lack of acknowledgments makes any guarantees of performance probabilistic. BLE advertisement networks can best be taken advantage of by applications that may need a high probability of data reception, but do not require the successful reception of

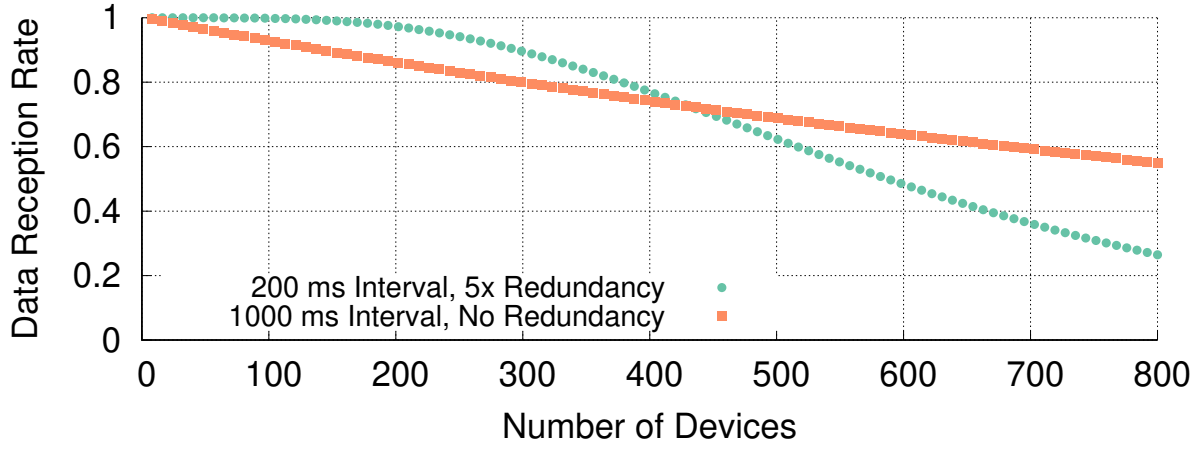

Figure 3.6: Data reception rate favors redundancy. Data reception rates are compared for one network configured to send a packet every 1000 ms versus another configured to send five redundant packets per second at 200 ms intervals across deployment sizes. A crossover point occurs where the additional likelihood of data reception due to redundancy is not enough to overcome the additional losses due to increased contention due to sending more packets, but in practice this point requires more transmitters than the expected maximum deployment size for many applications, making redundant transmission still useful.

any particular packet. For those scenarios, BLE itself may still be acceptable, but connections are likely a better communication mechanism. Devices may, of course, mix modes by using BLE advertisements for common, low-priority data and connections for rare, high-priority data.

3.1.4 Empirical Testing

To validate the analytical model of advertising network performance, we test networks of up to fifty devices and compare their performance with predictions from the model. For advertising, we use a programmable beacon platform built atop the nRF51822 BLE radio [60]. To recover packets, we use a Bluetooth Protocol Analyzer [61] to eliminate any potential variance from the Bluetooth stack. The analyzer listens on all three advertising channels concurrently. All nodes are placed in a single room in a 10×5 grid with approximately 15 cm spacing. The analyzer is placed approximately 1 m away from the grid. The experiment takes place in a residential building, with general interference from other devices such as WiFi or Bluetooth transmitters expected. We dwell in each configuration for several minutes, discarding the beginning and end of each trace.

The experiment varies multiple configuration parameters. We sweep the number of advertising devices from 3 to 50. For each deployment size, we test advertising intervals from 100 ms to 10 s. To detect missing packets, each advertising payload is sent with a

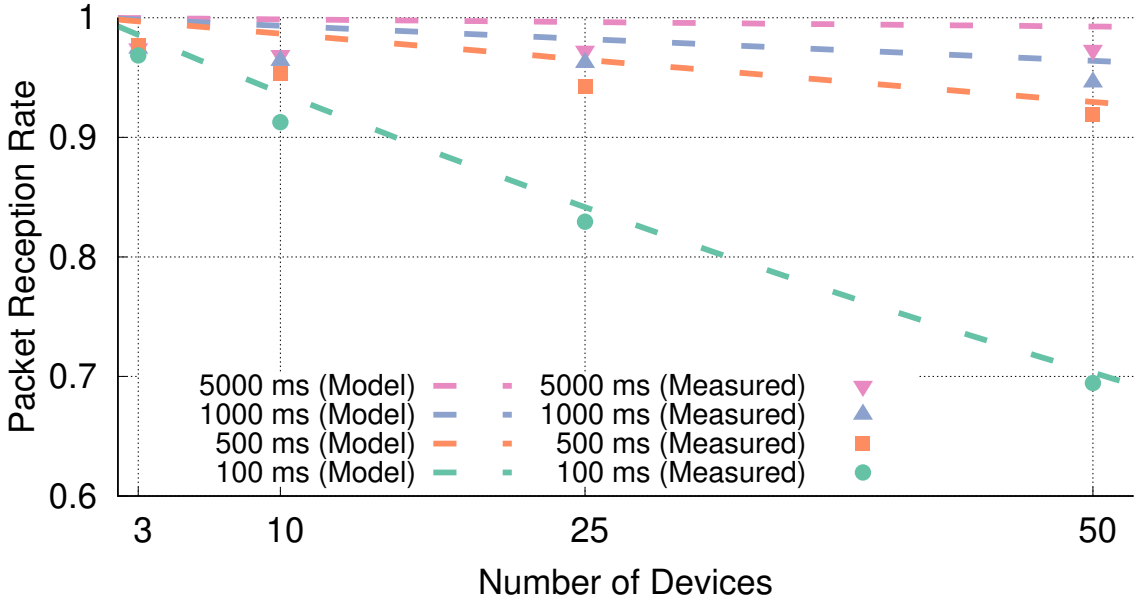

Figure 3.7: Analytical and experimental packet reception rate. Packet reception rate is measured across a range of number of transmitters and a selection of advertising intervals. Note that the y-axis ends at 0.6 PRR. We find that the analytical model tracks well with reality, but that it overestimates the true reception rate, possibly due to interference.

monotonically increasing sequence number. The remainder of the packet payload is padded to a total payload size of 31 bytes. Packets received with invalid CRC values are considered missing.

To reduce jitter induced by receiving scan requests, all transmitters are configured to not listen for scan or connection requests. Otherwise, the presence of nearby smartphones scanning for BLE packets may influence the behavior of the experiment. As a side effect, this limits the minimum advertising interval of the network to 100 ms per version 4.2 of the BLE specification [2].

Figure 3.7 shows the results of measuring packet reception rate for this experiment. Each point is the average reception rate across the three advertising channels for all deployed devices. While 95% confidence intervals are calculated, they are too small to visualize. Although the model for BLE advertising should be conservative (as it ignores the capture effect), we find the model strictly overestimates performance in this experiment. We hypothesize that this is due to interference with other transmitters such as WiFi, Bluetooth Classic, or other BLE devices. This error is minimal, however, accounting for less than 5% deviation from the expected result.

Figure 3.8 adapts the raw packet reception information to estimate data recovery rate with redundant packets. Integer division of raw sequence number by a redundancy rate (number of repeated packets) yields a new stream of possibly-redundant sequence numbers.

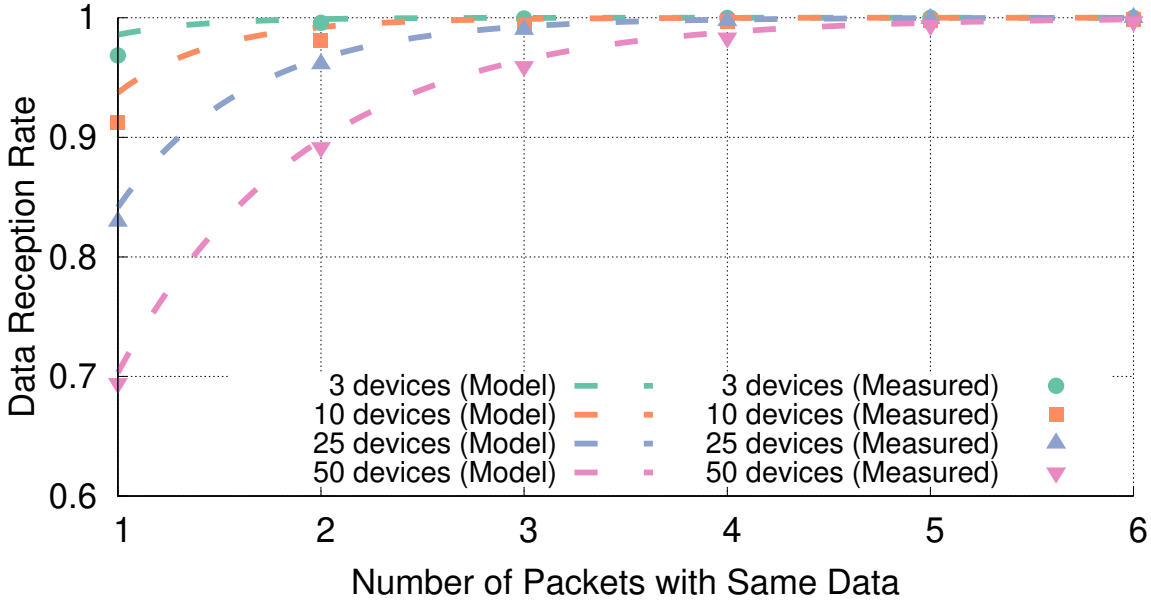

Figure 3.8: Analytical and experimental data reception rate. Packet reception rate is measured across a range of number of redundantly transmitted packets for a selection of deployment sizes. series of deployment sizes. Note that the y-axis ends at 0.6 DRR. To create an environment with many collisions, maximum sized packets are transmitted at 100 ms intervals. The experimental results closely match the analytical model.

If a sequence number is seen at least once in this adapted stream, we count that data item as having been received. For a redundancy rate of one, the results are identical to the 100 ms interval line from Figure 3.7. Adding even modest redundancy quickly results in 100% data reception rates. Again, we find that the experimental data tracks the analytical model.

3.2 Energy

BLE was created to be a "low energy" medium for devices. Particularly, the limited listening requirements are a significant part of this. BLE radios are entirely off for the majority of each second, reducing their power draw. This is important because energy is a first-class concern for many Internet of Things devices, with sensing and communication often dominating the energy budget. For devices on batteries, the less energy used in communication, the longer the device's overall lifetime.

While several works in literature investigate energy use in BLE [12, 62, 63], work by Schrader et al. [14] takes an in-depth look at energy use agnostic to transceiver implementation. They find six distinct power draw phases of BLE advertisements that recur in multiple different transceivers. These correlate to sleep, startup, transmit, tx-to-rx transition, receive, channel transition, and post-processing. The duration and cost of these phases varies

based on transceiver implementation and advertising parameters. Longer packets require more time transmitting, for example. Phases are sometimes completely omitted such as the receive phase when transmitting non-connectably without support for scan responses.

One takeaway about BLE advertisements is modifying advertising interval only affects the sleep phase. This means we can calculate the energy cost of an advertisement relative to packet length, transmit power, and transceiver IC without any knowledge of advertisement interval. The costs of redundancy are then a linear multiplier to this advertisement cost.

Using empirical data from Schrader et al. [14], we can determine the cost of an individual advertisement. We assume a maximum size advertisement (31 bytes of payload) transmitted at 0 dBm. For the nRF51822 [60], a popular microcontroller SoC with BLE support, sending a non-connectable, undirected advertisement uses 60 µJ of energy. For the nRF52832 [64], a generational improvement on the nRF51822, sending a non-connectable, undirected advertisement uses 52 µJ of energy.

Allowing BLE connections to be made adds an additional energy cost. This is due to short listening windows (less than 100 µs) where the node checks to see if a connection request is being made. That same reception window is used to listen for scan requests if they are enabled. Again assuming a maximum size advertisement transmitted at 0 dBm, the nRF51822 requires 77 µJ to send a connectable advertisement while the nRF52832 requires 66 µJ. These represent a 28% and 27% increase respectively in energy consumed per advertisement.

Given energy costs per advertisement, we can determine the expense of advertising at a given interval. As advertisement intervals are a continuous notion rather than a discrete event, it makes sense to present average power draw rather than energy consumption. The power draw of advertising at a certain interval is then the single advertisement energy cost, divided by the interval time. To be accurate, we must also take into account the sleep power draw of the microcontroller. On the nRF51822 sleep draws an average of 11 µW, while on the nRF52832 sleep draws an average of 5.8 µW. Figure 3.9 demonstrates average power draw for an nRF51822 transmitting connectable advertisements at 0 dBm. It covers a range of intervals from several packets per minute up to ten packets per second.

For some applications, the energy costs of scanning are also important. Listening is far more expensive than transmitting due to the increased duration of radio use. Scanning for advertisements uses an average of 30 mW when performed continuously. For applications that infrequently scan, such as adaptation as discussed in Section 4.3, the energy use of scanning can remain low despite the higher power costs.

To put the power draw into meaningful context, we can pair it with a battery and determine total device lifetime. Note that in this case, we are not accounting for the costs of sensing and computation. A CR2032 coin cell battery stores 325 mAh at 3 V of energy, 2500 J. For a device with an nRF51822 advertising connectably once per second, this results in a 334 day lifetime, about a year. Adding redundancy and transmitting three packets per second instead would result in a 121 day lifetime. This is a significant increase to support reliability, and in reality a CR2032 has insufficient energy to support this task long-term.

Energy costs are greatly reduced by sensing and communicating less often. Sending a

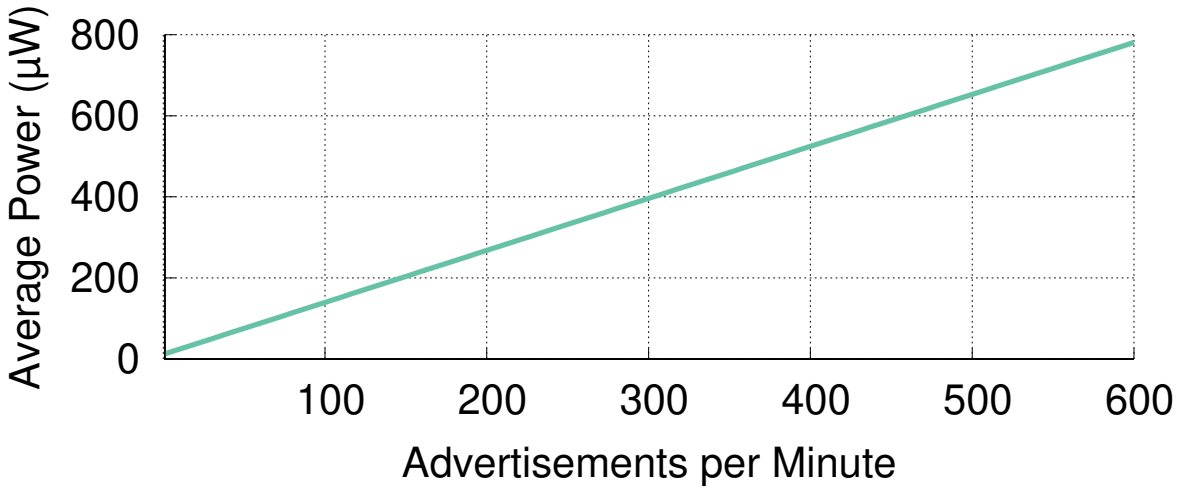

Figure 3.9: Average power consumption for transmitting at various rates. Connectable advertisements are transmitted at 0 dBm on an nRF51822, with a sleep power draw of 11 µW. Transmitting once per second results in an average power draw of 88 µW while transmitting every 100 ms results in an average power draw of 781 µW. These result in lifetimes ranging from a year to a month on a coin-cell battery.

single packet per minute leads to an average power draw of 13 µW with a lifetime of 2300 days. Since this lifetime is now dominated by sleep power draw, adding redundancy does not have such a large impact on total lifetime. Sending three packets per minute has an average power draw of 15 µW and a total lifetime of 1900 days.

3.3 Applying Models

Before we discuss deployments, it is valuable to take a look at what the models can teach us in the theoretical domain. First, we take a look at the BLE protocol itself to see how it could be modified and what the ramifications would be. Next, we take a look at an increasingly important domain, epidemiology and explore the use of BLE advertisements for contact tracing. Even before real-world deployments, we can gain valuable insights into protocol and application design through the use of our simple models alone.

3.3.1 BLE Protocol

A common desire in BLE advertisements is to fit a larger payload in each packet. Using our packet reception rate model, we can explore a change of payload from 31 bytes to 255 bytes as shown in Figure 3.10. We plot packet reception rate an advertisement interval of 1 packet per second for both payload sizes. At that rate, for up to 100 devices packet reception rate never falls below 90% for normally sized packets. However, packet receptions greatly

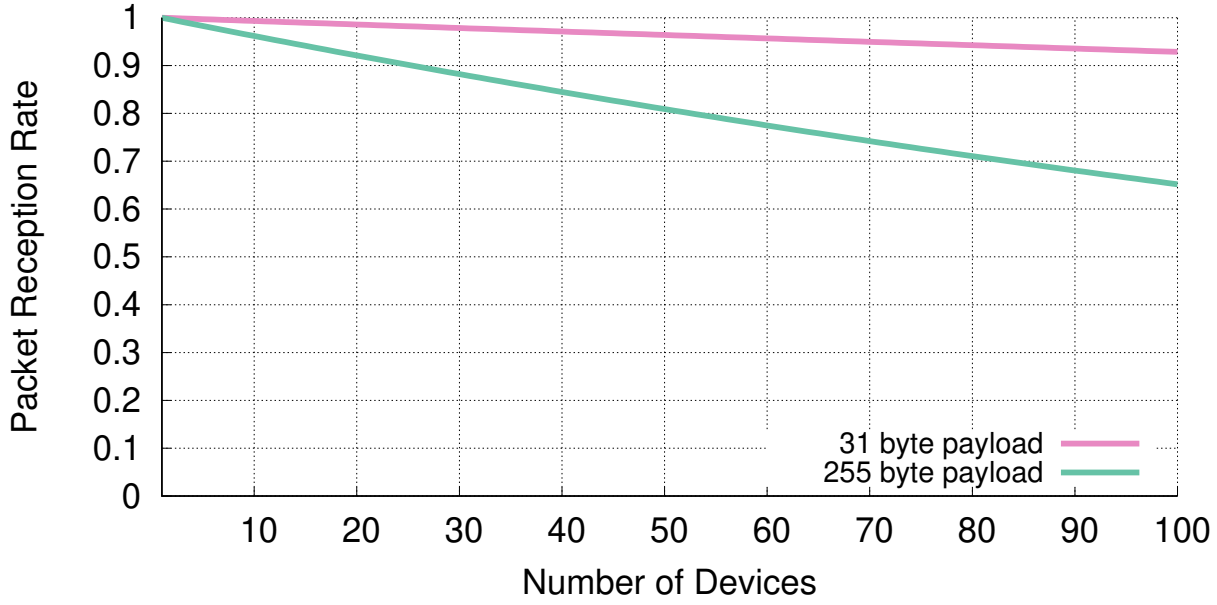

Figure 3.10: Packet reception rate for advertisements with expanded payload sizes. Normal BLE advertisements have a maximum payload size of 31 bytes. Increasing that payload to 255 bytes instead would have a large impact on packet reception rates and therefore discovery latency.

decrease for larger packet sizes due to increased collision windows, with only 65% of packet received with 100 deployed devices transmitting 255 byte payloads. The increased packet collisions caused by long payloads result in the loss data reliability, but would also result in increased discovery latency for beacons.

In version 5.0 of the BLE specification, we see that there is indeed an increased advertisement payload size. However, it cannot be used on the normal three advertisement channels and is instead relegated to the 37 connection channels now termed as "secondary advertisement channels". Devices broadcast normal-sized payload advertisements on the original advertisement channels that then point a receiver to another secondary channel for the increased length transmission. This somewhat strange protocol makes sense in the context of increased packet collisions. Devices using longer payloads in BLE 5.0 are kept from greatly reducing reception rates for non-longer-payload devices on the original advertising channels. The increased collision possibility from longer packets is greatly offset by the large number of secondary channels used.

Next, we are interested in exploring what a modification to the random transmission delay appended to each interval would result in. The delay is chosen from a uniform random distribution of 0 to 10 ms in all versions of the BLE protocol, between 26–78× the on-air duration of an advertisement packet. A shorter delay could allow faster advertisement rates, and therefore more data throughput.

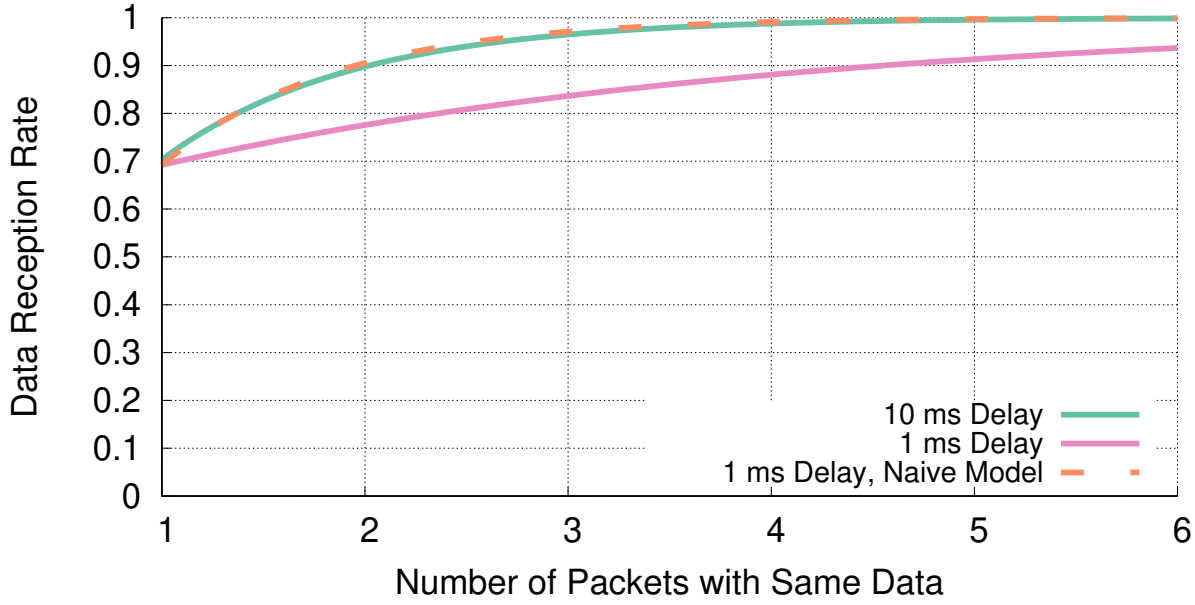

Figure 3.11: Data reception rates with modified random delay. The random delay appended to each advertising interval is normally selected from 0 to 10 ms. If its maximum value were to be reduced to only 1 ms, a significant increase in repeat collisions would occur, reducing the benefit of redundancy towards data reliability. Also shown is the naïve model of data reception rate, which cannot accurately account for increased repeat collision.

However, we find that reducing this delay from 10 ms to 1 ms would have a significant impact on data reception rates. Particularly, repeat collisions would have a large increase, which weakens the capability of two or three redundant packets to be sufficient for reliability. This is demonstrated in Figure 3.11 for 50 device deployments transmitting a packet every 100 ms. While with 10 ms delay, only two packets are needed to reach 90% data reliability, with delay reduced to 1 ms five packets are needed instead.

Also demonstrated in Figure 3.11 is the difference between the naïve and full models for data reception rate. The naïve method of multiplying P_c for each redundant transmission fails to account for increased repeat collisions in the 1 ms delay scenario.

Finally, we explore the use of multiple advertisement channels. As noted in Section 2.1.1, BLE advertisements are transmitted on three channels to increase robustness against RF interference. However, the three transmissions are always sent in the same order without any entropy in timing, which means they provide no robustness against advertisement collisions. A packet that collides on one channel will collide on the others as well.

A simple change, which is finally implemented in BLE 5.1, would be to allow advertisement channel order to be selected at random. Random advertisement channel ordering means that a packet collision can only occur if both devices are on the same channel during a transmission. We can modify Equation (3.1) to account for this new robustness. The odds

of a packet collision on a single channel are now reduced to a third of the original rate, as shown in Equation (3.7).

$$P_c = 1 - \left(1 - \frac{1}{3} \times \frac{2 \times t_{adv}}{T_{adv\_interval} + \mathbb{E}(t_{adv\_delay})}\right)^{N-1} \tag{3.7}$$

3.3.2 Contact Tracing

Models for BLE collisions and energy use can also inform emerging applications. One prime example is smartphone BLE-based contact tracing efforts. In epidemiology, contact tracing finds all recent contacts of someone who has become sick to test them for disease as well. This process is time consuming and manual, a clear target for technological improvement. In smartphone-based contact tracing systems, phones exchange information when they are in close proximity, creating a list of your recent contacts that can then later be notified or notify you in the case of disease spread. This use case is an example of a "proximal communication" application of BLE advertisements as discussed in Section 2.2.3.

A particular instance of smartphone-based contact tracing is the Apple/Google "Exposure Notification" system [54]. In this system, each smartphone transmits advertisements every 250 ms and scans for advertisers at least every five minutes, although the duration is unspecified. Opportunistically, the service will use scans requested by other applications for finer-granularity data.

Another take on this idea is the NIST Too Close for Too Long (TC4TL) challenge [65]. The competition seeks to explore the use of BLE advertisements for contact tracing by using RSSI, possibly in combination with other sensors or signals, to create an accurate range estimate for each possible contact. The end goal is to identify only contacts who have been within a certain range for greater than a certain period of time. The datasets provided by NIST lend some insight into their assumptions. They expect one data point per nearby BLE device per second, with a scanning duration of four seconds per minute. Combining these two contact tracing examples allows us to explore ramifications for likely contact tracing configurations.

One question of interest is how many nearby devices can be detected with high reliability in a single scan duration. We can investigate this question as an application of data reception rate. A detection is equivalent to a data reception in this scenario over the duration of the scan. So given an advertisement rate and scanning duration, we can determine maximum number of devices while meeting a desired reliability.

One unknown is the reliability of smartphone scanning. As will be discussed in Section 4.1.4, many scanners have challenges with reliable advertisement reception. Rather than using the default equation for DRR, we use the revised version from Equation (4.1) utilizing an optimistic loss of 7.8%. This is likely a significant overestimation as many smartphones share antennas for WiFi and BLE use, multiplexing them in time, and therefore puts an upper bound on number of devices.

Applying the DRR equation for a four-second scan and advertisements every 250 ms results in a redundancy of 16. Combining this with the gateway loss into Equation (4.1) results in detecting 432 devices with 99% reliability. This is an extraordinary number of devices in a single area for a scenario when contact tracing is necessary, suggesting that scanning duration could be greatly reduced. For example, scanning for a single second would allow 99% reliable detection of 98 devices, at one fourth of the energy cost.

This leads to a second question of how to determine the tradeoff between advertising and scanning. Phones could advertise more often and scan for a shorter duration, or advertise less frequently and scan for longer. The tradeoff here can be determined based on energy per models discussed in Section 3.2. Each second of scanning costs approximately 30 mJ. In comparison, each advertisement is approximately 60 uJ. So approximately 500 advertisements can be traded for each second of scanning. These numbers are for microcontrollers, but the order of magnitude applies to smartphones as well.

Assuming a desire to detect about one hundred devices in the nearby region with 99% reliability, multiple choices of scan and advertisement interval are possible. Scanning for one second every five minutes and continuously advertising every 250 ms results in a net average power draw of 340 uW. Alternatively, scanning for 800 ms every five minutes and continuously advertising every 100 ms results in a net average power draw of 680 uW, twice the cost. Or instead increasing advertising interval to 500 ms with a scanning duration of 1.45 s per five minutes results in an average power draw of 265 uW. The selection of these configurations could minimize the service's impact on overall phone lifetime while maintaining desired reliability and number of neighbors.

3.4 Summary

We have demonstrated that modeling advertisement collisions can accurately predict packet and data reception rates for real-world advertisement networks. Combined with a model for energy use, we have further demonstrated the capability to explore protocol modifications and application configurations. Our findings show that the introduction of modest redundancy should allow even large deployments to reliably communicate data over BLE advertisements. We follow through on this prediction in Chapter 4 with an analysis of real-world deployments. Our models will enable us to determine deployment issues when they arise and will allow the optimization of advertisement configurations for environmental conditions.

Chapter 4

BLE Deployment Studies

Powered with models that allow us to predict and introspect communication performance we now investigate real-world deployments of BLE advertisers. First, we explore PowerBlade, a plug-load power meter that uses BLE advertisements to broadcast measurements. We investigate existing data from PowerBlade deployments and identify gateway problems that led to unreliability. We next demonstrate the ability to apply models to statically plan deployments. Finally, we demonstrate an algorithm for deployed devices that enables them to automatically adapt transmission rates in the field for reliable communication.

4.1 Powerblade Deployment

PowerBlade is a research power meter that measures voltage and current waveforms in real time, transmitting the data via BLE advertisements [16]. PowerBlade's design results in a severely limited energy budget averaging less than 6 mW, making the use of traditional mesh networking architectures difficult. Moreover, it is valuable for PowerBlade to interact directly with users. BLE allows deployers to easily label devices and users to directly observe the power draw of metered appliances using a smartphone application.

We deployed PowerBlade in eight residential homes and one commercial office to study the contribution of various loads [15]. Table 4.1 describes the nine deployments, which include 335 PowerBlades over 608 deployment-days and result in 1.5 billion recorded measurements. Each location has occupants that live or work within it and includes other ambient wireless transmitters, including WiFi and other unrelated Bluetooth devices. However, these deployments took place before the deployment of Apple Continuity, so we expect few other BLE advertisers to exist in each household. We study the dataset collected from the PowerBlade deployment to evaluate the performance of its BLE advertisement sensor network and test the efficacy of our models on a real-world dataset.

In each location, one or more gateways are deployed to collect measurements. The gateway consists of a BeagleBone Black running Linux with an attached USB Bluetooth dongle [66, 67]. The Noble JavaScript library [68] drives the Bluetooth stack, and a simple

Location	Duration (Days)	Number of PowerBlades	Expected PRR	Expected DRR
1	66	68	63.5%	96.8%
2	168	84	57.0%	94.5%
3	7	12	92.8%	100.0%
4	110	23	86.1%	99.8%
5	87	37	78.3%	99.3%
6	83	35	79.4%	99.4%
7	40	29	82.7%	99.6%
8	24	21	87.3%	99.8%
9	23	26	84.4%	99.7%

Table 4.1: PowerBlade deployment overview. 355 BLE power meters are deployed in nine locations (averaging to 37 devices at each location for 68 days). Given the network configuration, our models predict the data reception rate for most deployment locations to be greater than 99%. This deployment provides an opportunity for measuring BLE advertisement network performance in the real world and comparing to the theoretical expectations.

application atop Noble parses advertisements into data packets. Packets are de-duplicated before being sent to a database backend. This data can be used to measure data reception rates for the deployments. At one location (Location 4), raw BLE packets are collected in addition to parsed measurements, allowing additional analyses of packet reception rates for that deployment.

Each PowerBlade generates a new power measurement once per second. It transmits with an advertising interval of 200 ms, sending four redundant packets for each measurement. The fifth packet each second is an Eddystone beacon [45] that points to a user application capable of interacting with the device. Each power measurement has an attached sequence number, allowing missing measurements to be detected. A PowerBlade data payload is 27 bytes while the Eddystone payload is 26 bytes. The redundancy amount for each device was chosen before the creation of data reception rate models, and was selected as an educated guess about redundancy needs.

PowerBlade also enables scan requests and responses. The device name field is too large to fit in the advertisement payload with the power data, so PowerBlade places it in the scan response, allowing smartphones to identify it as a PowerBlade. The scan response payload is 12 bytes. When transmitted, this can act as an additional source of collisions in the deployment, essentially increasing the transmission time of packets when a scan request is received. While the PowerBlade gateways do not need to collect scan response data, the Noble library used to collect advertisements does not provide an option for disabling scan

Figure 4.1: Data reception rate by deployment location. Data reception rate is determined by counting sequence numbers received throughout the deployment duration and dividing that by the expected count of sequence numbers. Expected DRR is marked as a black line above each bar. While we expect a DRR of greater than 99% for the majority of locations, we instead find that most locations receive between 50% and 80% of expected measurements.

requests. In retrospect, scan responses are unnecessary to the deployment. As discussed in Section 2.1.3, scan requests and responses have a backoff design that is problematic in large deployments. A redesign of the PowerBlade protocol with current knowledge would disable them to save energy and increase reliability.

4.1.1 Expected Reception Rates

We can apply the models from Section 3.1 to predict performance. We use an average value for t_{adv} of 342.4 µs and $T_{adv\_interval}$ is 200 ms. The number of redundant transmissions is four, and we assume that all devices in a deployment are within transmission range of each other.

While the actual probability of a scan request occurring is unclear due to scanner backoff policies, we can assume a worst case in which every advertisement has a corresponding scan request. We can further assume that every scan request is properly received and results in a scan response. We can model this worst case as an extension to the duration of the collision window. So rather than $2 \times t_{adv}$, the numerators of Equations (3.4) and (3.5) become $2 \times t_{adv} + 2 \times t_{IFS} + t_{scan\_req} + t_{scan\_resp}$ where t_{IFS} is the inter-frame spacing of 150 µs, t_{scan_req} is the scan request on-air duration of 176 µs, and t_{scan_resp} is the scan response duration for PowerBlade of 224 µs.

Accounting for scan requests and responses in this manner, Table 4.1 lists the expected packet and data reception rates for each deployment. While the deployment size is large enough to produce more than 40% packet error in the worst case, all deployments are ex-

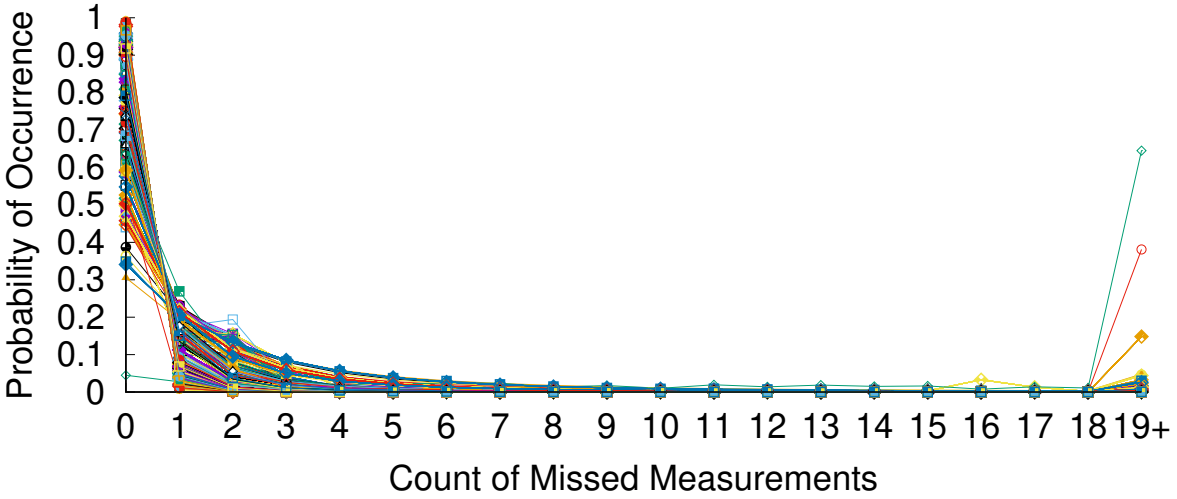

Figure 4.2: Measurement loss probability. For each received measurement, we count the number of immediately preceding measurements that were missed. A line is plotted for each device in the deployment. As expected, the most common count is zero (the previous measurement was received). However, brief streaks of one to several missed measurements are not rare. Longer gaps become less common, until we consider very long gaps—19 or more consecutive missed measurements. Large gaps suggest the possibility of infrastructure or device failure during the deployment.

pected to have data reception rates above 90% and most expect above 99%. The diversity of requirements leads to a tradeoff in system design where reliability can be guaranteed for even the largest scenario but only at the cost of wasted energy in the smaller common-case scenarios. This is an opportunity to apply an automatic adaptation scheme, as discussed in Section 4.3.

4.1.2 Measured Data Reception Rates

We can determine the data reception rate for each deployment location by observing the sequence number in each measurement, and then comparing how many unique measurements were received to how many were expected. As shown in Figure 4.1, data reception rate is significantly lower than expected. The best performance is in Location 4 with 83.4% DRR.

To attempt to identify why the network is underperforming, we can measure the data loss pattern. Figure 4.2 shows the probability of occurrence for increasing lengths of missed measurements for each device in the deployment. The most common number of missing data is zero for all devices except one. As run length increases the probability falls off, however there is a relatively high occurrence of very long runs of dropped measurements. It is possible that long periods of dropped measurements represent infrastructure failures rather than problems with devices. This includes problems such as gateway crashes and

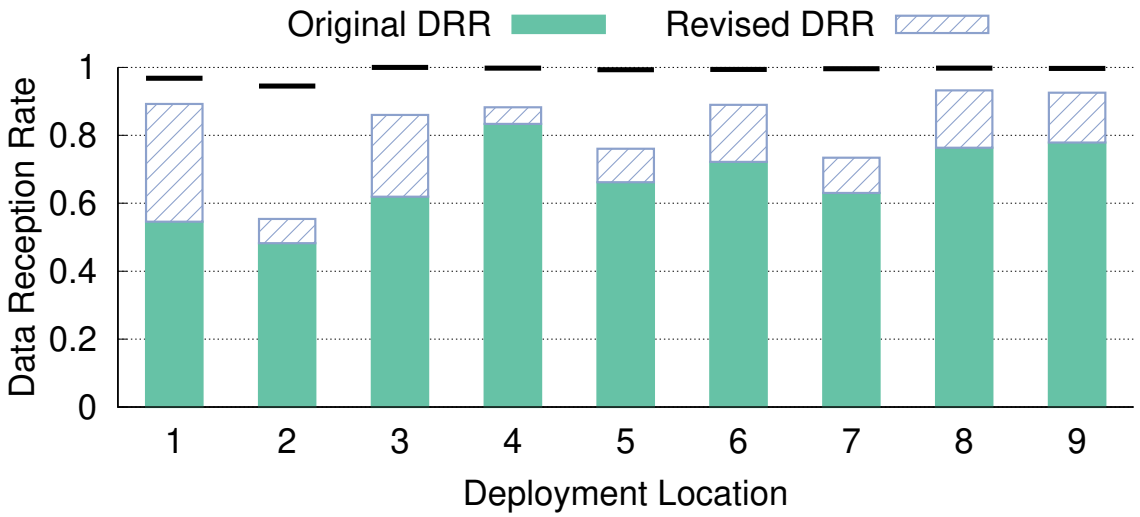

Figure 4.3: Data reception rate by deployment location with gaps removed. We liberally remove any contiguous gaps longer than one hour in duration from the expect packet receptions for each deployment. Expected DRR is marked as a black line above each bar. Even if all of these gaps represent true infrastructure failures, they fail to account for all of loss in reception rates.

network outages that are all too common in real world deployments [69].

To investigate whether infrastructure failures could be the source of most of the packet loss in these deployments, we investigate packet loss ignoring large gaps in data. Figure 4.3 displays the updated data reception rates if we discount all gaps in data of an hour in length of more. Upon investigation, Locations 1 and 3 each have multi-day gaps during their deployment period, which are likely true outages. However, even liberally removing all one-hour gaps does not result in satisfactory results, leading to suspicion that another factor is at fault.

4.1.3 Packet Reception

To gain a better understanding for the difference between predicted and measured performance, we can dig deeper into the performance of devices at Location 4. In that location, raw packets were recorded in addition to measurements. This includes the reception time, sequence number, and RSSI for each packet. We determine packet reception rate by counting the sequence numbers received, including duplicates, and dividing by the number of expected packets (four times the number of expected measurements). We discount any contiguous gaps of more than an hour to remove the effects of infrastructure failures. This deployment also tracks the room location in the residence where the PowerBlade was deployed. Location 4 contains a living room, kitchen, and bedroom. The living room and kitchen are adjacent, while the bedroom is farther away. The gateway for this deployment is placed in the living

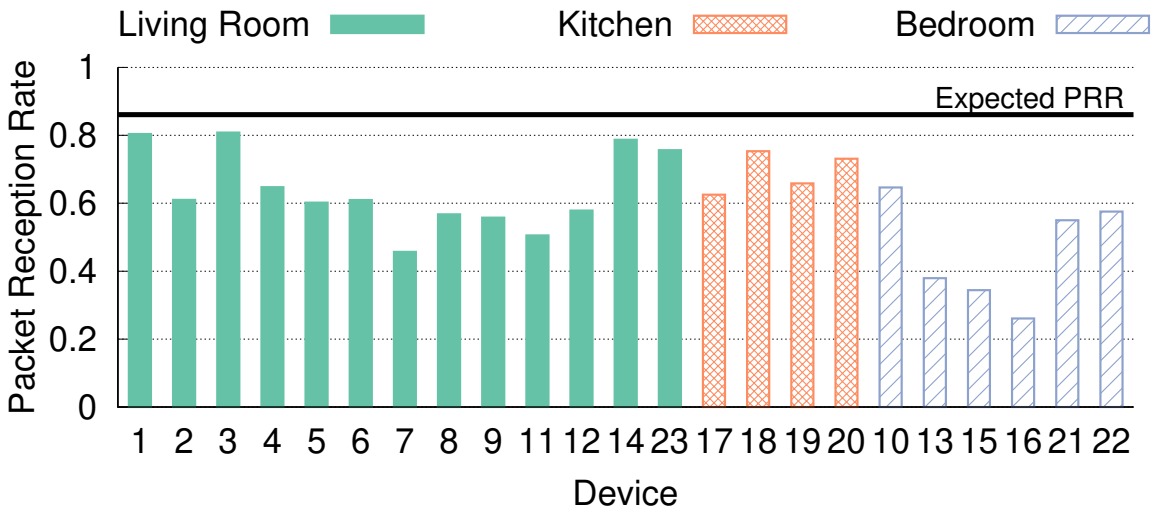

Figure 4.4: Packet reception rate by device at Location 4. Devices are grouped by the room where they are deployed, with the gateway being placed in the living room. Expected PRR is the same for each device and marked as a black line. While all devices are within a short distance from the gateway and should experience only about 14% packet loss, no device performs this well and many devices perform far more poorly.

room.

For Location 4, the model expects a packet reception rate of 86.1%. Figure 4.4 gives a breakdown of the measured PRR for each device, grouped by the room in which they were deployed. Devices in the living room and kitchen, closest to the gateway, have the highest reception rates in general. The maximum PRR seen is 80.8% for device number 3. Variation in PRR between devices implies differences in path loss each is experiencing, as is expected in real-world deployments. However, even the best performing devices are below the predicted PRR, suggesting possible issues with packet reception. This is made worse by the fact that expected PRR should be an underestimate. We would expect that due to the capture effect, some devices close to the gateway would do even better than predicted.

Reduced signal strength due to distance and obstructions certainly causes some of the packet loss that is observed. While we only have access to received signal strength indicator (RSSI) measurements for successfully received packets, rather than continuous signal strength measurements, we can examine the short-term trend of RSSI to see if variations are a likely cause of packet loss. Figure 4.5 views RSSI measurements over an arbitrary minute of the deployment for three devices in Location 4, including the best and worst performing devices. We see modest fluctuations in RSSI values, which account for some of the missing transmissions from Device 16. However, the magnitude of variation for the other two devices is small enough to make poor signal strength an unlikely cause of missed packets. We note that the conversion of actual signal to RSSI for this receiver is opaque, and it appears that higher RSSI values exhibit greater quantization. Even if there is not quantization, the RSSI

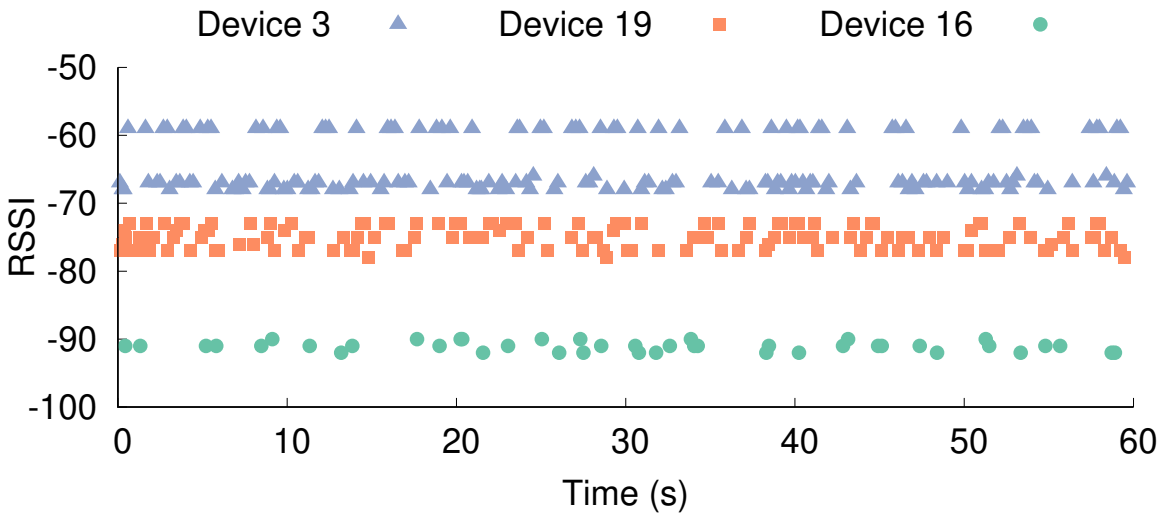

Figure 4.5: RSSI measurements from three devices over a minute. The received signal strength fluctuates over a small interval from packet to packet. For Device 16 these variations may be sufficient for its transmissions to fall below the sensitivity of the receiver (roughly -95 dBm). For the other devices, variation in signal strength seems unlikely to be the cause of missed packets.

of Device 3 would need to change by three times the amount seen in this trace to fall below the reported sensitivity. Only a minute out of the entire deployment is shown for brevity, but the results are similar on the scale of an hour, day, or week.

4.1.4 Gateway Analysis

With problems in data reception rate and packet reception rate among all devices, remaining causes include RF interference and gateway issues. RF interference, while likely accounting for some losses, is unlikely to affect all locations, especially when it has not been a significant factor in other indoor BLE experiments. This leads to a focus on the gateway used to receive BLE advertisements. It is possible that packets are being dropped by the USB dongle hardware or firmware, in the Linux BLE stack, or in the Noble BLE library. To test for this, we can compare the performance of various gateway configurations compared to the professional scanner in a controlled setting.

Re-using the experimental setup from Section 3.1.4, we place twenty-five nRF51822 BLE beacons in a single room. At five different advertising intervals, we simultaneously record the received packets from a sniffer and other gateway designs placed adjacent to each other. Table 4.2 shows the percent error from the theoretical model for each gateway configuration at each advertising interval tested.

"Original Gateway" is the configuration used in the original PowerBlade deployments, a USB BLE dongle with the Noble BLE library. We find that this gateway configuration

Advertising Interval (ms)	100	500	1000	5000	10000
	Deviation from Theoretical Performance (%)				
Original Gateway, Noble	-46.6	-42.8	-39.5	-36.4	-34.4
Modified Gateway, Noble	-32.9	-27.2	-23.9	-21.3	-18.3
nRF52DK Gateway, Noble	-33.3	-22.6	-17.1	-14.4	-12.3
ESP32 Serial	-8.3	-14.2	-12.9	-12.4	-11.6
nRF52DK Serial	-4.3	-9.9	-8.9	-8.5	-7.8
Professional Sniffer	-1.3	-3.0	-2.7	-3.4	-3.1

Table 4.2: Additional packet loss in tested receivers. 25 BLE beacons are set in a single room advertising data. For a series of advertising intervals, all packets are recorded by: a BLE gateway identical to the ones used in the PowerBlade deployment, a modified gateway with an increased scan interval, a gateway with an increased scan interval and different BLE hardware, an ESP32 BLE scanner over serial, a nRF52DK BLE scanner over serial, and a professional BLE sniffer. Packet reception rates are improved most by avoiding the Linux BLE stack and Noble library and instead streaming data directly from a scanning microcontroller. However all configurations tested do introduce some packet loss above that predicted by the theoretical models, which needs to be accounted for.

performs far more poorly than the professional scanner with a 30 to 40 percentage point lower reception rate.

A low scan interval on the gateway could result in frequent scanning gaps while channel-switching [31]. "Modified Gateway" increases the scan interval in the Noble library from 10 to 100 ms, resulting in some improvement, but still 20–30% loss. A different potential source is problems with the receiver hardware or firmware. Using an open-source HCI stack implementation [70], we configure a nRF52 development kit to receive packets for the Noble library with an increased scan interval. "nRF52DK Gateway" shows these results, which still demonstrate loss of 10–30%.

The most successful technique tested is to avoid the Noble library and Linux BLE stack altogether and instead stream advertisement data directly over a serial connection from a BLE scanner. We investigate the nRF52 development kit and the ESP32, both ARM microcontrollers combined with BLE radios. For the nRF52, we find 7.88% additional packet loss on average from the theoretical model.

Overall, we find loss due to both the scanning hardware and software. The particular choice of hardware and libraries used in the PowerBlade deployments, while the straightforward path, introduced the most packet loss of any configuration tested, likely accounting for many of the packet reception problems discovered in the deployment. Here we find that the ubiquity of BLE is a double-edge sword. Gateways are easy to build due to available commercial BLE dongles and installable radio stacks, but the presence of multiple layered

and opaque stacks makes understanding the flow of data and tracking down the source of possible losses difficult.

4.2 Statically Planned Deployments

While the results from the unplanned PowerBlade deployments are poor, combining knowledge of BLE advertisement models with the experimentally determined packet loss expected at the gateway can allow deployments to be accurately predicted. We demonstrate this through an additional two deployments, planned in advance to result in different reception rates.

4.2.1 Revising Deployment Parameters

An expected loss due at the gateway can be accounted for in our model of predicted data reception rates similarly to loss due to interference. Equation (4.1) shows the probability of receiving data sent M total times where P_l is probability of packet loss at the gateway, P_c is the probability of a collision, and P_{rc} is the probability of a repeat collision.

$$DRR = 1 - [1 - (1 - P_c)(1 - P_l)] \, [1 - (1 - P_{rc})(1 - P_l)]^{M-1} \tag{4.1}$$

Given this new model, we can now accurately plan deployments with knowledge of their data reception rates in advance. The first deployment we plan targets a 99% data reception rate and is referred to as the "99%" deployment. Assuming 7.88% loss at the gateway (the average for the nRF52 development kit over serial), 25 devices deployed, and new data once per second, we can solve for expected data reception rate given a number of redundant packets per second. Sending one packet per second (no redundancy) has a DRR of 90.6%, two has a DRR of 98.2%, and three has a DRR of 99.6%. We select three total packets per second (one every 333 ms) to reach a predicted data reception rate of over 99% for our planned deployment. For the second deployment, we target a deployment with 80% data reception rate and refer to it as the "80%" deployment. With the same number of devices and expected loss at the gateway, an advertising interval of 112.5 ms with no redundancy should lead to this outcome due to packet collisions.

Twenty-five PowerBlades are deployed throughout an apartment, attached to devices and plugged into outlets. A gateway using a nRF52 development kit over a serial connection is deployed at the intersection of the kitchen and living room. Nodes in the bedroom are the furthest away from the gateway, transmitting through a wall for the shortest-path link. The PowerBlades are first configured for the "99%" deployment and data is collected for 24 hours. Following that, devices are reprogrammed for the "80%" deployment and then replaced in the same locations for an additional 24 hours.

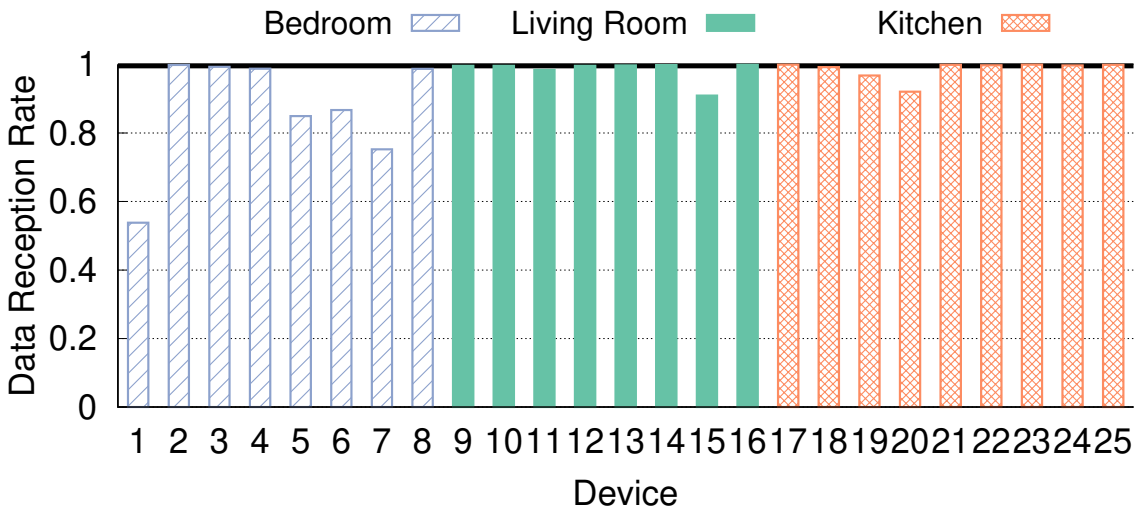

Figure 4.6: Data Reception rate for each device in the "99%" deployment. A gateway is deployed between the living room and kitchen of an apartment, with a bedroom adjacent through a wall. Expected DRR is marked with a black line. The majority of devices have greater than 99% data reception rate as predicted. Some devices, especially those further away, suffer degraded performance due to poor connectivity.

4.2.2 Deployment Results

The results of the real-world deployments are messier than the managed setup used for empirical testing, but far more predictable than the unplanned BLE deployment as we minimize and account for gateway issues. Figure 4.6 shows the results of the "99%" deployment. Out of the 25 deployed devices, 15 have a greater than 99% data reception rate, with another 3 greater than 98% and a median data reception rate of 99.6%. Devices in the bedroom are furthest away from the gateway and also have the highest prevalence of weaker than expected reception rates. Devices 19 and 20 are physically close to the gateway, but positioned behind a microwave, which likely results in weak signal strength.

We can also view the results of the "99%" experiment as average PRR for each minute of the deployment. Figure 4.7 demonstrates PRR for all 25 devices in the deployment. Many devices have high packet reception rates throughout the experiment, matching our expected PRR of 90.6%. An interesting event that can be observed in the PRR traces is the bedroom door which closes for the night around 10 hours into the experiment. Devices 1, 7, and 15 all respond to this event in different ways: improving, degrading, or stabilizing their reception rates.

Without changing the locations of deployed devices, the "80%" deployment has a predicted data reception rate of 80% and a measured median data reception rate of 78.5%. 8 devices have reception rates within the range of 76–84%, as displayed in Figure 4.8. An additional 7 devices have reception rates better than predicted, in the range of 84–90%.

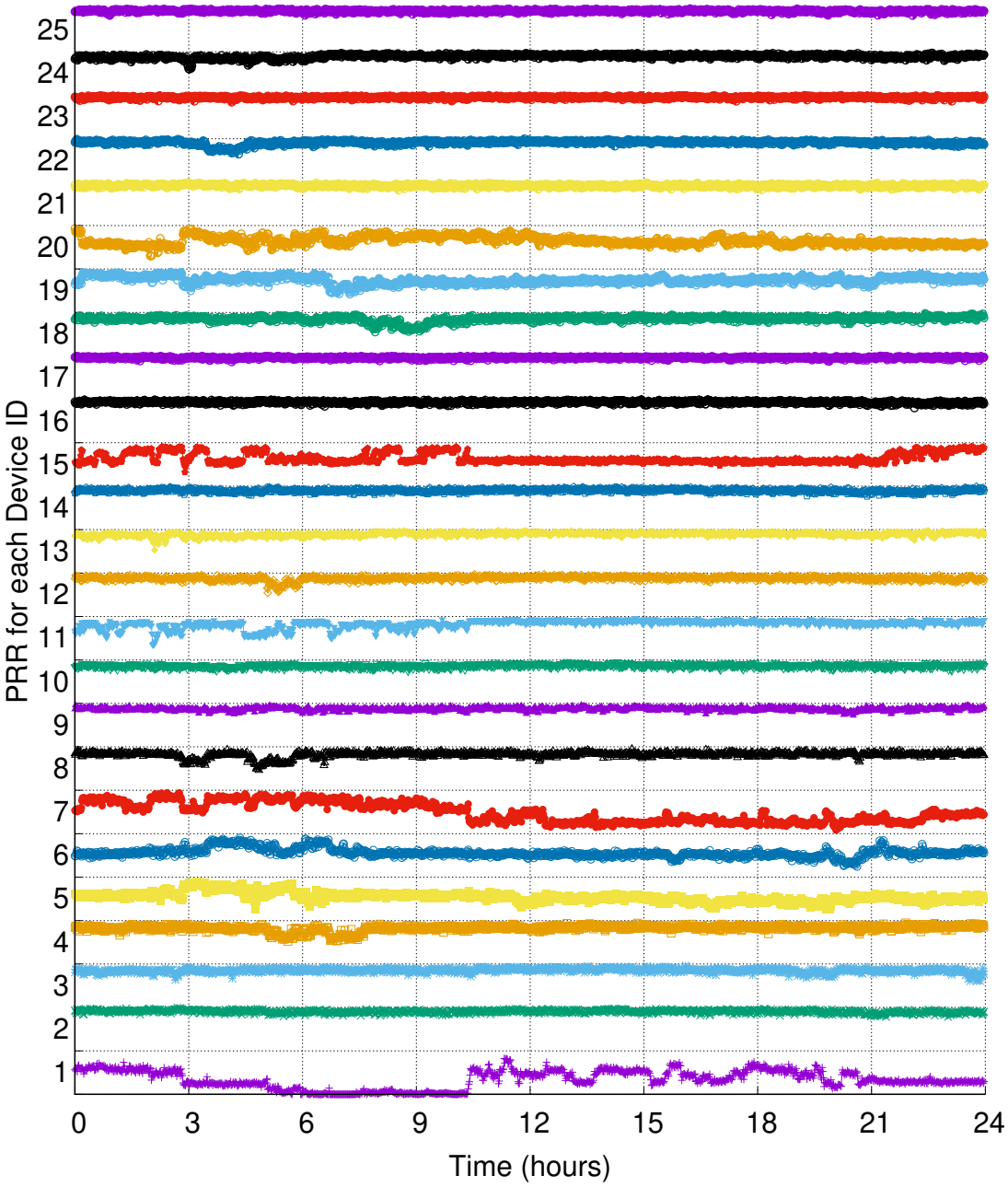

Figure 4.7: PRR over 24 hours for the "99%" deployment. Average PRR for each minute is plotted for all 25 PowerBlades in the deployment. The grid line above each ID is 100% reception rate for that device and 0% reception rate for the ID above it. With redundancy, 98% of data was received from 18 of the 25 devices. Poorly performing devices, like device 1, exhibit variability in PRR that suggests poor connection to the gateway.

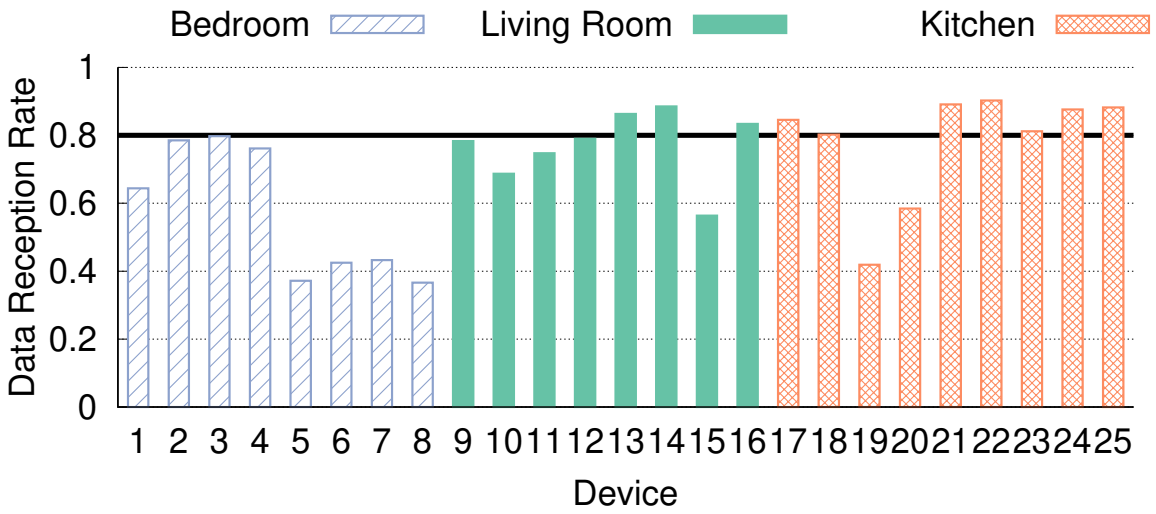

Figure 4.8: Data Reception Rate for devices in the "80%" deployment. The predicted reception rate is 80%, marked with a black line, and the majority of devices meet this expectation. Several devices close to the gateway perform better than expected, likely due to the capture effect. Without changing the locations of devices, the resulting data reception is poorer than in the "99%" deployment as expected due to more frequent packet collisions without redundant transmissions.

Looking at the received signal strength for all devices in the deployment, we believe the capture effect is the cause. Prior work evaluates the capture effect in BLE, finding that stronger colliding packets can be received successfully at any point during the weaker packet (not just if transmitted during the preamble) as long as the packet is 14 dB higher in strength [57]. We find that in our "80%" deployment, all devices above 84% reception rate are also above this threshold in comparison to at least two of the other deployed devices. Device 22 is the strongest case of this, with its transmissions averaging 14 dB higher signal strength than transmissions from 12 other deployed devices.

Neither deployment was successful in predicting the reception rates for all devices, and in each some devices have particularly poor reception. Although reduced reception does correlate with distance, determining in advance which devices will perform poorly is difficult. Even between the two planned deployments we find that which devices do poorly changes, for example devices 1, 8, and 19. Devices 1 and 2 are also deployed only inches from each other, but have wildly different results in the "99%" deployment case.

While prediction is difficult, detection of these poorly performing devices is made simple through the application of reception rate models. In these cases, devices could be adjusted or replaced to improve signal strength. Alternatively, a second gateway could be deployed in the bedroom, which would improve reception rates without any changes to the deployed devices.

4.3 Adapting to the BLE Environment

In the prior examples, we statically configured advertisement parameters based on our expectations of transmission contention. In many real-world scenarios however, the number of communicating devices is often not known in advance. This is especially true for long-term deployments, where new devices may be added or old devices removed over time. Automatically adapting to the environment would allow a device to ensure reliable communication under any scenario, while also minimizing transmissions and therefore conserving energy.

4.3.1 Measurement Method

The first step of adapting to the environment is measuring it. Many BLE radios support dual mode operation, where a device can be both an advertiser and a scanner. By entering scanner mode, the device can receive all valid BLE advertisements being broadcast around it. Doing so for a certain period will give a good sense of what the transmission contention looks like. Usefully, the ability to do a brief scan is part of the BLE API on microcontrollers and smartphones. Which means that this method will work on almost all devices.

One unfortunate issue with the basic scanning method is that it does not receive BLE advertisements with bad CRC values. This means that by default a scanner does not know how many transmissions were lost due to collisions. Unfortunately, packets with bad CRCs are usually discarded at the lower layers of the BLE library and not made available to applications. Modified BLE libraries would be needed to receive all packets. If an application has access to the radio hardware, another improvement would be to listen for all traffic on the channel, rather than just BLE advertisements. This would allow measurements to account for interference from other protocols as well.

4.3.2 Measurement Frequency

The next question is how frequently measurements should be taken. Knowledge of the deployment scenario could provide insight into how often the transmission environment is expected to change. The largest driver of rapid environmental changes are people. Wearable devices like Fitbits and personal devices like iPhones use BLE advertisements to communicate and go wherever their owner does. In locations like lecture halls, the environment can be expected to change hourly as groups of students come and go. In locations like shopping malls or subway stations, any particular device may only be in transmission range for a few minutes. For household and office buildings where the flow of people is smaller, their personal devices are unlikely to be the dominant source of transmission contention compared to a deployed sensor network. Knowing which of these environments are being measured can lead to a reasonable measurement frequency selection.

Another take is that measurements and adaptation should be based on how long a device can afford to either be unreliable or to expend excess energy. Devices that are extremely concerned with reliability should measure more frequently, and may transmit additional

redundant packets above modeled requirements to ensure reliable communication between measurements. Devices that are more concerned with energy savings should scan more frequently when transmitting a high number of packets, hoping the environment has changed and energy can now be saved.

Measurement frequency can also be automatically varied. A device would begin sensing periodically at some default rate. If the measurements are changing more than some acceptable margin, increase the sensing rate. If the measurements are stable over multiple periods, decrease the sensing rate. For this method it is important that the sensing period does not align with time-of-day. Measuring each day at 11:00 AM may result in a similar number of transmissions, while every six hours could show large variation due to the movement of people.

4.3.3 Measurement Duration

A final measurement question to discuss here is how long to measure for. In the original BLE specification, the longest allowable advertisement period was 10.24 s. Obviously a device could transmit slower than that if desired, and more modern iterations on the specification allow any maximum interval. Combine that with movement of devices and any listening duration will be insufficient to identify all devices in the area.

However, not all advertisement rates matter. If a device transmits once per second, collisions with devices transmitting once per minute will be incredibly rare. It should be sufficient to listen for up to some small multiple of the base desired transmission rate, where slower transmitting devices will be a negligible source of collisions.

It is also possible to listen to only a small window in time and to then extrapolate. Due to the uniform random delay, at steady state BLE advertisements are randomly distributed in time. So sampling any tenth of a second is likely to contain one tenth of the advertisements sent per second. Packet transmissions are not instantaneous, so this subsampling reduces in accuracy as the window becomes closer in size to a packet duration (maximum of 376µs). Also, since we count in discrete packet receptions, small variations can lead to large error when extrapolating.

Reductions in sampling duration result in advantageous energy savings, but ultimately result in loss of accuracy. Simulation can help us explore this tradeoff. Assuming perfect reception of all BLE advertisements transmitted, colliding or not, Figure 4.9 demonstrates the error in estimated collision probability for varying listen durations. Four different transmission rates are selected, which could transmitted by many or few devices interchangeably. Listening for at least 200 ms provides an error of less than 5% when estimating collision probability (and therefore decided transmission redundancy).

4.3.4 Measurement Energy Cost

While scanning is a much higher energy cost than transmitting, periodically performing a brief scan for advertisements can be a low energy cost if rare and brief enough. Scanning for

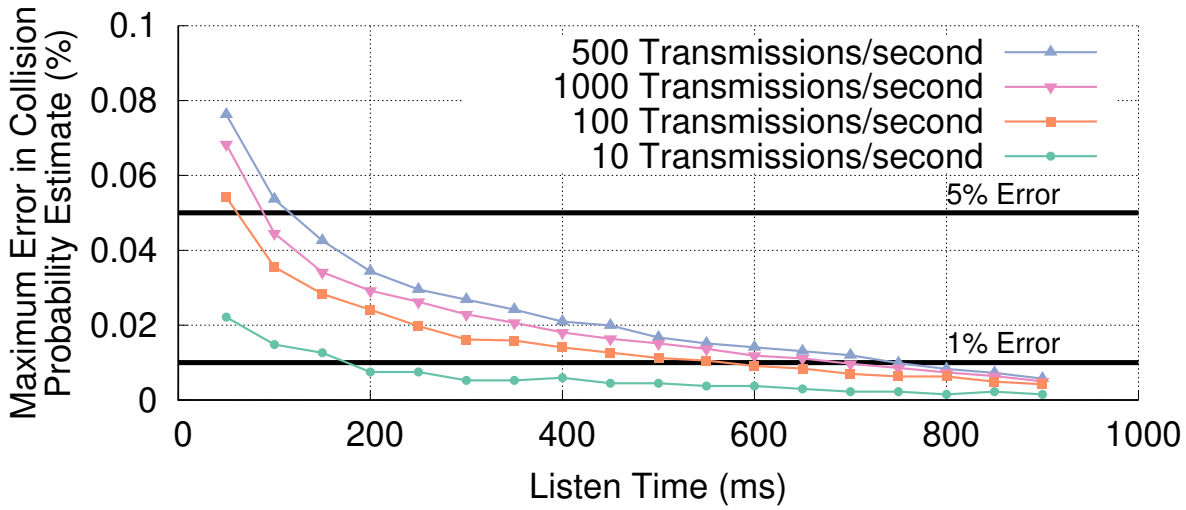

Figure 4.9: Error in collision estimation due to limited measurement windows. Sampling for an entire second gives a true measurement of transmissions per second. Sampling for a percentage of a second allows for extrapolation to an estimated measurement, at the cost of reduced accuracy. A measurement duration of a least 200 ms gives an approximation of true contention that is accurate to within 5% while only using roughly a fifth of the energy.

one second every ten minutes increases an nRF51822's average power draw by about 70 µW. This is a similar cost to that of increasing advertisement rate by one packet per second. For applications expecting rapid changes in the environment, this may be a reasonable price to pay. As discussed above, this scanning rate could be greatly decreased to daily or even weekly, diminishing the energy cost to a fraction of the cost of a transmission per second.

4.3.5 Adaptation

Two steps are needed for adaptation: estimation of collision probability and determination of redundancy. Given a reasonably accurate measurement capability, the receptions can be transformed into an estimated packet reception rate. Then that packet reception rate can be used, along with a desired data reception rate, to select transmission redundancy.

As calculations will be performed on microcontrollers, some simplifying assumptions are helpful. First, we can utilize the naïve model for packet collisions, as mentioned in Section 3.1.2. This will increase error slightly, but is unlikely to have a major impact on redundancy selection.

For another simplification, we can measure packets per second, rather than actual devices and advertisement intervals. The difference in collision rate between sending, for example, 200 packets from a single device or 2 packets each from 100 devices, is around 1%. Accounting for the duration of the packet is a more important factor for determining collisions, but packet size is provided to higher layers of most BLE libraries.

Combining these simplifications, Equation (3.4) can be revised to a product across all received packets of the device's own on-air duration (t_{adv_0}) plus the received on-air duration (t_{adv_i}), divided by the scan duration (t_{scan}).

$$estP_c = 1 - \prod_{i=1}^{Packets} \frac{t_{adv_0} + t_{adv_i}}{t_{scan}} \tag{4.2}$$

After calculating an estimated probability of collision, a desired data reception rate can be used to determine redundancy configuration. Algorithm 1 demonstrates the steps for doing so. Eliding repeated collisions, DRR is a product of PRR for each redundant transmission. In a microcontroller setting without floating point support, this same algorithm can be used with fixed point arithmetic instead without significant loss in accuracy. An upper limit on redundancy can also be imposed based on maximum acceptable energy use.

Algorithm 1 Pseudocode to determine redundant transmissions.

```
1: procedure CalculateRedundancy(estimatedPRR, desiredDRR)
2:     drr ← estimatedPRR
3:     advCount ← 1
4:     while drr < desiredDRR do
5:         drr ← 1 − ((1 − drr) ∗ (1 − estimatedPRR))
6:         advCount ← advCount + 1
7:     end while
8:     return advCount
9: end procedure
```

A concern with automatic adaptation is stability if many devices were employing the same algorithm. There are regions of stability where a number of devices can all be following the same algorithm and meet the same reliability. Adding enough devices will move the entire deployment to add an additional transmission per second. Eventually, this can go out of control, where the addition of transmissions harms overall reliability. The number of devices this occurs at depends on the desired DRR.

Table 4.3 demonstrates the maximum deployment size and stable regions for 90%, 95%, and 99% desired DRR. For example, 80 total devices are stable at 3 transmissions per second with 99% DRR (assuming data is generated once per second). 140 devices, however, are not stable at 99% DRR, but would stably meet 95% DRR with 2 transmissions per second. High reliability deployments of over 100 devices in a single area unsurprisingly need to investigate other protocols.

4.3.6 Experimental Results

We implement the discussed algorithm on an nRF51822 in order to test it. The "adaptation device" scans for one second every ten minutes, acting as a BLE scanner. It generates a new

Transmissions per Second	Desired DRR 90%	95%	99%
1	140	68	13
2	252	168	70
3	276	203	107
4	274	212	126
5	265	211	134
6	253	206	138
7	241	200	138

Table 4.3: Number of simultaneous devices supported at a desired data reception rate. Assuming all devices are following the same algorithm for determining redundancy, the number of devices that can be supported depends on the desired DRR. As more devices are added, additional transmissions are needed from each to maintain the same reliability. At a certain point, shaded grey in this table, additional devices cause a failure in the algorithm where more transmissions lead to reduced reliability. Deployments kept at less total devices in a single broadcast domain than this number will be stable.

data point each second and starts off transmitting one BLE advertisement per second. After performing each scan, it adjusts its transmission rate to reach 99% DRR. We additionally account for expected gateway losses when calculating packet loss as discussed in Section 4.2.1.

The environment the adapting device is deployed in has 20-100 BLE advertisements per second (background noise from nearby iOS devices). We additionally enable up to 49 other BLE transmitters, all placed within two meters of the adapting device, in order to increase noise in the environment. Additional transmitters are changed between ten-minute scan periods and transmit ten packets per second each. A gateway is placed within two meters of the adapting device to collect results.

Figure 4.10 demonstrates the results from this test. The top plot (in green) visualizes the number of BLE transmissions per second in the nearby environment. The spikes in transmission count near the start are due to nearby iOS devices, which increase advertisement rate considerably when being interacted with. From time range 15–35 minutes, twenty additional transmitters are added to the environment, from time range 35–45 they are disabled, and from time range 45–65 all transmitters are enabled.

The middle (orange) and bottom (blue) plots respectively display the number of advertisements per second sent by the adapting device and the resulting data reception rate for the adapting device. Vertical dashed lines every ten minutes denote when the adapting device scans the environment and then adapts its transmission rate. Grey regions from 0–12, 15–22, and 45–52 minutes mark points where the adapting device has underestimated the environment and poor performance is expected. Data reception rate is measured as a running average of the last 100 seconds, causing a lag after adaptation before reliability improves.

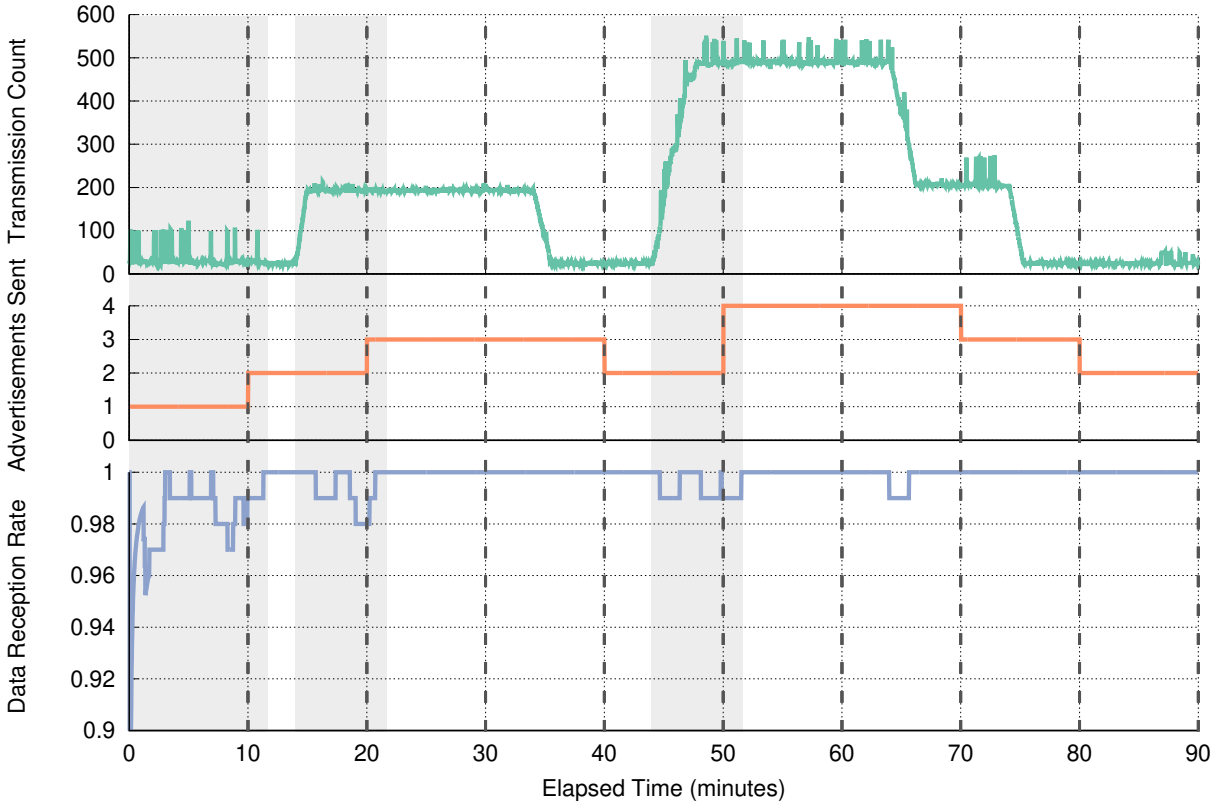

Figure 4.10: Runtime adaptation to the BLE environment to maintain target reliability. Over a 90 minute experiment, the number of BLE advertisements transmitted varies from around 20 to more than 500. One adapting device is deployed in this environment, scanning and modifying its behavior every ten minutes (marked by vertical dashed lines). Recorded for this adapting device are the number of advertisements it sends per second and the data reception rate (as a running average of the last 100 samples) for those advertisements. Regions marked in gray (before the 10, 20, and 50 minute marks) are periods when the adapting device is underestimating transmissions in the environment and poor performance is expected. After the device's next scan of the environment, it increases its redundancy to account for these additional transmissions in the environment and maintain 99% data reception rate. When the device has overestimated the environment, it reduces redundancy to save energy. The addition of simple adaptation capability allows the adapting device to maintain reliability even when the transmission in the environment change by an order of magnitude.

Initial background transmissions are enough to warrant the transmission of two packets per second to maintain 99% DRR. With more than 500 transmissions at once, sending four packets is necessary for 99% DRR. Sending only three packets would have resulted in 95% DRR. Sending two packets or a single packet would have resulted in 86% or 63% reception rates respectively. As the environment reduces in contention, the adapting device follows that

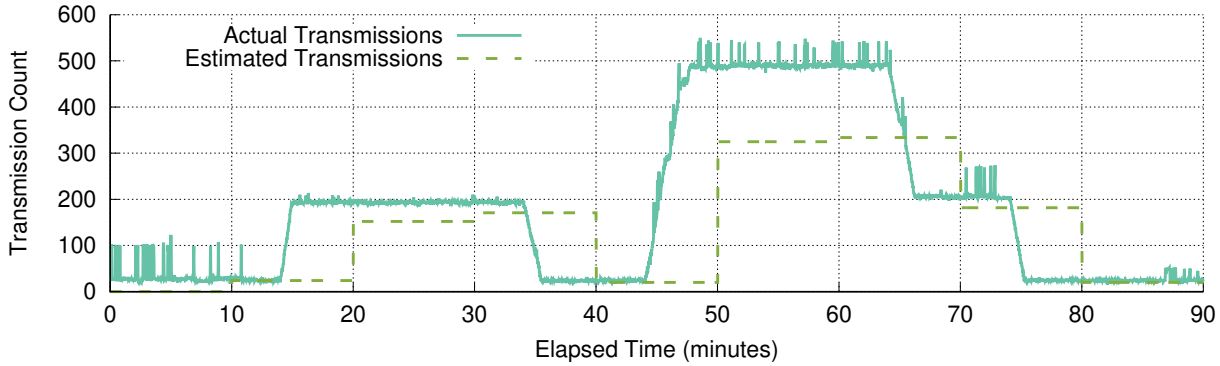

Figure 4.11: Actual and estimated transmissions during adaptation experiment. Actual transmissions are measured with a BLE gateway. Estimated transmissions are calculated by the adapting device every ten minutes based on the results of a one second BLE scan. Packet collisions will lead to invalid CRCs, which results in the packet being dropped rather than provided to the scanning device. This in turn results in an underestimate of the environment. With 500 total transmissions, this underestimate is as large as 35% error. For smaller transmission totals, collisions occur less frequently and the estimate is more accurate. Alternative scanning methods that receive packets with invalid CRCs may be necessary to support dense deployments.

as well, eventually returning to two transmissions per second at the end of the experiment.

The adapting device uses BLE scanning to receive nearby advertisements. This means that packets with invalid CRCs (due to collision for instance), are not provided to the device firmware. This leads to an underestimate of the actual transmission environment. Figure 4.11 demonstrates the actual versus estimated transmissions during the adaptation experiment. Note that the estimate is only calculated every ten minutes.

With a dense transmission environment, such as at 50 minutes into the experiment, the adapting device has 35% error in measuring the environment. This error could lead it to select less redundancy than necessary and ultimately reduce DRR. For very dense environments, reception of invalid packets may be necessary for reliable environment estimation.

4.4 Summary

Deployments of BLE devices illuminate two valuable takeaways about the future use of BLE advertisements for communication. First, that reliable communication is possible over advertisements. Second, that improvements to existing tools are needed to support this use case for advertisements. Here, we also propose future research directions that are revealed through this work. In Chapter 5 we turn our attention to the wide-area domain of wireless IoT communication.

Reliable communication is possible. Our deployments demonstrate that reliable data transfer over advertisements is possible, even in relatively crowded environments. Deployments can be statically configured in advance or can rely on runtime adaptation to meet reliability desires. In both cases, we demonstrate that 99% data reception rates are possible with 3–4 redundant transmissions. Combined with ease of use and the ability to communicate with smartphones and other personal devices, we believe this makes BLE a useful mechanism for many wireless sensing applications.

Improvements to tools are needed. In multiple cases, we discovered that the assumptions of various tooling were hampering the use of advertisements for scenarios other than discovery. Hardware and software for BLE communication should be capable of reliable packet reception, even with many transmissions per second. While it is possible to make a reasonably reliable scanner today, the obvious method of connecting a USB BLE dongle to a single-board Linux computer is not it.

Additionally, the assumptions of BLE libraries limit adaptation. An ideal measurement of transmissions would return all BLE traffic received, not just valid packets. This would allow a better estimate of the environment and better adaptation. While it is possible to use raw access to the hardware for microcontroller-based solutions, other systems such as smartphones are limited to the capabilities of the libraries they are provided. Removing artificial limitations on advertisement collection would better support additional uses, such as reliable communication.

Future research directions. One direction for future research is the improvement of reception rates in multi-gateway scenarios. Research from the LPWAN domain such as Choir [71] have the potential to be applicable here. Can packets which are only partially received by a single gateway be combined with data received by other gateways to reconstruct the data?

Another research direction is the combination of advertisements and connections for communication. In the general, periodic case data would be transmitted in advertisements. For rare data or infrequent large data uploads, a brief connection could be established to handle the transmission. What is the best protocol for managing multi-gateway deployments creating ephemeral connections with BLE devices? What are the energy costs of infrequent connections?

Chapter 5

LPWAN Background

The Internet of Things expands beyond the area of a single household and into city-scale deployments. To meet the communication needs of these devices, low-power wide-area networks (LPWANs) have been developed that promise communication ranges of kilometers with relatively low energy demands. We begin our exploration of LPWANs with an overview of what protocols exist in the space and what their capabilities and limitations are. A summary of protocols and their features is presented in Table 5.1. We will describe each of them, although the focus of this work is on Sigfox, LoRaWAN, and the cellular IoT protocols.

Network Technology	*Sigfox*	*LoRaWAN*	*Symphony Link*	*Weightless*	*802.11ah*	*LTE-M*	*NB-IoT*
Frequency Band	915 MHz	915 MHz	915 MHz	915 MHz	915 MHz	Licensed Cellular	Licensed Cellular
Signal Rate	0.6 kbps	1–22 kbps	0.2–38 kbps	0.6–100 kbps	100 kbps–347 Mbps	300–1000 kbps	27–65 kbps
Range	1–50 km	1–25 km	1–25 km	<10 km	<1 km	Several km	Several km
Modulation	DBPSK/GFSK	CSS	CSS	GMSK/OQPSK	BPSK/QPSK/16/64/256QAM	QPSK/16QAM	BPSK/QPSK
Access Scheme	Unslotted ALOHA	Unslotted ALOHA	Slotted ALOHA	TDMA/FDMA	TDMA slots with CSMA/CA	SC-FDMA/OFDMA	SC-FDMA/OFDMA
Active TX Power	50 mW	400 mW	400 mW	150 mW	Unknown	1800 mW	800 mW
Network Deployment	Managed	User or Managed	User	User or Managed	User	Managed	Managed
Protocol Standard	Proprietary	LoRa Alliance	Proprietary	Weightless SIG	IEEE	3GPP	3GPP

Table 5.1: Survey of LPWAN technologies. Each of these technologies provides low-bandwidth, low-power, long-range communications targeting IoT devices. The list includes unlicensed LPWANs as well as the cellular IoT protocols LTE-M and NB-IoT.

5.1 Unlicensed LPWANs

Low-power, wide-area network protocols, such as Sigfox [6] and LoRaWAN [5], utilize unlicensed communication bands to provide long-range communication. These technologies were the first to directly target large-scale machine-to-machine communications.

Not all unlicensed bands are the same. Different regions levy differing requirements on their use. In the US, technologies use the 915 MHz ISM band (902–928 MHz). This band requires frequency hopping across channels, with a maximum dwell time of 400 ms. There is no limitation on total usage though. In Europe, devices use channels around 868 MHz. Here the requirements focus on a limitation of duty cycle. Depending on the exact channel, the maximum transmission duty cycle of each device is limited to 10%, 1%, or 0.1%. In general, the European unlicensed band has far fewer channels and is more limited than the unlicensed bands in the US. Other regions have similar requirements, but vary in the specifics. The protocols we discuss here are therefore each families of protocols, with specific implementations for each region. We will focus on the capabilities of this protocols in the US unlicensed bands.

5.1.1 Sigfox

A major player in the unlicensed LPWAN space is Sigfox [6], a proprietary standard targeted at infrastructure monitoring. Sigfox utilizes ALOHA-style transmissions [11] from devices in the network which can be received by any gateway—any device may transmit whenever it has data. Radios are left off for the majority of the time, with messages sent to the device received during a brief listening window following each transmission. A 14 dBm Sigfox transmission draws 100 mW, which drops to 150 µW in idle mode [72].

Sigfox utilizes narrowband transmissions that trade bitrate for distance to an extreme extent. The physical layer bitrate for Sigfox is 600 bps in the US, and the protocol retransmits each packet on two additional channels to increase reliability [6]. To meet dwell time requirements with a limited bitrate, packets are correspondingly small. The maximum uplink payload size is 12 bytes, and the maximum downlink payload size is 8 bytes.

There is no particular limitation on the number of Sigfox devices connected to a single network. A single gateway is capable of receiving transmissions sent simultaneously by up to 270 devices [73], but in combination with an unslotted ALOHA MAC layer, Sigfox is only capable of about 4 kbps of goodput across an entire network.

Unlike many other unlicensed networks, Sigfox is solely a proprietary network. Sigfox does not support arbitrary users deploying their own networks. Instead partnered network operators deploy connectivity in various regions [74]. In turn, academic research focusing on the network has been limited, with almost no work that solely utilizes Sigfox. Commercial deployments continue, however, and as of 2020 Sigfox networks are deployed in most major cities in the US and in large regions surrounding Los Angeles and Chicago.

Sigfox's particularly low bandwidth targets limited data applications, such as gas and electric usage metering. The focus on a predominantly uplink network combined with slow

throughput makes firmware updates likely impossible.

5.1.2 LoRaWAN

LoRaWAN [5], another popular LPWAN protocol, is an open network standard built on top of the proprietary LoRa chirp-spread-spectrum physical layer. LoRaWAN, like Sigfox, uses an ALOHA access control model. Also like Sigfox, rather than solely communicating with a single gateway, devices broadcast data which can be received by any gateway on the network. Each transmission is followed by two listening windows which can contain an acknowledgement or any other downlink destined for the device. While transmitting at 20 dBm, a LoRaWAN transceiver draws about 400 mW, but otherwise it can remain in an idle state indefinitely in which it draws approximately 5 µW [75].

In the US, LoRaWAN transmissions hop across 64 channels and additionally select a spreading factor from five possibilities, allowing devices to trade off range and throughput. A single LoRaWAN device using the data rate with the most throughput (data rate three: 125 kHz bandwidth with spreading factor 7) is capable of about 5 kbps of goodput. Packet size varies based on the data rate (due to maximum dwell time constraints), but has a maximum of 256 bytes for upload or download.

Gateways are capable of receiving packets from different channels or spreading factors simultaneously if the gateway has a sufficient number of decoders. In practice, even if a gateway monitors all 64 channels it would only be capable of decoding at most 64 packets simultaneously. LoRaWAN's access control strategy of unslotted ALOHA reduces total capacity to 18% of maximum channel capacity in the optimal case [11], resulting in a little less than 60 kbps throughput for an entire network. Anyone who purchases a LoRaWAN gateway can operate their own network, but LoRaWAN is capable of managed network deployments as well, and several operators have deployed networks [76, 77].

LoRaWAN is an open standard managed by a nonprofit association, the LoRa Alliance. Combined with accessible transceivers and unlicensed network deployment, this has made LoRaWAN particularly attractive for academic research. A number of papers have been published recently that improve LoRaWAN gateways [71, 78], propose alternative access control methods for LoRaWAN [79, 80], or simulate LoRaWAN networks [81, 82]. One particularly active area of research involves combining LoRa and backscatter techniques to improve range of backscatter technologies [83–86].

LoRaWAN targets city-scale sensing applications such as parking enforcement and gas/-electric use metering. The ability to deploy user-controlled networks also lends towards use cases in agricultural and industrial settings, where a network can be deployed to service a controlled area.

5.1.3 Other protocols

The domain of unlicensed LPWANs is still a growing space. A number of competitors have risen and are continuing to mature. While they do not yet command the academic and

commercial interest of Sigfox and LoRaWAN, they present interesting selections of different configuration points in the unlicensed space.

Symphony Link Similar to LoRaWAN, Symphony Link is a proprietary network protocol built on top of the LoRa physical layer. It allows for long-range, low-bandwidth communications, on the unlicensed 915 MHz ISM band in the US. Using the same physical layer as LoRaWAN, it has a similar range and bit rate depending on configuration. Connections are made to a network, rather than an individual gateway, also similar to many existing LPWANs.

Unlike Sigfox and LoRaWAN, however, Symphony Link manages access control with a slotted ALOHA scheme. Downlink messages are sent at periodic intervals during which devices are listening and are followed by uplink windows which all devices can choose to contend for. Downlink messages contain acknowledgements for any messages from the previous interval that a gateway received. The maximum payload size for uplink and downlink packets is 256 bytes.

Weightless Weightless is a series of open protocols for long-range, low-bandwidth communications developed by Weightless-SIG. The Weightless-P is a protocol for bidirectional unlicensed-band communication. All communication in this protocol goes through gateways, with devices making connections to a single gateway in the network at a time. Weightless-P divides the spectrum into 64 channels and uses both channel and time division to schedule transmissions. The gateway broadcasts schedule information, allocating time slices and channel hopping sequences to devices for uplink and downlink.

The specification also includes Weightless-W which uses TV whitespace bands and Weightless-N which uses narrowband communications (similar to Sigfox). We find Weightless to be particularly interesting as it implements several of the recommendations we suggest in Section 7.2. Academic and commercial interest in Weightless, however, has lagged, as has the availability of hardware.

802.11ah The IEEE has also standardized an unlicensed-band LPWAN as part of the WiFi family: 802.11ah, branded HaLow. It focuses on shorter ranges, hundreds of meters, but also operates in the 915 MHz unlicensed band in the US. It is not interoperable with traditional WiFi, and requires new hardware to support the modified frequency band and modulation schemes.

Like traditional WiFi, 802.11ah devices connect to a single access point with no direct communication. Up to 8191 devices can connect to a single access point, which divides devices into groups and schedules their transmission slots with TDMA. Within a transmission slot, several devices contend for data transfer using traditional CSMA/CA techniques. Packets sent over 802.11ah use traditional WiFi security solutions, namely WPA2 and TLS. Expected bit rates range from hundreds of kilobytes per second to hundreds of megabytes per second.

802.11ah targets applications where machine-to-machine communication is needed, but WiFi is either too high power or too complicated to suffice. Smart home applications are particularly interesting. Like Weightless, HaLow adopts a more mature approach and energy-intensive approach to managing its devices in order to provide more throughput and reliability. However, lack of real-world hardware remains a large barrier to use in deployments.

5.2 Cellular Networks

Cellular technologies are also important to the LPWAN story. GPRS, part of 2G GSM, was a popular network for machine-to-machine communications before reaching end-of-life in the US, and is still popular in the rest of the world. After a standards lag, new cellular networks are now available that target IoT use cases by design. These technologies offer an interesting comparison to unlicensed-band LPWANs and they have begun to see adoption by the research community. Due to their use of licensed frequency bands, cellular networks cannot be deployed by arbitrary users, but rather must be managed networks run by telecommunications companies.

A challenge that remains for these managed networks is proper market pricing. Networks have been rolled out by Verizon, T-Mobile, and AT&T throughout the US [87–89], but pricing models vary wildly. Especially for exploratory deployments, costs need to be acceptable for both small and large deployments in order for cellular networks to meet application needs.

5.2.1 GPRS

The General Packet Radio Service (GPRS) as a part of 2G serves as waypoint from which long-range machine-to-machine communications began. GPRS is capable of providing payload uplink of up to 80 kbps for common class 12 modems. Combined with a large number of channels (124 for the US 900 MHz band and 374 for the US 1900 MHz band) this allows it to theoretically provide a network throughput of 30 Mbps shared among all devices in a cell. GPRS does not provide the same kind of low-energy operation that modern LPWANs do, however, as it requires frequent paging responses to stay connected to a network, which results in high average power. As an example, the SIM800H GPRS modem draws about 4 mW in its lowest power connected mode and over 1 W while transmitting at maximum rate [90].

5.2.2 LTE-M

LTE-M is a recent LTE protocol that targets machine communications, standardized in 3GPP release 13. Also known as LTE Cat-M, LTE Cat-M1, or eMTC, it provides communication through traditional cellular networks but with reduced power needs and throughput capability. LTE-M's maximum bandwidth is lower than traditional LTE protocols (1.4 MHz

vs 5–20 MHz), but it can otherwise coexist in the same band as traditional LTE categories. The range of LTE-M is improved over prior cellular technologies with a maximum coupling loss (link budget) of 156 dB [91] compared to 144 for GPRS and normal LTE, resulting in a several kilometer range.

Devices communicate directly to nearby cell towers, rather than with each other. Taking into account various protocol overheads, a single half-duplex LTE-M device is capable of sending payload data at a rate up to 375 kbps. For a 20 MHz bandwidth network, up to 16 devices could transmit simultaneously for a total network throughput of 6 Mbps.

One large difference of LTE-M compared to traditional LTE communication is the ability to enter power saving modes that enable a device to stay connected to the network for long durations (minutes to days) without activating its radio to send keep-alives or receive packets. While the power use during communication is high—1400 mW peak during transmissions and 380 mW while maintaining a connection—these power saving modes allow the total energy use to remain low with sleep modes of 30 uW [92].

5.2.3 NB-IoT

The other new LTE machine-to-machine protocol, LTE-M, supports even longer range and smaller bandwidth use at a cost of even lower throughput. A single device in an NB-IoT network is capable of a payload upload rate of 62.5 kbps while using 200 kHz of bandwidth. Maximum coupling loss is further increased to 164 dB resulting in a range that could extend beyond 10 km.

An advantage to operators is that NB-IoT is narrow enough to be deployed in the guard band at the edge of cellular channel allocations. This allows operators to support IoT needs without impacting their existing bandwidth allocations and is how T-Mobile has deployed NB-IoT throughout the US [89]. Assuming a single guard band deployment, the network throughput is equivalent to the single device throughput for NB-IoT. NB-IoT supports similar power saving modes as LTE-M, reaping similar energy savings when not in use.

5.3 Summary

In this chapter we have created a classification for LPWANs. This separation of unlicensed-band communication and cellular technologies will inform our thinking as we model capabilities. Next, Chapter 6 creates a metric that unifies the requirements of pervasive applications and the capabilities of wide-area networks.

Chapter 6

Modeling Wide-Area Communication

With the goal of exploring how well LPWANs can serve application needs, we set out to create models for understanding networks and applications. We first determine the throughput, range, and power characteristics of each network. Next, we introduce a new metric, bit flux, which unifies range and throughput for networks and applications. We apply this metric to both previously described networks and to a newly presented set of pervasive applications that cover a wide range of general application requirements.

6.1 Network Characteristics

Of particular interest for this study are the range, throughput, and energy costs of each network. We group maximum network throughput and range together, as they will eventually combine to create a bit flux value for the network we discuss in Section 6.2.

All of these measurements are theoretical rather than experimental. In practice, we are assuming more throughput capability and far more range than will be realized in real-world deployments. For LTE technologies in particular, range and throughput are traded to maintain connection with poor signal quality. Predicted upper bounds are still useful, however, in understanding capabilities. While deployments may or may not meet the desires of applications on the edge of serviceability, they certainly will not succeed for desires outside the upper bound of the network capabilities.

6.1.1 Throughput and Range

Figure 6.1 plots maximum throughput versus maximum range for the networks we describe. 2G GPRS is plotted as context for the capabilities of other networks. We select Sigfox and LoRaWAN as popular protocols within the unlicensed LPWAN space. Ranges span from 2–12 km while total network throughput spans 4–30,000 kbps.

We determine network throughput as the total uplink payload bits per second provided across many devices connected to a single gateway. While the maximum throughput for

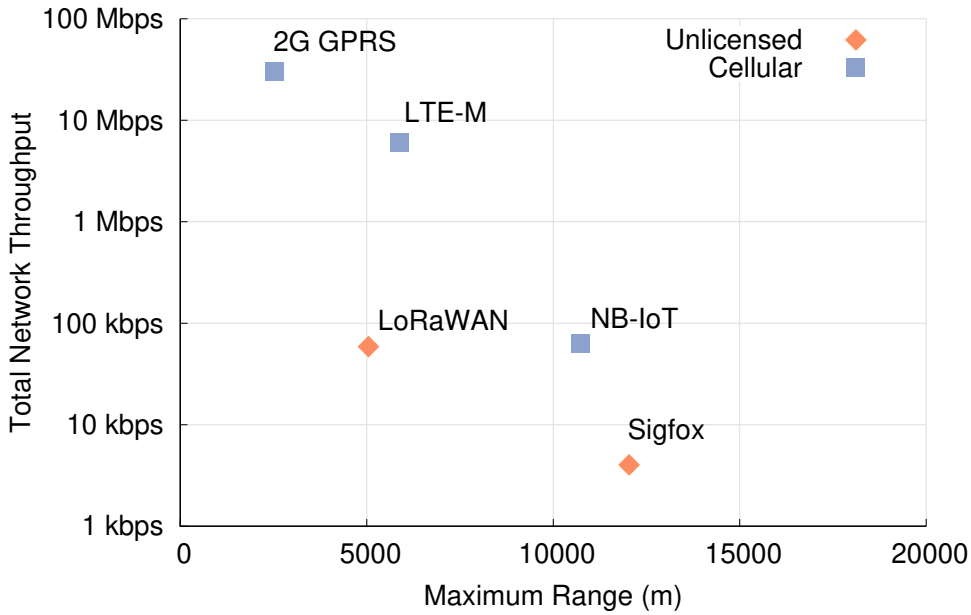

Figure 6.1: Range and network throughput for several IoT network technologies. Maximum range is estimated from uplink path loss using the Hata model [93]. Network throughput is the uplink payload bitrate shared by all devices connected to a single gateway, accounting for access control overhead. While all emphasize long range and low throughput, each network technology has different capabilities based on its particular protocol choices.

a single device in a network is important, for the kinds of large-scale, machine-to-machine applications we describe individual devices do not stress the network. Instead it is the deployment of many devices, each individually with small throughput needs, that cumulatively can exceed the network capacity. Calculating network throughput begins with the maximum goodput for a single device, increased by the number of devices a single gateway can concurrently support and reduced by the cost of contention with other deployed devices in the same network.

To determine maximum range, we start by determining the maximum path loss for each protocol given transmit power and receiver sensitivity for existing hardware. Transmit distance is then estimated using the Hata model [93] for protocols at 915 MHz and the Hata model PCS extension [94] for cellular protocols operating at 1900 MHz. In both cases the models are configured for medium-sized cities with end device and gateway heights of 1 m and 100 m respectively.

6.1.2 Power Comparison

While our evaluation of networks focuses on throughput and range, power is a first-order concern for the networks and applications we consider. Many IoT devices are battery-operated, with communication a large drain on their limited energy supply. A typical measurement

of communication energy use is bits per joule, but that metric functions particularly poorly for comparing protocols with very different amounts of overhead. While the cost of each bit transmitted over LTE protocols is quite low, the energy spent reestablishing a network connection upon wakeup from sleep needs to be accounted for as well for a fair comparison to protocols like LoRaWAN which do not have such connection overhead.

Instead, we compare protocols by presenting average power for sample application workloads: spanning 84 bytes per hour to 1000 bytes per day. This data requirement is on the low side for LTE-M, which would be more efficient with larger payloads. Meanwhile, this requirement is high for Sigfox, which must fragment the payloads across many packets. We believe this application workload is sufficient for at least providing a sense of the power tradeoff between these networks.

Estimating average power allows us to account for both the transmission and the additional overhead of maintaining the communication protocol. Calculating average power for Sigfox and LoRaWAN is straightforward and we do so based on datasheet numbers for existing hardware [72, 75]. For LTE-M and NB-IoT, we present numbers based on prior literature, which models expected power draw based on expected latencies at multiple total path loss choices [95]. We present the numbers for both a relatively good connection (144 dB total path loss) and at maximum range (164 dB). For reference, the maximum path loss for Sigfox is 155 dB while LoRaWAN (at data rate three) has a maximum path loss of 143 dB. Table 6.1 presents the average power for several applications ranging from one 84 byte payload every hour to one 200 byte payload every day. LoRaWAN has the lowest overall average power for each application case, around an order of magnitude better than LTE-M or NB-IoT in a good connection and two orders of magnitude better than the cellular technologies at maximum range.

6.2 Network Bit Flux

To understand how well a network can support pervasively deployed applications, we develop a new metric, *bit flux*, which measures a network's throughput over its coverage area. Specifically, we measure bit flux in units of bit per hour per square meter.

$$bit\ flux = \frac{network\ throughput}{coverage\ area} = \frac{bit/hour}{m^2} \tag{6.1}$$

This measure, which is the two dimensional version of a metric first proposed by Mark Weiser [96], is valuable because it considers how much capacity an application would require from shared networking infrastructure over a large geographical region. Importantly, *this metric captures both the capabilities of networks and the requirements of applications.* A network that provides a higher bit flux than the application requires is capable of serving the connectivity needs of that deployment.

Bit flux is more useful than network throughput alone for deployments with multiple gateways. Just comparing network throughput to application data rate is sufficient for deter-

Network Technology	Average Power (uW)			
	84 Bytes Per 1 Hour	84 Bytes Per 4 Hours	200 Bytes Per 24 Hours	1000 Bytes Per 24 Hours
Sigfox (155 dB)	110	29	11	56
LoRaWAN (143 dB)	12	3.0	1.1	5.1
LTE-M (144 dB)	50	25	12	13
LTE-M (164 dB)	2200	620	150	440
NB-IoT (144 dB)	62	22	13	15
NB-IoT (164 dB)	1800	520	100	240

Table 6.1: Average power for each network across example application demands. Expected power is presented for cellular protocols both with good connectivity (144 dB) and at maximum range (164 dB), while Sigfox and LoRaWAN are measured only at their maximum ranges. Application demands span from 84 Bytes each hour to 200 Bytes each day. LoRaWAN performs the best in all application cases, around an order of magnitude better than the cellular protocols in good connectivity. Sigfox must fragment payloads across many packets for all application examples, resulting in higher average power. The additional costs of more complicated physical layers and access control mechanisms lead to an increased power draw for the cellular protocols, particularly when at maximum range. NB-IoT performs better than LTE-M at maximum range, but both perform similarly otherwise.

mining whether a single gateway can support a deployment within its coverage area. Once an application spans the deployment regions of multiple gateways, however, this sufficiency analysis becomes difficult because the capability of each gateway needs to be individually compared to the needs of the devices deployed within its coverage area. By looking at throughput capabilities and needs averaged over an area, bit flux can be used to compare the needs of applications of any size to the capabilities of networks with any number of gateways, as long as the deployments of each are relatively homogeneous in density.

Bit flux also accounts for networks that take advantage of spatial reuse. Reducing gateway range increases network capacity by allowing for more concurrent transmissions at the cost of more deployed gateways. This concept is a common method for increasing cellular network capacity, and many LPWANs have some capability for power control to support it. Because bit flux accounts for coverage area, networks with the ability to shrink coverage area can increase their bit flux accordingly. This means that when applied to networks, bit flux does not result in a single value but a function.

Figure 6.2 demonstrates the increase in bit flux for long-range networks as maximum range decreases. For each network, there is both a maximum and a minimum range that can be achieved based on the maximum and minimum transmit power of existing hardware. As shown, reducing range for a network has the capability of improving its bit flux by several orders of magnitude. Due to much higher throughput, LTE-M has a higher bit flux,

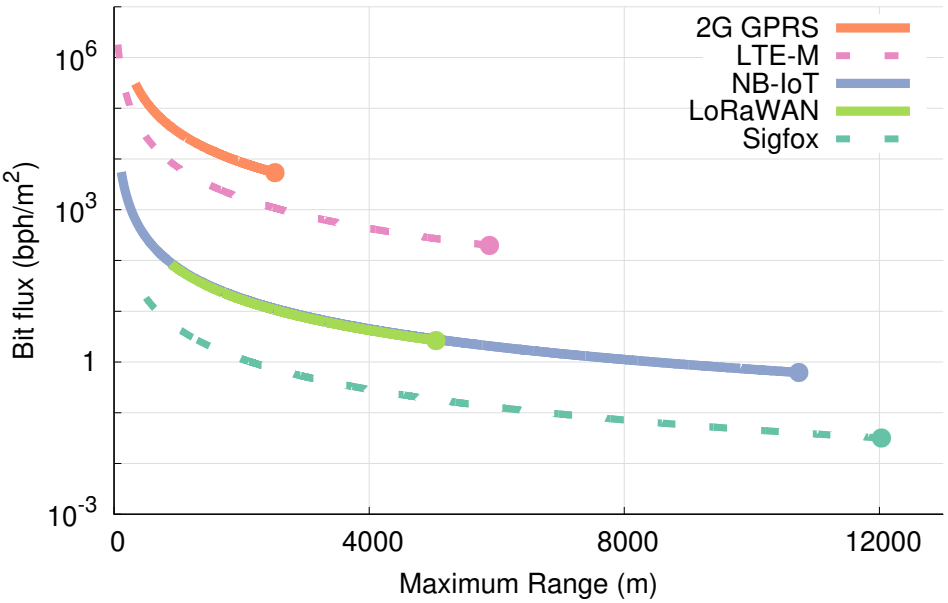

Figure 6.2: Throughput per unit area (bit flux) as range is varied through power control. Plotted are the bit per hour per square meter for each of the unlicensed-band and cellular LPWANs we discuss. Using power control, networks can reduce their coverage area, increasing their bit flux and allowing them to satisfy the needs of more applications at the cost of the deployment of additional gateways. The minimum and maximum ranges are limited to the power options found in existing hardware for each technology.

even at maximum range, than Sigfox or LoRaWAN at their minimum range. Additionally, LoRaWAN offers a subset of the capabilities of NB-IoT, which has a larger range of power control configurations.

An important limitation of bit flux is that it only measures technical capability, rather than feasibility. While a LoRaWAN network has similar throughput to local area networks such as 802.15.4, LoRaWAN by default covers a much larger area and therefore would provide a significantly lower bit flux. While short-range networks like 802.15.4 could service the throughput needs of city-scale deployments, the need to deploy many gateways or use high-power transmitters would likely make them unrealistic networking choices.

An additional problem lies in the amount of bandwidth required to support an application. A network may be able to support the throughput needs of an application but be doing so only by utilizing the entire bandwidth of the frequency band it occupies. In this case, no other networks could coexist within that same frequency band and geographic location. We explore network feasibility, in terms of number of gateways and bandwidth usage in Section 7.1.

Application	Single Location Throughput (bps)	Single Location Radius (m)	Pervasive Bit Flux ($\frac{bph}{m^2}$)
Zebranet [97]	53	75	0
Trash can monitoring [98]	0.38	370	0.003
Hospital clinic [99]	11	20	0.02
Volcano monitoring [100]	520	1,500	0.2
CitySee [101, 102]	20,400	5,700	1
Electricity metering [103, 104]	51,389	6,180	1.5
Habitat monitoring [105]	10	10	9
H1N1 [106]	18,000	60	43
IMT-2020 [107, 108]	35,556	564	128
Macroscope [109]	12	4	221
GreenOrbs [102, 110]	5,600	80	1,000

Table 6.2: Throughput, radius, and bit flux of sensing applications published in past sensor networking proceedings and the IMT-2020 standard [107]. The single location metrics show the requirements to deploy an instance of each application, while the pervasive metric assumes that the application is deployed at scale in its target environment. With throughput and bit flux spanning many orders of magnitude, these applications impose highly varying requirements on their underlying networks. While many networking technologies may meet the throughput requirements of a single application, they often do not have the capacity to support one or more of these applications at scale.

6.3 Pervasive Applications

The "Internet of Things" describes a wide and diverse range of applications. To understand and quantify their networking requirements, we survey notable application papers from the sensor networking literature, and consider their networking requirements in two deployment scenarios. The first, **single location** case, assumes the application is deployed to the fullest extent in a single location. We report the throughput and range required to support these deployments by multiplying the number of nodes in the deployment by the amount of data per measurement by the sampling interval.

A single instance of an application is often not consistent with the ubiquity targeted by the IoT. Therefore, we also consider the **pervasive** case for each application, which assumes that the application is scaled to be fully deployed in its target environment. For example, while a single location case may describe an application that monitors a single building, the pervasive case would including monitoring for all buildings of that type throughout a city. The applications vary tremendously in deployment area, so we employ the bit flux metric to compare them in terms of bits per hour per square meter. The networking requirements for the eleven applications we survey are shown in Table 6.2, and are described below, along with the assumptions for their pervasive deployments.

- **Zebranet** [97] is one of the earliest sensor network research deployments. It places GPS tracking collars on zebras that asynchronously send location data over a wide-area network. The incredibly low density of wild Grevy's Zebras results in near zero bit flux over a wide area, with peak throughput coming from monitoring all zebras in a large herd [111].

- **Trash Can Monitoring** [98] reports when trash cans are full in a deployment of 197 monitored trash cans throughout New York City's Times Square. Each trash can reports approximately twice a day, and we assume the same frequency and density for a pervasive deployment.

- **Hospital clinic** [99] measures patient vital signs in a 32 bed hospital clinic in St. Louis, USA. At scale all patients in the 2915 hospital beds in St. Louse would be monitored [112].

- **Volcano monitoring** [100] senses seismic tremors across 16 devices on Reventador volcano in Ecuador, streaming data when an event is detected. The pervasive case covers a volcanic area at the same sensor density.

- **CitySee** [101, 102] measures air quality from 1196 devices deployed in Wuxi, China, and we assume the same sensor density for a pervasive deployment.

- **Electricity metering** [103, 104] in San Francisco, USA. Approximately 370,000 smart meters throughout the city report 250 byte readings once every four hours.

- **Habitat monitoring** [105] measures microclimate and occupancy of bird burrows with 32 sensors on Great Duck Island off the coast of Maine, USA. A pervasive deployment would monitor the estimated 5000 Storm Petrel nests on Great Duck Island with 7500 sensors [113].

- **H1N1** [106] measures a single-day human contact graph of 850 people for modeling flu epidemiology in a school in San Francisco, USA. A full deployment would measure interactions for the 80,000 students in San Francisco [114].

- **IMT-2020** [107, 108] defines performance characteristics of 5G technologies. For machine-type communications it defines a connection density of one million devices per km^2 each transmitting a 32 byte packet every two hours.

- **Macroscope** [109] monitors the microclimate of a redwood tree with 33 sensors on a tree in Sonoma, USA. A full deployment would place sensors on all trees in an old-growth forest, at a density of about 20 trees/acre [115].

- **GreenOrbs** [102, 110] measures ecological data from 330 devices in a forest near Tianmu Mountain in China. We assume pervasive deployment at the same sensor density.

The eleven applications differ by many orders of magnitude in their throughput and bit flux requirements. Applications that cover a dense phenomenon, such as the redwoods Macroscope, have a relatively high bit flux even with low throughput, while applications that cover a large area (CitySee) or measure a sparse phenomenon have a low bit flux despite their high throughput. In terms of bit flux, the applications fall into two major categories. Sparse environmental or human monitoring require one bit per hour per square meter or less (Zebranet, trash can monitoring, volcano monitoring, CitySee, electricity metering, and hospital clinic). Denser monitoring (habitat monitoring, H1N1, Macroscope, GreenOrbs, and IMT-2020) requires one to several orders of magnitude more bit flux.

6.4 Summary

Bit flux is capable of measuring both networks and applications. Powered by our models of network throughput and range, any network with a higher bit flux than required by an application will technically be capable of serving that application's needs. However, the ability to theoretically serve an application does not mean that doing so in a real-world setting would be reasonable. Chapter 7 explores combinations of networks and applications to determine where problems will likely arise and to understand possible solutions to those problems.

Chapter 7

LPWAN Capabilities

Armed with the ability to model the capabilities of networks and the requirements of applications, we investigate how well they match up. First, we dive deeper into two case studies to see how well they are served by LPWANs. Identifying challenges for unlicensed LPWANs, we also explore possible modifications that could improve capabilities. Through the application of our models, we can determine capabilities and limitations of networks prior to significant deployments.

7.1 Network Suitability

To satisfy an application a network must provide equal or greater bit flux than the bit flux of the application in a pervasive deployment. This assumes a uniform distribution of application devices as well as a uniform distribution of gateways. If a network does not meet the bit flux needs at its maximum range, it can reduce range and increase the number of gateways deployed to increase capacity. For each application, Table 7.1 displays which networks could serve its data needs for the pervasive case. For the Zebranet application, all networks we investigate suffice due to the low data rate over an extremely wide area. Similarly, all networks can handle the load of the trash can monitoring application because its data rate is so low. On the other side, GreenOrbs can only be satisfied by the cellular IoT solutions due to the density of deployed sensors.

However, the capability to serve the needs of an application does not mean that doing so would be reasonable. The open circles on Table 7.1 denote circumstances where over 20% of a network's bit flux capacity would be spent on a single application or where reduced range would be required to meet the bit flux needs. Using a majority of total network capacity means using the majority of the bandwidth in the frequency allocation that network occupies. For example, saturating the capabilities of a LoRaWAN network means saturating the throughput of all 64 LoRaWAN channels, a significant portion of the 915 MHz ISM band. This is not a realistic scenario for wide-area deployments in urban locales.

Application	Sigfox	LoRaWAN	NB-IoT	LTE-M	2G GPRS
Zebranet [97]	●	●	●	●	●
Trash can monitoring [98]	●	●	●	●	●
Hospital clinic [99]	○	●	●	●	●
Volcano monitoring [100]	○	●	○	●	●
CitySee [101, 102]	○	○	○	●	●
Electric metering [103, 104]	○	○	○	●	●
Habitat monitoring [105]	○	○	○	●	●
H1N1 [106]		○	○	○	●
IMT-2020 [107, 108]			○	○	●
Macroscope [109]			○	○	●
GreenOrbs [102, 110]			○	○	●

Table 7.1: Sufficiency of a networking technologies to meet the pervasive bit flux requirements of each application. A circle indicates sufficiency, however an open circle indicates that range reduction is required for suitability, where suitability is defined as providing greater than five times the bit flux required for each application. The degrees of range reduction required to meet these cases varies significantly. For instance LTE-M can easily meet 5× the capacity of the IMT-2020 standard with greater than 4000 m range, however NB-IoT must reduce its range to less than 1000 m to provide this same capacity.

7.1.1 H1N1 Case Study

To put this idea into more concrete numbers, we first take a deeper dive into the H1N1 application. In the pervasive example, we imagine the application deployed throughout the city of San Francisco. Figure 7.1 displays the bit flux measurements for each network and the H1N1 application. The application is plotted as a horizontal line at 43 bph/m^2. Any network above the line is capable of meeting application needs. LTE-M and 2G GPRS are capable of meeting application needs at any range. LoRaWAN and NB-IoT must reduce range and deploy additional gateways in order to serve this application. Sigfox, hurt by limited throughput, is incapable of serving the H1N1 application at any range.

Figure 7.2 shows for each network the number of gateways that would be necessary to serve its throughput needs. The number of gateways is plotted as a line rather than a point because the network could vary its range through power control, changing the number of gateways necessary for coverage. This is plotted against the proportion of total network capacity the application would be using for gateways deployed at that density with optimal power control. LoRaWAN, for instance, could serve the H1N1 application throughout all of San Francisco with only 24 gateways, but it would use all of its capacity for this application

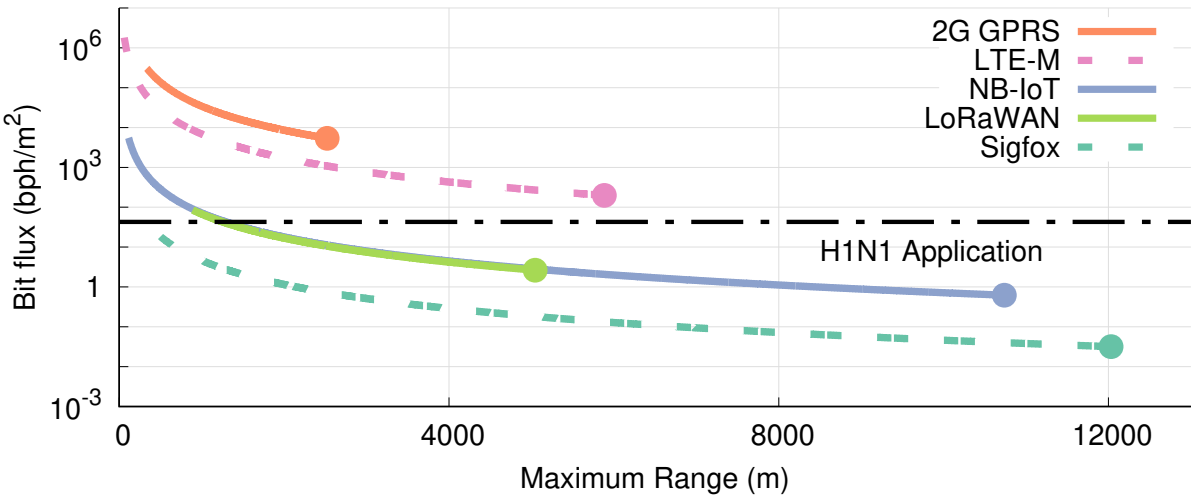

Figure 7.1: Bit flux for networks and the H1N1 application. LTE-M satisfies application needs, but only while devoting most of the network to the application. NB-IoT and LoRaWAN are capable of satisfying application needs with range reduction and the deployment of additional gateways. Sigfox is incapable of meeting the needs of the H1N1 application under any configuration.

alone. Even with the greatest reduction in range, LoRaWAN would still use 50% of its capacity for the H1N1 application. In practice, this would dedicate a significant chunk of the ISM band towards this application alone.

Another case where capability does not equate to reasonableness is in terms of the number of gateways required to cover an application deployment. For example, NB-IoT could cover the H1N1 application case while only using 1% of its network bit flux, however, doing so would require the deployment of 2000 gateways throughout San Francisco. While this deployment size is not totally implausible based on new femtocell efforts [116], sufficient motivating applications would need to exist before a service provider would invest in such a dense deployment.

The resulting range of the network after power control should be used as a final consideration for a reasonable deployment. Several of the networks we describe are capable of reducing end device power until the resulting range is only several hundred meters. In these situations, a deployment of WLAN technologies, such as WiFi or 802.15.4 should be considered instead of an LPWAN as they can provide much greater throughput at lower energy budgets.

7.1.2 Electricity Metering Case Study

We also take a similar deep dive into the electricity metering application. In this application, each household electric meter sends approximately 250 bytes every four hours. While this is a

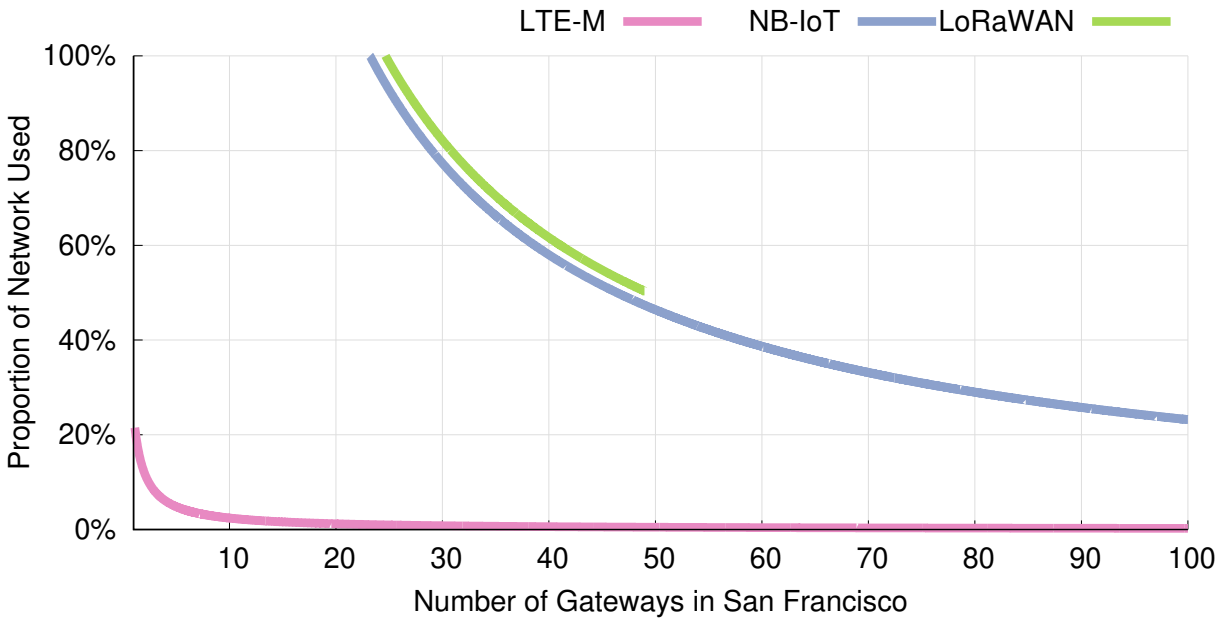

Figure 7.2: The proportion of the network capacity used by the H1N1 application for varying gateway density. As shown in Figure 6.2 and Table 7.1, networks can increase bit flux through power control to service certain applications at the cost of a decrease in range and a subsequent increase in gateway deployment density. LTE-M networks can service the application throughout San Francisco, USA (120 km^2) with only a few gateways and a small proportion of their total network capacity. LoRaWAN and NB-IoT can also serve the application, but only by allocating a significant proportion of their capacity to it or deploying many gateways.

rather low data rate, it is made up for with scale. The deployment area of this application is all of San Francisco, which has approximately 370,000 deployed electricity meters, resulting in a bit flux of 1.5 bph/m^2.

As this application has a lower bit flux requirement, more networks are capable of servicing it. LoRaWAN, LTE-M, and 2G GPRS all meet its needs without range reduction. NB-IoT can do so with modest reduction, while Sigfox needs significant range reduction but is still capable of meeting the required throughput, as displayed in Figure 7.3.

When viewed as a matter of gateways and proportion of network usage in Figure 7.4, we see an interesting story. To reach a reasonable percentage of the network devoted to this application, Sigfox must deploy dozens of gateways. LoRaWAN could cover the city with only two or three gateways, but doing so would devote 50% of the ISM band towards the electricity metering application. NB-IoT has a near identical tradeoff. LTE-M has an easier go of it, with less than 1% of network capacity spent on this application even with only one or two gateways.

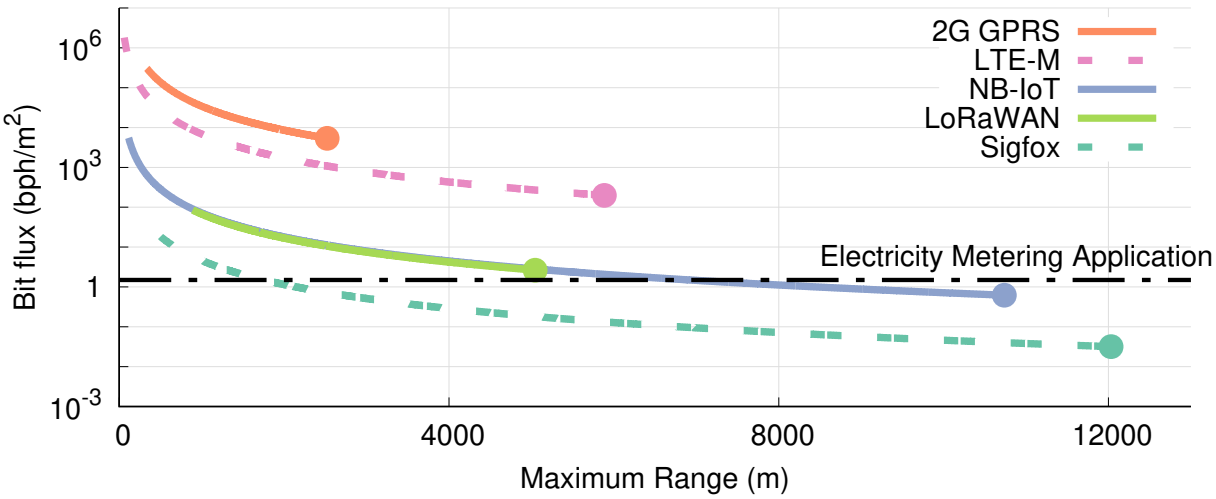

Figure 7.3: Bit flux for networks and the electricity metering application. Networks above the application requirement line, such as 2G GPRS, LTE-M, and LoRaWAN meet its requirements without modification. NB-IoT and Sigfox are also capable of servicing this application, but require a range reduction.

7.1.3 Are LPWANs Sufficient?

The first takeaway from this analysis is the relative success of cellular IoT technologies. NB-IoT and LTE-M could, at least conceptually, meet the needs of every application we describe. Their high throughput and wide coverage areas allow them to cover the needs of a low-throughput applications with a single deployment, but also affords them the opportunity to over-provision gateways to handle high-throughput applications. In the H1N1 case, LTE-M could cover all of San Francisco with a single gateway using 22% of the capacity of that entire band. Or, it could cover all of San Francisco with 24 gateways and only allocate 1% of the band capacity to that single application.

Even though their use for pervasive applications seems promising from a capacity standpoint, cellular technologies do have their own challenges, the most notable being fees to access the network and the higher average power requirements discussed in Section 6.1.2. Some IoT devices may have flexibility in their design to accommodate an increased energy demand in trade for network performance and reliability, but this will not be true for all applications.

For unlicensed LPWANs, we find two challenges in network suitability. The first is an issue of *capacity*. For several of the applications we describe, LoRaWAN deployments are unable to transport the data necessary even at minimum range. Even when LoRaWAN and Sigfox can meet the bit flux requirements of an application, they only do so with a dense deployment of gateways using a majority of the unlicensed bandwidth available. To handle pervasive application needs, unlicensed-band LPWANs will need to increase their capacity. This problem is primarily one of implementation. The selection of the ALOHA access control

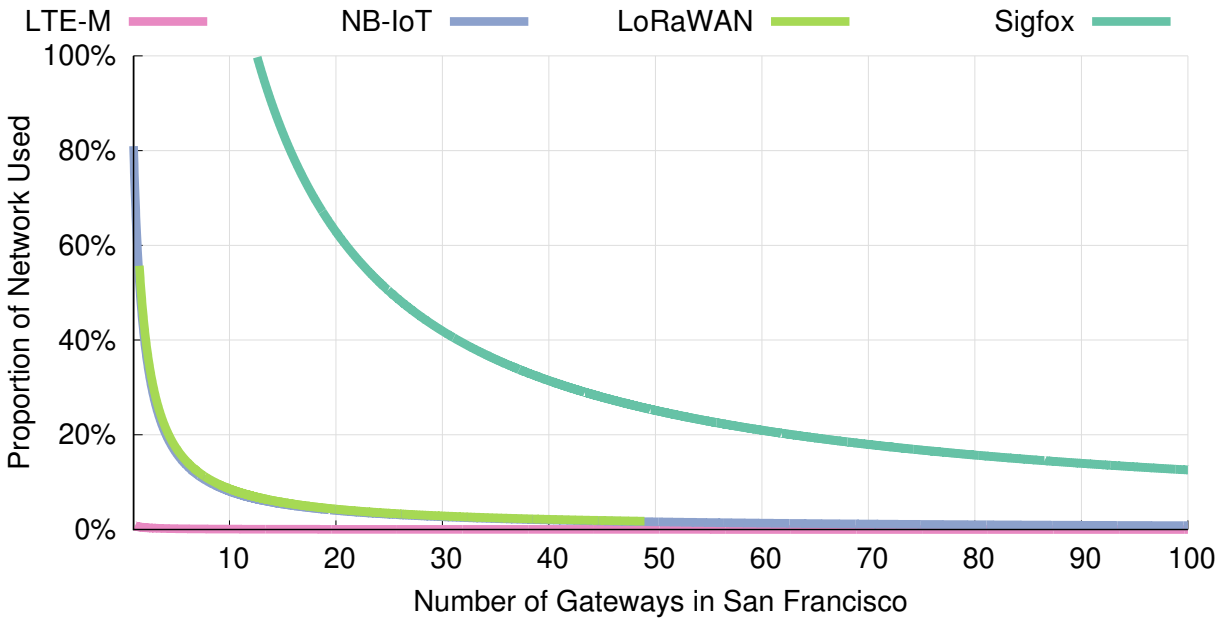

Figure 7.4: The proportion of the network capacity used by the electricity metering application for varying gateway density. LoRaWAN and NB-IoT have similar tradeoffs, where a single gateway could cover the entire deployment region but would require the devotion of more than half of the network's throughput. They may also deploy additional gateways, with the deployment instead of ten gateways utilizing less than 10% of network capability.

mechanism, for instance, greatly reduces network throughput.

The second challenge, as demonstrated by the high percentage of bit flux needed to satisfy some applications, is one of *coexistence*. A network cannot assume that it is deployed in isolation. Especially in the context of city-scale deployments, many networks will be running in the same physical region. For long-term success, technologies making use of the unlicensed band are going to need to share it, either by using so little of the band that multiple networks can naturally coexist or by actively coexisting with other networks. This problem is a fundamental one for long-range networks in unlicensed bands.

These two problems, capacity and coexistence, do not mean LoRaWAN or other unlicensed-band, long-range technologies are unsuitable for all applications; with little contention, sensing deployments in remote areas with sufficiently low bit flux are well-provided for by unlicensed LPWANs. However, to succeed for pervasive applications in urban areas, solutions to these challenges will be necessary.

7.2 Network Solutions

If capacity of the unlicensed band and the networks that use it is not sufficient to provide for the desired applications, contention within a network and between networks, can lead to

poor throughput and unpredictable reliability. A number of solutions have been proposed to increase the capacity of individual LPWAN networks, some of which would also decrease the impact of collisions with coexisting networks. Researchers have also proposed active coordination between networks and widening the unlicensed band.

This section enumerates these techniques. We focus on LoRaWAN as the majority of research projects target it, but the resulting techniques are applicable to many unlicensed-band technologies. Weightless [117], for example, already utilizes TDMA and FDMA techniques for managing access control.

7.2.1 Improving Transmission

Modifying LoRaWAN's access control mechanism could result in greatly increased capacity for a single network, although it would not greatly increase the ability of a network to coexist with other networks beyond the decrease in total channel usage. Polonelli et al. describe a method for layering ALOHA slots on top of the existing LoRaWAN protocol [79]. They utilize acknowledgements for device synchronization with the gateway similar to Symphony Link.

Alternatively, channel access can be explicitly scheduled. Trüb et al. demonstrate two possible TDMA schemes that could be employed for LoRaWAN systems which could improve network throughput to 60%, or three times that of unslotted ALOHA [80]. Neither work measures the increase in energy cost for implementing such schemes. For dense networks which would experience many packet collisions under ALOHA-style access control, however, the energy cost for scheduling may be lower than the cost of repeated transmissions. Access control changes would need modifications to software on both gateways and devices, a challenge for existing deployments.

7.2.2 Resilient Reception

Methods for better receiving packets in the presence of noise not only increase the capacity of a network, but also increase resiliency to the presence of coexisting networks.

DaRe [118] performs convolutional erasure coding on LoRaWAN application-layer data such that a lost packet can be recovered from other packets. Application layer coding may be one method for increasing data reception rates while requiring software changes to the gateways and end devices. Applying a code rate to payloads would increase energy use as on-air time increases.

Choir [71] leverages radio imperfections in frequency, time, and phase to simultaneously receive several transmissions. While Choir suggests that this could enable as much as a 30× increase in network capacity, NetScatter finds that this technique would enable no more than 5-10 simultaneous transmitters [86]. Charm [78] uses coherent combining to increase reception rates for transmissions weak in signal strength. This could also serve to increase resiliency to collisions, however we do not know the exact magnitude of this improvement. Deployment of systems like Choir and Charm require modifications to gateway hardware and

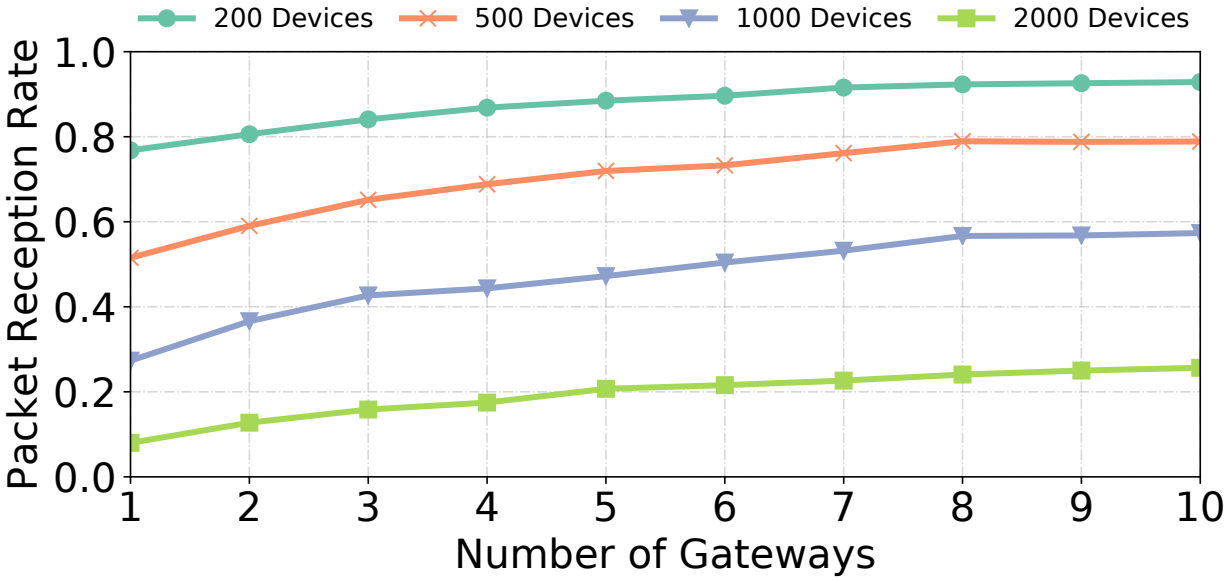

Figure 7.5: Increased deployment of gateways results in higher packet reception rate due to the capture effect. Shown is the reception rate for packets sent by 100 devices on the target network. As the total number of deployed devices, most not on the network, increases, collisions cause packets to be lost. Increasing the number of gateways deployed throughout the same area results in more packets received as some overcome collisions due to the capture effect.

software, but can be deployed without modifying existing devices and without any increase in energy.

We may also rely on the capture effect to improve reception rates. This causes a packet with stronger signal strength to be received despite a collision with a weaker signal, and can be achieved by densely deploying gateways without reducing the transmit power of end devices. To evaluate this technique, we simulate a deployment of LoRaWAN devices coexisting on a single channel using a modified version of LoRaSIM [81]. In our simulation, devices and gateways are deployed randomly across a 5 km region and devices send a 20 byte packet once per minute on average. As shown in Figure 7.5, when a network of a single gateway and 100 devices is deployed alongside 1000 devices on another network, the gateway receives 27% of transmissions, however when 10 gateways are deployed in the same scenario, the reception rate increases to a 57% due to the capture effect.

7.2.3 Increasing Bandwidth

An increase to the amount of bandwidth usable by unlicensed systems would result in increased capacity for all networks operating in the band. There has been progress towards making additional unlicensed spectrum available in the US, specifically TV bands around

600 MHz [119]. The amount of TV white space bandwidth available for unlicensed use depends on local channel usage, which varies widely based on location and population density. Still, one simulation finds that an average of 80 MHz could be available in cities in the US [120], triple the 26 MHz currently allocated to the 900 MHz ISM band.

While existing LPWAN transceivers have support for some of the TV whitespace frequencies, firmware changes on both devices and gateways would be necessary to exploit this hardware. Additionally, protocols would need to adopt the ability to determine which TV channels are available in a particular deployment area.

7.2.4 Coexisting through Coordination

Ultimately unlicensed-band collisions are inevitable between networks, and all the more inevitable due to the number of stakeholders present within the range of an LPWAN. If the underlying capacity of the band is not enough to meet the needs of its users, some form of coordination may be the only hope of increasing capacity and creating predictable and fair performance.

This coordination could be done in both the frequency and time domains. Some protocols, such as WiFi, divide the frequency domain with protocol-specified limits of single-network bandwidth, ensuring some minimum number of networks can coexist. This technique is more difficult for unlicensed LPWANs, which may have to coexist with many more networks due to their range, and would additionally further limit their throughput.

Coordination becomes simpler for managed networks run by only a few service providers. A limited number of network operators could provide communication in the unlicensed bands, such as is occurring with Sigfox. A dominant regional provider could result in de facto ownership of the band, as other smaller network deployments would have to work around them. Removing the option for anyone to put up a gateway and create their own network would be a loss of some of the value of the unlicensed bands, however.

Techniques to enable coordination through inter-network communication have also been considered. De Poorter et al. present a design for an LPWAN management framework that includes cross-network and cross-technology optimization [121], however it requires synchronization of heterogeneous devices to enable layered TDMA schemes. WiSHFUL demonstrates a similar scheme, which changes access control mechanisms to reduce cross-technology interference [122].

To be successful, coordination needs buy-in from the majority users of network capacity. Regulations, such as the 1% per-device duty cycle limits in the EU are one mechanism to enforce this coordination, however per-network rather than per-device limits may be necessary to prevent takeover by a deployment of many devices.

7.3 Summary

Our investigation of LPWANs has illuminated problems for unlicensed LPWANs. They face challenges in both capacity and coexistence. Capacity issues may already be solved, if modifications from the research world are pulled into the protocols. Coexistence problems remain unsolved and represent a valuable domain for future research. In contrast, cellular IoT networks suffer from neither of these concerns, positioning them to possibly dominate the LPWAN domain. Here, we summarize the problems facing unlicensed and cellular LPWANs along with possible solutions. We also present future research directions.

Capacity problems in unlicensed LPWANs can be addressed. Capacity issues are a purely technical one. To minimize energy use and come to market quickly, Sigfox and LoRaWAN kept to simple design choices such as ALOHA access control. Research solutions already approach capacity issues from a number of directions. The remaining difficulty, and it is a challenging one, is to decide which are most important and most palatable. Are additional energy costs for a scheduled medium too burdensome for battery-powered sensor? Is the addition of TV white spaces acceptable for US deployments?

A combination of existing research could suffice for improving network capacity. Reviewing the techniques we have described, simple TDMA mechanisms could increase capacity by 3×, simultaneous reception by 5×, coherent combining by 1-2×, and increasing bandwidth to the predicted TV white space availability by 4×. Together these implementation changes might therefore generate a two order-of-magnitude improvement to LoRaWAN capacity, pushing it to be on par with LTE-M in capacity (and probably looking a lot like an LTE protocol in design).

Few of these techniques are free. Many, particularly modifications to access control mechanisms, would increase the energy cost of communication. One of the biggest strengths of existing unlicensed band networks like LoRaWAN is how low power they are, but the reality is that additional energy costs will need to be paid in order to provide connectivity for higher-throughput applications in urban areas. Exactly where in the tradeoff space between capability and energy cost future networking technologies ought to fall is still unclear, but we believe the question is worthy of exploration.

Coexistence concerns for unlicensed LPWANs are more challenging to address. Unfortunately, capacity solutions alone would be insufficient to solve unlicensed LPWAN problems. Coexistence solutions, which licensed-band cellular technologies can ignore, will also be necessary. Especially in urban areas, the range of LPWANs mean that multiple stakeholders will necessarily overlap. Cellular technologies avoid this through management of which devices are allowed on their network and how much traffic they can send. Too many stakeholders may overwhelm cellular capacity, but the result will be a total failure for some stakeholders rather than a partial failure for everyone.

The coordination mechanisms necessary to overcome co-existence problems are less clear and less studied in prior work than capacity issues. An additional concern is that buy-in

from all deployed networks is necessary. One bad actor could cause a source of interference for all coordinating networks.

One approach is to avoid coexistence problems altogether. In industrial or agricultural deployments, it is quite possible that a little power control could limit the network to covering an area solely owned by a single stakeholder. This may be a successful domain for LoRaWAN networks deployed with the current protocol.

Cellular IoT technologies are well situated to serve pervasive applications. While they were much slower in development, LTE-M and NB-IoT are now deployed throughout the US and many other regions. LTE-M particularly has a much higher capacity than unlicensed LPWANs like Sigfox or LoRaWAN, which allows it to successfully serve more applications. NB-IoT plays an interesting role of longer range and lower throughput. Due to their managed nature and deployment on licensed bands, cellular networks avoid the issues of coexistence, allowing market forces to decide which applications can use their frequency resources.

Two concerns that could hold cellular IoT back are power and access fees. Transmission remains relatively high power, possibly too high for some battery-based systems to source. Sleep periods can lead to low-energy operation, but only with the selection of higher latency as the radio is off for minutes or hours at a time. Finally, for deployments to choose cellular IoT technologies, access costs for both small and large deployments must be reasonable. What this means in terms of dollars is still very undecided, and it is unclear if network providers will adapt to deployment needs.

Future research directions. One direction of future research is the creation of protocols for enabling reliable coordination of shared mediums, like unlicensed bands. How can such a design ensure good performance even without complete buy-in from all users of the band? What are the energy costs of coordination mechanisms?

Deployments of cellular IoT devices to understand their performance in real-world scenarios is another strong direction for future research. How should devices best leverage high-power communication to provide low-energy, high-latency communication? How does the tradeoff of throughput and range perform in different environments? Can indoor deployments be augmented with cellular connections?

Chapter 8

Conclusion

In this work, we have explored the capabilities and weaknesses of several recent IoT-focused networks. We have demonstrated that modeling aspects of these networks allows us to gain insights into their performance prior to large-scale deployments.

With BLE advertisements, we have demonstrated that simple models of packet collisions can accurately predict network performance for real-world deployments. We have used these models to investigate the potential impacts of modifications to the protocol and explore possible configurations for real-world applications. We have also demonstrated that using models for BLE advertisement collisions, devices can automatically adapt their configurations to network conditions, ensuring reliability with minimal wasted energy.

BLE advertisements are frequently used in development because the enable simple communication with smartphones. This work shows that reliable communication can be built on top of these simple advertisements. This supports the development-to-deployment cycle by allowing the same protocol to be used in both testing and real-world scenarios. Advertisements also provide benefits by allowing simultaneous reception by any number of gateways and smartphones in range, a redundancy win. As we show, however, tooling to support advertisements for communication is still lacking. The simple, straightforward method of gateway construction leads to significant packet loss. With modifications, and future engineering effort, BLE advertisements can be leveraged for reliable deployments.

We then turned our attention towards the domain of city-scale sensing. Our new metric, bit flux, combines data throughput and range to model the capabilities of networks and requirements of applications. We use this metric to investigate unlicensed-band and cellular networks that target machine-to-machine use cases. We first demonstrate issues with existing networks and then explore potential modifications that could improve them.

For unlicensed communication like Sigfox and LoRaWAN, we demonstrate that the protocols as existing today support too little throughput for many wide-scale applications. This is a design choice. Recent research has demonstrated a variety of methods for improving network capacity An additional problem is that the unlicensed band must by necessity be shared by any deployments wishing to use it. With kilometers of range and urban deployments, this means many networks will need to coordinate their use of the band. This

is a fundamental problem for unlicensed communication, and it is one worthy of additional research. Without solutions, deployments will find that their expected capacity is greatly reduced in the real world. This may lead to an exodus to cellular technologies, which now support the requirements of machine-to-machine communication with LTE-M and NB-IoT. We hope the research community will rise to this challenge to support city-scale applications on unlicensed bands.

Through our investigation of low-energy wireless networks, we have presented methods for predicting network performance at both the local and wide area scales. We have shown that these models can predict successful use cases and potential failure points for networks, enabling successful deployments and demonstrating the importance of improvements to the protocols. We hope this work will allow researchers to better navigate the tradeoffs available to them and guide protocol designs that will enable the next generation of Internet of Things devices.

Bibliography

[1] Nest. *Nest Create a Connected Home*. 2019.

[2] Bluetooth SIG. *Bluetooth Core Specification 4.2*. 2014.

[3] ZigBee Alliance. *Zigbee-2006 specification*. http://www.zigbee.org/. 2006.

[4] Thread Group. *Thread Specification 1.1.1*. 2017.

[5] LoRa Alliance. "LoRaWAN 1.1 Specification". In: (2017).

[6] JC. Zuniga and B. Ponsard. *Sigfox System Description*. Internet Requests for Comments. 2017.

[7] Roy Want, Bill N Schilit, and Scott Jenson. "Enabling the internet of things". In: *Computer* 48.1 (2015), pp. 28–35.

[8] Bin Yu, Lisheng Xu, and Yongxu Li. "Bluetooth low energy (BLE) based mobile electrocardiogram monitoring system". In: *Information and Automation (ICIA), 2012 International Conference on*. ICIA'12. IEEE. 2012, pp. 763–767.

[9] F. Lin et al. "A Self-Powering Wireless Environment Monitoring System Using Soil Energy". In: *IEEE Sensors Journal* 15.7 (July 2015), pp. 3751–3758. ISSN: 1530-437X. DOI: 10.1109/JSEN.2015.2398845.

[10] Ramsey Faragher and Robert Harle. "Location fingerprinting with bluetooth low energy beacons". In: *IEEE journal on Selected Areas in Communications* 33.11 (2015), pp. 2418–2428.

[11] Norman Abramson. "The ALOHA System: Another Alternative for Computer Communications". In: *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*. ACM. 1970, pp. 281–285.

[12] Wha Sook Jeon, Made Harta Dwijaksara, and Dong Geun Jeong. "Performance Analysis of Neighbor Discovery Process in Bluetooth Low-Energy Networks". In: *IEEE Trans. on Vehicular Technology* 66.2 (2017), pp. 1865–1871.

[13] Christine Julien et al. "BLEnd: Practical continuous neighbor discovery for Bluetooth low energy". In: *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IPSN'17. ACM. 2017, pp. 105–116.

[14] Raphael Schrader et al. "Advertising power consumption of bluetooth low energy systems". In: *2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*. IEEE. 2016, pp. 62–68.

[15] Samuel DeBruin. "Enabling Visibility Into Building Energy Consumption Through Novel Metering Designs and Methods". PhD thesis. University of Michigan, 2017.

[16] Samuel DeBruin et al. "PowerBlade: A Low-Profile, True-Power, Plug-Through Energy Meter". In: *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. SenSys'15. Seoul, Republic of Korea: ACM, 2015. DOI: 10.1145/2809695.2809716. URL: http://doi.acm.org/10.1145/2809695.2809716.

[17] National Science Foundation. *NSF commits more than $60 million to Smart Cities Initiative*. 2016.

[18] European Commission. *Using EU funding mechanism for Smart Cities*. 2013.

[19] China Business Review. *Smart City Development in China*. 2014.

[20] Joshua Adkins et al. "The Signpost Platform for City-Scale Sensing". In: *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IPSN'18. New York, NY, USA: ACM, Apr. 2018.

[21] Charlie Mydlarz, Justin Salamon, and Juan Pablo Bello. "The implementation of low-cost urban acoustic monitoring devices". In: *Applied Acoustics* 117 (2017), pp. 207–218.

[22] Charles E Catlett et al. "Array of things: a scientific research instrument in the public way: platform design and early lessons learned". In: *Proceedings of the 2nd International Workshop on Science of Smart City Operations and Platforms Engineering*. SCOPE'17. ACM. 2017, pp. 26–33.

[23] Yun Cheng et al. "AirCloud: a cloud-based air-quality monitoring system for everyone". In: *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. SenSys'14. ACM. 2014.

[24] A. Ledeczi et al. "Multiple simultaneous acoustic source localization in urban terrain". In: *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* IPSN'05. Apr. 2005, pp. 491–496. DOI: 10.1109/IPSN.2005.1440982.

[25] Branden Ghena et al. "Challenge: Unlicensed LPWANs Are Not Yet the Path to Ubiquitous Connectivity". In: *Proceedings of the 25th Annual International Conference on Mobile Computing and Networking*. MobiCom'19. Los Cabos, Mexico, Oct. 2019.

[26] Jia Liu et al. "Adaptive device discovery in bluetooth low energy networks". In: *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*. IEEE. 2013, pp. 1–5.

[27] Android. *Scan Settings*. https://developer.android.com/reference/android/bluetooth/le/ScanSettings.html. 2014.

[28] Joshua Adkins and Prabal Dutta. "Monoxalyze: Verifying Smoking Cessation with a Keychain-sized Carbon Monoxide Breathalyzer". In: *Proceedings of the 14th ACM Conference on Embedded Networked Sensor Systems*. SenSys'16. Stanford, CA, USA, Nov. 2016.

[29] Joakim Linde. *Advertising and scanning states in Bluetooth Low-Energy*. https://lists.apple.com/archives/bluetooth-dev/2012/Feb/msg00037.html. 2012.

[30] Apple Inc. *Accessory Design Guidelines for Apple Devices*. https://developer.apple.com/accessories/Accessory-Design-Guidelines.pdf. 2018.

[31] David Pérez-Diaz de Cerio et al. "Analytical and experimental performance evaluation of BLE neighbor discovery process including non-idealities of real chipsets". In: *Sensors* 17.3 (2017), p. 499.

[32] Nordic Semiconductor. *nRF52832 Product Specification—Radio Timing*. https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.nrf52832.ps.v1.1%2Fradio.html&cp=2_1_0_22_14_7&anchor=unique_1024511272. 2018.

[33] Bjorn Spockeli. *nRF52832 Scan Channel Switch Time*. https://devzone.nordicsemi.com/f/nordic-q-a/25553/nrf52832-scan-channel-switch-time/100747#100747. 2017.

[34] Sandeep Mistry. *Noble – Default Scan Window*. https://github.com/noble/noble/blob/2afa49a798e067d84970f97778a14aa07b986ad8/lib/hci-socket/hci.js#L267. 2018.

[35] Ángela Hernández-Solana et al. "Proposal and evaluation of BLE discovery process based on new features of Bluetooth 5.0". In: *Sensors* 17.9 (2017), p. 1988.

[36] Albert F Harris III et al. "Bluetooth Low Energy in Dense IoT Environments". In: *IEEE Communications Magazine* 54.12 (2016), pp. 30–36.

[37] Robin Kravets, Albert F Harris III, and Roy Want. "Beacon trains: blazing a trail through dense BLE environments". In: *Proceedings of the Eleventh ACM Workshop on Challenged Networks*. ACM. 2016, pp. 69–74.

[38] Sandeep Mistry. *Noble*. https://github.com/noble/noble#maximum-simultaneous-connections. 2015.

[39] Apache. *NimBLE, Mynewt's Bluetooth 5 compliant stack*. https://mynewt.apache.org/pages/ble/. 2018.

[40] Bluetooth SIG. *Bluetooth Core Specification 5.0*. 2016.

[41] Lohit Yerva et al. "Grafting Energy-harvesting Leaves Onto the Sensornet Tree". In: *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*. IPSN'12. Beijing, China: ACM, 2012, pp. 197–208. ISBN: 978-1-4503-1227-1. DOI: 10.1145/2185677.2185733. URL: http://doi.acm.org/10.1145/2185677.2185733.

[42] Brandon Lucia et al. "Intermittent Computing: Challenges and Opportunities". In: *SNAPL*. 2017.

[43] Google Inc. *The Physical Web*. https://google.github.io/physical-web/. June 2017.

[44] Apple Inc. *iBeacon - Apple Developer*. https://developer.apple.com/ibeacon/. 2013.

[45] Google Inc. *Eddystone*. https://github.com/google/eddystone. Apr. 2017.

[46] Tile Inc. *Find Your Keys, Wallet & Phone with Tile's App and Bluetooth Tracker Device*. https://www.thetileapp.com. 2018.

[47] Apple Inc. *iCloud - Find My - Apple*. https://www.apple.com/icloud/find-my/. 2020.

[48] Philipp Bolliger. "Redpin-adaptive, zero-configuration indoor localization through user collaboration". In: *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*. ACM. 2008, pp. 55–60.

[49] Yifei Jiang et al. "Ariel: Automatic wi-fi based room fingerprinting for indoor localization". In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM. 2012, pp. 441–450.

[50] Giorgio Conte et al. "BlueSentinel: a first approach using iBeacon for an energy efficient occupancy detection system". In: *1st ACM International Conference on Embedded Systems For Energy-Efficient Buildings (BuildSys)*. 2014.

[51] Apple Inc. *macOS - Continuity - Apple*. https://www.apple.com/macos/continuity/. 2019.

[52] Jeremy Martin et al. "Handoff all your privacy–a review of apple's bluetooth low energy continuity protocol". In: *Proceedings on Privacy Enhancing Technologies* 2019.4 (2019), pp. 34–53.

[53] Guillaume Celosia and Mathieu Cunche. "Discontinued Privacy: Personal Data Leaks in Apple Bluetooth-Low-Energy Continuity Protocols". In: *Proceedings on Privacy Enhancing Technologies* 2020.1 (2020), pp. 26–46.

[54] Apple Inc and Google Inc. *Exposure Notification Bluetooth Specification v1.2*. Apr. 2020.

[55] Joshua Adkins et al. "Demo: Michigan's IoT Toolkit". In: *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. SenSys'15. Soeul, Republic of Korea, Nov. 2015.

[56] Albert F Harris et al. "Smart LaBLEs: Proximity, Autoconfiguration, and a Constant Supply of Gatorade (TM)". In: *Edge Computing (SEC), IEEE/ACM Symposium on*. SEC'16. IEEE. 2016, pp. 142–154.

[57] Coen Roest. "Enabling the Chaos Networking Primitive on Bluetooth LE". MA thesis. Delft University of Technology, 2015.

[58] PrithviRaj Narendra, Simon Duquennoy, and Thiemo Voigt. "BLE and IEEE 802.15.4 in the IoT: Evaluation and Interoperability Considerations". In: *International Internet of Things Summit*. InterIoT. Springer. 2015, pp. 427–438.

[59] Mobashir Mohammad, XiangFa Guo, and Mun Choon Chan. "Oppcast: Exploiting Spatial and Channel Diversity for Robust Data Collection in Urban Environments". In: *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. Apr. 2016, pp. 1–12. DOI: 10.1109/IPSN.2016.7460681.

[60] Nordic Semiconductor. *nRF51822 Product Specification v3.1*. http://infocenter.nordicsemi.com/pdf/nRF51822_PS_v3.1.pdf. Oct. 2014.

[61] Frontline. *Frontline BPA low energy Bluetooth Protocol Analyzer*. http://www.fte.com/products/BPAlowenergy.aspx. 2012.

[62] Jia Liu et al. "Energy analysis of device discovery for bluetooth low energy". In: *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*. IEEE. 2013, pp. 1–5.

[63] Mohammad Ghamari et al. "Detailed Examination of a Packet Collision Model for Bluetooth Low Energy Advertising Mode". In: *IEEE Access* 6 (2018), pp. 46066–46073.

[64] Nordic Semiconductor. *nRF52832 Product Specification v1.4*. https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.4.pdf. Oct. 2017.

[65] NIST. *NIST TC4TL Challenge*. https://tc4tlchallenge.nist.gov/. 2020.

[66] BeagleBone.org Foundation. *BeagleBone Black*. https://beagleboard.org/black. 2014.

[67] Kinivo. *Kinivo BTD-400 Bluetooth 4.0 Low Energy USB Adapter*. https://kinivo.com/products/kinivo-btd-400-bluetooth-4-0-usb-adapter-for-windows-10-8-7-vista. 2014.

[68] Sandeep Mistry. *Noble*. https://github.com/noble/noble. 2015.

[69] Timothy W Hnat et al. "The hitchhiker's guide to successful residential sensing deployments". In: *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. SenSys'11. ACM. 2011, pp. 232–245.

[70] Linux Foundation. *Zephyr Project*. https://www.zephyrproject.org/. 2018.

[71] Rashad Eletreby et al. "Empowering Low-Power Wide Area Networks in Urban Settings". In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM'17. ACM. 2017, pp. 309–321.

[72] Atmel. *ATA8520E Datasheet*. 2016.

[73] Sigfox. *Number of orthogonal SigFox channels*. https://ask.sigfox.com/questions/3194/number-of-orthogonal-sigfox-channels.html. 2017.

[74] Sigfox. *Sigfox Operators*. https://partners.sigfox.com/companies/sigfox-operator. 2019.

[75] Semtech. *SX1276/77/78/79 Datasheet*. 2016.

[76] MachineQ. *Improving How the World Operates by Bridging the Physical and Digital Worlds - MachineQ*. https://machineq.com/. 2019.

[77] The Things Industries. *The Things Network*. https://www.thethingsnetwork.org/. 2019.

[78] Adwait Dongare et al. "Charm: Exploiting Geographical Diversity Through Coherent Combining in Low-Power Wide-Area Networks". In: *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE. 2018, pp. 60–71.

[79] Tommaso Polonelli, Davide Brunelli, and Luca Benini. "Slotted ALOHA Overlay on LoRaWAN-A Distributed Synchronization Approach". In: *2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC)*. IEEE. 2018, pp. 129–132.

[80] Roman Trüb and Lothar Thiele. "Increasing Throughput and Efficiency of LoRaWAN Class A". In: *UBICOMM 2018*. ThinkMind. 2018, pp. 54–64.

[81] Martin C. Bor et al. "Do LoRa Low-Power Wide-Area Networks Scale?" In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM. 2016, pp. 59–67.

[82] Floris Van den Abeele et al. "Scalability Analysis of Large-Scale LoRaWAN Networks in NS-3". In: *IEEE Internet of Things Journal* 4.6 (2017), pp. 2186–2198.

[83] Vamsi Talla et al. "Lora backscatter: Enabling the vision of ubiquitous connectivity". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (2017).

[84] Ambuj Varshney et al. "Lorea: A Backscatter Architecture that Achieves a Long Communication Range". In: *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. ACM. 2017, p. 18.

[85] Yao Peng et al. "PLoRa: A Passive Long-range Data Network from Ambient LoRa Transmissions". In: *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. ACM. 2018, pp. 147–160.

[86] Mehrdad Hessar, Ali Najafi, and Shyamnath Gollakota. "Netscatter: Enabling Large-scale Backscatter Networks". In: *Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation*. NSDI'19. Boston, MA, USA: USENIX Association, 2019, pp. 271–283. ISBN: 978-1-931971-49-2. URL: http://dl.acm.org/citation.cfm?id=3323234.3323258.

[87] Verizon Enterprise. *Transitioning your IoT Environment to 4G LTE*. http://www.verizonenterprise.com/products/internet-of-things/4g-lte-iot/next-gen/#ps-overlay. July 2018.

[88] AT&T Business. *LTE-M Network Solutions*. https://www.business.att.com/solutions/Service/internet-of-things/networks/lte-m/. 2018.

[89] T-Mobile Newsroom. *T-Mobile Launches Nation's First Plan for Narrowband IoT*. https://www.t-mobile.com/news/narrowband-iot. Jan. 2018.

[90] SIMCom. *SIM800H Hardware Design*. 2013.

[91] Mads Lauridsen et al. "Coverage and capacity analysis of LTE-M and NB-IoT in a rural area". In: *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. IEEE. 2016, pp. 1–5.

[92] ublox. *SARA-R4/N4 Series Datasheet*. 2019.

[93] Masaharu Hata. "Empirical formula for propagation loss in land mobile radio services". In: *IEEE transactions on Vehicular Technology* 29.3 (1980), pp. 317–325.

[94] Preben Elgaard Mogensen and J. Wigard. "COST Action 231: Digital Mobile Radio Towards Future Generation System, Final Report." English. In: *Section 5.2: On antenna and frequency diversity in GSM. Section 5.3: Capacity study of frequency hopping GSM network*. 1999.

[95] Pascal Jörke, Robert Falkenberg, and Christian Wietfeld. "Power Consumption Analysis of NB-IoT and eMTC in Challenging Smart City Environments". In: *2018 IEEE Globecom Workshops*. IEEE. 2018, pp. 1–6.

[96] Mark Weiser. "Some Computer Science Issues in Ubiquitous Computing". In: *Commun. ACM* (1993), pp. 75–84.

[97] Pei Zhang et al. "Hardware Design Experiences in ZebraNet". In: *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. 2004, pp. 227–238.

[98] Bigbelly, Inc. *New York City's Times Square Efficiently Manages 26,056 Gallons of Waste and Recycling Each Day with Bigbelly*. http://info.bigbelly.com/case-study/times-square-new-york-city. 2019.

[99] Octav Chipara et al. "Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit". In: *Proceedings of the 8th ACM conference on embedded networked sensor systems*. ACM. 2010, pp. 155–168.

[100] Geoff Werner-Allen et al. "Fidelity and yield in a volcano monitoring sensor network". In: *Proceedings of the 7th symposium on Operating systems design and implementation*. USENIX Association. 2006, pp. 381–396.

[101] X. Mao et al. "CitySee: Urban CO2 monitoring with sensors". In: *2012 Proceedings IEEE INFOCOM*. 2012, pp. 1611–1619.

[102] Xiaolong Zheng. *Personal Communication*. July 2018.

[103] Dominic Fracassa. *CleanPowerSF tripiling households served with municipal electricity*. https://www.sfchronicle.com/bayarea/article/CleanPowerSF-tripling-households-served-with-13618155.php. Feb. 2019.

[104] Pacific Gas and Electric Company. *EPIC 1.14 - Next Generation SmartMeter Telecom Network Functionalities*. Nov. 2016.

[105] Alan Mainwaring et al. "Wireless sensor networks for habitat monitoring". In: *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. Acm. 2002, pp. 88–97.

[106] Maria A Kazandjieva et al. "Experiences in measuring a human contact network for epidemiology research". In: *Proceedings of the 6th Workshop on Hot Topics in Embedded Networked Sensors*. ACM. 2010, p. 7.

[107] ITU-R. *Minimum Requirements Related to Technical Performance for IMT-2020 Radio Interface(s)*. Nov. 2017.

[108] ITU-R. *Guidelines for Evaluation of Radio Interface Technologies for IMT-2020*. Oct. 2017.

[109] Gilman Tolle et al. "A macroscope in the redwoods". In: *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM. 2005, pp. 51–63.

[110] Lufeng Mo et al. "Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest". In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. SenSys '09. 2009, pp. 99–112.

[111] P.D. Moehlman, D.I. Rubenstein, and F. Kebede. *The IUCN Red List of Threatened Species. Grevy's Zebra*. http://www.iucnredlist.org/details/7950/0#sectionPopulation. 2015.

[112] Missouri Department of Health and Senior Services. *MO Hospital Profiles By County*. https://health.mo.gov/safety/healthservregs/pdf/MOhospbyCounty.pdf. July 2018.

[113] College of the Atlantic. *Seabirds at Great Duck Island*. https://www.coa.edu/islands/great-duck-island/seabirds-at-gdi/. 2018.

[114] Lapkoff and Gobalet Demographic Research Inc. *Demographic Analyses and Enrollment Forecasts for the San Francisco Unified School District*. https://www.sfusd.edu/en/assets/sfusd-staff/enroll/files/DemographicReport3182010.pdf. Mar. 2010.

[115] Redwood National and State Parks. *Del Norte Coast Redwoods*. https://www.parks.ca.gov/pages/414/files/DelNorteSPFinalWebLayout2015.pdf. 2015.

[116] Takehiro Nakamura et al. "Trends in Small Cell Enhancements in LTE Advanced". In: *IEEE Communications Magazine* 51.2 (2013), pp. 98–105.

[117] Weightless SIG. "Weightless-P System Specification". In: (2015).

[118] Paul J. Marcelis, Vijay S. Rao, and R. Venkatesha Prasad. "DaRe: Data Recovery Through Application Layer Coding for LoRaWAN". In: *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation*. IoTDI'17. IEEE. 2017, pp. 97–108.

[119] Federal Communications Commission. *In the Matter of Unlicensed Operation in the TV Broadcast Bands, Additional Spectrum for Unlicensed Devices Below 900 MHz and in the 3 GHz Band*. Report and Order, FCC-08-260. Nov. 2008.

[120] Farzad Hessar and Sumit Roy. "Capacity Considerations for Secondary Networks in TV White Space". In: *IEEE Transactions on Mobile Computing* 14.9 (2015), pp. 1780–1793.

[121] Eli De Poorter et al. "Sub-GHz LPWAN Network Coexistence, Management and Virtualization: An Overview and Open Research Challenges". In: *Wireless Personal Communications* 95.1 (2017), pp. 187–213.

[122] Peter Ruckebusch et al. "Wishful: Enabling Coordination Solutions for Managing Heterogeneous Wireless Networks". In: *IEEE Communications Magazine* 55.9 (2017), pp. 118–125.