

2017

A Platform for Large-Scale Regional IoT Networks

Keyi Zhang

Bucknell University, kz005@bucknell.edu

Follow this and additional works at: https://digitalcommons.bucknell.edu/honors_theses

Recommended Citation

Zhang, Keyi, "A Platform for Large-Scale Regional IoT Networks" (2017). *Honors Theses*. 407.
https://digitalcommons.bucknell.edu/honors_theses/407

This Honors Thesis is brought to you for free and open access by the Student Theses at Bucknell Digital Commons. It has been accepted for inclusion in Honors Theses by an authorized administrator of Bucknell Digital Commons. For more information, please contact dcadmin@bucknell.edu.

**A PLATFORM FOR LARGE-SCALE REGIONAL IOT
NETWORKS**

by

Keyi Zhang

A Thesis

Presented to the Faculty of
Bucknell University


in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science with Honors in Computer Science

May 3, 2017

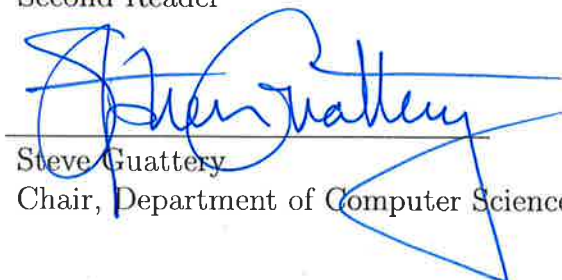
Approved:



Alan Marchiori
Thesis Advisor



Luiz Felipe Perrone
Second Reader



Steve Guattery
Chair, Department of Computer Science

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Prof. Alan Marchiori for the continuous support of my undergraduate study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I would also like to thank my friends for their encouragement and company through the years, although without them my research and thesis work could have finished much sooner.

Last I would like to thank my family for their continuous support.

Contents

Abstract	ix
1 Introduction	1
2 Background	4
2.1 Internet of Things and the Gateway Problem	4
2.2 Common Wireless Network Protocols	5
2.2.1 ALOHA	6
2.2.2 Time Division Multiple Access	7
2.2.3 Carrier Sense Multiple Access	8
2.2.4 Code Division Multiple access	9
2.3 Radio Communication and Properties	10
2.3.1 FCC Regulation	10
2.3.2 Capture Effect	11
2.4 Mathematical Models and Theorems	12

<i>CONTENTS</i>	iv
2.4.1 Bloom Filter	12
2.4.2 Queueing Theory	13
3 Related Works	14
3.1 State-of-Art Technology in Industry	14
3.2 Academic Approaches	17
4 Crowdsourcing IoT Networks	20
4.1 Crowdsourcing LPWAN Networks	20
4.2 PlanIt, Realistic LPWAN Planning	21
4.2.1 Selecting IoT Device Locations	21
4.2.2 Path Loss Estimation	25
5 Distributed Queueing with Multiple Data Slots	30
5.1 Distributed Queueing	30
5.2 DQ-N for csLPWANs	35
5.2.1 Simulation	38
5.3 Implementation	41
6 Conclusion	46
Appendices	52
A DQ-N Protocol Specification Version 0.27	53

A.1 Protocol Overview	53
A.1.1 Node Synchronization	54
A.1.2 Node Joining Process	54
A.1.3 Node Sending and Receiving Data	54
A.1.4 Protocol Sequence Example	55
A.2 Frame Details and Equation	55

List of Tables

4.1	ITWOM parameter values used to compute path loss.	27
5.1	Parameters used to calculate BER	39
5.2	Components and costs for single DQ-N base station and end device	42
A.1	MessageID values table for DQ-N version 0.27	56
A.2	TR structure for DQ-N version 0.27	57
A.3	Feedback structure for DQ-N version 0.27	57
A.4	TR-JOIN structure for DQ-N version 0.27	57
A.5	JOIN-REQ structure for DQ-N version 0.27	57
A.6	JOIN-RESP structure for DQ-N version 0.27	58
A.7	Message ID values for TR structure	58
A.8	Encoding details of frame parameters in feedback structure	58

List of Figures

2.1	The 7 layers of the OSI model.	6
2.2	Pure ALOHA protocol. Boxes indicate frames. Shaded boxes indicate frames which have collided.	7
2.3	Slotted ALOHA protocol. Boxes indicate frames. Shaded boxes indicate collided frames.	8
2.4	TDMA frame structure. The medium is shared by 3 devices and each communicates at designated slots.	9
4.1	Probability density map of $\rho(x, y)$, $\mathbf{G} \cdot \langle \frac{\partial g}{\partial x}, \frac{\partial g}{\partial y} \rangle$, and P_{IoT} . Notice the change of the map at bottom left corner due to geographic information. The right plot shows the 2500 IoT device locations generated based on P_{IoT}	23
4.2	The data from 1,000 test points generated in Philadelphia. Census tracts are shown in colored polygons and test points are shown in dots.	24
4.3	Census tract area and population distributions for the city of Philadelphia, Pennsylvania containing 381 census tracts.	25
4.4	The locations of 1,000 sample points generated for the city of Parker, Pennsylvania (red) and Armstrong County Census Tract 9503 (blue).	26

4.5	The path loss distribution from 100 rounds of evaluating 1,000 test points with randomly assigned base station nodes in two Pennsylvania cities. The dashed line at 158 dB indicates the threshold for reception.	28
4.6	The coverage rate for all cities in Pennsylvania with one base station and a receive threshold of 158 dB. The median connectivity, 99.2%, is indicated by a dashed line.	29
4.7	Screenshot of online PlanIt service.	29
5.1	DQ frame structure	31
5.2	Example of how DQ distributes the transmission bursts into following frames.	32
5.3	Burst contention resolution time, L_n	33
5.4	Comparison between M/D/1 and DQ.	35
5.5	DQ-N frame structure.	36
5.6	Simulation results for the ideal environment.	41
5.7	Simulation results for the realistic environment.	42
5.8	Radio duty cycle in the realistic environment.	43
5.9	Setup with 3 Adafruit Feather M0 RFM95 boards and 1 Raspberry Pi 3 with LoRa Radio Hat.	43
A.1	DQ-N protocol sequence diagram.	56

Abstract

The Internet of Things (IoT) promises to allow everyday objects to connect to the Internet and interact with users and other machines ubiquitously. Central to this vision is a pervasive wireless communication network connecting each end device. For individual IoT applications it is costly to deploy a dedicated network or connect to an existing cellular network, especially as these applications do not fully utilize the bandwidth provided by modern high speeds networks (e.g., WiFi, 4G LTE). On the other hand, decades of wireless research have produced numerous low-cost chip radios and effective networking stacks designed for short-range communication in the Industrial, Scientific and Medical Radio band (ISM band). In this thesis, we consider adapting this existing technology to construct shared regional low-powered networks using commercially available ISM band transceivers. To maximize network coverage, we focus on low-power wide-area wireless communication which enables links to reliably cover 10 km or more depending on terrain transmitting up to 1 Watt Equivalent Isotropically Radiated Power (EIRP). With potentially thousands of energy constrained IoT devices vying for extremely limited bandwidth, minimizing network coordination overhead and maximizing channel utility is essential. To address these challenges, we propose a distributed queueing (DQ) based MAC protocol, DQ-N. DQ-N exhibits excellent performance, supporting thousands of IoT devices from a single base station. In the future, these networks could accommodate a heterogeneous set of IoT applications, simplifying the IoT application development cycle, reducing total system cost, improving application reliability, and greatly enhancing the user experience.

Chapter 1

Introduction

We are living in a world where billions of devices are connected to the Internet, collecting an enormous amount of data every second. As the manufacturing cost for microcontrollers and sensors has been significantly reduced due to the advancement of technology resulting from Moore's Law, our surroundings are becoming more connected and smarter. Since most devices will not be directly wired to the Internet, wireless networks are the major platform for developing an extremely connected physical world. Therefore, improving the performance of the wireless networks is essential to develop a connected physical world.

As a result, different ideas have been proposed to categorize the usages, model the behavior, and ultimately profit from the blooming connectivity. Ideas such as the Internet of Things (IoT), Low-Power Wide-Area Network (LPWAN), ultra-narrow band transmission (discussed in Chapter 2 and 3) are among those early pioneering practices. These ideas have brought a new perspective to the problems of this ever-connected world, inspiring new technologies to provide a more efficient and secure platform for these connected devices.

Currently, industry companies are competing with each other to capture the emerging market. Different standards and protocols have been proposed and most of them are proprietary, such as those from SemTech™ and Ingenu™. These companies race to solve challenging problems such as Internet connectivity and low-power communications. They have built several regional networks either experimentally or in practice, trying to offer a better communication platform for connected devices.

However, if we take a step back and examine these standards and protocols, we can find and summarize several existing problems, namely the dilemma in balancing the network scale with a reasonable budget, difficulty in planning network services, and the usage of inefficiency protocols.

Setting up and maintaining a large-scale regional network is not easy and cost-effective. Taking traditional cellular networks as an example: it was estimated that three percent of the worlds annual electrical energy consumption is contributed by the infrastructure in 2011 [1]. Since the industry is pushing standards from 4G to 5G, the cost to maintain a faster network is higher. As cellular communication is generally battery consuming, low-power oriented technology has gained more favor for connected devices development [2]. Yet these new attempts fall into another pitfall that might prevent the network scaling up: unless these industrial companies have enough financial capability, it is difficult to set up a large-scale regional network comparable to traditional cellular networks. In the end, the dilemma turns out to be a question about how we can set up a cost-effective network that covers as much area as possible.

These newly connected devices are deployed around us as their original purpose is to make our life more convenient. As a result, their deployment cannot be oversimplified as a model where uniform distribution is used. Deploying more base stations to the areas where more devices are connected can be beneficial to alleviate high traffic load, and deploying fewer base stations to other areas can reduce cost without jeopardizing connectivity. However, there are not any conventional tools to our knowledge that can capture this human-centric information embedded in the device locations and use it in simulation prior to deployment. In other words, it is important for these industrial companies to capture market as fast as possible, yet deployment without reasonable simulation might also introduce extra cost for infrastructure maintenance.

After deployment, the communication protocol directly determines the overall efficiency of the network. Providing a generic protocol stack can be essential if the network wishes to support heterogeneous application domains. Using traditional protocols can significantly reduce development time and ensure its usability as they have been tested numerous times. However, one drawback is that these traditional protocols were designed in a different era where the design goal was slightly different, e.g. providing necessary wireless connectivity (ALOHA for instance) as compared to maximize the channel usage of low-power networks. Researchers have discovered inefficiency in protocols used in recently developed regional IoT networks as they are based on traditional-yet-inefficient protocols, for instance, LoRaWAN™ [3]. Such inefficiency will increase the infrastructure cost as more base stations are needed to

provide desired throughput and connectivity.

In this thesis, we will lay out the foundations for a novel large-scale regional IoT network. It aims to shed a light on the problems mentioned above and offer better solutions to these problems. Readers can find our take on how a large-scale IoT network should be built and how we address these problems. Additionally, the platform presented in this thesis is published as open-standard and open-sourced, as opposed to proprietary platform employed by the industry. This is not meant to be overwhelmingly inclusive, but to serve as a way to open discussion and inspire future research.

The structure of the thesis is as follows: in Chapter 2 we will give a general background for the following discussion; in Chapter 3 we will discuss related work; in Chapter 4 we will present our novel solution to coverage and simulation problem; in Chapter 5 we will present our improved design of DQ-based protocol.

Chapter 2

Background

This chapter is intended for readers who are not familiar with recent advancements in IoT technologies or wireless communication. It covers topics on the Internet of Things, conventional MAC layer protocols such as ALOHA, FCC regulations, capture effects, Bloom filter, and queueing theory. Readers who have a strong background in these topics can skip to Chapter 3.

2.1 Internet of Things and the Gateway Problem

The Internet of Things (IoT) is an emerging paradigm in which information and communication systems are embedded in our surroundings [4]. It is the concept of connecting devices such as coffee makers, washing machines, headphones, lamps, and wearable devices to the Internet. The application areas of IoT include intelligent healthcare, environmental monitoring, precision agriculture, and smart cities [5, 6, 7]. For instance, the smart lock is a pioneering IoT product which has drawn much public attention. One of its outstanding features is how it makes physical keys obsolete and instead uses your smartphone to lock or unlock your house. Homeowners can give visitors a guest pass and monitor who enters or leaves the house. The smart lock will also lock your house automatically when your smartphone leaves your property [8]. Smart watches can also take part in household activities. It can switch on an Internet-enabled coffee machine to brew coffee before playing your favorite music on an Internet-enabled speaker to wake you up [9, 10]. These demonstrations are just

a few samples of how IoT allows normal objects to communicate with the Internet and thus to interact with people and assist their daily routines.

Currently, there are 9 billion interconnected devices and the number is expected to reach 24 billion devices by 2020 [4]. These devices are typically supported by cloud-based infrastructure services providing data storage and data analysis [11]. Furthermore, the IoT ecosystem spans monitoring, storage, communication, and analytical tools, it is common practice to use clouds as a service for data storage and analysis [4]. However, such an approach poses a problem that is addressed in this thesis: the communication between IoT devices and the cloud is a challenge as the device must establish connection to a gateway when communicating with the cloud. This is also known as the “IoT Gateway Problem” [12]. For instance, many smart watches and personal fitness trackers rely on communication with smartphones via Bluetooth Low Energy (BLE). In order to synchronize personal data with the cloud, these devices have to be physically within the proximity of BLE, which has a range of approximately 100 meters. Therefore these devices are limited by the physical location of the smartphone. In other words, if the device is out of BLE proximity, no active communication can occur between the device and the cloud. In addition, the applications installed on users’ smartphones are vendor dependent and proprietary. For instance, Fitbit and Garmin have their own applications and protocols for data synchronization. Such heterogeneity introduces additional challenges for IoT developers as they need to design their own secured communication protocols in addition to the hardware design and manufacture. Although BLE has a desirable battery profile, long-term active communication between the IoT device and the smartphone still induces a considerable battery drain [13], especially when the smartphone pushes notifications to and synchronizes with the IoT device actively.

2.2 Common Wireless Network Protocols

As Internet protocols become more and more complicated, researchers have divided the functionality into protocol layers. Each layer operates independently and can be replaced without affecting other layers. One of the most common models to represent this layered structure is the OSI model, as shown in Figure 2.1. The physical (PHY) layer deals with bit-level transmission. The layer above PHY is the data link layer, also known as the Medium Access Control (MAC) layer. Protocols such as Internet Protocol (IP) and Transmission Control Protocol (TCP) are commonly used above MAC layers. This thesis mainly focuses on MAC layer protocol design. Therefore, it

is critical to know the main MAC protocols used in practice. In addition, the MAC layer has been an active research focus for the past twenty years. As a result, numerous protocols have been proposed to increase bandwidth utility (defined as correctly transmitted data per maximum data possibly transmitted) in different application domains. This section will examine some of the popular classes of protocols from which others derive.

Application (Layer 7)
Presentation (Layer 6)
Session (Layer 5)
Transport (Layer 4)
Network (Layer 3)
Data Link (Layer 2)
Physical (Layer 1)

Figure 2.1: The 7 layers of the OSI model.

2.2.1 ALOHA

ALOHA was a pioneering MAC layer protocol developed by the University of Hawaii. There are two variations of ALOHA protocols, namely, pure ALOHA and slotted ALOHA. As shown in Figure 2.2, pure ALOHA is very simple [14]:

1. If you have a data packet to send, you should send it immediately.
2. If there is any contention while transmitting that data packet, you should resend it later.

It has been proved in many studies that the maximum channel utility of pure ALOHA is $e^{-2} = 13.5\%$ [15]. Slotted ALOHA is an improvement of pure ALOHA as it divides the time domain into discrete small slots, as shown in Figure 2.3 [16]. Each data transmission has to be aligned within the time slot, thus reducing contention significantly. It is also well-known that the maximum channel utility is $e^{-1} = 36.8\%$. Though ALOHA is not efficient in terms of channel efficiency compared to other

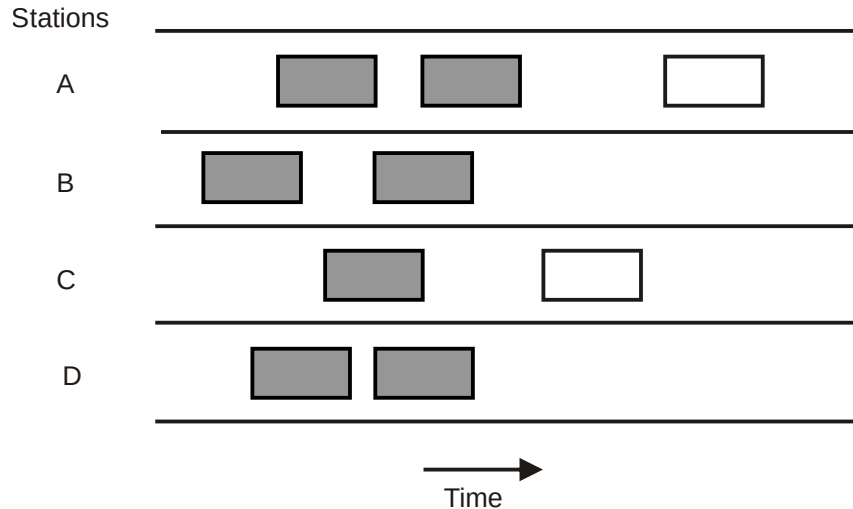


Figure 2.2: Pure ALOHA protocol. Boxes indicate frames. Shaded boxes indicate frames which have collided.

modern MAC protocols, it inspired a generation of wireless protocols, such as Carrier Sense Multiple Access (CSMA), which will be explained in Section 2.2.3.

2.2.2 Time Division Multiple Access

Time Division Multiple Access (TDMA) is a channel access control method, on which many MAC layer protocols are based. Each device shares the same frequency by dividing the channel into small time slots [17]. In other words, if a group of devices wish to communicate with the base station, they need to use the same channel in turns for a fixed amount of time. The sequence they use is typically determined either statically or dynamically. Static channel allocation is illustrated in Figure 2.4. TDMA is notably used in digital 2G cellular systems, such as the Global System for Mobile Communications (GSM).

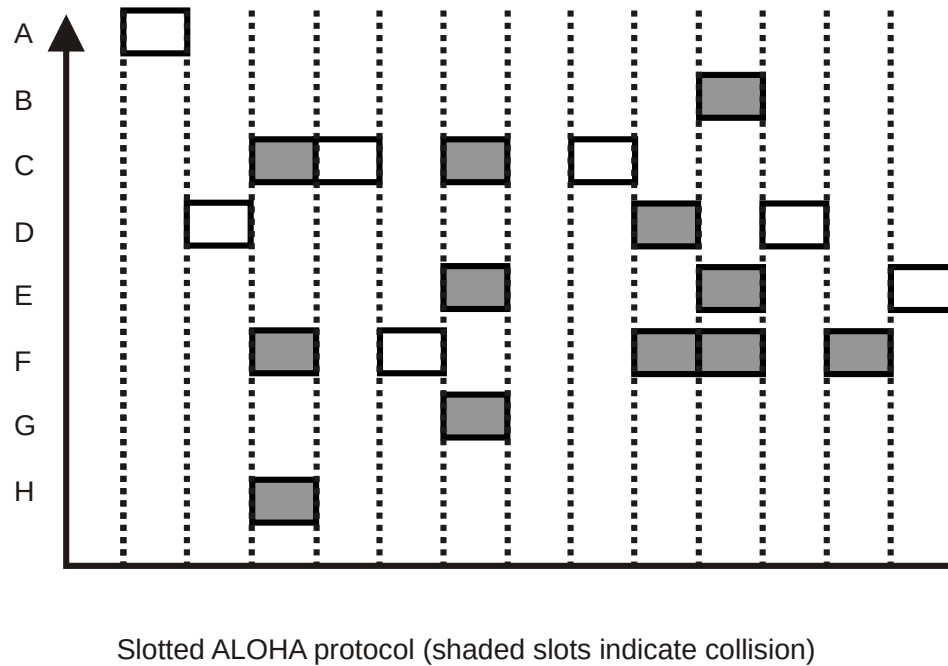


Figure 2.3: Slotted ALOHA protocol. Boxes indicate frames. Shaded boxes indicate collided frames.

2.2.3 Carrier Sense Multiple Access

Carrier Sense Multiple Access (CSMA) is a significant improvement to ALOHA. Before sending a data packet, the device has to “sense” the transmission medium and make sure that there is no transmission going on. Once verifying that the transmission medium is not busy, the device can then send the packet.

There are a few variations of CSMA. CSMA-CD (Collision Detection) requires devices that abort transmission as soon as a collision is detected. CSMA-CA (Collision Avoidance) lets the device avoid collision by not transmitting for a certain amount of time (called backing off) if the device senses that the transmission medium is busy. There are different transmission modes of CSMA-CA where the probability of deferring may vary. In general, p -persistent CSMA represents the probability, p , of sending a packet directly without backing off.

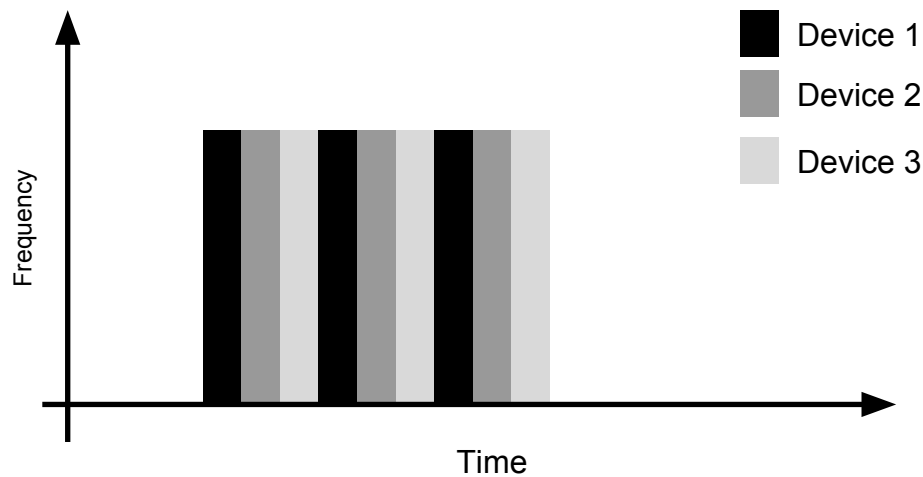


Figure 2.4: TDMA frame structure. The medium is shared by 3 devices and each communicates at designated slots.

2.2.4 Code Division Multiple access

Code Division Multiple access (CDMA) is a medium access control protocol that allows multiple devices to transmit data simultaneously over a single communication channel. CDMA employs spread-spectrum technology and uses a special coding scheme. It is achieved by exploiting mathematical properties of orthogonality between vectors representing the data strings. Due to its complexity, we will only introduce a simple example of synchronous CDMA, adapted from *Computer Networking: A Top-Down Approach* [15].

Imagine we have two devices (d_0 and d_1) that want to transmit a packet simultaneously. For simplicity, we will use -1 to represent 0 and let's assume d_0 wants to send 1-bit value 1 and d_1 sends value -1 . Before sending the bit, we need to encode the bit using our special coding scheme. To do so, we will multiply the bit with the device's unique Gold code, a binary sequence named after Robert Gold [18]. Let the the Gold code for d_0 be $\langle 1, 1, 1, -1, 1, -1, -1, -1 \rangle$ and d_1 be $\langle 1, -1, 1, 1, 1, -1, 1, 1 \rangle$. Notice that $\langle 1, 1, 1, -1, 1, -1, -1, -1 \rangle \cdot \langle 1, -1, 1, 1, 1, -1, 1, 1 \rangle = 1 - 1 + 1 - 1 + 1 + 1 - 1 - 1 = 0$, which means two Gold

codes are orthogonal to each other. The multiplication result is

$$\begin{aligned} d_0 &: \langle 1, 1, 1, -1, 1, -1, -1, -1 \rangle \\ d_1 &: \langle -1, 1, -1, -1, -1, 1, -1, -1 \rangle \end{aligned}$$

As a result of multiplication, each device will send their computed 8-bit value. The receiver will receive a superimposed result of these two signals, yielding

$$\langle 0, 2, 0, -2, 0, 0, -2, -2 \rangle .$$

To decode, the receiver needs to compute the normalized projection of received data on the same Gold code. Thanks to orthogonality, each Gold code can properly decode the signal, as shown below:

$$\begin{aligned} d_0 &: \frac{\langle 0, 2, 0, -2, 0, 0, -2, -2 \rangle \cdot \langle 1, 1, 1, -1, 1, -1, -1, -1 \rangle}{|\langle 1, 1, 1, -1, 1, -1, -1, -1 \rangle|} = \frac{8}{8} = 1 \\ d_1 &: \frac{\langle 0, 2, 0, -2, 0, 0, -2, -2 \rangle \cdot \langle 1, -1, 1, 1, 1, -1, 1, 1 \rangle}{|\langle 1, -1, 1, 1, 1, -1, 1, 1 \rangle|} = \frac{-8}{8} = -1. \end{aligned}$$

This illustrates how CDMA can successfully decode two signals sent simultaneously

2.3 Radio Communication and Properties

This section gives the reader a basic introduction of radio communication regulations imposed by FCC as well as some known effects in the PHY layer.

2.3.1 FCC Regulation

In this thesis, we will discuss radio communication in 902-928 MHz, also known as Industrial, Scientific, and Medical radio band (ISM band). The Federal Communications Commission (FCC) strictly regulates on how the radio system broadcasts messages. Here is a list of terms relevant to our thesis work taken from Semtech [19]:

1. In order for Frequency hopping spread spectrum (FHSS) to be compliant with a frequency hopping system, the device or system must meet the following requirements:
 - (a) The channel carrier frequencies must be separated by a minimum of 25 kHz, or the 20 dB bandwidth of the hopping channel, whichever is the greater.
 - (b) The channel hopping frequencies which are selected at the system, hopping rate must be pseudo-random in nature. On average, each channel hopping frequency must be used equally.
 - (c) If the 20 dB bandwidth is less than 250 kHz, the system shall use at least 50 channels. The average dwell time on a particular channel shall not exceed 400 ms within a 20 second period. If the 20 dB bandwidth is 250 kHz or greater, then the system shall use at least 25 channels. The average dwell time shall not exceed 400 ms within a 10 second period.
2. The FCC allows a device to comply with these regulations without necessarily implementing Direct Sequence Spread Spectrum (DSSS), provided that the following requirements are met:
 - (a) The minimum 6 dB bandwidth of the signal shall be at least 500 kHz.
 - (b) The maximum permitted peak conducted output power is +30 dBm (1 W).
 - (c) If the antenna used has a directional gain in excess of 6 dBi then the conducted output power described shall be reduced by the amount in dB that the directional gain of the antenna exceeds 6 dBi.

We will discuss how these regulations are going to influence our design of LP-WAN protocols in Chapter 5.

2.3.2 Capture Effect

In radio communications, typically frequency modulation (FM), when multiple signals near the same frequency arrive at the receiver at the same time, only the strongest signal can be demodulated and received. Such phenomenon is called the capture effect. As an example, suppose there are three signals, s_0 , s_1 , and s_2 arriving at the same time and all of them share similar frequencies. However, since the transmitters are located at different places, the strength of the signals are ranked as $s_1 > s_2 > s_0$.

Due to the capture effect, packets with weaker signal strength, s_0 and s_2 will not be amplified and thus attenuated by the receiver. Therefore, s_1 can be correctly received by the receiver even though a collision has occurred.

The capture effect plays an important role in radio communication. Sometimes the capture effect is ideal as the content of strong signal will not be corrupted by other weaker interferences. It is possible to deploy more base stations to recover collided signals. However, in case of collision detection at a single base station, the capture effect will prevent a Cyclic Redundancy Check (CRC) from detecting packet corruption, hence failing contention detection. In addition, the capture effect is a very complex process in the PHY layer and the technical details are out of the scope of this thesis. We will explore some implications of the capture effect in Section 5.2.

2.4 Mathematical Models and Theorems

This section focuses on giving readers some basic mathematical models and theorems useful to understand the discussion of this thesis. To focus on the major contributions of this thesis, this section will not delve into details and convoluted proofs. Instead, it will point out useful references if readers are interested.

2.4.1 Bloom Filter

Set and testing membership plays an important role in many standard algorithms. A Bloom filter is a space-efficient probabilistic data structure for representing a set to support membership queries. Although the Bloom filter allows false positives, it forbids false negatives and can reduce the storage size significantly while preserving the controlled error rate [20].

The mathematical representation for the classical Bloom filter is very simple. Let $S = \{x_1, x_2, \dots, x_n\}$ be a set of n elements represented by m bits, which are initially set to 0. There are k independent hash functions, $\{h_1, h_2, \dots, h_k\}$ whose range is $\{1, 2, \dots, m\}$. To add an element x , we need to feed in each individual hash function to obtain $H_x = \{h_1(x), h_2(x), \dots, h_k(x)\}$. Then we set each bit at position $h_i(x)$ to 1 for all i .

Querying is very similar to adding an element. For element y , we need to feed y into k hash functions and obtain $H_y = \{h_1(y), h_2(y), \dots, h_k(y)\}$. If any of these bits are 0, then y is definitely not in set S , otherwise y might be in the set S . Notice that by relying entirely on the bit values we have introduced a false positive into the Bloom filter, which is a result of a space-accuracy trade-off.

To obtain an optimal size for the Bloom filter, we can use probability to prove the lower bound for a Bloom filter with n entries. Let the false positives fraction be ϵ and the size be m . m has to satisfy

$$m \geq n \frac{\log_2(1/\epsilon)}{\ln 2} = n \log_2 e \cdot \log_2(1/\epsilon).$$

There are many network-related applications of the Bloom filter. Abhishek Kumar *et al.* proposed a new Bloom filter based algorithm for network traffic measurement [21]. Haoyu Song *et al.* used the Bloom filter to build a fast hash table for network processing [22]. We will use the Bloom filter to handle the capture effect discussed later in Section 5.2.

2.4.2 Queueing Theory

Queueing Theory is the mathematical study of queues. It focuses on constructing a model so that queue length and waiting time can be predicted [23]. We will focus on single queueing nodes, which can be described using Kendall's notation, in the form of $A/S/C$. A describes the time between arrivals to the queue, S describes the size of jobs, and C describes the number of servers in the queue.

One of the simplest models is $M/D/1$, where M stands for Markov and the distribution forms a Poisson process, D stands for deterministic and in our thesis will be a constant, and 1 stands for a single server. Due to limited space and a concern about oversimplification, we will further our discussion in Chapter 5 but leave curious readers to check out *Probability, Statistics, and Queueing Theory* for more information [23].

Chapter 3

Related Works

Since part of the thesis is a demonstration of a series of improvements we have made in LPWAN research, it is important to examine the current state-of-art technologies. As mentioned in the Chapter 1, these technologies have enjoyed a wide range of deployments and testing, yet have also shown their limitations and disadvantages. Particularly, all of them are proprietary and patented. In Section 3.1, we will discuss in detail about each technology: what makes it suitable to build LPWAN, and why we need to consider a better solution. In Section 3.2, we will talk about some recent researches related to our thesis discussion.

3.1 State-of-Art Technology in Industry

IngenuTM (previously On-Ramp Wireless) is a San Diego-based company that is actively deploying low-power wide-area networks across the country. IngenuTM has designed a proprietary protocol called Random Phase Multiple Access Direct Sequence Spread Spectrum (RPMA-DSSS) for its networks. RPMA is used favorably in the upstream link while conventional CDMA is used for the downstream traffic. IngenuTM has deployed more than 38 large-scale networks, offering network connectivity to millions of IoT devices [24]. IngenuTM claims to be the world's largest IoT network provider [25].

In contrast to the other standards discussed in this chapter, IngenuTM uses the 2.4

GHz band, making it universal as some countries have different regulations on the 900 MHz ISM band, such as South America and Africa [24]. This allows one single device to join a global network without network reconfiguration, thus enabling users to reach a global market. Another selling point for Ingenu is its claimed long battery life. The company claims that the average battery life for an Ingenu™ device is 20 years [24]. Although the measurement details are unknown, it is still impressive for an IoT device.

After introducing its claims and achievements, we now proceed to analyze its communication protocol. RPMA is a variation of conventional CDMA and differs from CDMA in two ways [26]. First, all the Ingenu™ devices share the same gold code to spread the bits before transmitting [26]. Second, after selecting a transmission slot, the transmitter performs a random delay. As long as two transmissions do not arrive simultaneously, RPMA allows correct decoding of these two transmissions [27]. This ability is achieved by brute force despreading of all waveforms at the receiver side in parallel. This is achievable as the receiver is typically connected to the power grid and it has much more computation power than average nodes.

However, Ingenu™ is not perfect. As with other contention-based MAC protocols, when the network load increases collisions will be more frequent, causing a reduction in the available network bandwidth. In addition, sharing the same Gold codes will also have security risks. For instance, a malicious node can join the network and potentially obtain the shared Gold codes. After acquiring the Gold codes, the attacks can either eavesdrop or analyze the traffic. Furthermore, since the downstream link employs traditional CDMA, each device needs to generate a Gold code from its MAC address and exchange it with the base station. By jamming the network, the attack can force the device to continuously attempt to join the network, hence giving the attacker a chance to crack the shared Gold codes [27].

Another industrial technology is LoRaWAN™ a protocol designed by the LoRa Alliance for low-cost IoT networks [28]. There are three classes of end-point devices in LoRaWAN™, namely Bi-directional end-devices (Class A), Bi-directional end-devices with scheduled receive slots (Class B), and Bi-directional end-devices with maximal receive slots (Class C) [28]. Class A operation uses the lowest power as it requires minimum coordination between the device and the base station.

The physical layer of LoRaWAN™, LoRa, uses Chirp Spread Spectrum modulation (CSS). CSS is very useful to recover data from weak signals, therefore increasing the effective range of LoRaWAN™ transmissions. This advantage attracts other vendors to build their IoT networks on LoRaWAN™, such as The Things Network, which

will be discussed later. LoRa has very strong capture effects, meaning that if we deploy enough base stations in a region, potentially eliminating the negative effect of collisions. It also claims to be low-power and wide-range.

However, the MAC layer of LoRaWAN™ is very lightweight and essentially implements pure ALOHA with two receive windows (dedicated time slot to receive packets) to ensure data integrity. This results in low channel utility under high traffic load due to packet collisions if there are insufficient base stations or geographic diversity to recover the collided packets. Furthermore, LoRaWAN™ may not scale as LPWAN™ claims in all cases. As demonstrated by M.C. Bor *et al.*, LoRaWAN™ does not scale very well if there is only one sink (base station). However, they also showed that LoRaWAN™ can scale if there are multiple sinks. This shows the trade-off between scalability and efficiency.

In addition, to resolve contention among signals, LoRaWAN™ recommends the deployment of more base stations within the region to take advantage of the capture effect. Recall that the capture effect allows the receiver to decode the strongest signal regardless of the contention. As a result, if enough base stations are deployed, every transmission signal can be decoded. However, It is not cost-effective as we could have used extra available alternative bands provided by these base stations to increase the network throughput. Commercial LoRaWAN™ radios are widely available and implement the LoRaWAN™ MAC in software. Replacing the LoRaWAN™ MAC layer while keeping the CSS physical layer is an attractive option for developing a more efficient platform.

SigFox is another major LPWAN technology provider. It claims to have deployed networks in 32 countries, providing services to more than 512 million people [29]. Their network coverage is primarily in Europe and is operated by different partner companies in each country. Due to its wide deployment, it has attracted industrial investors such as Intel [30].

SigFox developed Random-FDMA (R-FDMA) with the goal to minimize the manufacturing cost for IoT devices [31]. By using ultra-narrow band transmitters and sophisticated wideband base stations, R-FDMA allows each IoT device to transmit using a random frequency. Although the lack of contention resolution introduces the possibility of interference within the same channel when two nodes are transmitting simultaneously, R-FDMA does not impose any constraints on how the node chooses an operating frequency. It is typically set in manufacturing allowing relaxed oscillator stability constraints. R-FDMA depends on each ultra-narrow-band channel being lightly utilized and forgoes typical medium access control mechanisms. This design

simplifies device nodes but requires a more sophisticated base station to receive the random frequency signals.

A typical R-FDMA base station would be implemented using a software defined radio (SDR) to sample the available spectrum and then perform all RF signal processing in software. This requires significant bandwidth between the SDR and host processor and enough processing power to process the received signal in real-time. In other words, R-FDMA base station is relatively expensive to deploy. Until a low-cost base station platform is developed, R-FDMA is less attractive.

Another company worth mentioning is The Things Network (TTN). TTN uses LoRaWAN™ up to the MAC layer and pre-package the entire protocol stack into read-to-use base station boxes and devices [32]. Users can order these premade base stations and set it up by themselves easily. One innovation TTN contributes is that it employs a crowdsourcing approach to set up its network. Users volunteer to share their base stations and computation power. By doing so, TTN has deployed over 827 regional networks. However, TTN's deployment lacks formal analysis and modeling. The cost of their base station is over \$ 1,000 USD and therefore is formidable to many people [32]. Furthermore, TTN uses LoRaWAN™ which suffers from the MAC efficiency and scalability problems mentioned previously. This thesis will formalize the crowdsourcing idea and provide a much more efficient and cost-effective platform than TTN.

The approach used by TTN echoes a novel solution to solve the IoT connectivity problem, proposed by T. Zachariah *et.al* [12]. It utilizes a person's smartphone as a public gateway device. The so-called Universal Gateway can be implemented using Bluetooth Low Energy (BLE) on personal smartphones to remove the restriction that one IoT device has to connect to one specific smartphone to communicate with the Internet. Individuals can set up gateway proxies on their smartphones to increase the network coverage. Financial or other incentives can be used to increase user participation. We borrowed this idea and explored the participation rate needed to achieve regional network coverage our proposed platform.

3.2 Academic Approaches

For low-power networks, the conventional MAC protocols may not work very well as they were designed for different needs. We need to look for protocols that are

fundamentally different yet can be derived to more efficient protocols for IoT networks. Distributed queueing (DQ) is one of the protocols this thesis will focus on.

In the DQ literature, there are several recent attempts to adopt DQ into the IoT domain, most notably LPDQ and DQ over LTE [33, 34, 35]. We will cover the basics on distributed queueing in Section 5.2. Both simulation and real world experiments demonstrate the superiority of DQ over ALOHA-based MAC protocols such as CSMA in terms of latency and throughput under various traffic patterns. LPDQ also has its own time synchronization and frequency hopping mechanisms. However, LPDQ is designed for high-frequency, higher-bandwidth wireless links and is not intended for low-rate IoT networks. As we will demonstrate in this thesis, in a low data rate environment, the protocol overhead adversely affects the channel utilization. In addition, LPDQ suffers larger latency under bursty traffic, as discussed in Section 5.2. Finally, LPDQ only supports upstream packets, which may be unacceptable in some IoT applications. We will present a more efficient DQ-based protocol in this thesis.

To minimize the deployment cost, the network operator needs to know the expected performance for a planned deployment and make changes if necessary. Given the fact that an actual deployment of large-scale wireless network costs millions of dollars, simulation plays an important role in designing and evaluating a wireless network.

Low-power IoT coverage planning and network simulation are emerging topics [36, 37, 38, 39]. B. Reynders *et.al* simulated both the physical and MAC layer for low-power networks in a square arrangement to evaluate the packet delivery ratio. The simulation explores the difference between wideband spread spectrum (LoRa-like) and ultra narrowband (Sigfox-like) networks [36]. M. Centenaro *et.al* deployed a LoRa network in the real world and proved the feasibility of complementing the problems of IoT networks with long-range radio links [37]. SCALECycle was designed to solve intermittent and varying coverage using a mobile agent actively collecting data [38]. Y. Al Mtawa *et al.* proposed an algorithm to identify the holes in an IoT deployment [39].

Since the world is not ideal, we are guaranteed to encounter radio propagation delay, path loss, and possibly interference. As a result, properly understanding the radio propagation model is critical to planning and simulation. Among radio propagation based simulations, a number of models are proposed to provide path loss estimation for a higher network layer simulation, such as ITM, Hata, and ITWOM [40, 41]. S. Kasampalis *et al.* demonstrated that ITWOM, though not perfect, gives more ac-

curate results than previous models within a radius of 20 Km [40]. However, to our knowledge, there is no known simulation using demographic information to generate the test points; test points have been generated at random from uniform distributions. This thesis provides a more accurate representation of IoT deployment topology.

Chapter 4

Crowdsourcing IoT Networks

4.1 Crowdsourcing LPWAN Networks

Crowdsourcing is the practice of obtaining information or input into a task or project by enlisting the services of a large number of people, either paid or unpaid, typically via the Internet. Crowdsourcing is commonly used in environmental sensing and monitoring as it provides ubiquitous sources of environmental data. Borrowing the ideas from crowdsourced sensing, if we let LPWAN users to set up base stations by themselves, we can reduce our deployment and maintenance cost, that is, crowdsourcing gateways for LPWAN. We define a *csLPWAN* as any LPWAN where the base stations are randomly deployed by users of the system rather than deployed in a coordinated fashion by a network operator.

To incentive *csLPWAN* users, we can make sure each base station is affordable and users can get paid by the volume of data transmitted through their base stations. As a result, a single IoT device can rely on the network provided by another user. Let's imagine that a *csLPWAN* user jogging around the neighborhood. His smartwatch will connect to his neighbor's base station and then his neighbor's neighbor's. His smartwatch will always stay connected and updated and therefore there is no need for him to bring a smartphone for synchronization. The IoT gateway problem is solved!

However, using crowdsourcing gateways introduces new uncertainties. For instance, how many base stations are needed to cover a give area? What is the dis-

tribution of these crowdsourced gateways? What is the average link quality in the csLPWAN? How do we simulate the network performance of a csLPWAN? To answer these questions, We need a mathematical model so that we can estimate performance, which will be discussed in next section.

4.2 PlanIt, Realistic LPWAN Planning

Estimating wireless coverage is a challenging problem. The most basic approach is to assume that the wireless devices are uniformly distributed in a square and the terrain is flat with only free-space path loss [36]. These are obviously not realistic assumptions. Since many IoT devices are designed to assist or improve human activities, it is reasonable to assume that their deployment shares similar characteristics as local demographic information. For instance, a city is more likely to have a denser deployment of IoT devices than a rural region. Therefore, a uniform distribution of test points does not represent the realistic deployment of human-centric IoT devices, thus reducing the credibility of these network simulation results.

To address these challenges, we have designed a planning tool called PlanIt. Our approach in PlanIt is to select potential IoT device locations (Section 4.2.1) within a region that reflects the local demographic characteristics. From the generated locations, we randomly select a subset to be crowdsourced gateway devices. Then we use the Irregular Terrain with Obstructions Model (ITWOM) 3.0 to compute the path loss from each device to every gateway (Section 4.2.2). If there is a gateway device within the link budget of the radio, a network connection is possible. The path loss information can then be used in a network simulator to produce realistic packet errors for all links in the network.

4.2.1 Selecting IoT Device Locations

Given longitude x and latitude y , the probability of an IoT device is $P_{IoT}(x, y)$, where P_{IoT} is the joint probability function defining the probability of all IoT devices. For a given region Ω , $\int_{\Omega} P_{IoT}(x, y) dx dy = 1$. PlanIt can consider several factors that can affect P_{IoT} :

- population density, $\rho(x, y)$,

- geographic-related information, such as $g(x, y)$ for topological effects,
- and demographic-related information, I , such as the influence of average age or income on IoT device use.

For simplicity, we define the adverse effect of topography as

$$P_{gt}(x, y) = -\mathbf{G} \cdot \nabla g = -\mathbf{G} \cdot \left\langle \frac{\partial g}{\partial x}, \frac{\partial g}{\partial y} \right\rangle,$$

where \mathbf{G} is a vector representing strength of the effect.

Therefore, unnormalized P_{IoT} , P'_{IoT} can be written as

$$P'_{IoT}(x, y) = \rho(x, y) - \mathbf{G} \cdot \left\langle \frac{\partial g}{\partial x}, \frac{\partial g}{\partial y} \right\rangle + I\rho(x, y)$$

Hence the probability function P_{IoT} is

$$P_{IoT}(x, y) = \frac{P'_{IoT}(x, y)}{\int_{\Omega} P'_{IoT}}$$

Although in the real world these functions are continuous given a large region and population, for computational simplicity, we compute discrete values over a small interval.

To generate random IoT test points based on the probability distribution P_{IoT} , we will use tiles when computing P_{IoT} . For tile (x, y) , P_{IoT} is locally uniform, hence we can easily generate random points in the tile. For those boundary tiles, additional caution should be taken as we might generate points outside the region. If this is the case, we can simply discard these points.

By correctly choosing \mathbf{G} and d , we can realistically approximate the probability function describing the IoT device locations. Below is a demonstration of how we generate IoT locations given population and geographic information. We choose $\rho(x, y) = 50((x - 0.5)^2 + (y - 0.5)^2)^{-1}$, $\Omega = (0, 1) \times (0, 1)$, $g(x, y) = ((x - 0.3)^{1.5} + (y - 0.3)^{1.5})^{-1}$. For illustration purpose we choose $\mathbf{G} = \langle 20, 20 \rangle$. The first map in Figure 4.1 shows the probability density map of ρ , $\mathbf{G} \cdot \left\langle \frac{\partial g}{\partial x}, \frac{\partial g}{\partial y} \right\rangle$, and P_{IoT} . These functions are chosen to model a situation where population is centered in a city near a mountain. As we can see, the raw population is centered in the city yet influenced

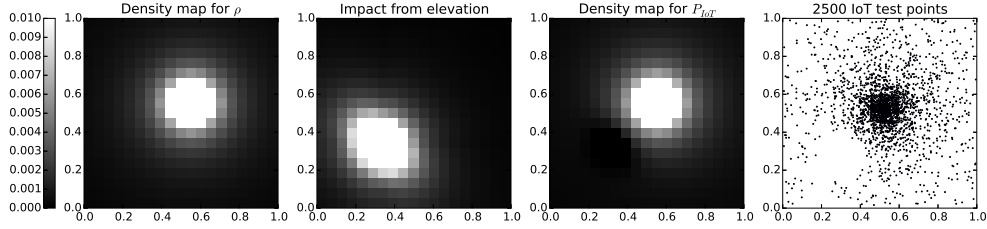


Figure 4.1: Probability density map of $\rho(x, y)$, $\mathbf{G} \cdot \langle \frac{\partial g}{\partial x}, \frac{\partial g}{\partial y} \rangle$, and P_{IoT} . Notice the change of the map at bottom left corner due to geographic information. The right plot shows the 2500 IoT device locations generated based on P_{IoT} .

by the existence of the mountain, as indicated by the dark corner in P_{IoT} density map. As a result, the IoT device location map reflects the population distribution in the P_{IoT} density map.

To select points in the United States, PlanIt uses demographic data from the U.S. Census 2010 Summary File 1 (SF1) [42] to estimate the population density (ρ). Census data is provided in several hierarchical levels from coarse to fine-grained: State, County, Subdivision, Place, Tract, Block group, and Block. For csLPWAN planning, we want to estimate network coverage over entire census-designated places. These census-designated places include cities, towns, boroughs, districts, municipalities, and townships. To further refine the test points within a place, we use data from one level down the hierarchy and break each place into the underlying census tracts. Each census tract has a distinct population density that is used to bias the sampling of points within the place.

As a real-world example, we consider the city of Philadelphia, Pennsylvania. Philadelphia is the most populated city among Pennsylvania’s 57 cities with 1.526 million inhabitants as of the 2010 census. The city covers 365 square kilometers and is divided into 381 census tracts. Each census tract is fully contained within the city. We use the point selection algorithm considering only population density without any other geographic-related information (topological or demographic effects) to generate 1,000 test points in Figure 4.2. Visually we can see the test points are not uniformly distributed. Some census tracts have multiple test points while others have no test points at all. The distribution of land area and the population of the census tracts are shown in Figure 4.3. From this we can see there are 10 census tracts with fewer than 1,000 inhabitants and the majority of census tracts are smaller than 3 square kilometers. The large area near $(-75.20^\circ, 40^\circ)$ contains Fairmount Park, a large urban park surrounding the Schuylkill River containing the Philadelphia Zoo and several

other museums. The areas to the south are industrial zones with rail and shipyards to access the Delaware river. The area in the northeast contains the appropriately named Northeast Philadelphia Airport (the International airport, PHL, is outside of the city limits). With these facts in mind, the generated IoT test point distribution does closely follow the population distribution.

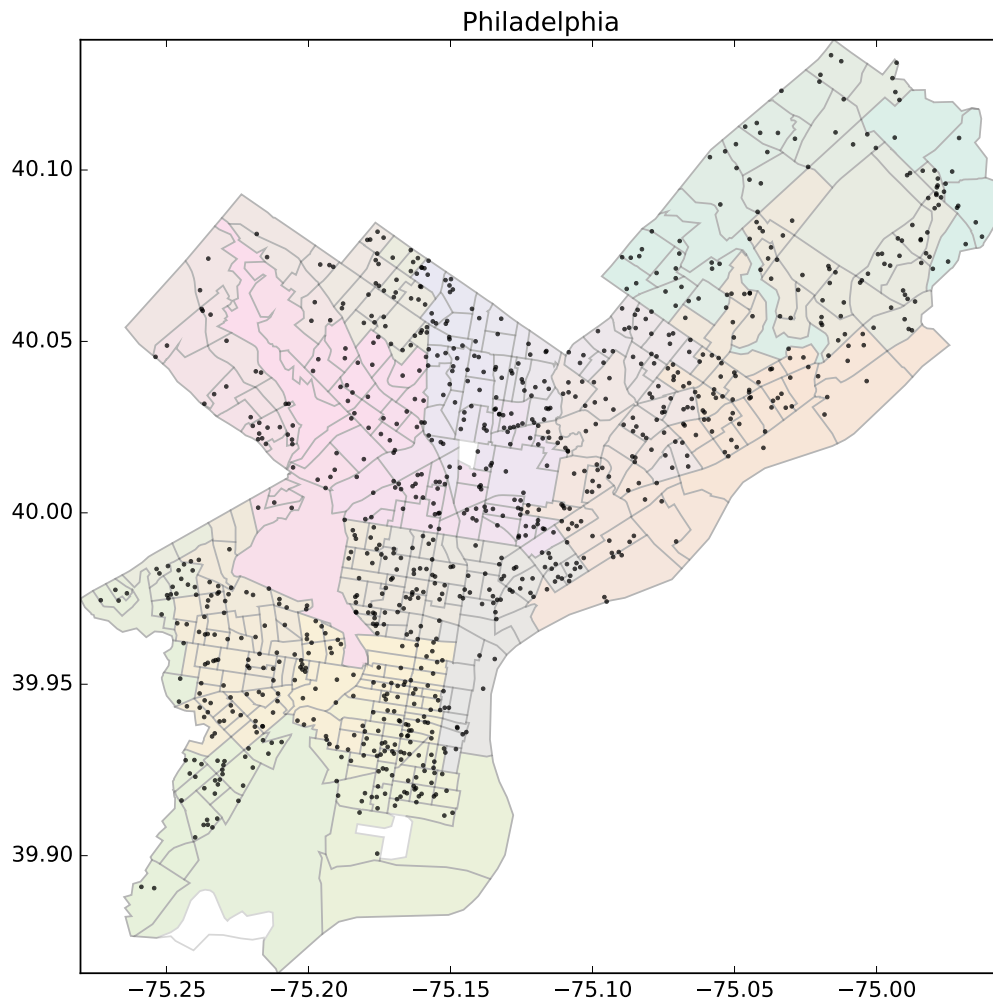


Figure 4.2: The data from 1,000 test points generated in Philadelphia. Census tracts are shown in colored polygons and test points are shown in dots.

When considering other cities in Pennsylvania, we found that census tracts are not always fully contained within a city boundary as in Philadelphia. There are some cases when a city is contained within a census tract and the census tract is larger than the city. This is a common phenomenon when examining smaller census-designated

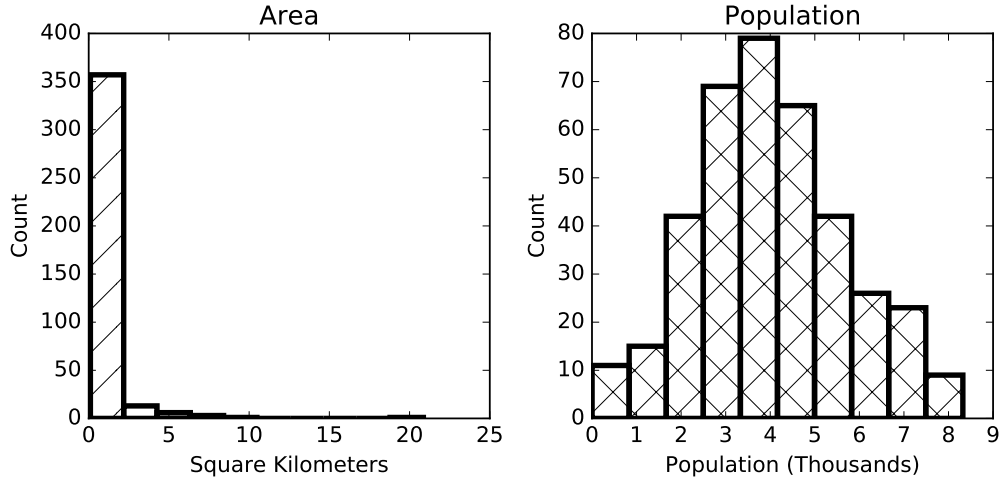


Figure 4.3: Census tract area and population distributions for the city of Philadelphia, Pennsylvania containing 381 census tracts.

places. The smallest city in Pennsylvania, Parker, falls into this category. In these cases, we select points within the census tract using a normal distribution with mean at the centroid of the city and a standard deviation $\sigma = \frac{1}{3}$ of the smallest distance from the city centroid to the bounding box created by the census tract, so there are at least three standard deviations to the nearest border. The goal is to generate points near the city center while still spanning the entire census tract with some probability. Any point generated outside of the census tract is rejected and a new random point is drawn. The resulting test points for Parker, Pennsylvania are shown in Figure 4.4. This shows the desired clustering of points near the city (red) while still representing the entire census tract (blue).

4.2.2 Path Loss Estimation

From a set of generated test points in a city or other region, we uniformly select n test points to be crowdsourced base stations. Because the test points were generated using population density and possibly other information, the selected base stations will also share this distribution. Then we compute the path loss from each point to all of the base stations using the Irregular Terrain with Obstructions Model (ITWOM) 3.0. This model improves on the Longley-Rice (ITM) model and estimates path loss taking into account topographic and ground clutter information. Although these models

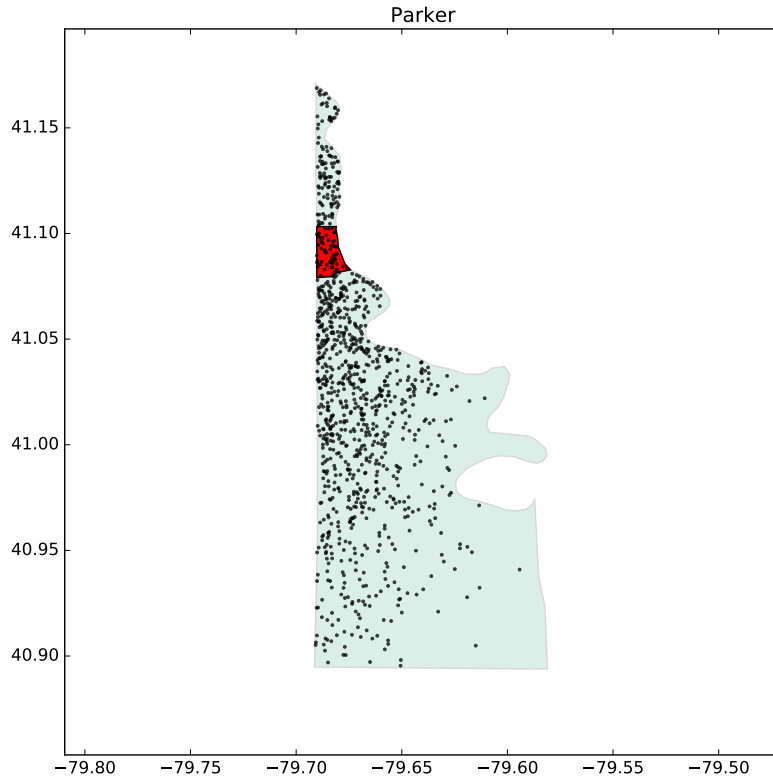


Figure 4.4: The locations of 1,000 sample points generated for the city of Parker, Pennsylvania (red) and Armstrong County Census Tract 9503 (blue).

were developed for predicting Digital Television (DTV) and frequency modulation (FM) broadcast coverage, they can also be applied to the ISM frequency bands. It is because DTV broadcast coverage estimation tends to be very conservative about its result. Hence using ITWOM can give us an estimated lower bound of the network performance.

Topographic information for the model was obtained from the Shuttle Radar Topography Mission (SRTM) dataset with 1 arc-second resolution (approximately 30-meter resolution on the ground). This data has been previously shown to have good accuracy [43]. For each source/destination pair, the line of sight geodesic path is constructed by taking no more than 30-meter steps, to match the SRTM data resolution, from the source to the destination until reaching the destination. The final point along the path is always the destination. The elevation at each point along the path is retrieved from the SRTM dataset. Before passing to the ITWOM library, if there are any negative elevation values, the entire path is shifted up by the absolute value

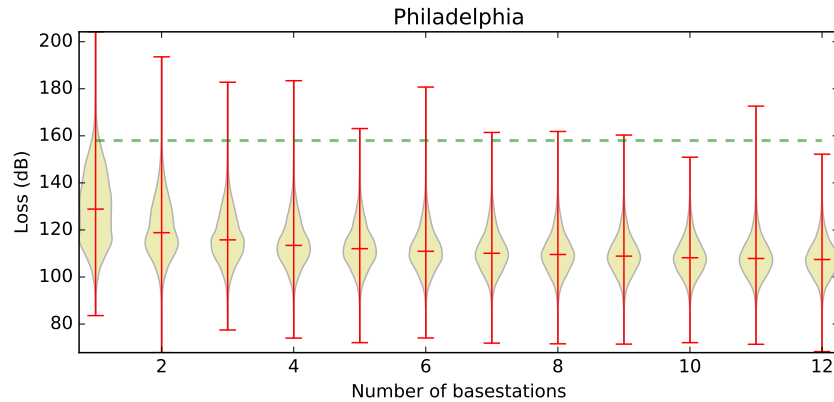
of the most negative point so the minimum elevation is zero to prevent errors in the path loss calculation.

The complete set of ITWOM parameters are listed in Table 4.1. The values for dielectric constant and conductivity are typical for a city environment. The transmitter height was set to 5 meters above ground level (AGL) which is easily reached from the roof of a 1-story building. The receiver height was set to 1 meter AGL to represent a device near ground level. Because of the reciprocity principle, it is irrelevant whether the IoT device or base station is the transmitter for path loss estimation.

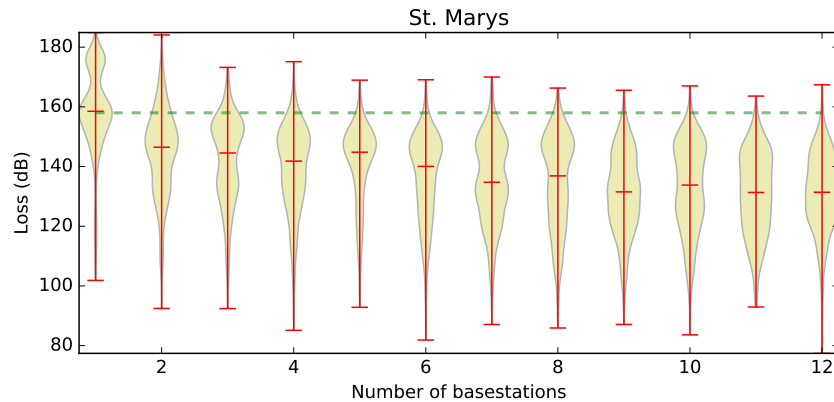
Table 4.1: ITWOM parameter values used to compute path loss.

Parameter	Value
Transmitted Height	5 meters
Receiver Height	1 meter
Earth Dielectric Constant	5.0 (city)
Earth Conductivity	0.001 (city)
Atmospheric Bending Constant	301.0
Frequency	900 MHz
Polarization	Horizontal
Location Variability	50%
Time Variability	50%

Using these parameters, we generate 100 sets of 1,000 test points and randomly select $\{1 \dots b\}$ points from each set to act as base stations, then compute the path loss from all points to every base station keeping only the lowest loss path. Figure 4.5a shows the path loss distribution for Philadelphia, Pennsylvania along with a hypothetical 158 dB receive threshold. This threshold was selected as a fair estimate from available LPWAN radio specifications which vary from 149 to 175 dB. This shows that a single randomly located base station has a very good chance of providing connectivity to most of Philadelphia with a median path loss of 130 dB. This result is somewhat surprising and we should keep in mind that ITWOM only considers the elevation data measured from space provided by the SRTM dataset. It is possible that the reported elevations reflect the top of buildings, especially in a dense city, resulting in urban canyon effects on the surface. However, this result gives us confidence that csLWPANs can cover a large population using very few base stations.



(a) Path loss distribution in Philadelphia, Pennsylvania.



(b) Path loss distribution in St. Marys, Pennsylvania.

Figure 4.5: The path loss distribution from 100 rounds of evaluating 1,000 test points with randomly assigned base station nodes in two Pennsylvania cities. The dashed line at 158 dB indicates the threshold for reception.

Analyzing other cities in Pennsylvania yields similar results to Philadelphia. Figure 4.6 shows the estimated connectivity using one base station and a 158 dB receive threshold for every city in Pennsylvania. The median connectivity is 99.2% and in all but two cities the median connectivity is greater than 90%. The worst performing city in Pennsylvania is St. Marys, shown in Figure 4.5b. Investigating St. Marys reveals although classified as a city, it is a geographically large area of 257 km² in the Allegheny Mountain region spanning an elevation range of more than 250 meters. For comparison, Philadelphia covers 367 km² but only has an elevation range of 100 meters with more gradual elevation changes. These two factors combine to yield a

challenging region to cover.

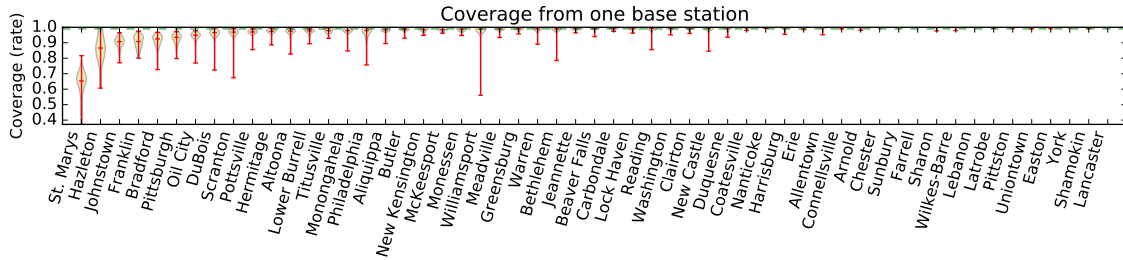


Figure 4.6: The coverage rate for all cities in Pennsylvania with one base station and a receive threshold of 158 dB. The median connectivity, 99.2%, is indicated by a dashed line.

We have implemented PlanIt as a web service (much credit to Dr. Alan Marchiori) and it is available to the public ¹, as shown in Figure 4.7. Users can log in with their Google credential and compute path loss, determine the number of base stations needed for network coverage, and calculate the path loss. While performing radio propagation simulation, users can also choose different radio parameters to obtain better network coverage.

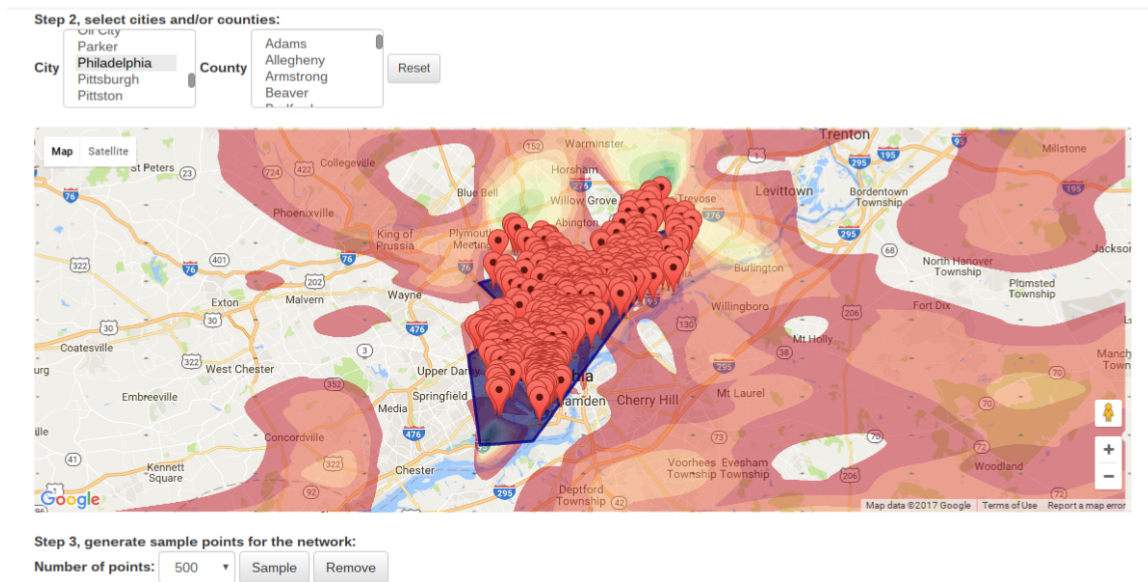


Figure 4.7: Screenshot of online PlanIt service.

¹<http://www.ebucknell.edu/planit>

Chapter 5

Distributed Queueing with Multiple Data Slots

In this chapter, we will first introduce Distributed queueing (DQ) and present its mathematical model and analysis. Then we will propose an improved DQ protocol designed for csLPWAN that we call DQ-N.

5.1 Distributed Queueing

DQ is a hybrid contention reservation medium access approach where the coordinator broadcasts contention-free transmission queue values to individual devices in response to contention-based transmission requests [44]. In contrast to most conventional media access approaches (i.e., CSMA, ALOHA, TDMA as discussed in Section 2.2), the coordinator maintains minimal information about the network nodes and is only responsible for processing and broadcasting the results of the contention resolution process. Each device in the network maintains two queue lengths, namely the length of the contention resolution queue (CRQ) and the length of the data transmission queue (DTQ). Using only this information, devices can compute contention-free transmit times in a fully distributed fashion, thus called distributed queueing.

Although DQ can be implemented in the frequency domain, this thesis focuses on the time domain. Users can exploit different frequency channels to increase the

base station capacity. In DQ, the channel is divided into multiple fixed-length frames. Each frame has three components:

1. several minislots for transmission requests (TR),
2. a contention-free data transmission slot,
3. and a feedback slot.

The structure of the frame is shown in Figure 5.1, where m is the number of TR minislots. Each device in the network is synchronized so that at the beginning of each frame devices can transmit a TR message using one of the minislots, send data contention free at the data transmission slot, and listen at the feedback slot.

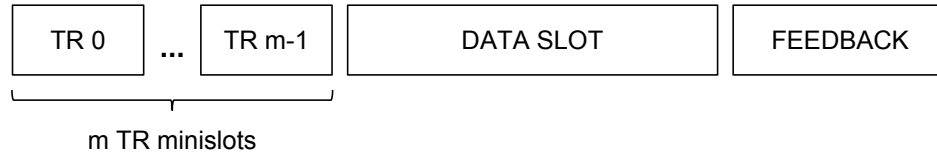


Figure 5.1: DQ frame structure

The benefit of DQ is that contention is greatly reduced by the coordinator broadcasting feedback containing the current CRQ and DTQ length as well as TR results. The TR results consist of an array of states for each of the minislots, namely, *idle*, *success*, and *contention*. These minislot states are used to compute the queue lengths at each node. The base station maintains a counter to calculate CRQ and DTQ length respectively. To do so, the base station will decrement each counter by one for each frame and then adjust the counters based on the results of each minislot. In other words, the queue lengths provided in the feedback enable each device to calculate their position in the contention resolution or data transmission queues individually.

To request a data slot, a device independently selects a minislot from $\{0, 1, \dots, m-1\}$ at random and then sends a TR at that chosen minislot in the next frame. Since there may be many devices attempting to transmit simultaneously and there are a fixed number of minislots, a contention may occur at that chosen minislot. Contention is assumed if the base station receives a corrupt packet (i.e., invalid CRC). If TR minislot is contended, the device will enter the CRQ, otherwise it will enter the DTQ. To calculate the device's CRQ position, it will scan the minislot states provided in the feedback from lower index to higher index. If a contention occurs in the scanned slot, the device will increase the current CRQ length by 1. The scanning process continues

until it reaches its requested minislot index. The value obtained by this process is the position of the device in the CRQ. Devices in the CRQ back off for the computed CRQ number of frames and repeat the TR process. Analogously, the position of DTQ is calculated by scanning success states in the feedback and incrementing the base DTQ value for each successful TR minislot. Devices in the DTQ wait for the indicated number of frames (the DTQ length) and then communicate without contention.

An example of DQ ($m = 3$) resolving a traffic burst caused by 7 devices sending packets simultaneously, as shown in Figure 5.2. At time $t = 0$, all of the devices send TR to the base station. Device 1, 2, 3, and 4 transmit TR at the same time, thus causing a collision, which will be detected by the base station. Since there is only one device (device 5) sending TR at mini slot 1, device 5 will enter DTQ immediately. Device 6 and 7 will enter CRQ following device 1, 2, 3 and 4. At $t = 1$, since device device 1, 2, 3, and 4 are at the head of CRQ, they will send TR again to the base station, which causes more collision and all of them will enter CRQ again. As device 5 is at the head of DTQ, it will transmit at this time frame. At time $t = 2$, device 6 and 7 will send TR and then enter DTQ as their TRs are successfully received by the base station, and so on. When $t = 8$, all the devices have finished transmission and the channel utility is $7/9 = 77.8\%$.

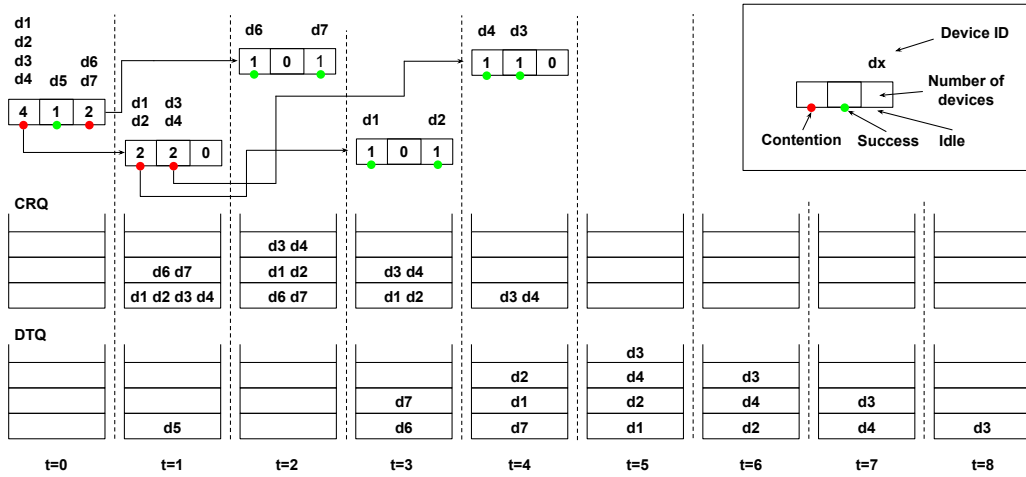


Figure 5.2: Example of how DQ distributes the transmission bursts into following frames.

The contention resolution algorithm of DQ is essentially a tree-splitting algorithm. All devices that transmitted a TR in the same minislot will compute the same CRQ value, hence they occupy a common branch in the contention tree. We can use this to calculate the expected waiting time in the system for a transmission burst. After showing the general equation for expected time, we will prove that if there are n

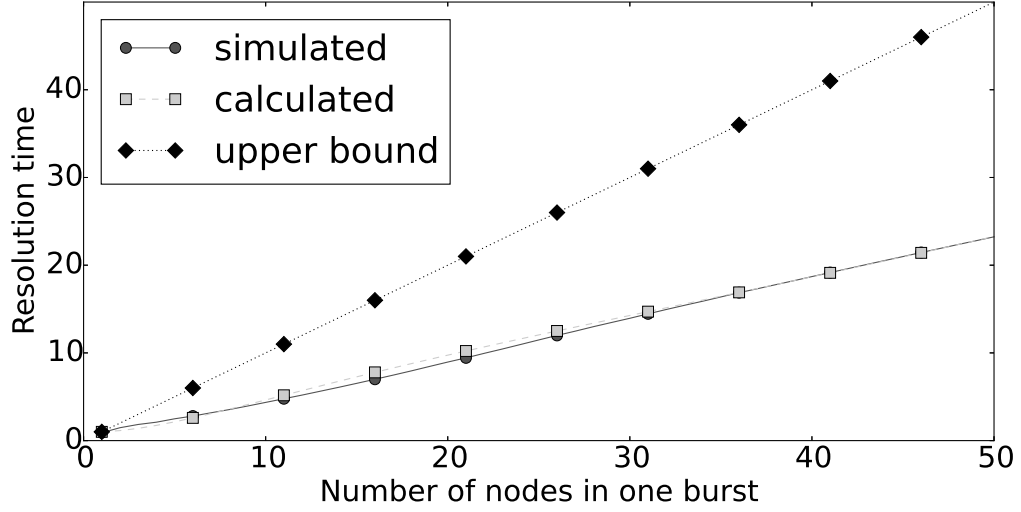


Figure 5.3: Burst contention resolution time, L_n .

nodes in a burst and the number of minislots, $m \geq 2$, then it takes no more than n frames to assign each node a unique DTQ value, this resolving the contention, as shown in Theorem 1. The proof for Lemma 1 is shown in the original DQ protocol paper [44] while Theorem 1 is part of this thesis' contribution.

Lemma 1. Let L_n be the expected time waiting in the queue. If the contention resolution queue is empty and there is a burst of n request signals, then

$$L_n = \begin{cases} 1 & \text{if } n \leq 1 \\ \frac{m^{n-1} + \sum_{k=2}^{n-1} \binom{n}{k} (m-1)^{n-k} L_k}{m^{n-1} - 1} & \text{otherwise.} \end{cases}$$

Proof. If there is no request or only one request, then by default L_n will be the minislot request time 1. If we have more than one request, then according to the tree-splitting algorithm, we need to consider the subsets. Suppose k out of n devices choose the minislot i , then the expected time for these k devices will be pL_k , where p forms a binomial distribution. Therefore we have

$$L_n = 1 + m \sum_{k=2}^n \binom{n}{k} \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{n-k} L_k.$$

After simplification we will obtain the desired equation. □

Theorem 1. It takes no more than n frames to resolve a burst of n messages when $m \geq 2$.

Proof. We will proceed the proof using mathematical induction. The base case is trivial to check. Suppose for some n the statement holds, then for $n + 1$ bursts, we have $L_{n+1} = 1 + (m - 1)\frac{L_n - 1}{m} + \frac{L_{n+1}}{m^n}$ given in Lemma 1. Hence we have

$$L_{n+1} \leq \frac{m^{n-1}}{m^n - 1} + \frac{m^n - m^{n-1}}{m^n - 1}n$$

using the induction hypothesis. Notice that $\frac{m^{n-1}}{m^n - 1} \leq 1$ and $\frac{m^n - m^{n-1}}{m^n - 1} \leq 1$ for all $m > 1$ and $n > 1$. Hence $L_{n+1} \leq 1 + n$ and this concludes the proof. \square

Figure 5.3 demonstrates the expected delay time versus the number of nodes in a burst. This is a very important feature to IoT applications as it guarantees the stability of the system during bursty loads.

To further explore the latency in a DQ system we let the expected arrival (input) rate be ρ . If we assume the arrivals can be determined by a Poisson process and the service time takes 1 unit time, the system can be lower bounded by an $M/D/1$ queue with wait time:

$$W_{M/D/1} = 1 + \frac{\rho}{2(1 - \rho)} \quad (5.1)$$

Figure 5.4 compares the $M/D/1$ delay and throughput by simulating DQ with various numbers of minislots, m ($m = 2$ is not shown due to poor performance). The initial gap between the simulation and the ideal $M/D/1$ results from DQ requiring at least two frames to transmit data: one containing the TR and feedback plus one for the actual data transmission. Nevertheless, DQ demonstrates near ideal channel utility under heavy traffic loads, a feature not shared by ALOHA-based protocols.

An important feature of DQ is that once the CRQ and DTQ values are computed, the device can switch to sleep mode to save energy and wake up at the scheduled time to transmit data. There is no need to continuously sense the network traffic load. This is an advantage to the IoT domain where channel sensing typically consumes significant device energy [45]. If a device detects that both CRQ and DTQ are empty, it may use any unused data slot, accepting the possibility of contention. This behavior is similar to slotted ALOHA and reduces the latency significantly when the network utilization is low.

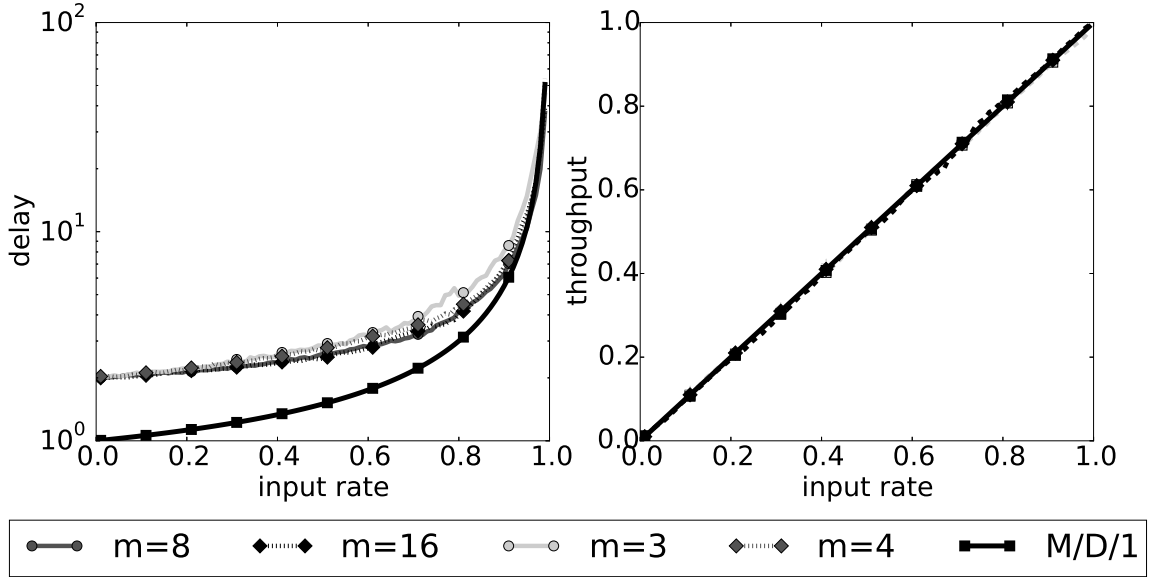


Figure 5.4: Comparison between M/D/1 and DQ.

5.2 DQ-N for csLPWANs

We decide to take advantage of DQ’s high network utility and adapt it into a more efficient under LPWAN environment, which we call DQ-N. The goal of DQ-N is to support N data transmission slots between each TR and feedback message.

Because current LPWAN radios use relatively small packet sizes, the data slot size is constrained by the maximum radio packet size in practice. As the TR and feedback messages are overhead, the overall protocol efficiency with small data slots is low. In addition, most of the LPWAN protocols cannot handle upstream and downstream in a single channel using the same encoding and protocol (e.g. Ingenu).

Figure 5.5 shows the DQ-N frame structure. The overall protocol efficiency can be tuned by varying N and m .

There are several improvements of DQ-N that are noteworthy:

1. N data slots in a single frame;
2. both upstream and downstream share the same protocol format and transmis-

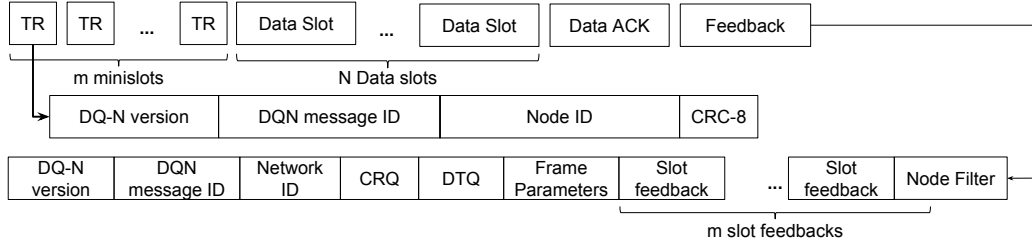


Figure 5.5: DQ-N frame structure.

sion channel;

3. a fixed-sized Bloom filter is appended at the end of feedback to provide better TR status feedback compensating for LoRa’s strong capture effect.
4. most of the parameters used in DQ-N, such as N and m , are encoded in the feedback frame so that the network configuration can be changed at runtime;
5. and Quality of Service (QoS) can be obtained through the introduction of an ACK frame.

In the TR packet, each node can request either 1 or 2 data slots. This will be encoded together with the TR type, such as upstream, downstream, or network join. Similarly, the slots status in the feedback has three corresponding statuses: idle (0), number of slots successfully requested (1 or 2), and contended (3). Upon receiving the feedback, the node should scan each slot status and compute CRQ and DTQ length. Contrary to traditional DQ, which increments DTQ by 1 for each successful slot, DTQ in DQ-N can be increased by either 1 or 2, depending on how many data slots are requested. Although CRQ is decremented by 1 for each frame, DTQ will be decreased by N , where N is the total number of data slots in the frame.

To support upstream and downstream traffic simultaneously, each node needs to register their unique hardware address (typically 6 bytes) with the base station so that base station can keep track of downstream requests from the upper layer protocols. However, since LPWAN uses an extremely low data rate, it is impractical to send entire hardware address in every TR. Our solution to this is to use the DTQ to handle node joining and registering. To join the network, the node sends a TR requesting to join the network and request 2 data slots. If these two data slots are requested successfully, the node will enter the DTQ (as with normal upstream traffic). When it is the node’s turn to transmit, it will first transmit a join packet containing its complete hardware address, then in the next data slot, the base station will respond

with an assigned node ID (2 bytes). To reduce node ID waste, if the node is inactive for a certain amount of time, the base station will recycle the node ID and disconnect the associated node.

Although we have CRC-8 in the TR packet, due to capture effect, even though there is packet collision, the TR with the strongest signal strength arriving at the base station will be demodulated correctly. As a result, even there are multiple TRs in a single minislot, the base station may fail to detect contention. Our solution is to use a standard Bloom filter. The base station adds the node ID, whose TR has been successfully received to a Bloom filter contained in the feedback message. Once the node receives the Bloom filter contained in the feedback, the node can test if its own TR was received correctly.

To reach a mutual agreement about DQ-N parameters between the base station and nodes, these parameters, called frame parameters, are encoded in the feedback packet. As the node needs to synchronize the clock when starting up and it is usually done by listening to the feedback packet, the node can learn all the frame parameters during synchronization stage. Therefore, once the node finishes synchronizing the clock, it can send the joining request immediately. In addition, encapsulating frame parameters also allows the base station to change the network configuration at runtime, for instance, increase the number of mini slots, m , to reduce contention. To do so, the base station will mark every upcoming TR as contended and assign CRQ to a large value enough to make all node IDs expire. After receiving all the pending upstreaming traffic (DTQ reaches 0), the base station can then proceed to change its frame parameters and reset CRQ.

DQ-N also provides support for QoS by sending acknowledgements for upstream packets. For each transmitted data slot, if the base station receives the packet successfully, it will set the corresponding bit in the ACK message to 1. Nodes that require QoS to deliver important content can use the ACK message to ensure packet delivery. However, receiving the ACK message is optional and nodes can just skip it if they do not require QoS.

More detailed specification of DQ-N protocol can be found in Appendix A.

For further analysis, since DQ-N supports requesting multiple data frames in each TR, it is reasonable to assume the total number of requested data slots per TR forms a distribution with mean value λ and standard deviation σ_a . Accordingly, we compare DQ-N to an M/G/1 queue with input ratio $\rho = n\lambda/N$. The following list defines constants used in the analysis of DQ-N:

ρ : server utilization (traffic load).

N : number of data slots per frame.

m : the number of minislots per frame.

λ : average number of data slots per TR.

σ_a : standard deviation of data slots for each TR.

σ_s : standard deviation of service time.

γ : average number of TR per frame.

By Little's Law and the Pollazcek-Khintchine formula, the average delay time in DQ-N based $M/G/1$ queue is: $W_{M/G/1} = \frac{L_q}{\lambda} + \frac{\lambda}{N} = \frac{\lambda^2 \sigma_s^2 + \rho^2}{2\lambda(1-\rho)} + \frac{\lambda}{N}$, where $\sigma_s^2 = \sigma_a^2/N^2$. If we know the the expected value of λ and σ_a is relatively small compared to N , we can choose the number of mini-slots based on λ , as given in Theorem 2.

Theorem 2. It takes no more than n frames to resolve λn TRs if $m \geq \gamma + 1$.

Proof. Since the server can process N data slots for each unit time, one can show that it is sufficient to prove that $L_{(\lambda n + \lambda)/N} \leq 1 + L_{(\lambda n)/N}$. Hence it is equivalent to show $L_{n+1} \leq \frac{N}{\lambda} + L_n$. According to DQ theory we need $\frac{m^{n-1}}{m^n - 1} \leq \frac{\lambda}{N} = \frac{1}{\gamma}$. Since $\frac{m^{n-1}}{m^n - 1}$ is monotonically decreasing, and $m \geq \gamma + 1$, we have $\frac{m^{n-1}}{m^n - 1} \leq \frac{1}{\gamma}$, as desired. \square

5.2.1 Simulation

To simulate the performance of DQ-N, we first use PlanIt to generate device locations in three counties in central Pennsylvania, namely Union, Northumberland, and Snyder County. The test area covers 2,976 km² with a population of 180 thousand. 5,000 locations are randomly drawn based on the procedure discussed earlier. We filter out any locations with a negative signal to noise ratio (SNR), as those devices are unlikely to transmit to or receive any valid information from the base station. We also assume each device communicates to the base station with the lowest path loss. The path loss values are converted to bit error rate (BER) using the following equations.

$$SNR = TI + GT - NJ - L + G - NF$$

$$\frac{E_b}{N_0} = SNR - 10 \log_{10} \left(\frac{f_b}{B} \right)$$

$$BER = \frac{1}{2} e^{-\frac{1}{2} \frac{E_b}{N_0}}$$

The meanings and values of the parameters are listed in Table 5.1.

For the base station transmitter, we use $\frac{1}{2}$ Watt as the transmitter power and 200 ms for the duration of the time slot, to comply with FCC regulations for frequency hopping systems in the 900 MHz band. We select 1200 bps datarate and 12.5 KHz channel bandwidth to improve receiver sensitivity and therefore increase the effective range. The remaining transceiver parameter values are obtained from the TI CC1120 and CC1190 narrowband transceiver and range extender datasheets [46].

Table 5.1: Parameters used to calculate BER

Name	Value
RF power delivered to the transmitter (TI)	-3 dBW
Transmitter antenna gain (GT)	3 db
Johnson Noise (NJ)	-114 dBW
Path loss (L)	Computed by PlanIt
Receive antenna gain (G)	3 db
Receiver noise figure (NF)	7 db
Channel data rate (f_b)	1200 bps
Channel bandwidth	12.5 KHz
PHY model	BPSK

For comparison, we also simulate P-persistent CSMA, statically allocated TDMA, and LPDQ in addition to DQ-N. These protocols are chosen because they form the basis for many popular MAC layer protocols. P-persistent CSMA operates like traditional CSMA but a node only transmits on idle channels with probability P (we use $P = 0.001$) to reduce contention. Statically allocated TDMA assumes a static round-robin transmit schedule for all nodes. LPDQ is a previous implementation of distributed queuing for low-power wireless networks. DQ-N improves upon LPDQ by scheduling multiple data slots per frame resulting in reduced protocol overhead

for the short frame sizes common in LPWAN systems and supporting both upstream and downstream traffic.

We performed two different simulations, the ideal case without packet loss and a realistic case with packet loss/corruption caused by the path loss predicted by PlanIt. For both cases, we generate sufficient upstream traffic to saturate the network evenly distributed among all of the nodes (e.g., if there are 100 nodes and 1,200 bps available upstream bandwidth, each node would generate 12 bps of upstream traffic) and then sample the network utility and latency after a warm up period. Network utility is defined as the ratio of successfully received data without contention or corruption and the possible maximal data. The latency is defined as the time from when a node begins to send a packet to the time when a base station successfully receives it, thus including the contention and waiting time. No retransmissions are used in the simulations as such effort will confound the simulation results. In addition, retransmission typically belongs to upper layer protocols and we are only interested in MAC layer performance. We also measure the duty cycle for each node to demonstrate the energy intensity of each protocol.

Transmitted packet sizes are selected from a normal distribution with a mean of 240 bytes and standard deviation of 120 bytes. Since the total available bandwidth is 1,200 bps and the maximum time slot is 200 ms, each data slot contains 30 bytes of data. We choose $N = 16$ and $m = 8$ for the DQ-N parameters. We also subtract the protocol overhead when calculating maximum channel utility, and therefore use 0.94 for DQ-N and 0.8 for LPDQ which saturates the network using these parameters.

Figure 5.6 shows the simulation results for an ideal environment where there is no packet loss. In this case, DQ-N achieves similar performance to LPDQ but demonstrates lower latency. However, since DQ-N reduces protocol overhead, it has higher channel utility.

Figure 5.7 shows the simulated utility and delay considering packet loss. The latency for all protocols except TDMA increases when compared to the idea case. The channel utility for all protocols is reduced due to packet loss. For DQ-N and LPDQ, in addition to lost data messages, a lost TR or feedback message may result in an unscheduled data slot, further reducing utility. However, DQ-N still outperforms LPDQ and P-CSMA in terms of channel utility and latency.

Figure 5.8 shows the radio duty cycle distribution in the realistic environment with 2,500 nodes. Although TDMA exhibits the optimal duty cycle ($1/2500$), it is not practical as the number of nodes has to be fixed and dynamically changing

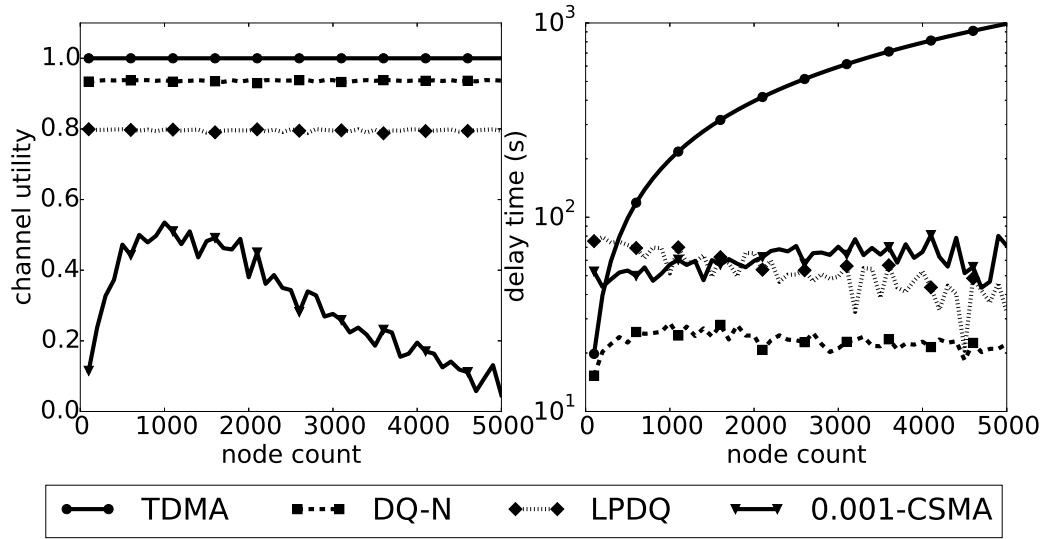


Figure 5.6: Simulation results for the ideal environment.

the TDMA schedule requires a more complex protocol. Nevertheless, the simulation shows the improvement of DQ-N over LPDQ as it eliminates the need for nodes to make multiple TRs for messages longer than one data slot.

5.3 Implementation

We implemented the DQ-N v0.27 (specified in Appendix A) on hardware. Table 5.2 shows the components and cost of the base station and end devices. The PHY layer is provided by a LoRa radio. The reason why we choose Adafruit Feather development boards is that they are Arduino compatible. Therefore, the node and base station can share the same core DQ-N implementation, thus reducing the prototyping time.

As we can see from Table 5.2, the total cost for a base station is \$85. This is less than a tenth of what TTN charges for a base station. As a result, we believe people will be more willing to set up base stations contributing to the crowdsourced network.

We tested our implementation of DQ-N with one base station and 3 nodes, as shown in Figure 5.9. These nodes can register their unique hardware addresses, send data to the base station, and receive data from the base station. Listing 5.1 shows the output from an example run of a base station communicating with a single node.

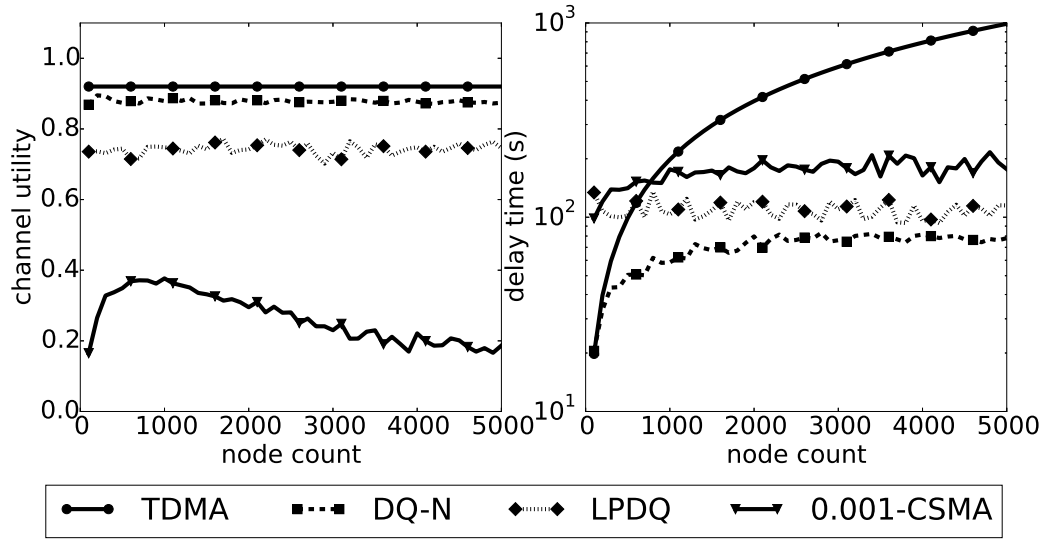


Figure 5.7: Simulation results for the realistic environment.

Table 5.2: Components and costs for single DQ-N base station and end device

Base Station	
Name	Cost
Raspberry Pi 3A	\$35
LoRa Radio HAT	\$50
Total	\$85
Node	
Adafruit Feather M0 RFM95	\$34.95

The network configuration is shown at the beginning.

The node first tried to register in the network. It sends a registration TR to the base station with minislot 13, then enter DTQ at 0. After sending its hardware address 42 : 43 : 44 : 45 : 46 : 47, it obtained its node ID, which is 37532 obtained from a hash function. Then it slept for a short period and started to transmit a 48-byte packet. After transmitting the packet, it continued to send another 56-byte packet.

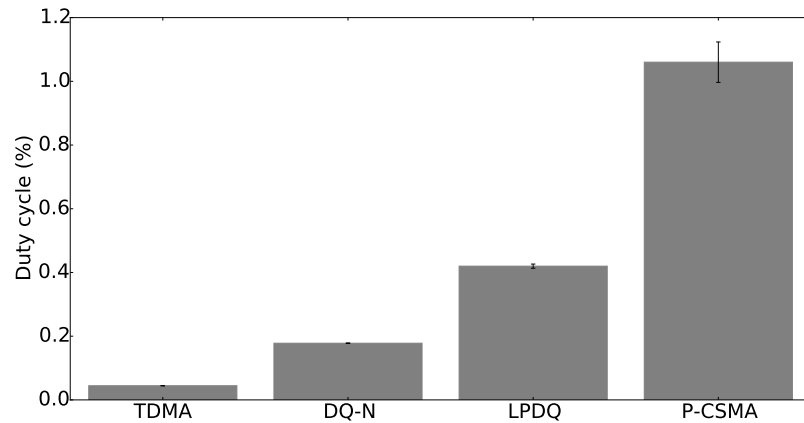


Figure 5.8: Radio duty cycle in the realistic environment.

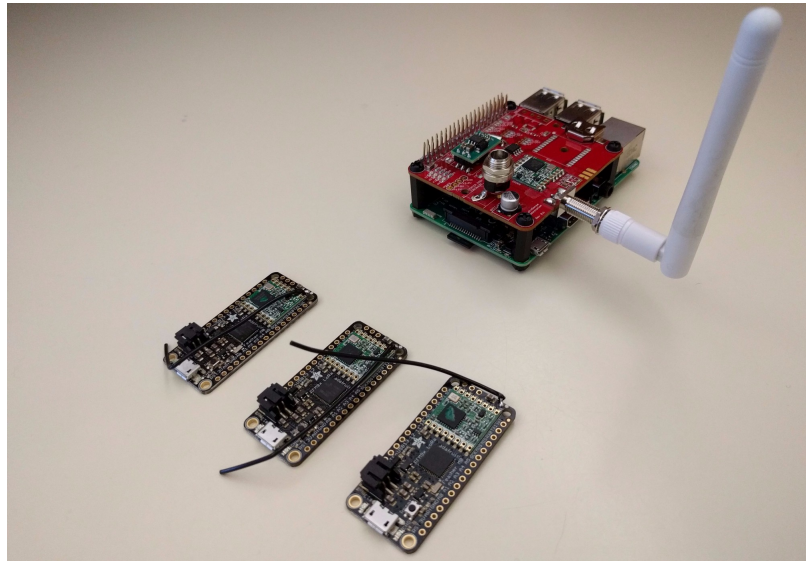


Figure 5.9: Setup with 3 Adafruit Feather M0 RFM95 boards and 1 Raspberry Pi 3 with LoRa Radio Hat.

Listing 5.1: Base station’s console output for a test run, where a single device registered and sent data to the base station.

```

----- DQN information -----
frame length: 6253
num of TR: 16    num of data slots: 6
data length: 478 ms    feedback length: 609 ms ACK length: 216
-----

TR received at 13 (time: 15953). Version: 27 message id: 92
-----FEEDBACK-----
    
```

CHAPTER 5. DISTRIBUTED QUEUEING WITH MULTIPLE DATA SLOTS 44

```
frame param: 4601 CRQ: 0 DTQ: 0
TR result:
00000000 00000000 00000000 00100000
```

```
-----
assign nodeid: 37532 for HW: 42:43:44:45:46:47
```

```
-----FEEDBACK-----
frame param: 4601 CRQ: 0 DTQ: 0
TR result:
00000000 00000000 00000000 00000000
```

```
-----
TR received at 9 (time: 27839). Version: 27 message id: 81
```

```
-----FEEDBACK-----
frame param: 4601 CRQ: 0 DTQ: 0
TR result:
00000000 00000000 00010000 00000000
```

```
-----
total len is 2
42:43:44:45:46:47 sent a message
00 01
```

```
-----FEEDBACK-----
frame param: 4601 CRQ: 0 DTQ: 0
TR result:
00000000 00000000 00000000 00000000
```

```
-----
TR received at 8 (time: 40193). Version: 27 message id: 82
```

```
-----FEEDBACK-----
frame param: 4601 CRQ: 0 DTQ: 0
TR result:
00000000 00000000 10000000 00000000
```

```
-----
total len is 48
42:43:44:45:46:47 sent a message
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29
```

```
-----FEEDBACK-----
frame param: 4601 CRQ: 0 DTQ: 0
TR result:
00000000 00000000 00000000 00000000
```

```
-----
TR received at 7 (time: 52544). Version: 27 message id: 82
TR: 7 num of slots: 2
```

```
-----FEEDBACK-----
frame param: 4601 CRQ: 0 DTQ: 0
TR result:
00000000 00000010 00000000 00000000
```

```
-----
total len is 56
```

CHAPTER 5. DISTRIBUTED QUEUEING WITH MULTIPLE DATA SLOTS 45

```
42:43:44:45:46:47 sent a message
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37
```

Chapter 6

Conclusion

In this thesis, we have presented our platform for large-scale regional IoT networks. Within the network, thousands of devices can connect directly without resorting to a local gateway device such as a smartphone or a WiFi router. As a result, an IoT device can freely roam inside a region without losing network connectivity.

To substantiate our vision, we have proposed several novel concepts, frameworks and protocols, namely, csLPWAN, PlanIt, and DQ-N. We have defined the term csLPWAN as a LPWAN where a subset of users provide base stations without coordinating the location of the base stations. To better understand csLPWAN behavior, we have created a csLPWAN network planning tool, PlanIt. PlanIt combines topographic and demographic information to give a more realistic representation of real-world csLPWAN connectivity. Using PlanIt, we found the median connectivity over all cities in Pennsylvania was 99.2% from a single randomly located base station within each city. Using the path loss data from PlanIt, we have also simulated and compared the efficiency of three different LPWAN protocols. To simplify csLPWAN gateways and improve network utilization, we have designed and analyzed DQ-N, an extension of the DQ protocol optimized for highly utilized low-rate csLPWAN networks. By presenting mathematical analysis and numeric simulation results, we show the superior performance of DQ-N and demonstrate the ability of a single DQ-N base station to support thousands of nodes in a csLPWAN.

We believe these results will help catalyze the deployment of future csLPWAN networks. We are currently developing a low-cost base station to support networks of thousands of LoRa devices using the DQ-N protocol. More information on this work

and a web-based version of PlanIt are available at <http://cslpwan.me>. Current implementation of DQ-N conforming the current v0.27 specification is available at github.com/Kuree/DQN.

Future work includes refining the DQ-N protocol and polishing its implementation. In addition, we would like to deploy the entire csLPWAN platform in real world to validate its feasibility and performance. We would also like to conduct a large-scale survey on IoT device deployment to verify our mathematical model used in PlanIt. This survey will focus on IoT device owner's home address, their daily usage of the devices, number and type of IoT devices used in their houses and so on. We believe collecting this information will help us furthermore understand csLPWAN and improve PlanIt's model.

References

- [1] E. Oh, B. Krishnamachari, X. Liu, and Z. Niu, "Toward dynamic energy-efficient operation of cellular network infrastructure," *IEEE Communications Magazine*, vol. 49, no. 6, 2011.
- [2] G. Y. Li, Z. Xu, C. Xiong, C. Yang, S. Zhang, Y. Chen, and S. Xu, "Energy-efficient wireless communications: tutorial, survey, and open issues," *IEEE Wireless Communications*, vol. 18, no. 6, 2011.
- [3] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "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*, pp. 59–67, 2016.
- [4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013.
- [5] F. Bing, "Research on the agriculture intelligent system based on IoT," in *Processing of IEEE International Conference on Image Analysis and Signal*, pp. 1–4, 2012.
- [6] N. Bui and M. Zorzi, "Health care applications: a solution based on the internet of things," in *Proceedings of the 4th ACM International Symposium on Applied Sciences in Biomedical and Communication Technologies*, p. 131, 2011.
- [7] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of iot for environmental condition monitoring in homes," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3846–3853, 2013.
- [8] A. H. Inc., "August smart lock." <http://august.com/products/august-smart-lock/>, 2016.

- [9] T. Huntington, “Nightstand central - a music alarm clock with sleep timer, weather, and photos.” <https://itunes.apple.com/us/app/nightstand-central-music-alarm/id392479477>, 2016.
- [10] Scanomat, “Topbrewer apple-watch.” <http://www.scanomat.com/topbrewer/apple-watch>, 2016.
- [11] M. Aazam, I. Khan, A. A. Alsaffar, and E.-N. Huh, “Cloud of things: Integrating internet of things and cloud computing and the issues involved,” in *Proceedings of 11th IEEE International Bhurban Conference on Applied Sciences & Technology*, pp. 414–419, 2014.
- [12] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, and P. Dutta, “The internet of things has a gateway problem,” in *Proceedings of the 16th ACM International Workshop on Mobile Computing Systems and Applications*, 2015.
- [13] S. Trifunovic, A. Picu, T. Hossmann, and K. A. Hummel, “Slicing the battery pie: Fair and efficient energy usage in device-to-device communication via role switching,” in *Proceedings of the 8th ACM MobiCom Workshop on Challenged Networks*, CHANTS ’13, pp. 31–36, ACM, 2013.
- [14] W. Commons, “ALOHA,” 2017. File: `Pure_ALOHA1.svg`.
- [15] J. F. Kurose and K. Ross, *Computer networking: A top-down approach*. Pearson, 2005.
- [16] W. Commons, “ALOHA,” 2017. File: `Slotted_ALOHA.svg`.
- [17] G. Miao, J. Zander, K. W. Sung, and S. B. Slimane, *Fundamentals of Mobile Data Networks*. Cambridge University Press, 2016.
- [18] R. Gold, “Optimal binary sequences for spread spectrum multiplexing (corresp.),” *IEEE Transactions on Information Theory*, vol. 13, pp. 619–621, October 1967.
- [19] Semtech, “FCC regulations for ISM band devices: 902 - 928 MHz,” tech. rep., Semtech, June 2006.
- [20] A. Broder and M. Mitzenmacher, “Network applications of bloom filters: A survey,” *Internet mathematics*, vol. 1, no. 4, pp. 485–509, 2004.
- [21] A. Kumar, J. Xu, and J. Wang, “Space-code bloom filter for efficient per-flow traffic measurement,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2327–2339, 2006.

- [22] H. Song, S. Dharmapurikar, J. Turner, and J. Lockwood, "Fast hash table lookup using extended bloom filter: An aid to network processing," in *Proceedings of the ACM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 181–192, 2005.
- [23] A. O. Allen, *Probability, statistics, and queueing theory*. Academic Press, 2014.
- [24] Ingenu™, "Ingenu - dedicated machine connectivity for IoT." <https://www.ingenu.com/>, 2017.
- [25] J. Schneider and K. Garvin, "Ingenu announces launch of the machine network," *Ingenu Press Release*, 2015.
- [26] T. J. Myers, D. T. Werner, K. C. Sinsuan, J. R. Wilson, S. L. Reuland, P. M. Singler, and M. J. Huovila, "Light monitoring system using a random phase multiple access system," 2013.
- [27] G. Margelis, R. Piechocki, D. Kaleshi, and P. Thomas, "Low throughput networks for the IoT: Lessons learned from industrial implementations," in *Proceedings of 2nd IEEE World Forum on Internet of Things*, 2015.
- [28] N. S. (Semtech), M. L. (Semtech), T. E. (IBM), T. K. (IBM), and O. H. (Actility), "LoRaWAN specification," tech. rep., LoRa Alliance, January 2015.
- [29] SigFox™, "Sigfox - The Global Communications Service Provider for the Internet of Things (IoT)." <https://www.sigfox.com/>, 2017.
- [30] Intel™, "Sigfox - Intel Capital Portfolio Company." <http://www.intelcapital.com/portfolio/company.html?id=16892>, 2017.
- [31] M. T. Do, C. Goursaud, and J. M. Gorce, "On the benefits of random FDMA schemes in ultra narrow band networks," in *Proceedings of International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2014.
- [32] Intel™, "The Things Network." <https://www.thethingsnetwork.org/>, 2017.
- [33] P. Tuset-Peiro, F. Vazquez-Gallego, J. Alonso-Zarate, L. Alonso, and X. Vilajosana, "Experimental energy consumption of frame slotted ALOHA and distributed queuing for data collection scenarios," *Sensors*, 2014.
- [34] P. Tuset-Peiro, F. Vazquez-Gallego, J. Alonso-Zarate, L. Alonso, and X. Vilajosana, "LPDQ: A self-scheduled TDMA MAC protocol for one-hop dynamic low-power wireless networks," *Pervasive and Mobile Computing*, 2015.

- [35] A. Laya, C. Kalalas, F. Vazquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Goodbye, ALOHA!," *IEEE Access*, 2016.
- [36] B. Reynders, W. Meert, and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," in *Proceedings of 23rd IEEE International Conference on Telecommunications*, May 2016.
- [37] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios," *CoRR*, 2015.
- [38] Q. Zhu, M. Y. S. Uddin, Z. Qin, and N. Venkatasubramanian, "Upload planning for mobile data collection in smart community internet-of-things deployments," in *Proceedings of IEEE International Conference on Smart Computing*, May 2016.
- [39] Y. A. Mtawa, H. S. Hassanein, and N. Nasser, "Identifying Bounds on Sensing Coverage Holes in IoT Deployments," in *Proceedings of IEEE Global Communications Conference*, Dec 2015.
- [40] S. Kasampalis, P. I. Lazaridis, Z. D. Zaharis, A. Bizopoulos, S. Zetlas, and J. Cosmas, "Comparison of Longley-Rice, ITM and ITWOM propagation models for DTV and FM broadcasting," in *Proceedings of International Symposium on Wireless Personal Multimedia Communications*, June 2013.
- [41] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE transactions on Vehicular Technology*, 1980.
- [42] US Census Bureau, "United states census," 2010.
- [43] P. Guth, "Geomorphometric comparison of ASTER GDEM and SRTM," in *A special joint symposium of ISPRS Technical Commission IV & AutoCarto in conjunction with ASPRS/CaGIS*, 2010.
- [44] W. Xu and G. Campbell, "A distributed queueing random access protocol for a broadcast channel," in *Proceedings of ACM Communications Architectures, Protocols and Applications*, 1993.
- [45] M. A. Razzaque and S. Dobson, "Energy-efficient sensing in wireless sensor networks using compressed sensing," *Sensors*, 2014.
- [46] Instruments, "CC1120 high-performance rf transceiver for narrowband systems," *Datasheet Available Online at: <http://www.ti.com/lit/ds/symlink/cc1120.pdf>*, 2015.

Appendices

Appendix A

DQ-N Protocol Specification Version 0.27

There are three major components in a DQ-N frame, namely, TR slots, data frames, and a feedback frame, as shown in Figure 5.5. Currently two transmission rates are supported. Without explicitly mentioning transmission rate, all packets are transmitted using the slower radio modem configuration. Each node is allowed to request up to 2 data slots.

One thing to notice is that the entire network can be reconfigured at runtime, which can be done at the synchronization state, as shown in Section A.1.1. This is done by encoding network information into the feedback packet. In addition, since this version is implemented on the LoRa radio, there are some extra details we need to take care of. LoRa radio by default will include a header and CRC-8 at the end. To save space and enable collision detection, the protocol requires no header and no CRC mode in TR transmission. However, for normal transmission, we can use the built-in CRC to drop corrupted packets automatically.

A.1 Protocol Overview

This section describes the behaviors of the node and the base station, including registration and how to send and receive data from the network.

A.1.1 Node Synchronization

Before transmitting any data to the base station, the node has to synchronize its clock to the base station's. To do so, it will continue listening to the channel until it receives a valid feedback packet. It then calculate all the DQ-N parameters encoded in the feedback packet and length for each sub-frames. Using these values the device can compute the time when the frame starts and thus finish the synchronization.

To take clock drifting into account, the node needs to remember the last time when it synchronized with the base station. If it has been too long, i.e. the current timestamp minus last synchronized time exceeds a predefined threshold, the node has to synchronize again before sending any data.

To change the network configuration at runtime, the base station needs to make sure DTQ to be empty so that no data will be lost. To do so, the base station will set CRQ length value in the feedback to a reasonably large value and mark all TRs as contended. Therefore every device has to enter CRQ and sleep. Given the size of CRQ values, it is easy to put all devices out of synchronization. Hence once these nodes try to sending data to the newly reconfigured network, they have to synchronize again and obtain the new network configuration.

A.1.2 Node Joining Process

To join, the node has to send a special TR request, asking for two data slots. Once TR is successfully sent, the device will enter DTQ. At the first data slot, the device will send a join packet containing its hardware address. At the second data slot, the base station will send a join response with a node ID assigned to that node. Once the node obtains its node ID, the joining process is finished. The TR is sent without a header and CRC using slow transmission rate.

A.1.3 Node Sending and Receiving Data

Sending and receiving data is very similar to the joining process except the Message ID attributes (see Section A.2) are different. Currently there is no downstream flag in the feedback and the node has to actively request downstream packets by sending

corresponding TRs.

A.1.4 Protocol Sequence Example

An overview of the joining and sending process is shown in Figure A.1. Figure A.1a shows that the node listens to a feedback packet and decodes the network information as well as synchronizing the clock. Then it sends a TR-JOIN (a special TR) to the base station. The TR-JOIN is successfully received by the base station and the device enters DTQ immediately. Upon sending packets, the node first sends its entire hardware address, then receives the assigned node ID. Figure A.1b shows how a device sends data to the base station. First it sends a TR. Unfortunately there is a contention and the device has to enter CRQ. After another TR request the device successfully enters DTQ. When it is the device's turn to transmit, it sends data to the base station.

A.2 Frame Details and Equation

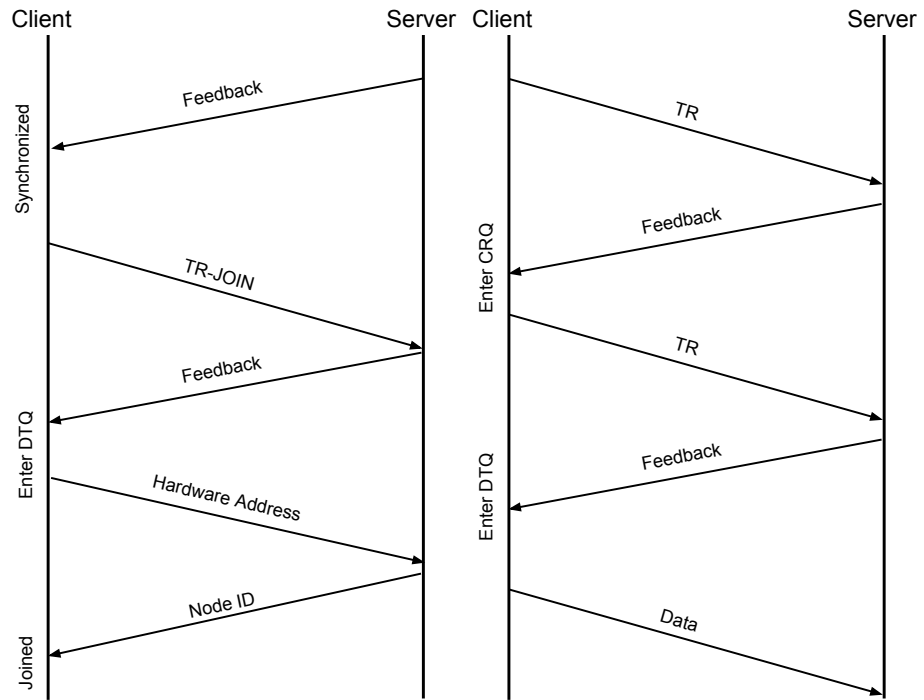
To calculate the size of a Node Filter, which is essentially a bloom filter, the following equation is used to calculate the size given a specific error rate and entry size.

$$m = n \log_2 e \cdot \log_2(1/\epsilon), \quad (\text{A.1})$$

where ϵ is the false positive (error) rate and n is the number of entries. The error rate will be encoded into the feedback back and n is simply the number of TRs per frame.

For all protocol messages, the version has to be 0x27 and the message id has to be one of the following listed in Table A.1.

The structures for TR, feedback, JOIN-REQ, and JOIN-RESP are shown in Table A.2 A.3, A.5, and A.6, respectively. Notice that TR-JOIN shares the same structure as TR, yet the Node ID will be set to 0 and the message ID will be different. For a TR, upper 4 bits of message ID are fixed and the lower 4 bits carry the TR request information, as shown in Table A.7.



(a) DQ-N joining procedure. (b) DQ-N send procedure. Notice that the first TR has a contention and as a result, a second TR needs to be sent.

Figure A.1: DQ-N protocol sequence diagram.

Table A.1: MessageID values table for DQ-N version 0.27

Type	Value/Mask
TR	0x8X
Feedback	0x01
TR-JOIN	0x92
JOIN-REQ	0xA0
JOIN-RESP	0xA1

Table A.2: TR structure for DQ-N version 0.27

Attribute	Size
DQN version	1 Byte
DQN MessageID	1 Byte
NodeID	2 Bytes
CRC	1 Byte

Table A.3: Feedback structure for DQ-N version 0.27

Attribute	Size
DQ-N Version	1 Byte
DQ-N MessageID	1 Byte
DQ-N NetworkID	4 Bytes
Unix Timestamp	4 Bytes
CRQ Length	2 Bytes
DTQ Length	2 Bytes
Frame Parameters	2 Bytes
Slots	Determined by Equation
Node Filter	Determined by Equation A.1

Table A.4: TR-JOIN structure for DQ-N version 0.27

Attribute	Size
DQN version	1 Byte
DQN MessageID	1 Byte
Hardware Address	6 Bytes

Table A.5: JOIN-REQ structure for DQ-N version 0.27

DQN version	1 Byte
DQN MessageID	1 Byte
Hardware Address	6 Bytes

Table A.6: JOIN-RESP structure for DQ-N version 0.27

Attribute	Size
DQN version	1 Byte
DQN MessageID	1 Byte
Hardware Address	6 Bytes
NodeID	2 Bytes

Table A.7: Message ID values for TR structure

Name	Bits	Description
Num slots	0-1	0, 1, 2
Up/down	2	0 = upstream to base, 1 = downstream to node
Rate	3	0 is low trans rate and 1 is high rate

The frame parameter in Feedback is defined in Table A.8.

Table A.8: Encoding details of frame parameters in feedback structure

Name	Bits	Description
FPP	0-1	False Positive Probability (0=0.1%, 1=1%, 2=2%, 3=5%)
TRF	2-7	TRs per Frame = $16 + 4 \times TRF$
DTR	8-11	Data Slots per TR, number of data slots = $\lfloor \frac{DTR}{15} (16 + 4 \times TRF) \rfloor$
MPL	12-15	Max payload in bytes = $6 \times (MPL + 1)$