EMPOWERING INTERNET OF THINGS WITH THE EMERGING WIRELESS INFRASTRUCTURES AND TECHNOLOGIES

By

Deliang Yang

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering – Doctor of Philosophy

2021

ABSTRACT

EMPOWERING INTERNET OF THINGS WITH THE EMERGING WIRELESS INFRASTRUCTURES AND TECHNOLOGIES

By

Deliang Yang

Wireless technologies have been evolved rapidly, whose infrastructures are built and delivered speedily. The emerging wireless technologies offer new solutions for data communication, monitoring, sensing, and edge computing, etc. The fast growth of wireless networks generates not only opportunities for new applications, but also issues in high energy consumption, unexpected latency, and potential privacy breach. In this dissertation, we propose two novel cyber-physical systems to demonstrate the possibility of empowering new IoT services and applications by leveraging the emerging wireless charging infrastructures and benchmarking the energy performance of end nodes in low-power wireless networks, respectively.

First, we present QID, the first system that identifies a Qi-compliant device during wireless charging in real-time using wireless charging fingerprints. QID employs a 2-dimensional motion unit to emulate a variety of multi-coil designs of Qi, which allows for fine-grained device fingerprinting. With the novel mobile coil design and a set of novel fingerprints from oscillator and controller patterns, QID achieves high device recognition accuracy by using ensembled Machine Learning algorithms. With the prevalence of public wireless charging stations, our results also have important implications for mobile user privacy.

Second, we develop a novel benchmarking ecosystem, called *NB-Scope*, to study the energy performance of the Narrowband Internet of Things (NB-IoT) network. NB-Scope adopts a hierarchical design, resolving the heterogeneity in network operators, node module vendors, and location profiles, to allow for the fusion of fine-grained diagnostic traces and current measurement. We then conduct a large-scale field measurement study con-

sisting of 30 nodes deployed at over 1,200 locations in 3 regions for three months. Our in-depth analysis of the collected 49 GB traces showed that NB-IoT nodes yield significantly imbalanced energy consumption in the wild, up to a ratio of 75:1, which may lead to short battery lifetime and frequent network partition. By extensive data analysis, we identify several key factors, including diverse network coverage levels, long-tail power profile, and excessive control message repetitions, that lead to high variance in the energy performance.

This dissertation is dedicated to my family and friends.

ACKNOWLEDGEMENTS

There would be no possibility for me to finish the work presented in this thesis without the guidance from my dissertation guidance committee, the help from my colleagues, as well as the support from my family members. I owe my sincere gratitude to all these people who make this thesis possible.

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Guoliang Xing, for his guidance and generous support throughout my entire doctoral study. He guided me to build up my research vision and capability. He has always been my role model in conducting high-standard research. I would not be an independent and enthusiastic researcher without his guidance. This rewarding experience will benefit my entire life.

Besides my advisor, I would like to thank the rest of my guidance committee members: Dr. Richard Enbody, Dr. Jian Ren, and Dr. Mi Zhang, for their strong encouragement, critical comments, and helpful suggestions. In particular, Dr. Ren provided me important suggestions on my early-stage projects. I also discussed many important research topics with Dr. Enbody in the middle of my program. Dr. Zhang gave me important feedback on my research projects and suggestions on my career paths.

Special thanks to Dr. Xiaobo Tan, who provided me much help and suggestions during the short transition period between my research projects.

I sincerely thank my fellow lab-mates in the eLANS lab: Dr. Ruogu Zhou, Dr. Dennis Philips, Dr. Mohammad Moazzami, Dr. Jinzhu Chen, Dr. Yu Wang, Dr. Tian Hao, Dr. Chen Qiu, Dr. Jun Huang, Dr. Chongguang Bi, and Linlin Tu. Particularly, I would like to thank Dr. Zhou for his tremendous support at my early stage of research. I cannot express how much I have learned from him. To me, he is a sincere friend, an enthusiastic collaborator, and an inspiring mentor. I also thank Dr. Huang, for his professional comments that steered me in the right paper direction. He also inspired me with a wide range

of possible directions during the exploration stage of my research projects. I would not forget the time that I spent with Ruogu and Chongguang both inside and outside the lab, which was a wonderful memory during my Ph.D. study.

To all my friends, thank you all so much for always standing by me and delighting my journey. I wish I could name you all. I wish you the best of luck wherever you are.

Particularly, I would like to sincerely thank my parents, whose unconditional love, sacrifice, and encouragement accompany me since the beginning of my life.

Special thanks to my soul-mate, Mengying Sun. We stand together, explore the world together. We celebrate the good times, encourage each other in the bad times, and blueprint the future. I found another meaning of life through the past five years we spent together. Finally, I thank my two little cats, Huita and Taiji for their companion all along with my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The wireless technologies have been evolved speedily, whose infrastructure is built and delivered rapidly. The fast growth of wireless networks generates not only opportunities for new applications, but also issues in high energy consumption, unexpected latency, and potential privacy breach. Our research sheds the light upon two of the emerging wireless technologies, namely wireless charging and Narrowband Internet-of-things, empowering IoT services and applications.

1.1 Device Recognition Using New Wireless Infrastructures

Recent years have witnessed the increasing penetration of wireless charging base stations in public areas like office buildings, restaurants, and airports, etc. [19]. There is also a trend to embed wireless charging base stations in furniture like desks and tables [14, 66]. It is estimated that nearly 500 million wireless charging devices were shipped during the year 2017, fulfilling only 20% of the potential world market [58]. This emerging wireless charging infrastructure has presented new opportunities for precise user localization, where the base station learns the location and identification of the mobile device being charged. Many RF- or ultrasonic-based approaches have been proposed for indoor localization [40, 45, 67, 73, 77, 82, 85, 86]. Designed for providing the continuous location of a moving user, they often incur significant overhead, e.g., due to the need for large-scale wardriving for collecting fine-grained signal fingerprints. In this work, we exploit wireless charging for a specific application scenario, where the user stays right next to the wireless charger for a certain amount of time, waiting for the phone to be charged. Therefore, the wireless charger localizes a mobile phone by simply referring to the already-known location of the registered charger. More importantly, this process does not change natural users' behavior, which allows for easier adoption of the proposed system.

Leveraging pervasive wireless charging stations to provide high localization accuracy, high reliability at low deployment cost enables a wide range of applications. For instance, a coffee shop may recognize its customers when they charge their phones on the coffee table and provide customized services or location-based advertisements. For another example, when users charge their phones on the table instrumented with wireless charging during a meeting or lecture, the precise sitting positions of the users can be determined, which enables interesting interactions such as sharing documents in an ad-hoc group, sending instant messages, exploring nearby people [49], or establishing ad-hoc voting or commenting groups for the attendees to express their opinions. In addition to mobile device localization, the popularity of wireless charging infrastructure also provides the opportunity for user authentication. For a paid wireless charging service, the charger can identify the phone and process the payment automatically. Lu et al. [56] proposed a wireless charging network system, where multiple wireless chargers communicate with the server or adjacent wireless chargers to provide pay-per-use charging service. Reliable charging device identification is the basic building block for such applications.

To leverage the wireless charging infrastructure for user localization and identification services, a key challenge is to reliably identify the wireless charging unit of mobile devices. Unfortunately, unlike network interfaces such as Wi-Fi and Bluetooth that have unique and fixed hardware addresses, the wireless charging unit of commercial off-the-shelf (COTS) mobile devices typically does not have a fixed hardware ID. For instance, according to the Qi standard [83], the identity of a power receiver is defined by a Basic Device ID, which can be a software-generated random sequence that may change each time the power receiver is booted.

1.2 Performance Evaluation of New LPWAN Technology

In the last decade, we have witnessed the rapid development and wide adoption of a variety of low power wide area networks (LPWAN) technologies, such as Sigfox [48],

LoRa [54], and Narrowband Internet-of-Things (NB-IoT). While LoRa and Sigfox have attracted much attention in the research community due to their utilization of the Industrial, Scientific, and Medical (ISM) bands, NB-IoT technology received inadequate research focus.

NB-IoT was developed by the 3rd Generation Partnership Project (3GPP) in 2016. NB-IoT envisions an anytime, anything connectivity paradigm [46] for a wide spectrum of low data rate, large volume, and long lifetime IoT applications, including smart grid [53, 65], smart streetlamp [15], parking management [72], air quality sensing [26, 80], and intelligence agriculture [37]. Currently, NB-IoT has been launched globally with 93 commercial networks [35], while there are 140 operators in 69 countries investing in NB-IoT network deployment [34]. According to Ericsson [29], the number of global IoT shipments will grow from one billion in 2018 to 4.1 billion by 2024.

Unfortunately, to date, the key aspects of NB-IoT networks, such as radio access performance and power consumption, have not been well understood, especially to developers and academic researchers. This is due to three key challenges. First, NB-IoT is a closed cellular network deployed by operators on the licensed spectrum, where the base stations can not be accessed for public measurements. The message-level interactions between the node and base station are largely inaccessible to the developers and researchers. Second, NB-IoT measurement is fundamentally different from 3/4G cellular network measurement, where the latter could be conducted through mobile applications [50, 52] installed on massive mobile devices. In contrast, an IoT application may consist of numerous nodes embedded in the environment over a large geographic region, which presents a high barrier for understanding the performance of NB-IoT in the wild. In particular, NB-IoT networks differ significantly due to variations of operator configurations, modules from different vendors, and location profiles. Finally, there lack effective tools that can expose the low-level diagnostic traces from NB-IoT nodes, support large-scale measurement studies, and capture the high level of heterogeneity of network operators, NB-IoT modules, and

location profiles.

1.3 Contribution

In this thesis, we propose two novel cyber-physical systems to demonstrate the possibility of enabling new applications by leveraging the emerging wireless charging infrastructures, and benchmark the network performance of end nodes in an emerging LPWAN, respectively.

First, we present the design, implementation, and evaluation of QID – the first practical system that reliably identifies Qi-compliant mobile devices based on the hardware fingerprints. Specifically, QID augments standard-compliant wireless charging base station to extract features from the oscillator, coil, and controller of a Qi-compliant power receiver, while requiring no retrofitting or modification to existing Qi-compatible mobile devices. QID employs a 2-D motion controller to emulate the coil array in the Qi reference design (described in Section 3.1) and regulate the inductive coupling between the power transmitting and receiving coils, which allows for fine-grained fingerprinting of the power receiver while optimizing the efficiency of power transfer. Experimental results based on 52 Qi-compatible devices show that QID achieves an overall identification accuracy of up to 89.7%, with an average of 85.3%. Our results also have important implications for user privacy. With the increasing prevalence of wireless charging stations in public areas, how to prevent the leakage of the user's location opens up new research questions.

Second, we develop NB-Scope – the first hardware NB-IoT diagnostic tool that supports fine-grained fusion of power consumption and protocol message traces for both real-time in-lab benchmarking and field testing. With NB-Scope, we conduct a large-scale field measurement study based on the deployment of 30 NB-Scope nodes at over 1,200 locations in 3 regions of 2 countries during three months. Our in-depth analysis of the collected 49 GB debug logs and current consumption traces reveals several important insights into the power consumption of NB-IoT in the wild. We showed that nodes yield significantly im-

balanced energy consumption across different locations, operators, and module vendors, which can lead to unexpected short battery life and frequent network partitions. For instance, the ratio of the highest and lowest energy consumption of nodes can be 75:1. By decomposing the energy consummation by radio access phrases in a fine-grained manner, we showed that such performance variance can be attributed to several key factors including poor network coverage level, long-tail power profile due to conservative inactivity timer settings, and excessive control message repetitions during random access control.

Third, we explore the optimization space of the NB-IoT base station on a software-defined eNodeB testbed and propose several optimizations that may save up to 66.4% of the packet energy in signal-limited coverage areas. Then, we discuss several important design aspects that can be considered by future NB-IoT specifications and chipsets for optimizing energy consumption. Moreover, we present our upgrades to the NB-Scope system, including in-device real-time message decoding and LoRa-NB-IoT dual-mode support, which provide new paradigms for the LPWAN research community.

1.4 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 reviews the related works to this dissertation, especially on device identification as well as network measurement and optimization. Chapter 3 presents QID, the first system that identifies mobile devices via wireless charging fingerprints. Chapter 4 presents NB-Scope, the first embedded network measurement system for the NB-IoT network, addressing the heterogeneity in field measurement, as well as the statistical analysis of a large amount of field measurement data. Chapter 5 proposes and evaluates multiple energy optimizations to NB-IoT and presents the important upgrades to the NB-Scope platform. Finally, Chapter 6 concludes this thesis.

CHAPTER 2

RELATED WORK

2.1 Device Fingerprinting, Localization, and Wireless Charging Infrastructure

Device identification has been studied for a wide range of networked communication systems. The existing developed techniques can be broadly classified into three categories. One category uses the fingerprints of the RF signal introduced by the hardware imperfection of the frequency generator on the devices. The next category uses temporal features, i.e. the clock skew introduced by the minor difference in the oscillator among the devices. The clock skew mainly affects the time interval of the transmitted packets. The last category utilizes the sensor hardware fingerprints on mobile devices. Besides the device identification, we briefly compare our proposed system QID with other localization methods and motion-assisted cyber-physical systems. Finally, we discuss previous work that enables applications using wireless charging infrastructure.

RF Signal Fingerprinting. PARADIS [13] identified the source network interface card (NIC) of an IEEE 802.11 frame through passive radio-frequency analysis. Specifically, it uses I/Q origin offset, frequency error, and SYNC correlation to distinguish the devices. Caraoke [2] separated devices by their carrier frequency offset differences to avoid wireless collisions in an e-toll transponder network. Similarly, Danev et al. [22] achieved wireless sensor recognition using RF transient characteristics. Eletreby et al. proposed Choir [28], a system that disentangles collisions in LoRa LP-WAN by distinguishing the sensor nodes using their time, frequency, and phase offsets caused by hardware imperfection. Despite the previous research effort to explore RF signal in the device fingerprinting, these techniques cannot be applied to wireless charging, because on one hand, diving into the details of the RF signal for the fingerprints requires expensive equipment, for example, [13] used

the Agilent 89641S vector signal analyzer; on the other hand, wireless charging adopts resonant coupling to transfer energy, where both the carrier frequency and amplitude are variable. Thus, one is almost not able to infer the device identity using the RF signal in wireless charging.

Clock Skew Fingerprinting. Kohno et al. [44] used the TCP timestamp option to estimate a device's clock skew. Similarly, Cristea and Groza [21] studied how to fingerprint smartphones remotely via the Internet Control Message Protocol (ICMP) timestamp response. While these two studies focused on traffic and driver-level signatures, other systems explored hardware-level features to distinguish devices. Huang et al. [38] used temporal features of Bluetooth baseband embedded in the chipset firmware to fingerprint Bluetooth devices. However, one key difference between these scenarios and wireless charging is that the device placement affects the clock skew fingerprints of the mobile device. Moreover, the placement of the device on the charger pad is unpredictable, which casts much difficulty in building a precise model for each device. We will further discuss the challenges in detail in Section 3.2.

Sensor Fingerprinting. Another device identification techniques use fingerprints in the sensors, such as acoustic sensor [9,23], camera [31,78], and inertia sensor [24,88]. For example, Das et al. [23] utilized manufacturing imperfection in the microphone and speaker to distinguish different mobile devices using classical machine learning algorithms. Valsesia et al. [78] proposed a compressed camera fingerprint algorithm to reduce the complexity in feature computation and storage requirements, improving the device recognition performance using photo-response nonuniformity. Zhang et al. [88] utilized per-device inertial sensor factory calibration data, embedded inside a mobile device firmware, to extract fingerprints. Das et al. [24] performed an in-depth study on device recognition using combined features extracted from accelerometer and gyroscope sensor streams with machine learning techniques. Although these methods may achieve acceptable accuracy, they require reading the data or sensor samples from the phone directly, which can be intrusive.

Localization. A major application of our system is to identify and localize mobile devices. Previous localization systems were based on GPS, angle of arrival (AoA) [32, 86], time of flight (TOF) [55, 67, 79], received signal strength (RSS) [6, 87], ultra wideband (UWB) [47, 71], and multipath signal aggregation [81]. Wireless charging-based localization is similar to landmark-based approaches such as Wi-Fi AP or Cell-IDs [76], where the location of the charger (access point) is already known or registered. In this manner, the location of the mobile device can be acquired immediately after the users put the phone on the charging station. Such a user-initiated localization approach is highly precise. While Cell-ID based methods usually have an error of up to tens or hundreds of meters, wireless charging infrastructure-based localization can achieve centimeter-level accuracy. Wireless charging-based localization is also robust compared to AoA/TOF/RSS-based methods because the location of the former is not affected by dynamics of wireless systems like signal strength.

Motion-Assisted System Augmentation. Many sensing systems are augmented with the assistance of motion control. Graefenstein et al. [33] used a rotating antenna on a mobile robot to localize wireless sensor node basing on RSSI. They improved the robustness of the measured RSSI and increased localization accuracy. Similarly, Malajner et al. [57] employed a stepper motor to rotate the antenna array to estimate the AoA of the RF transmitter, achieving low cost and high accurate AoA estimation. Chou et al. [17] added a rotating four-bar linkage to a mobile platform to extend the 2D laser range finder to support 3D environmental sensing and mapping. Although motion platform has been applied to various sensing systems, to the best of our knowledge, this idea has not been adopted by wireless charging systems. We will show in Section 3.7.3 that, with the assistance of a 2-D motion platform, we improve the device recognition accuracy substantially.

Application Based on Wireless Charging Infrastructure. In our previous work, we have developed QiLoc [49], a system that extracts the software ID of the charging device and localizes its location based on the known deployment location of the charging station.

Comprehensive API is provided to implement location-based service, occupancy analysis, and authentication. It also provides API for location-based services, occupancy analysis, and authentication. However, according to the Qi standard [83], the identity of a power receiver can be a software-generated random sequence that may change each time the power receiver is booted. Thus, the fingerprinting techniques presented in this dissertation can be integrated with QiLoc to provide a reliable localization service. Lu et al. [56] proposed a wireless charging network architecture, where multiple wireless chargers communicate with the server or adjacent wireless chargers to provide pay-per-use charging service. However, device recognition during charging is not studied in [56]. Although their work envisions potential wireless charging applications, they did not consider the situation where the software ID is unreliable. Our project focuses on how to accurately recognize the device under Qi wireless charging with reliable hardware features.

2.2 Wireless Infrastructure Benchmark and Diagnostics

Several prior works on cellular networks focus on the design and implementation of 4G LTE measurement toolkits [51, 60]. Nikravesh et al. [60] proposed an open platform for conducting mobile network measurement experiments in a principled manner. Li et al. [51] proposed MobileInsight, a diagnostic tool that collects and analyzes information from the operational cellular network at a fine-grained message level. However, the measurement of 4G LTE is usually conducted through massively deployed smartphone applications, which is not viable for NB-IoT due to the embedded nature of NB-IoT applications and the limited compute capability and battery life. The tools that can provide both power measurement and extraction of debug logs of NB-IoT are not currently available which makes large-scale measurement of NB-IoT very challenging.

Power measurement and modeling. A lot of early studies have investigated the energy efficiency of cellular networks over smartphones through power modeling [7, 16, 25, 39]. Balasubramanian et al. [7] developed an energy consumption model for the 3G network as

a function of transfer size and inter-transfer times. Ding et al. [25] modeled power based on wireless signal strength. However, they focus on modeling the download link without considering the uplink data service. All these power measurement studies of the cellular network are based on smartphones. In [11] and [54], the authors designed new LoRa end nodes for measuring the energy consumption of LoRa. Bouguera et al. [11] presented a power model for different LoRaWAN modes and scenarios. Liando et al. [54] focused on characterizing the energy consumption of the LoRa module. Different from LoRa, NB-IoT is deployed by operators on licensed bands and the base stations cannot be accessed for public measurements.

Power consumption optimization. Sehati et al. [70] proposed an online bundling algorithm to reduce the impact of small packets on the long tail energy consumption in IoT applications. Jang et al. [43] presented a novel access control mechanism for cellular IoT networks to meet various performance metrics. Azari et al. [5] proposed a model to analyze the impact of channel scheduling on latency and energy consumption for nodes under different coverage levels. Astudillo et al. [3] proposed a probabilistic re-transmission approach to avoid collisions of random access control messages and to reduce delay and energy consumption. However, these optimizations are based on analytical results of network protocols which fall short in capturing the power consumption patterns and direct factors that cause energy waste in a real deployment. In our work, we identify the main factors causing low energy efficiency through field tests and propose effective optimization strategies correspondingly.

CHAPTER 3

IDENTIFYING MOBILE DEVICES VIA WIRELESS CHARGING FINGERPRINTS

3.1 Background

Qi is an open standard that defines wireless power transfer over short distances. A typical Qi system consists of a power transmitter (PTx) installed on a Qi base station and a power receiver (PRx) attached to or installed in a Qi-compliant mobile device. The PTx comprises a transmitting coil, called Primary Coil, which generates an oscillating magnetic field, which induces an alternating current in the receiving coil, namely the Secondary Coil, of the PRx. The PRx communicates with the PTx via backscatter modulation of the current draw, and primarily sends two types of messages to optimize the power transfer. The control error packet (CEP) carries an integer that indicates the difference between the desired power level and the received power level. The received power packet (RPP) reports the average level of power received in the past period. Throughout the process of charging, CEPs and RPPs are transmitted periodically. Table 3.1 shows the CEP and RPP intervals specified by the Qi standard. Based on the information in CEPs and RPPs, the PTx adjusts the carrier frequency and amplitude of the primary power signal to optimize the coupling between the coils of PTx and PRx. The combination of the carrier wave frequency and amplitude is defined as the operating point. Figure 3.1 exemplifies a set of operating points collecting in an experiment. When the phone position changes, the coupling between the PTx and PRx coils varies, such that the PRx informs the PTx with CEPs to regulate the wireless power output. Then, the wireless charging adapts to a new operating point eventually. We note that, if the phone placement remains unchanged, the corresponding operating points will aggregate around a specific small region in Figure 3.1.

Qi specifies multiple reference receiver and coil designs [84]. Figure 3.2 exemplifies one

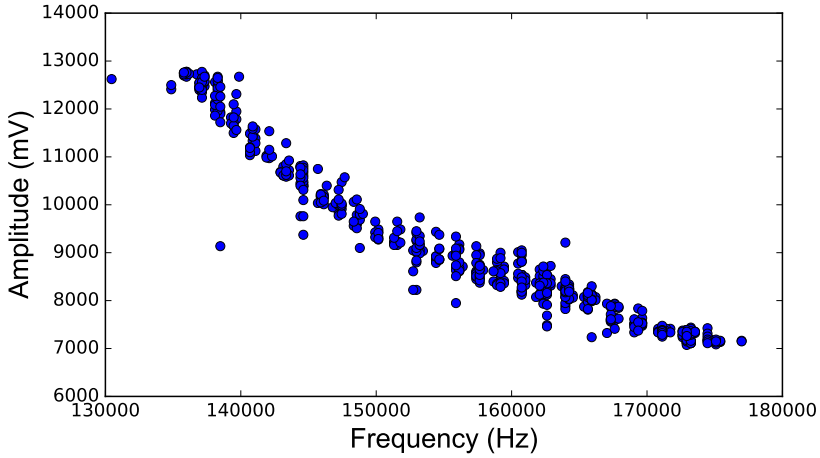


Figure 3.1: Operating points collected in an experiment by manually changing the mobile device placement.

Table 3.1: PRx timing constraints during the Qi power transfer phase.

Parameter	Symbol	Target (ms)	Max (ms)
CEP Interval	t_{interval}	250.0	350.0
RPP Interval	t_{received}	1500.0	4000.0

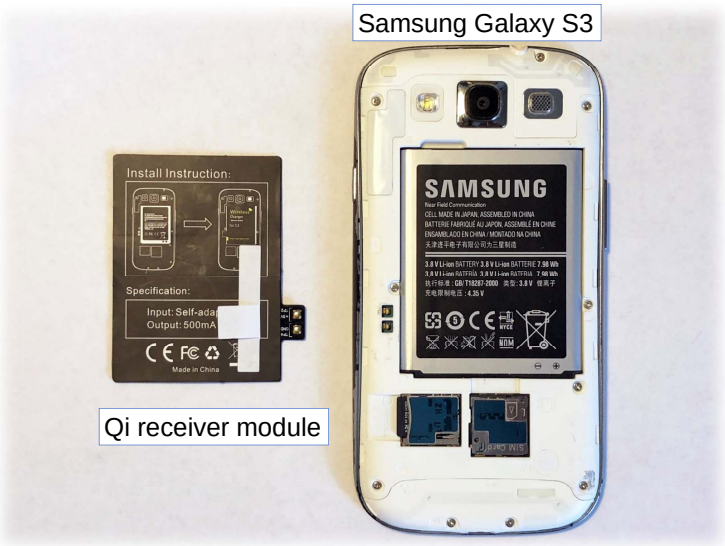
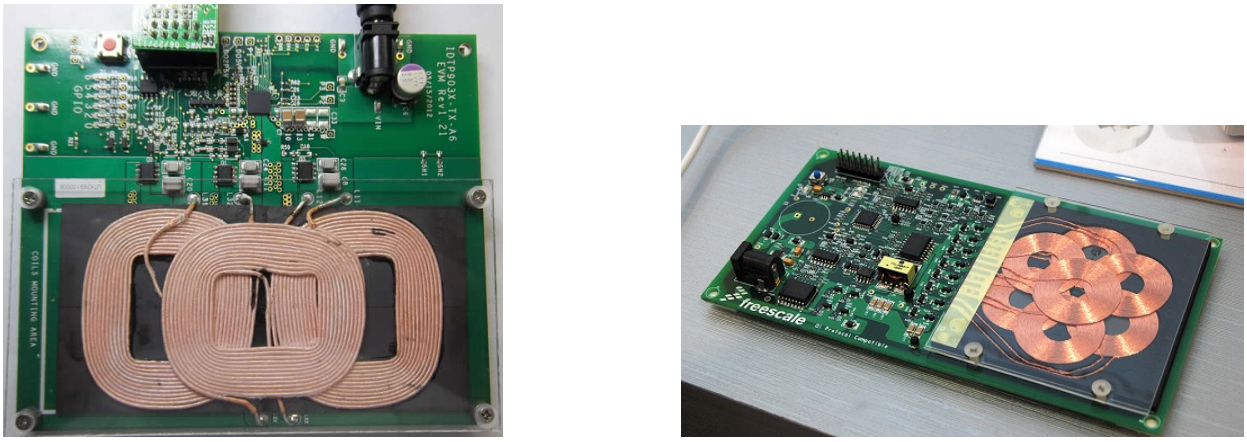


Figure 3.2: An attachable Qi-compatible power receiver for Samsung Galaxy S3.

of the available designs and Samsung Galaxy S3 that supports attachable Qi-compatible PRx modules. It has a pair of terminals (`+5V` and `GND`) that connects to the output of the

PRx. In such a design, the PRx is an independent module and does not communicate with the phone. The attachable PRx modules provide the wireless charging capacity to those devices that originally don't ship with the wireless power receivers. We consider each of the modules to represent a user identity since it is an independent component (either attached outside or pre-installed inside). It outputs a stable current at 5 V for charging with its maximum capacity most of the time.

In addition to the PRx design, Qi also specifies more than 30 Type A and 7 Type B PTx designs, where type A designs have one or more Primary Coils but *only one* of them can be activated at a time, while type B designs support an array of Primary Coils and *one or more* Primary Coils can be activated to provide wireless power to multiple PRxs simultaneously. Figure 3.3 [18] shows two examples of the multi-coil chargers. Compared to the single-coil designs, multi-coil PTx enlarges the possible coupling area with the PRx, thus providing more flexibility in the device placement. As a result, the coil array PTx designs become more prevalent on the market.



(a) Charger coil movement illustration (b) The mobile coil array

Figure 3.3: Examples of the multi-coil designs in Qi.

Qi also supports a serial number of at least 20 bits, also known as the Basic Device ID. However, a PRx can also use a random number generator to dynamically change the Basic Device ID, so that every time the user puts the mobile device onto a PTx, the Basic Device ID updates. Such a random ID invalidates device ID based applications, which inspires

us to design a system to identify a device using its hardware fingerprints.

3.2 Design Challenges and System Overview

3.2.1 Design Challenges

It is challenging to recognize a mobile device from the Qi charger due to the intrinsic characteristics of the wireless charging environment. Specifically, the following challenges need to be addressed:

Noise in temporal features caused by power transfer. The wireless power transfer happens in the rapidly-changing high power electromagnetic field between the two coupling coils, casting more noise than typical RF wireless systems. For instance, as shown in Figure 3.4(1), the measured packet intervals have significant fluctuations. The standard deviation of the CEP time interval is more than 4.4 ms, corresponding to 1.7% error, which makes it difficult to distinguish between different charging devices.

Undesirable stable operating point. Qi wireless charging has a well-designed feedback control loop. The wireless power transfer process is usually stabilized at an operating point within hundreds of milliseconds (3 to 5 CEPs). Although this is a desired feature in terms of maintaining high charging efficiency, it brings a major challenge in recognizing the target device. Figure 3.4(1) and Figure 3.5(1) show the experimental result of such a scenario, where the phone remains stationary on the charging pad. As shown, the selected features, both the CEP time interval and CEP values, remain unchanged during the charging process, which eventually raises the recognition error rate.

Unconstrained device placement. The third challenge, the most difficult one in our case, is that the selected features are dependent on the phone placement. In other words, if the user alters the PRx position, the feature values can change dramatically. Figure 3.4(2) and Figure 3.5(2) demonstrate how the CEP time interval and control error value change with respect to the phone placement. During the measurement, after we rotate the phone with a random angle, the CEP time interval decreases from about 250 ms to 160 ms, and

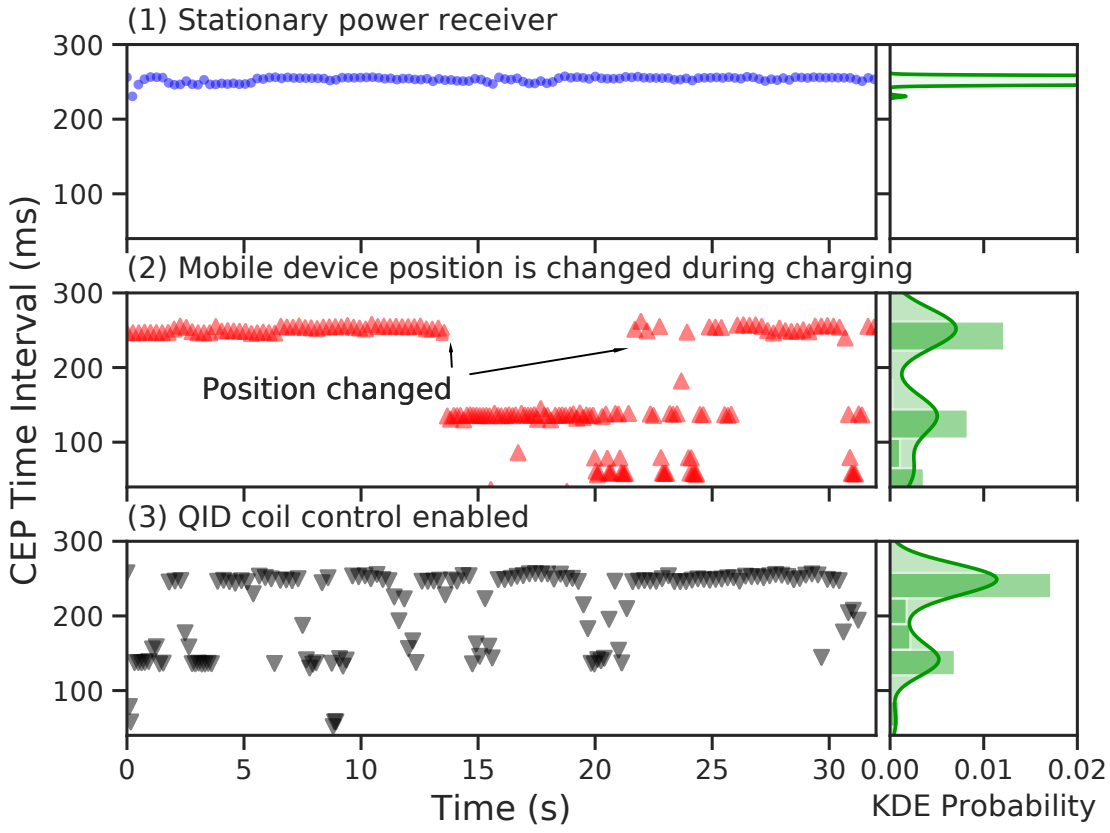


Figure 3.4: The CEP time interval vs. sampling time in 3 independent feature acquisition experiments: (1) stationary PRx; (2) PRx position changed during charging; (3) QID is enabled.

the Control Error value increases dramatically from 0 to 30. In a real-world scenario, the placement of a mobile phone is often unpredictable. As a result, the errors are accumulated over time, which eventually renders the recognition unsuccessful.

3.2.2 System Overview

We now provide an overview of the QID system. The system architecture is shown in Figure 3.6. It consists of 3 components, namely a COTS Qi wireless charger, the QID sensor, and the QID server. The QID sensor is responsible for collecting a selected set of features from wireless charging and uploading the data to the server, while the QID server is responsible for the feature extraction and device classification. The QID server can connect

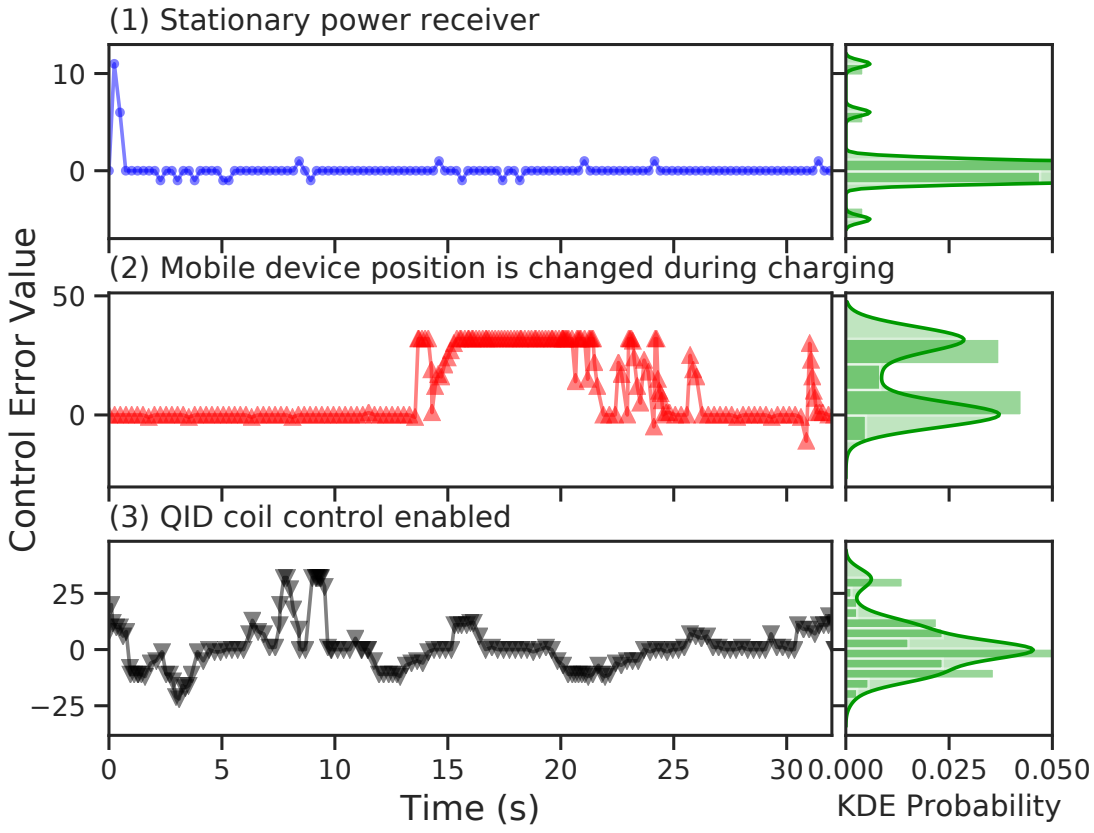


Figure 3.5: The CEP value vs. sampling time in 3 independent feature acquisition experiments: (1) stationary PRx; (2) PRx position changed during charging; (3) QID is enabled.

to the QID sensor directly (e.g., through UART) or resides on the cloud and communicates with multiple QID sensors through the Internet, enabling tracking the target device at different charging locations.

The QID sensor can work with most Qi-compliant chargers. It does not modify any of the charger pad circuits. What QID sensor needs from a Qi-compliant charger is a *test pin* that outputs the filtered data bit flow. We note that such a data pin is indispensable for the Qi charging system because the PTx requires feedback from the PRx. Reading data flow from the pin does not affect the operation of the Qi charging system. Therefore, thanks to the minimal modification requirement, QID can be easily integrated with off-the-shelf Qi chargers. After connecting the test pin and mounting the charger coil to the QID sensor, the platform is ready for device fingerprinting. The QID sensor consists of a

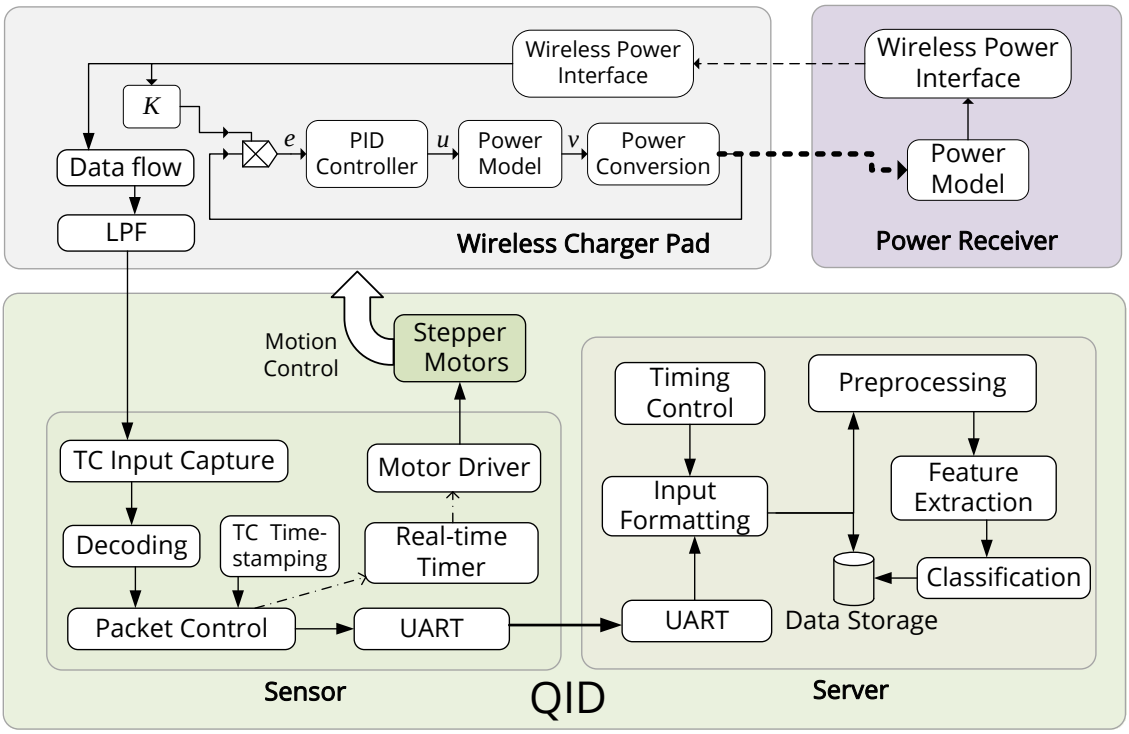


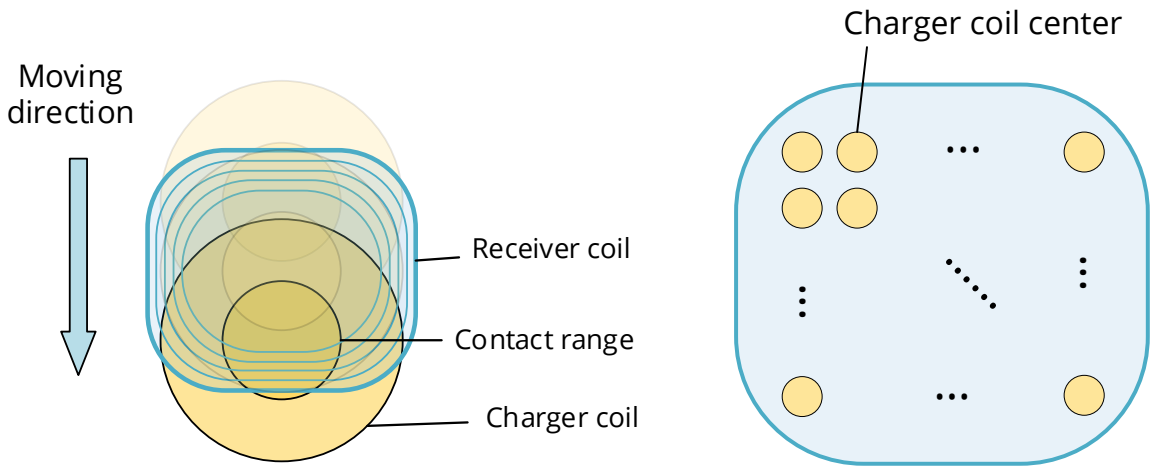
Figure 3.6: The system architecture of QID. It consists of QID sensor and QID server. QID sensor is responsible for controlling the motion of the charger coil, capturing the signal from the wireless charger, as well as sending the timestamped packets to QID server. QID server extracts the features from the packet sequence and classifies the device.

motion control hardware component and a software component for feature collection. The design of the motion unit is discussed in Section 6.1. The motion unit hosts the charging pad and moves it according to a certain pattern, within a range of 10 cm. This allows to fingerprint PRx dynamically at different relative positions between the PTx and PRx coils, resulting in higher identification accuracy. We note that the motion unit is connected to a separate control module, and it does not require a wire connection with the charger itself. As a result, it can be integrated with any off-the-shelf charger easily.

The motivation for adopting the motion unit is two-fold. First, it can be easily integrated with single-coil chargers and improve the performance of classification accuracy as well as power delivery. Second, many emerging new chargers are based on multi-coil Qi-compliant power transmitter design [18, 42, 61]. As described in Section 3.1, each coil

on such transmitter is controlled by an individual switch or a separate bridge. The PTx can select the optimal coil to deliver the wireless power to the PRx. Thus it enlarges the coupling area between the PTx and PRx and provides more flexibility in the device placement. Compared with the single-coil design, the multi-coil design offers several key advantages for charger fingerprinting. First, the noise within the temporal features can be controlled and even filtered out in post-processing, because the fingerprints are collected from multiple coil locations, a more complete device profile can be built. Second, the wireless power transfer process to hop between different operating points when the charging coils are switched. Thus we can infer the PRx control scheme from the transient states between the operating points, which can be used to differentiate different Qi modules. Finally, the feature uncertainty caused by the phone placement can be essentially mitigated because the coil array covers a range of device positions on the charger pad. Despite these advantages, it is difficult to exploit the multiple coils of Qi chargers for fingerprinting in practice, since there exists a large number of heterogeneous designs as specified by Qi [84]. Usually, the existing multi-coil charger pads use large coils with small overlapping areas. The fingerprint sampling granularity is thus low even it provides the coil switching capability. To address this challenge, the QID sensor extends the design of the physical coil array to a *mobile coil* by equipping the primary transmitter coil with a motion unit. Figure 3.7a illustrates the relative movement between the PTx and the PRx. When the relative alignment of the PTx and PRx is adjusted in a 2-dimensional plane, we can construct a fine-grained mobile coil as shown in Figure 3.7b. Such a design effectively emulates a variety of different multi-coil designs of Qi. However, we note that the motion unit is not necessarily required if the Qi system already adopts a multi-coil design. In such a case, the rest of our system can be integrated into a small module that connects to the charger through two pins, namely the data flow test pin and `GND`, allowing for easy deployment.

In addition to the motion unit, the QID sensor also extracts and timestamps every packet in the data flow. A challenge in the design is to ensure the PRx is correctly located



(a) Charger coil movement illustration (b) The mobile coil array

Figure 3.7: Fingerprinting a PRx coil with a movement unit.

and measured. The details of the QID sensor are discussed in Section 3.3. The last component, namely the QID server, reads all the data sent by the QID sensor. As the packet is in byte representation, the server needs to parse each field in the packet. Then, the server performs feature extraction and classification, which are discussed in Section 3.5.

3.3 Feature Selection and Acquisition

3.3.1 Selecting Hardware Fingerprints

To reliably identify Qi-compliant devices, QID leverages hardware fingerprints extracted from the following three PRx components. The selected fingerprints should be device-specific, time-invariant, and discriminative.

Onboard oscillator. The PRx controller chip of Qi utilizes an internal oscillator to generate the clock signal. It is well known that oscillators have distinctive drifts due to factors like hardware manufacturing variations [21,38,44]. We thus exploit the drift of the PRx oscillator as a feature to identify the device in charging. For example, Panasonic AN32258A [63], a commercially available Qi receiver IC, utilizes an internal oscillator. NXP MWPR1516 [62] also uses an internal Low-power Oscillator (LPO) as the clock source. We note that the

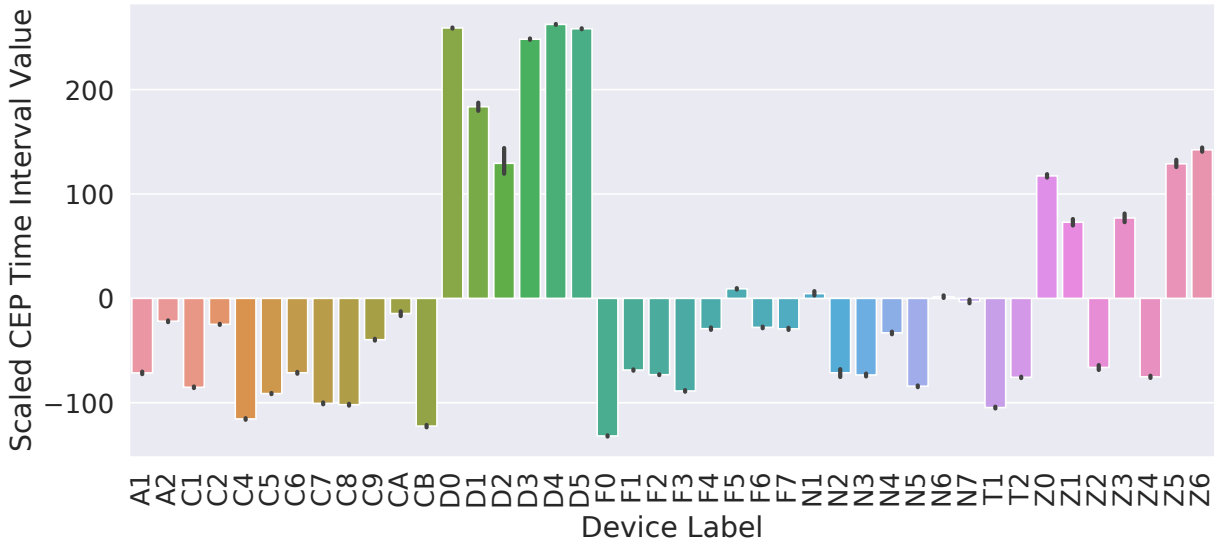


Figure 3.8: The scaled and zero-meaned CEP time interval distribution of 42 evaluated devices. The first letters of the devices represent the brands of the receivers, while the following digit represents the specific label in its brand. The error bar shows the standard deviation of the CEP time interval. The corresponding actual time interval spans a range of (238, 270) ms.

Qi receiver ICs typically have low clock accuracy as they are not designed for data communication. For instance, the receiver IC NXP MWPR1516 has a clock accuracy tolerance of as high as ±5%; Rohm BD57011AGWL data sheet [69] also indicates that the driving frequency of the communication signal is between 1.92 and 2.08 kHz, which corresponds to a 4% frequency error tolerance. In comparison, the clock frequency tolerance is ±50 ppm for Bluetooth [8] and ±40 ppm for Zigbee [1]. Therefore, the clock drift effect of Qi is highly device-dependent and much more significant than other wireless communication systems. Although drift variations like this can be used to differentiate different devices, it is difficult, if not impossible, to directly measure the clock drifts in COTS devices. Our key observation is that the Control Error Packet (CEP) time interval yields high variance among the devices around the target value specified in Qi (see Table 3.1). Figure 3.8 shows that the CEP time interval distribution of 42 devices spans a range of (238, 270) ms in the time domain. Therefore, the PRx oscillator can be inferred and fingerprinted by measuring the period drift of the control packets. However, some devices, for example, "A2" and

"C2", or "F6" and "F7", yield close CEP time interval values.

Figure 3.9: Heterogeneous power receiver coils. The size and shape of the coils result in different contact range mean and standard deviation.

Receiving coil. Different Qi-compliant devices may have different coil shapes, diameters, and layouts. Generally, a larger PRx coil has a larger contact area between the PRx and the PRx coils, leading to more flexibility in placing the device. In our scenario, the receiving coil diameter can be fingerprinted based on the area that the PTx interacts with the PRx. Figure 3.9 exemplifies different fingerprints extracted from heterogeneous PRx coils. It can be a determinant for device brand, but not a good indicator of a specific device because it is almost identical among the same type of device. We discuss how to measure the contact range of the PRx coil in Section 3.4.2.

PRx controller. The Qi standard does not specify the exact period of control packet transmission. We observe that the periods of the CEPs do differ across devices of different manufacturers. Such vendor-dependent controller implementations can be exploited as a fingerprint to differentiate devices from different manufacturers. For example, Texas Instruments bq51013B [75] sends the CEPs with an interval of 240 ms, while Panasonic AN32258A sends the CEPs at a period of 160 ms. This feature is not related to clock error but a value chosen at design time by the manufacturer. Intuitively, determining the IC manufacturer improves the recognition accuracy by narrowing the categories of the devices. In addition to the packet period, the value carried in the CEP is also specified by the receiver IC manufacturers. For example, we observe that the maximum control error value sent by brand "C" devices is 30, while the brand "Z" devices can send the control

error values as high as 80. Therefore, it is another feature that may distinguish the device brand.

3.3.2 Temporal Feature Acquisition

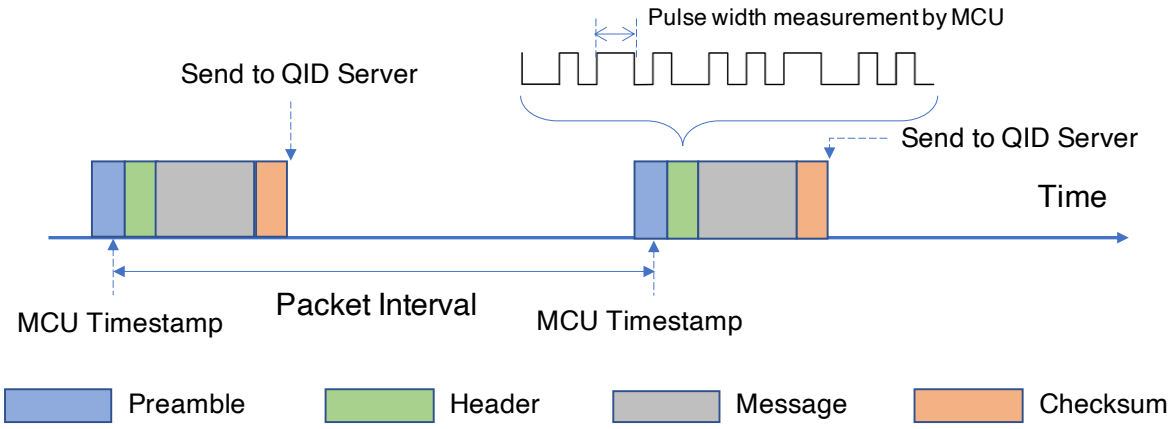


Figure 3.10: Temporal Feature Acquisition. Each packet is decoded and timestamped by the microcontroller (MCU).

We now discuss how the QID sensor collects temporal features of Qi packets. Figure 3.10 illustrates such a procedure. First, to decode the bits, the QID sensor uses a timer to measure the width of each pulse. As the data sent by the PRx is encoded with a differential bi-phase scheme, we can convert the pulse widths to bit values. Then, the decoded bits are then grouped into bytes.

A Qi packet consists of preamble, header, payload, and checksum. The QID sensor timestamps the packet after the 11th bit in the preamble phase of each packet. The corresponding packet time interval is then the difference between two consecutive timestamps. As described in Section 3.1, the Qi protocol defines two types of packets that have fixed time intervals. QID sensor mainly observes and analyzes the CEP time interval to infer the PRx oscillator because the CEP is the most frequent type of packets that are sent during the wireless charging process. Even if a packet is corrupted due to decoding errors, its

timing information is still valid if only the packet interval is in the desired range. Finally, the timestamped packets are transmitted to the QID server for further processing.

3.4 QID Motion Control

3.4.1 Motion Platform Design

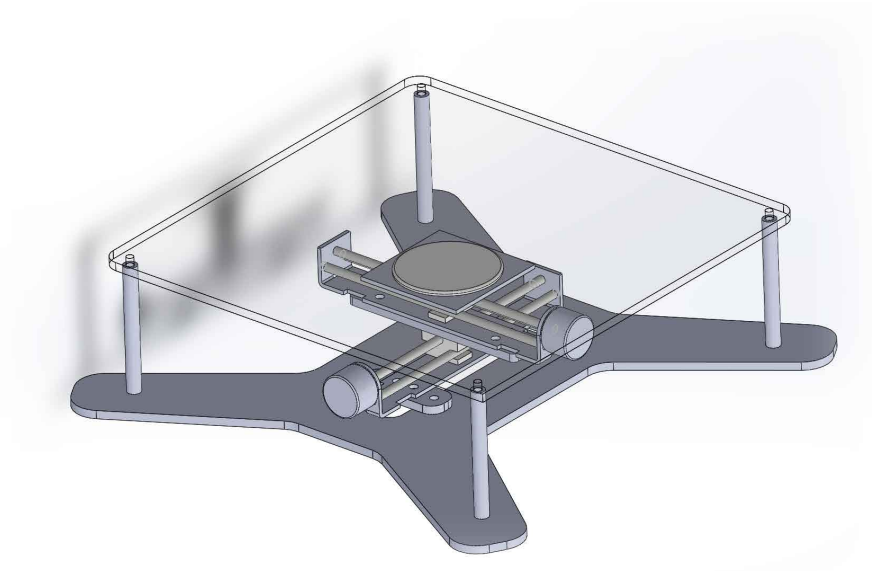
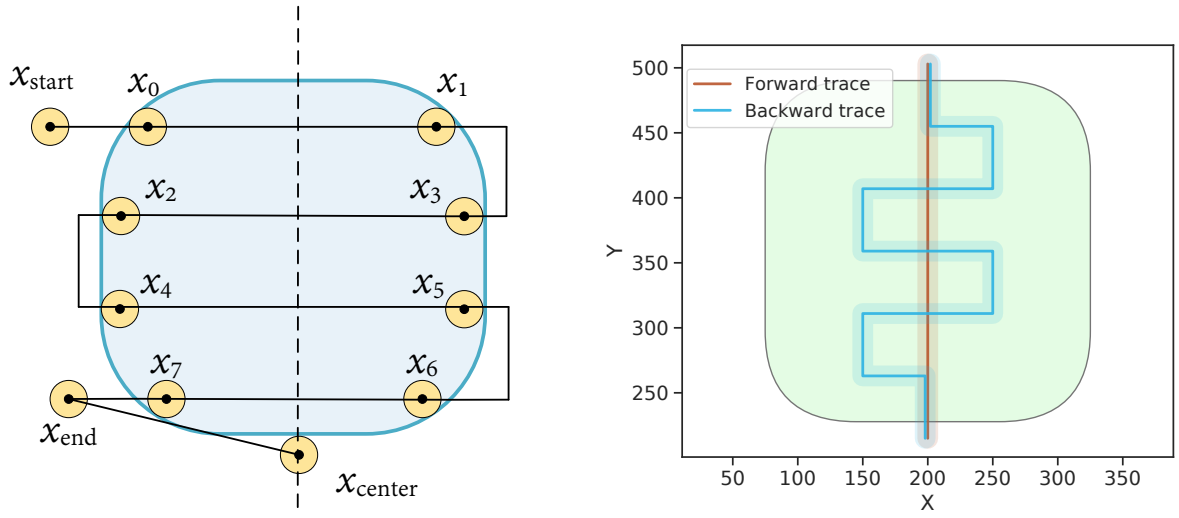


Figure 3.11: Mechanical design - the charger pad is controlled by two stepper motor linear slides, moving in a 2-D surface.

In this subsection, we present the mechanical design to enable the movement of the charger coil, as illustrated in Figure 3.11. It requires two linear slides powered by a stepper motor individually. First, the bottom slide is fixed on a surface. Next, the upper slide is placed with its axis direction perpendicular to that of the bottom one. The upper one is attached to the bottom one's slider. Finally, the charger coil is attached to the slider of the upper linear slide. The two stepper motors are controlled independently to drive the coil in an X-Y plane to form a mobile coil. It thus allows for more flexibility in the PRx placement. The user can place the device at any location and any angle in the designated area. Correspondingly, we define the axes of the bottom slide and the upper slide as the x axis and y axis respectively. As the lengths of the two slides are the same, the working



(a) Symmetric axis searching. The upper left point is the starting point.

(b) The designed trajectory of the charger coil center within one complete fingerprinting scan.

Figure 3.12: Trajectory design of the QID sensor.

space of the PTx coil is defined as a square area. Our measurement results indicate that the motion platform can achieve a control accuracy up to 0.2 mm, which is adequate to emulate heterogeneous PTx coil designs. We envision that the QID motion platform can enable other applications, such as locating the PRx and searching for the optimal charging operating points, which are critical for optimizing the charging efficiency [83]. We leave these applications for open research work.

3.4.2 QID Sensor Motion Control

In this subsection, we discuss the control schemes for the proposed QID sensor motion platform.

3.4.2.1 Contact area boundary detection

The detection of the PRx boundary allows for better coil movement control. For example, the stepper motor can adapt to a higher speed if the PRx is out of the contact range, or the trajectory can be optimized to avoid unnecessary moves, such that the total time needed

for collecting sufficient features can be reduced. The QID sensor utilizes a timer to detect the contact boundary. Each time the QID sensor receives a new packet, it reads the real-time timer (RTT) to update a value t_{last}. In the meantime, the QID sensor reads the RTT with a period of 10 ms to fetch the current time t_n and compares it with t_{last}. Then the condition that the PRx is out of the contact range is given by:

$$t_n > t_{last} + T_{timeout},$$

where $T_{timeout}$ is the allowed time that the PRx does not send any feedback. We choose $T_{timeout}$ to be 350 ms, which is the maximum CEP time interval in the Qi standard, as presented in Table 3.1. If the condition is met, the QID sensor determines that the PRx loses its contact with the PTx.

3.4.2.2 PRx symmetric axis alignment

Next, we discuss how the QID sensor finds the symmetric axis of the PRx coil along the y axis. Finding the symmetric axis is important because it is the reference for the fingerprinting trajectory.

We assume that the device is in the contact range once the user puts it on the charger pad. The PRx symmetric axis alignment is achieved as follows. In the beginning, the PTx coil moves along the positive direction of the y axis until it is out of the contact range. Next it moves to the negative direction of both x and y axis, reaching the starting point shown in Figure 3.12a (upper left corner). From there, the coil starts to move in an "S" pattern and sweeps across the PRx coil four times, generating a sequence of the detected boundary $[x_0, x_1, ..., x_7]$. Then the x value of the symmetry axis is

$$x_{center} = \frac{1}{8} \sum_{i=0}^{7} x_i$$

Finally, the PTx coil is aligned to the x_{center} with its y coordinate value right out of the contact range boundary. This location is the starting point of the coil in the fingerprinting phase.

We note that, for a multi-coil PTx, this phase can be achieved by switching between the coils and identifying the one with the highest coupling.

3.4.2.3 Fingerprinting trajectory planning

Now we present the PTx coil trajectory design when the QID sensor collects fingerprints from the PRx. We take two factors into account when designing it. On the one hand, it is crucial to ensure the QID sensor captures adequate data from the mobile coil in Figure 3.7b, such that QID records the complete feature profile of the PRx. On the other hand, the more data points are measured, the more time it takes. Typically 4 or 5 packets can be collected during one second. If we plan to record 3,000 CEPs, it may take more than 10 minutes and exceed the time one leaves the phone on the charger pad. Therefore, we need to find a trade-off between spatial data diversity and measurement delay. In our design, we assume that the PRx will be left stationary on the platform for a time window of at least 90 seconds, such that the QID sensor captures adequate fingerprints for device identification.

The designed fingerprinting trajectory of the QID sensor is shown in Figure 3.12b. It comprises two sessions, namely the forward one and the backward one. During both sessions, the QID sensor records all the packets sent by the PRx and uploads them to the server. The forward session starts from the endpoint of the PRx symmetric axis alignment phase. The PTx coil is driven to move along the positive direction of the y axis. Once it enters the contact range, the coordinate value y_{start} is recorded. In the meantime, the moving speed of the coil is set to a slower speed. It stops for 1 second each time after moving forward for about 8 mm (corresponding to 40 control units in Figure 3.12b) to wait for the operating point hopping, which also mimics the coil switching in an array. Once the PTx loses contact with the PRx, i.e., the PRx is out of the contact range, the boundary point y_{end} is recorded, which also marks the end of the forward session. The backward trace is similar to the center alignment one, which is in "S" shape. The major difference

is that the distance Δx between the farthermost point and the symmetric axis is about 10 mm along the x axis, corresponding to 50 control units. The movement along the negative y direction is divided into 6 sections. Each of them is

$$\Delta y = \frac{y_{end} - y_{start}}{6}$$

At each turning point, the PTx coil stops for 1 second to wait for the operating point hopping. After the coil reaches the forward session starting point (x_{center}, y_{start}), the fingerprinting phase finishes. We define a complete scan as the completion of both the forward and backward sessions, and the data collected during this phase are defined as a *scan sample* correspondingly. It is later post-processed at the QID server.

There are several reasons why such a trajectory is designed. First, it covers the central area of the contact range. Even if the PRx changes its placement angle next time, the central region still largely remains overlapping. As a result, the two independent scan samples do not deviate significantly. Second, the distance Δx is carefully chosen to accommodate different coil shapes and sizes. No matter how the PRx is placed, the planned trajectory falls within the contact range for most of the time. The trajectory is also able to tolerate the symmetric axis alignment error up to about 8 mm (corresponding to 40 control units). Finally, the same location is fingerprinted for at most once, as the forward and backward traces do not overlap except the region where the coil center enters or exits the contact area.

3.5 Feature Extraction and Device Classification

In this section, we present how the QID server extracts the features from the measured data and classifies the features into different device classes.

3.5.1 Feature Extraction

We note that the contact range diameter of the receiving coil physical feature can be simply calculated as $y_{end} - y_{start}$. The oscillator features and the PRx control schemes require

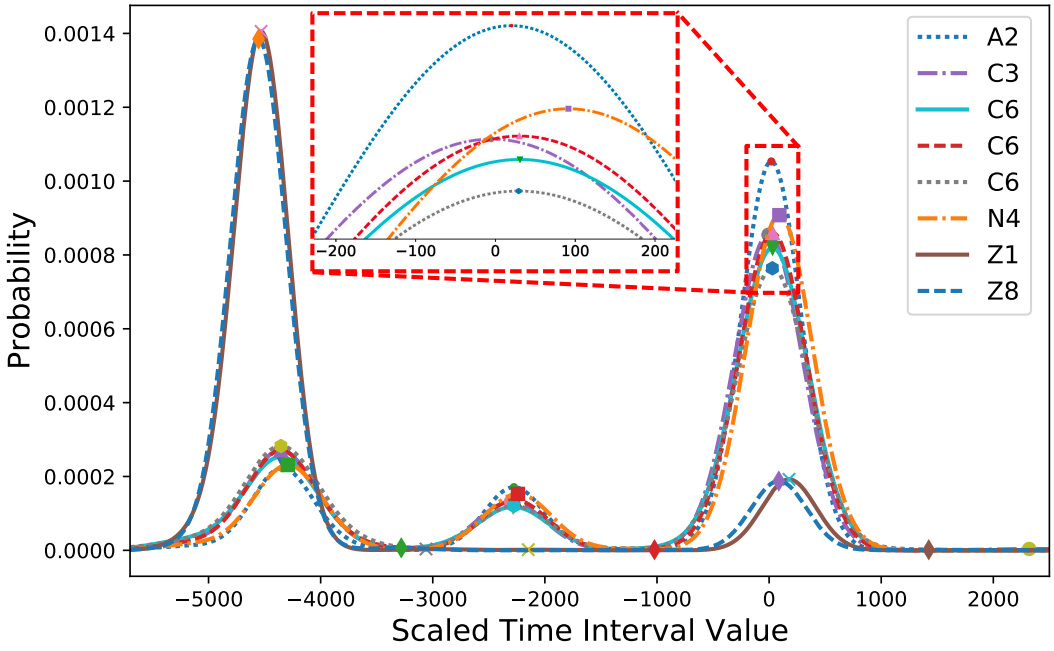


Figure 3.13: Gaussian kernel density estimation of CEP time intervals. The letters indicate the device brands. The number indicates each unique device of its brand. The 0 in the horizontal axis corresponds to 240 ms in real time scale.

additional processing to obtain. We discuss them as follows.

3.5.1.1 CEP interval features

The QID server uses a local maximum searching algorithm to extract the CEP time intervals that represent the feature of the PRx onboard oscillator. First, all the CEP time intervals are processed with a fixed bandwidth Gaussian kernel density estimator (KDE). Unlike the histogram, the Gaussian KDE represents the probability of each point in the feature space with a fixed-variance Gaussian distribution, i.e., the Gaussian kernel, and then the estimation output is the averaged sum of all the individual kernel probability estimations. The output of the KDE is actually the probability density function (PDF) of the CEP time interval.

$$d(t) = pdf(t), t \in [40, 270] \text{ ms}$$

Note that the variable t here is in time domain representation, while the feature values correspond to the QID sensor controller timer values.

Next, the QID server extracts the CEP time intervals that achieve local maximums in the PDF. We observe that the CEP time intervals typically fall into 3 domains, corresponding to [40, 60], [140, 160], [235, 270] ms in time domain (as shown as the three local maxima in Figure 3.13). Therefore, it is feasible to find the peaks directly without calculating the derivative of the CEP time interval PDF. Specifically, the 3 domains are denoted as D_1, D_2, and D_3 respectively. Then the feature CEP time intervals and their corresponding log-probabilities are

$$t_{pi} = \arg\max_{t \in D_i} d(t), i = 1, 2, 3.$$

$$p_{Li} = \log d(t_{pi}), i = 1, 2, 3.$$

The detected peaks of 8 independent complete scan samples are marked in Figure 3.13. The CEP time interval that corresponds to 240 ms in the time domain is shifted to 0. Although some of the curves appear close to each other in the figure, their peak-indexed CEP time interval features actually span a considerable wide range in the feature space, as shown in Figure 3.8 and the zoom-in sub-figure in Figure 3.13. In the zoom-in figure, we see that the indexed time interval values have little intra-class variability and inter-class similarity. For example, although "C3" and "C6" are from the same brand, their peak indexes deviate from each other with a distance up to 20, while the three peak indexes of device "C6" are consistent during the three independent scans that are collected in a wide time span.

3.5.1.2 CEP value features

QID also fingerprints the controller of a PRx based on the statistic of the recorded CEP values during a complete scan. What the CEP value differs from the CEP time interval is that the former can only take integer values. We analyze CEP values using the probability

mass function (PMF). Specifically, we compute the PMF of the CEP values as

$$p(V) = \frac{\sum_{i=1}^{N} 1\{v_i == V\}}{N}$$

where V is the absolute CEP value, ranging from 0 to 127, $1\{a == b\}$ is an indicator function, and N is the total number of CEPs in a scan sample. In particular, N is chosen to be one of the CEP value features because it is an indirect measurement of the frequency that the PRx sends CEPs. As we note in Section 3.3.1, it is a PRx controller implementation-specific feature rather than the oscillator drift. We find that the PMF of the CEP values is not a good feature, as the PMFs span a wide range of [0, 127], which introduces high variance in the output features. We also observe that even the same device may generate different PMFs within different scan samples. To reduce the variance in the output CEP value features, we further group the CEP values into 7 ranges. For each range, the probabilities of the CEP values within the range are summed up. Specifically, the 7 ranges are: 0, [1, 5], [6, 10], [11, 20], [21, 30), 30, (30, 127]. We choose these ranges empirically by correlating the statistical patterns of the ranges with the device brand.

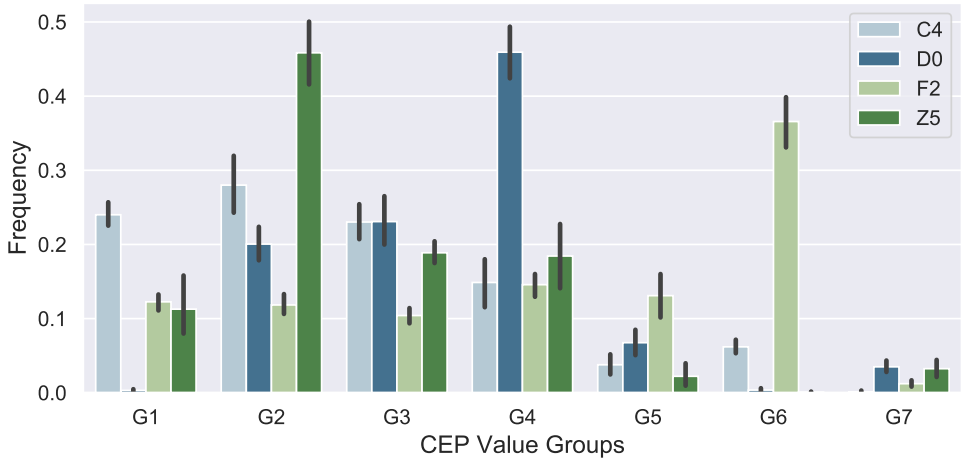


Figure 3.14: Comparison of the CEP value frequency for 4 difference divices. G0 to G7 correspond to the seven PRx controller feature frequency range in Table 3.2.

Table 3.2 summarizes the features used in classification. These features are collected in both steady charging states and operating point switching transient states, through which

Table 3.2: The list of features extracted from a complete scan.

Feature Group	Feature	Value Range
Oscillator feature	Domain 1 CEP interval	[40, 60] ms
	Domain 1 CEP interval peak log-probability	
	Domain 2 CEP interval	[140, 160] ms
	Domain 2 CEP interval peak log-probability	
	Domain 3 CEP interval	[235, 270] ms
	Domain 3 CEP interval peak log-probability	
Sample time	Time needed in a complete scan	>0
Contact range	The range that PRx interacts with PTx	[220, 260] control unit
PRx controller feature	Number of packets	>0
	CEP value = 0 frequency	[0, 1]
	CEP value $\in$ [1, 5] frequency	[0, 1]
	CEP value $\in$ [6, 10] frequency	[0, 1]
	CEP value $\in$ [11, 20] frequency	[0, 1]
	CEP value $\in$ [21, 30) frequency	[0, 1]
	CEP value = 30 frequency	[0, 1]
	CEP value $\in$ (30, 127] frequency	[0, 1]

QID extracts the fingerprints in the oscillator, coil and the controller for classification. We envision that the PRx controller features can separate the device brand and the oscillator features can then further separate the devices within the same brand. Figure 3.14 shows the frequency distribution grouped by the CEP value range for 4 different devices. As we can see, different PRx brands exhibit significantly distinct CEP value preferences during the wireless charging feedback control. Therefore, these PRx controller features can be used to categorize the device brand effectively.

3.5.2 Classification

The QID server classifies Qi-compliant devices by an ensemble classifier, also known as "bagging classifier", comprising of Support Vector Machine (SVM) [10], AdaBoost [30] with decision tree as weak learner, decision tree classifier [12], k-Nearest Neighbor (kNN) [20], and Linear Discriminant Classifier (LDC) [59]. The bagging algorithm in our design utilizes a voting system, as shown in Figure 3.15. When 3 or more classifiers are outputting

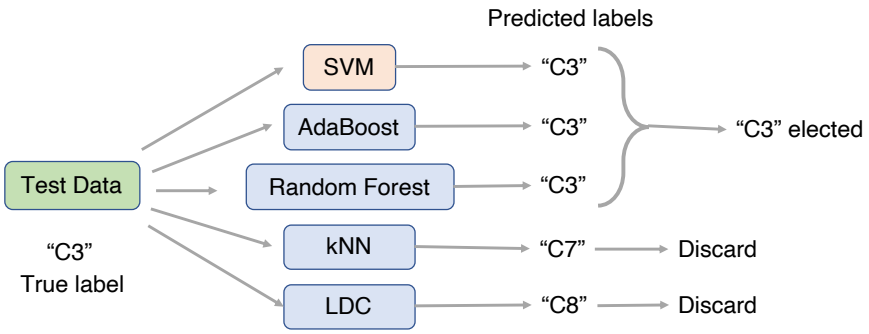


Figure 3.15: Classification process with a bagging classifiers in QID server.

the same device label, the bagging classifier chooses it as the final decision. Otherwise, the output of the SVM is chosen because it has relatively higher accuracy than the other classifiers. We design this classifier architecture by tuning the classifiers empirically.

The QID server stores all the extracted features and their corresponding device labels in a feature table. Then it utilizes the repeated random sub-sampling cross-validation, also known as Monte Carlo cross-validation [27], to split the data into training and testing set randomly. Finally, the classifier models are trained with the training set and validated with the testing set. The mean and the standard deviation of accuracy from the results of the sub-sampling experiments are recorded. It is shown [36] that cross-validation evaluation introduces neighborhood bias to the time-continuous sliding window frame data, which results in overly optimistic model evaluation estimations. However, in our experiments, each of the samples is collected in a wide span in the time domain. In other words, all the features extracted from a complete experiment scan are independent of each other. As a result, the cross-validation is suitable for our experiments.

The objective of the QID server is focused on classifying the device into one of the known classes. This design is applicable to the scenarios where the devices are already fingerprinted. For instance, a company may register and fingerprint all the work devices of employees, and then use QID to track the location of each device. However, QID can be easily extended to recognize new devices via online learning. For instance, by setting a detection threshold in the classifier, QID can identify whether the newly collected sample

corresponds to any device that is already recorded. If the sample's probability of corresponding to an existing device is low, QID can recognize the device as a new one.

The QID server stores all the extracted features and their corresponding device labels in a feature table. Then it utilizes the repeated random sub-sampling cross-validation, also known as Monte Carlo cross-validation [27], to split the data into training and testing set randomly. Finally, the classifier models are trained with the training set and validated with the testing set. The mean and the standard deviation of accuracy from the results of the sub-sampling experiments are recorded.

The objective of the QID server is focused on classifying the device into one of the known classes. This design is applicable to the scenarios where the devices are already fingerprinted. For instance, a company may register and fingerprint all the work devices of employees, and then use QID to track the location of each device. However, QID can be easily extended to recognize new devices via online learning. For instance, by setting a detection threshold in the classifier, QID can identify whether the newly collected sample corresponds to any device that is already recorded. If the sample's probability of corresponding to an existing device is low, QID can recognize the device as a new one.

3.6 Implementation

In this section, we present the implementation of the QID system. Figure 3.16 shows a QID sensor prototype.

QID sensor base station. The base station is the mechanical component of the QID sensor, which is built on a clear acrylic board. The two stepper motor linear sliders enable the motion along the x and y direction respectively, with a moving distance of 90 mm each. A switch is added to one end of each screw, such that the MCU can reset the position each time the system is powered on. Another clear acrylic board (not shown in the figure) supported by four nylon hex spacers is the surface that the mobile device is put on. The cost of the mechanical components is less than $20. Therefore it is feasible to be massively

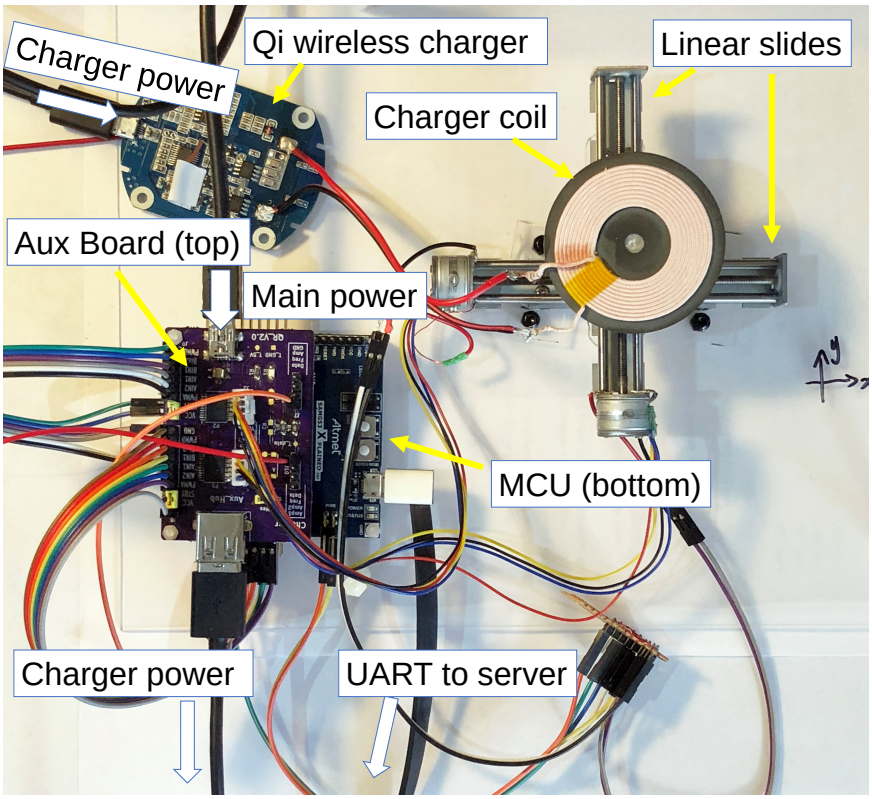


Figure 3.16: A prototype of the QID sensor.

deployed in the public area.

Embedded controller and motor driver. At the center of the QID sensor, the Atmel SAMG53N19 [4] MCU is employed, which is responsible for decoding and timestamping packets, driving the stepper motors, and sending collected data to the QID server. The MCU supports the UART communication with the server via a USB virtual COMM port. The motor driver IC is Toshiba TB6612FNG, which shares the power with the Qi PTx. The peak motor driving current is around 150 to 200 mA, which is negligible to the Qi wireless charging system because a typical COTS USB charger can provide 2000 mA current at 5 V.

Qi-compatible power transmitter. We choose a COTS GMYLE Mini Qi Charging Pad as the PTx, which is connected to the MCU via a data flow debug pin. The PTx coil is extended with a pair of wires, which provides extra flexibility, such that the coil moves without dragging the charger circuit board around.

The QID server. At the server side, the feature extraction and classification modules are

implemented using approximately 1,100 lines of Python codes, including the `pySerial` UART library for the QID sensor communication handler and the machine learning library `scikit-learn` [64] for classification.

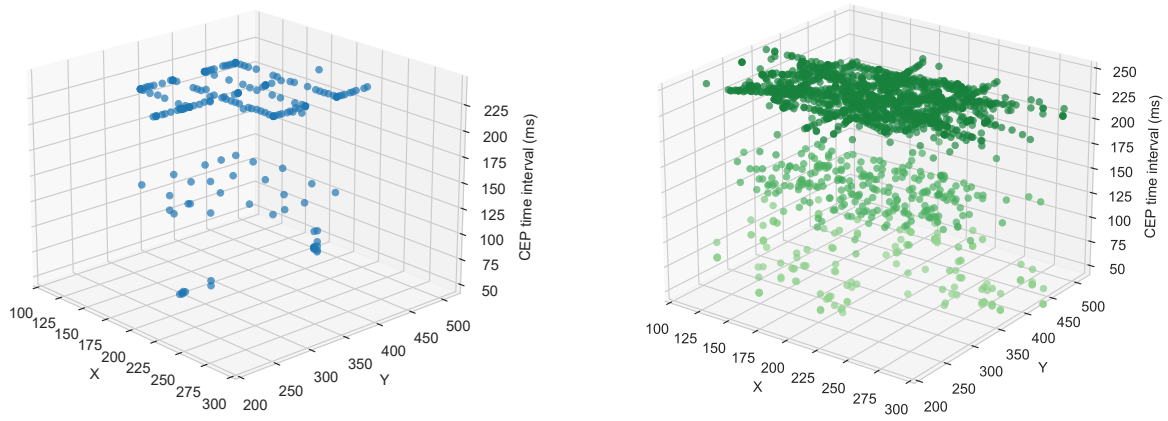
3.7 Evaluation

In this section, we present the performance evaluation of QID based on 52 Qi-compliant devices. We first present the evaluation settings and then discuss feature analysis, measurement delay analysis, classification accuracy, feature backward search, and accuracy breakdown test.

3.7.1 Evaluation Settings

We evaluate 52 Qi-compliant devices in total, including 7 Google Nexus 4 (labeled as "N") and 45 attachable PRx modules from 6 different manufacturers, including DigiYes, Hugchg, and RAVPower. We note that the ICs in these modules are widely used in mainstream mobile devices. For example, the Texas bq51013B in the DigiYes modules is also adopted by Google Nexus 5. For each device, we conduct 10 complete independent scans to collect fingerprints. In total, there are 520 scan samples. To simulate the users' device placement behavior in the real world, we alter the phone placement manually. Specifically, for the first scan, the phone is aligned with the x axis of the motion plane. For the next seven scans, the device is rotated counter-clockwise for 45° each time. For the last two scans, the phone is placed on the pad at a random angle.

We visualize the CEP packet intervals with respect to the PTx coil location in Figure 3.17. As shown in Figure 3.17a, merely using the sample collected in one experiment may lead to biased results because the features are extracted from a limited number of couplings between the two coils. While the data points from multiple measurement rounds are aggregated, as shown in Figure 3.17b, the point cloud covers the majority of the contact range between the two coils, providing adequate ground truth for classification, which can



(a) Point cloud in 3D space for one sampling experiment.

(b) Point cloud after aggregation from multiple experiments. Different "layers" of the points correspond to different peaks in Figure 3.13.

Figure 3.17: Point cloud illustration.

accommodate unpredictable device placements.

To quantify the contribution of the motion platform, we repeat this whole process for once *without* moving the charger coil with respect to the PRx. We consider this as our *baseline* and will discuss it in Section 3.7.3.

In classification, the training-testing split ratio is 7: 3. In other words, 7 out of the 10 samples for each device are randomly chosen to train the QID classifier model, and the remaining scan samples are for testing. Such a process is repeated 10 times. The average accuracy and the standard deviation of each classifier are reported. In addition, the hyper-parameters in all the implemented classifiers are tuned by the grid search. As discussed in Section 3.5.2, we assume all the devices are already fingerprinted and recorded in the database.

3.7.2 Measurement Delay

The measurement delay is defined as the time delay from the moment when the power receiver is booted to the moment that the server produces a device label. Specifically, the

Table 3.3: The measurement delay in the QID system

Symbol	Definition	Mean (s)	Std (s)
T_1	Coil symmetric axis alignment time	8.050	0.927
T_2	Fingerprinting time	55.478	6.092
T_3	Feature extraction time	0.147	0.006
T_4	Classification time	0.001	N/A

measurement delay T_M is

$$T_M = T_1 + T_2 + T_3 + T_4$$

where T_1 is the coil symmetric axis alignment time, T_2 is the fingerprinting time, T_3 is the feature extraction time, and T_4 is the classification time. The means and standard deviations of these measurement delay terms are presented in Table 3.3.

As one can see, the fingerprint phase time T_2 is around 55.5 s, which contributes the most to the total measurement delay. Thus reducing the time T_2 is crucial in further optimizing the measurement delay T_M.

The measurement delay is actually acceptable due to the characteristics of wireless charging. First, unlike other wireless communication systems where the user is usually in mobility, the charging process usually takes more than 10 minutes, during which the user device remains stationary. Second, in the targeted scenarios, such as a coffee shop that offers location and personalized services to customers, the users need to register their devices before using such user-identification service. During the registration process, QID can collect 8-10 different samples for future recognition. Finally, previous systems that exploit clock drifts for device identification have similar delay performance. For example, BlueID [38] takes 21 seconds for data traffic or 65 seconds for voice traffic to guarantee the low measurement error. In [44], it takes the system hours to collect enough packets in order to distinguish devices. Therefore, the 60-second measurement delay in QID is acceptable.

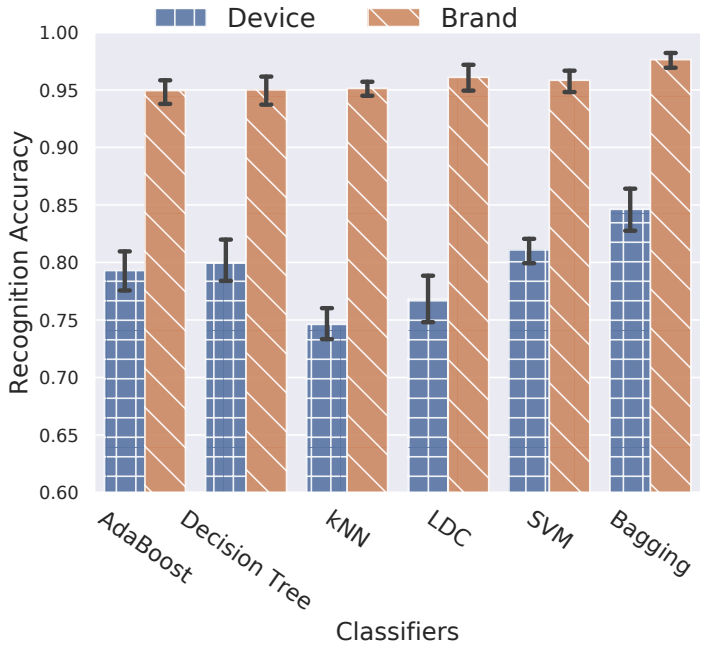


Figure 3.18: The cross-validation score and device brand detection accuracy of different classifiers.

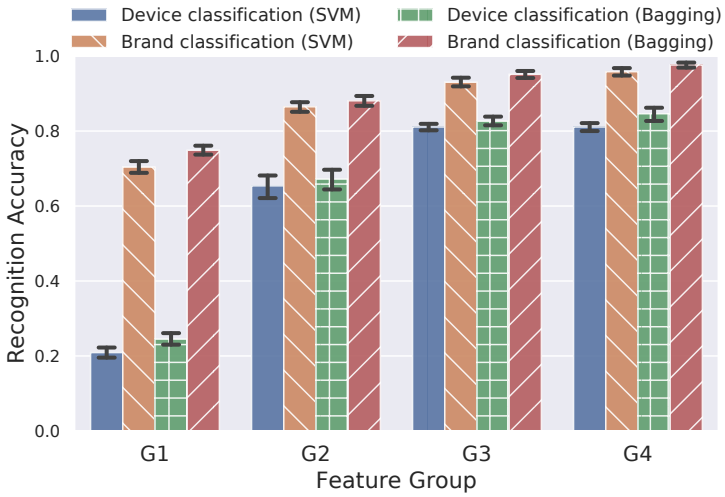


Figure 3.19: The impact of feature selection on the classification accuracy. G1: classification without the CEP time interval features; G2: all features are included, but they are measured without the motion platform; G3: classification performance using the CEP time interval features only; G4: all features are included.

3.7.3 Classification Accuracy

We first present the overall test accuracy of the cross-validation study. The overall accuracy is the ratio of the number of correctly classified scan samples to the size of the test

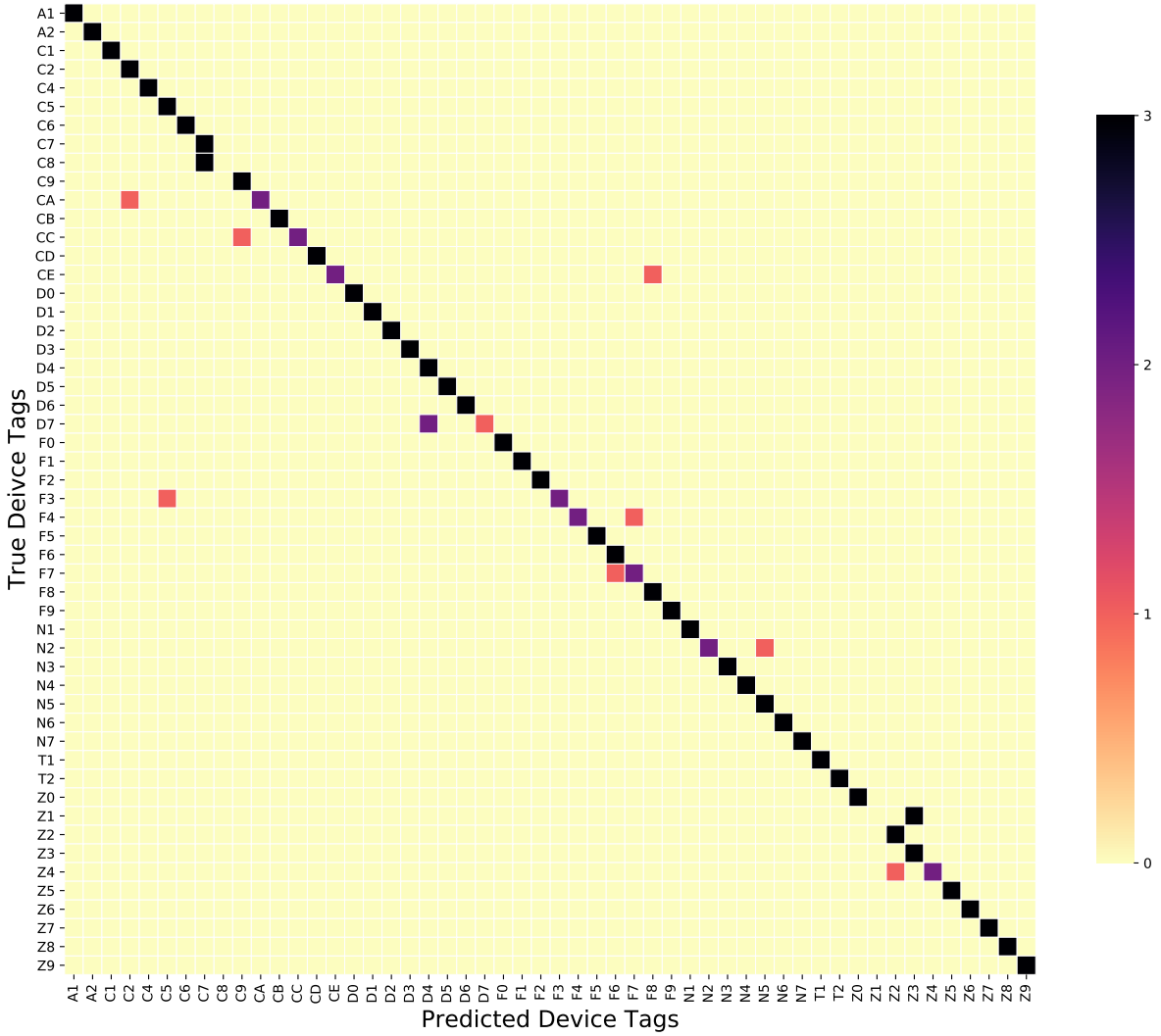


Figure 3.20: Confusion matrix of the 52 evaluated devices.

set. Figure 3.18 shows the means and standard deviations of the implemented classifiers from 10 repeated random sub-sampling cross-validation folds. As shown in the figure, all the implemented classifiers can recognize the device brand with a mean validation accuracy up to 96.1%. Particularly, the bagging classifier identifies the brand with up to 97.9% mean accuracy. The mobile device brand classification accuracy is of interest because of the following two reasons. First, it is the foundation of device recognition. As mentioned in Section 3.3.1, although the CEP interval is a good device separator, it con-

fuses some devices from different brands. If the device brand is successfully identified, the QID server can reduce the range of device candidates and increase the overall device classification accuracy. Moreover, brand recognition can enable applications like device brand-specific advertisement. For device identification, the bagging classifier achieves an average accuracy of 85.2%. The highest accuracy achieved by QID is 89.7%. To illustrate the classifier performance, a confusion matrix is plotted in Figure 3.20. Generally, the misclassified samples are from the devices of the same brand that have close clock drifts. For instance, there are two devices, namely "C8" and "Z3", whose all 3 test samples are misclassified. However, some devices are classified into different brands due to their close values in feature space.

Next, we quantify the performance of the motion platform, namely the multi-coil array. The G2 in Figure 3.19 shows the baseline result, where the motion unit is not enabled. Each device is sampled without the motion control for 55 seconds, corresponding to the delay T_2 in Section 3.7.2. We plot the classification accuracy of QID in Figure 3.19 G4 for comparison. As we can see, the device recognition rate increases by 17% by both SVM and bagging classifiers when the motion platform is enabled. The brand recognition accuracy is also boosted by 8%. It is indicated that the motion platform plays an important role in achieving reliable and sufficient device features for classification due to its ability to extract fingerprints from more spots, which captures a more complete profile of a device.

3.7.4 Impact of Feature Selection

Not all features are equally important. Figure 3.21 and Figure 3.22 shows the distribution of two features respectively, obtained from 42 out of the 52 devices, namely the total number of packets per scan and the frequency of the CEP value 0. The distribution of other CEP value frequencies yields similar trends as shown in Figure 3.22 and is thus omitted. These two features contain more noise than the CEP interval (as shown in Figure 3.8). Nonetheless, intuitively, these features are able to separate device classes to some extent. We next

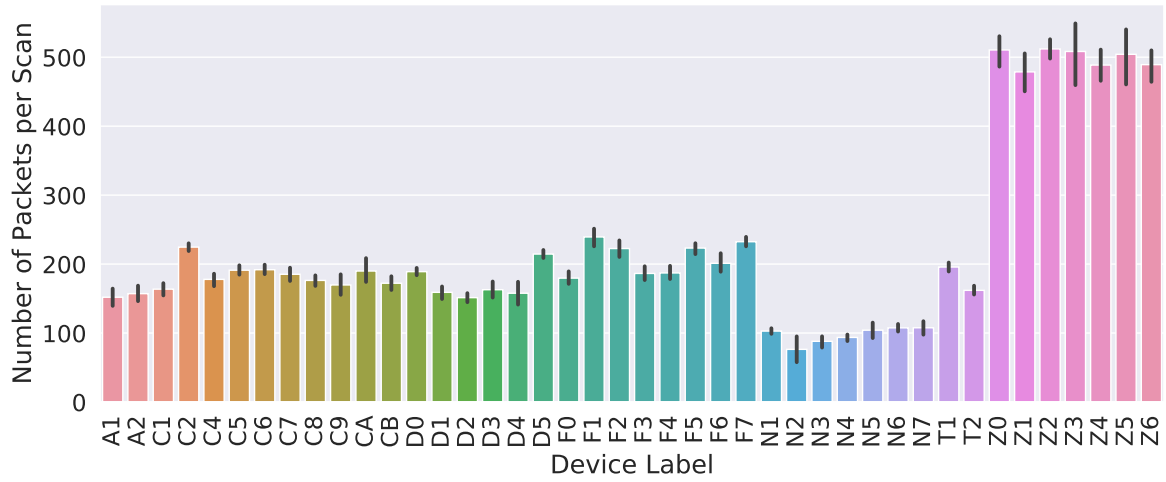


Figure 3.21: Number of packet per scan feature distribution of 42 devices (10 samples per device).

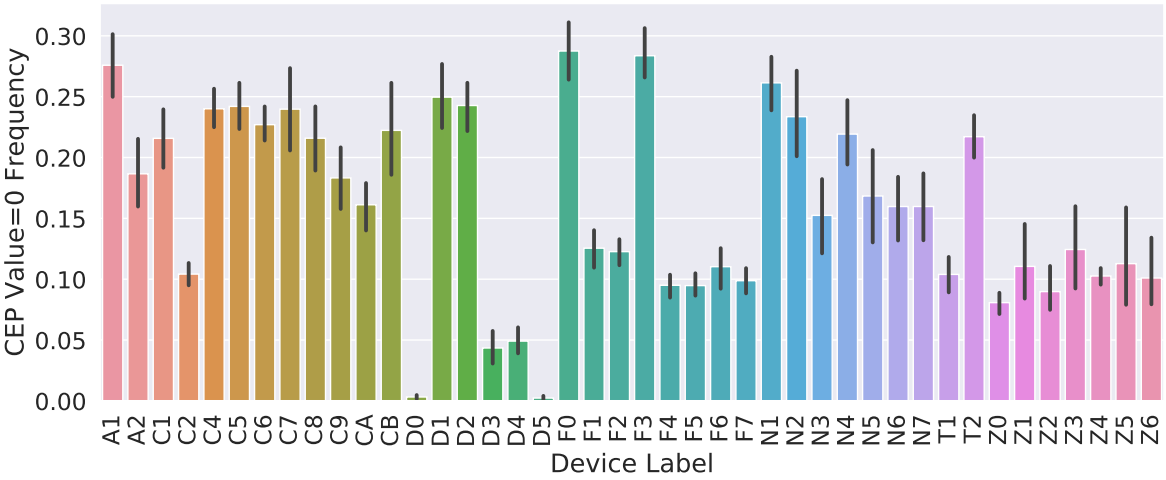


Figure 3.22: The frequency of CEP value equaling 0 distribution of 42 devices (10 samples per device).

conduct the backward search to evaluate the effectiveness of each selected feature.

We perform two case studies. In the first case "G1", we evaluate the bagging classifier without the CEP time interval features, i.e., the onboard oscillator fingerprints. In other words, only the CEP value features are used. In the second case "G3", we evaluate the bagging classifier with only the CEP time interval features. The results of these two case studies are shown in Figure 3.19. We observe that the device recognition accuracies

of both the bagging and the SVM classifier degrade to about 21% in G1, which indicates the significant contribution of the onboard oscillator fingerprint to device identification performance. Another observation is that, although the CEP value features fail in device recognition, they are still able to distinguish the brands with 75% accuracy by the bagging classifier. Next, we compare G3 and G4. We can see that both the device and brand recognition accuracies of the bagging classifier are improved by about 2.5% by adding the CEP value features to the CEP time interval features. This indicates that although the CEP value features are not as important as the onboard oscillator fingerprints, it helps QID to reduce uncertainty and achieve higher accuracy. However, as the number of devices increases, the chance of CEP interval (PRx oscillator) feature overlapping is expected to increase. In such a case, the PRx controller fingerprints will provide the necessary device brand identifies, thus reducing the collision in the feature space.

3.7.5 Recognition Accuracy Breakdown

Another characteristic of the QID system is that how robust the selected fingerprints are. In other words, how is the classifier performance influenced when more devices are added to the feature database? Here we present the results of the recognition accuracy breakdown. In this case study, we evaluate both the device and brand recognition accuracy of the bagging classifier concerning a sequence of the device numbers from 5 to 50, with an increment of 5. For each device number N_i in the list, we first randomly choose N_i devices out of the 52 devices and then perform the Monte Carlo cross-validation on their scan samples 10 times to obtain the mean test accuracy. For each N_i, such a process is repeated for 6 rounds. Finally, the mean and standard deviation from the results of these 6 rounds are recorded, as shown in Figure 3.23. The brand recognition accuracy keeps at a high level around 97% regardless of the increase in the number of devices. However, the device recognition accuracy decreases by about 1.3% each time the number of devices increases at increments of 5. If this trend continues, when the number of the device reaches about

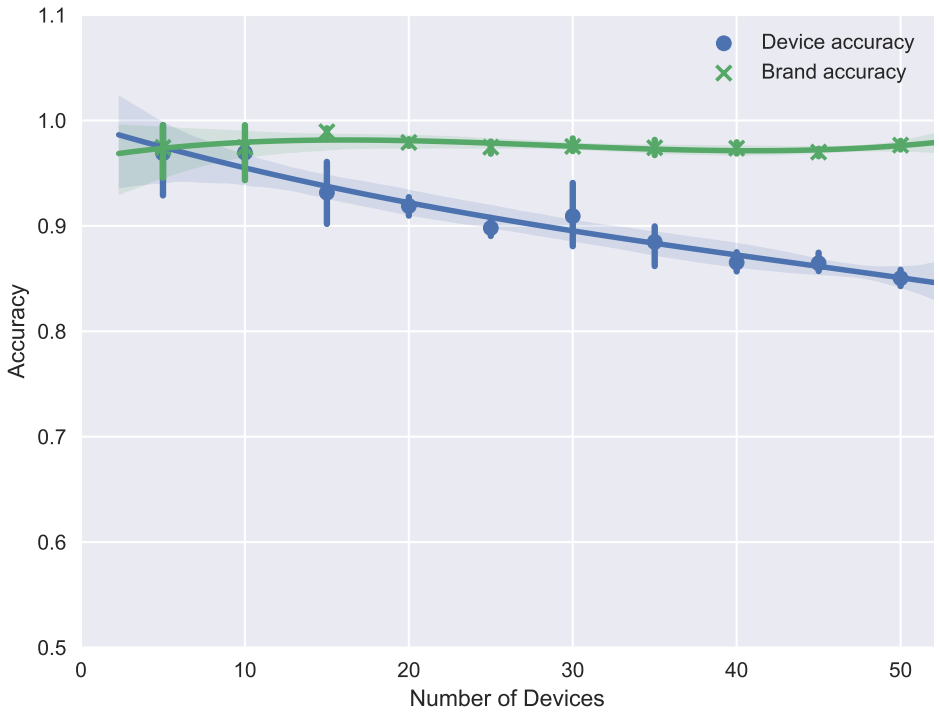


Figure 3.23: Device recognition accuracy changes with the number of devices

180, the device accuracy decreases to 50%. However, we note that the power receivers may have much higher diversity regarding the device brand in real-life scenarios than that in our evaluation setting. Therefore, the QID can potentially benefit from the high brand accuracy and thus accommodate more devices than the number of device 180 calculated above.

3.8 Conclusion and Discussion

In this research work, we present our design and implementation of QID, the first system that recognizes Qi power receivers during wireless charging using fingerprints from the onboard oscillator, coil characteristics, and control scheme of the wireless charging system. QID also employs a movement unit to emulate multi-coil power transmitters, which allows for fine-grained fingerprinting. Our evaluation results show that QID achieves an overall identification accuracy of up to 89.7%, with an average of 85.2%. Moreover, QID can recognize the device brand with an average accuracy of 97.9%. Therefore, we demonstrate

the feasibility of leveraging public wireless charging infrastructure for tracking mobile users and providing ID/location-based services. Our results also open up new research questions on how to prevent the leakage of user's location with the increasing wireless charging station deployment in public.

QID has several limitations. First, unlike other wireless communication systems where passive remote sensing and recognition are possible, QID adopts a user-initiated device recognition approach. This narrows its applications because it requires physical contact between the device and the sensor. However, this could also be an advantage because it preserves the user's awareness and thus protects the user's privacy. Second, at this stage, QID requires motion parts to achieve fine-grained fingerprinting. We envision achieving device recognition in wireless charging using only stationary multi-coil chargers. However, since the commercially available multi-coil chargers are not ready for device recognition yet, our QID sensor implementation mainly focuses on emulating a multi-coil array with motion control. We expect no motion unit is needed in the real-world implementation and deployment. Nevertheless, the mechanical structure could be possibly re-designed to achieve a smaller form factor. Finally, we note that the charging process may cause several short-time charging disruptions (about 1 to 2 seconds each) due to the decoupling between the PTx and PRx coils. In such a case, the user experience may be potentially impacted. However, as discussed above, the measurement delay is about 55 seconds. After a device is successfully identified, the PTx coil will move to a position (or switch to a particular physical coil) that achieves the maximum coil coupling to continue the power delivery process.

Our findings have important implications for user privacy. Privacy is a primary concern in device recognition systems like QID. In fact, the Qi specification itself keeps device information securely. For example, QID can only identify a device based on the physical fingerprints of the Qi PRx module. It will not and can not record any information about the phone itself including operating system, phone number, as well as battery-related values,

such as voltage level, energy percentage, and health record.

From another perspective, the public should be aware of their privacy when using wireless charging because it would possibly leak the location information of a particular device. Hackers may precisely localize a targeted user for malicious purposes. There are sufficient reasons for the Qi PRx module to offer users the choice to generate their Device ID randomly. The mobile phone manufacturer and the mobile phone operating system should also offer the choice to shut down the wireless charging functionality when the user intends to. To the best of our knowledge, they are not implemented in any Qi-based wireless chargeable device yet. The location of a mobile device may be tracked when the user charges mobile devices in public wireless chargers. Mitigating such possible user privacy breaches is open to future problems.

Another open problem is to design a compact coil antenna to extract the data directly from the wireless power interface, such that the QID sensor can be non-intrusive to the PTx. It is also possible to explore new QID sensor fingerprinting trajectories to further reduce the measurement delay. Although the users usually initialize their mobile device registration by themselves in our targeted scenarios, we call for the design of efficient on-line machine learning algorithms to classify unknown devices, such that QID provides an easy-to-use interface and enables a wider range of applications. We can achieve this by quantifying the similarity between the incoming sample features with the ones already in our database. If the difference exceeds a threshold, QID determines the sample belongs to a device that has not been seen before.

CHAPTER 4

UNDERSTANDING POWER CONSUMPTION OF NB-IOT IN THE WILD: TOOL AND LARGE-SCALE MEASUREMENT

4.1 Chapter Introduction

Narrowband Internet-of-Things (NB-IoT) is a low-power wide-area network (LPWAN) specification developed by the 3rd Generation Partnership Project (3GPP) in 2016. NB-IoT envisions an anytime, anything connectivity paradigm for a wide spectrum of low data rate, large volume, and long lifetime IoT applications, including smart grid, smart street-lamp, parking management, air quality sensing, and intelligent agriculture. Currently, NB-IoT has been launched globally with 93 commercial networks, while there are 140 operators in 69 countries investing in NB-IoT network deployment. One of the key features of the NB-IoT network is the promise of long battery life, up to 10 years. Understanding how the energy is spent and identifying possible energy loopholes in the NB-IoT network are thus of great importance.

Unfortunately, to date, the key aspects of NB-IoT performance and power consumption have not been well understood, especially to developers and academic researchers. This is due to three key challenges. First, NB-IoT is a closed cellular network deployed by operators on licensed spectrum, where the base stations can not be accessed for public measurements. The message-level interactions between the node and base station are largely inaccessible to the developers and researchers. Second, NB-IoT measurement is fundamentally different from 3/4G cellular network measurement, where the latter could be conducted through mobile applications installed on massive mobile devices. In contrast, an IoT application may consist of numerous nodes embedded in the environment over a large geographic region, which presents a high barrier for understanding the performance of NB-IoT in the wild. In particular, NB-IoT networks differ significantly due to

variations of operator configurations, modules from different vendors, and location profiles. Finally, there lack effective tools that can expose the low-level diagnostic traces from NB-IoT nodes, support large-scale measurement studies, and capture the high level of heterogeneity of network operators, NB-IoT modules, and location profiles.

In this chapter, we propose our design of NB-Scope, the first hardware NB-IoT diagnostic tool that supports fine-grained fusion of power consumption and protocol message traces for both real-time in-lab benchmarking and field testing. NB-Scope is a powerful tool that can advance the research of NB-IoT by allowing to instrument large-scale operational NB-IoT networks and experiment with various protocol-level optimizations. Next, we conduct a large-scale field measurement study based on the deployment of 30 NB-Scope nodes at over 1,200 locations in 3 regions of 2 countries during a period of three months. Our in-depth analysis of the collected 49 GB debug logs and current consumption traces reveals several important insights into the power consumption of NB-IoT in the wild. We showed that nodes yield significantly imbalanced energy consumption across different locations, operators, and module vendors. For instance, the ratio of the highest and lowest energy consumption of nodes can be 75:1. By decomposing the energy consummation by radio access phrases in a fine-grained manner, we showed that such performance variance can be attributed to several key factors including poor network coverage level, long-tail power profile due to conservative inactivity timer settings, and excessive control message repetitions during random access control.

The rest of this chapter is organized as follows. First, Section 4.2 briefly introduces the NB-IoT technology, as well as the radio access procedures and the energy management. Nest, Section 4.3 presents NB-Scope design, a platform for large scale NB-IoT measurement study. Finally, Section 4.4 presents our discovery through a large-scale measurement study.

4.2 NB-IoT Primer

In this section, we provide the background of NB-IoT technology, including its main features, frame structure, physical channels, random access procedure, and energy management.

4.2.1 Features of NB-IoT Technology

NB-IoT has several advantages over many other wireless communication technologies, such as 4G-LTE, Wi-Fi, and ZigBee.

- **Wide coverage**. Compared to GPRS and LTE, the maximum coupling loss budget of NB-IoT increases by 20 dB, which improves its coverage ability in signal-limited environments, such as underground parking lot, basement, and garage.
- **Low power consumption**. By streamlining the message exchanges and leveraging multiple power-saving methods, the end node has the chance to reach 5-years or even 10-year battery life in specific scenarios.
- **Low cost**. The narrow bandwidth, low power consumption, and low data rate lead to the low complexity in the NB-IoT modem and chipset design, which lowers the cost at the node side. In the meantime, NB-IoT can reuse the existing 4G-LTE infrastructures, which largely decreases the deployment cost at the network operator side.
- **Massive connection**. One single NB-IoT cell can support more than 50,000 nodes connecting to the network core, which is 50 to 100 times larger than the 2G, 3G, and 4G network.
- **Licensed bands**. NB-IoT supports inband mode deployment, which coexists with the current LTE network. In the meantime, it can reuse the licensed band of 2G and

3G. All these bands are licensed, meaning that there is less interference than other ISM-based technologies, and the network coverage in a wide area can be guaranteed.

- **Security**. NB-IoT leverages the security approaches from 4G LTE, with 2-way authentication and data encryption over the air, providing a more secure pipeline for the packet data than other LPWAN technologies.

4.2.2 Frames and Channels

In an NB-IoT network, an end node, commonly known as a User Equipment (UE), is connected to the Evolved Packet Core (EPC) via an eNodeB base station. To achieve this, NB-IoT specifies a dedicated frame structure and a set of channels both physically and logically to ensure the synchronization and effective communication between the UEs and the eNodeB. We briefly introduce them as follows.

- **Frames and time-frequency domain resources**. Figure 4.1 shows the frame structure of NB-IoT in time domain. It is a hierarchical structure, from hyperframes to radio frames, then to subframes and slots as the granularity increases. Each slot consists of 7 Orthogonal Frequency Division Multiplexing (OFDM) symbols. In the frequency domain, NB-IoT utilizes 12 subcarriers, 15 kHz each (in Single-tone mode), 180 kHz bandwidth in total (200 kHz in reality due to the employment of guardband).

  To allow the UE to synchronize before attaching to the network, the eNodeB broadcasts two synchronization signal *Narrowband Primary Synchronization Signal* (NPSS) and *Narrowband Secondary Synchronization Signal* (NSSS) periodically in the radio frames, as shown in Figure 4.2 in red and yellow color, respectively. After synchronization in clock signals, UE starts to listen to the system information from the downlink channels.

- **Downlink physical channels**. There are three physical channels in the downlink (DL) direction, i.e., from eNodeB to the UEs, namely *Narrowband Physical Broadcast*

Channel (NPBCH), *Narrowband Physical Downlink Control Channel* (NPDCCH), and *Narrowband Physical Downlink Shared Channel* (NPDSCH). NPBCH carries the system configuration information to guide the UE to attach to the network. NPDCCH and NPDSCH carry the scheduling or controlling messages and the DL data packets for the UE, respectively. NPDCCH and NPDSCH are the most commonly allocated channels in the NB-IoT downlink, as shown in Figure 4.2.

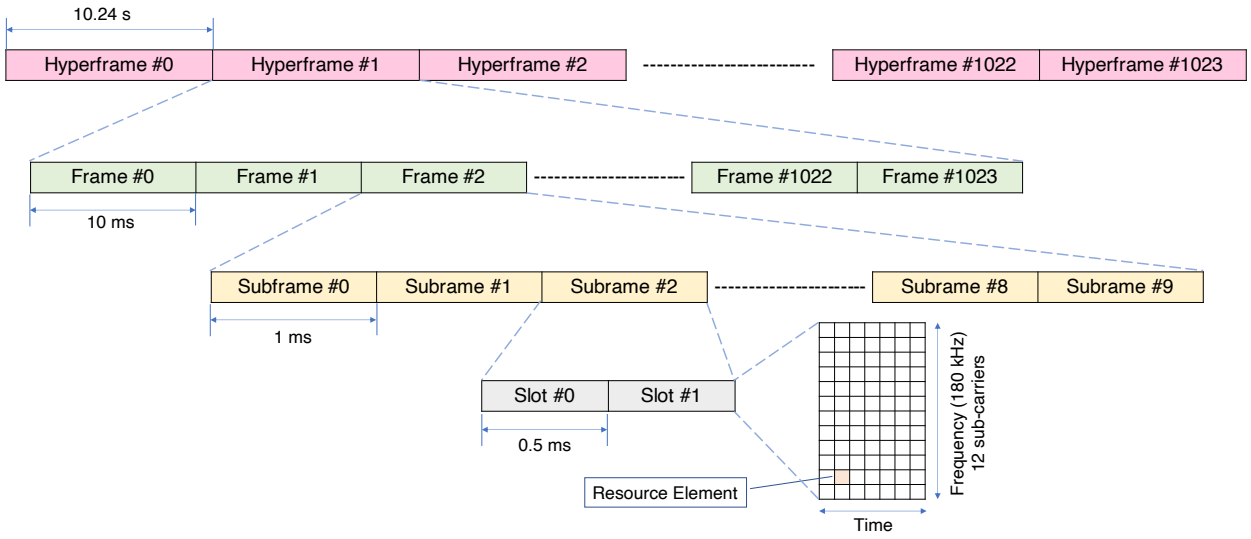
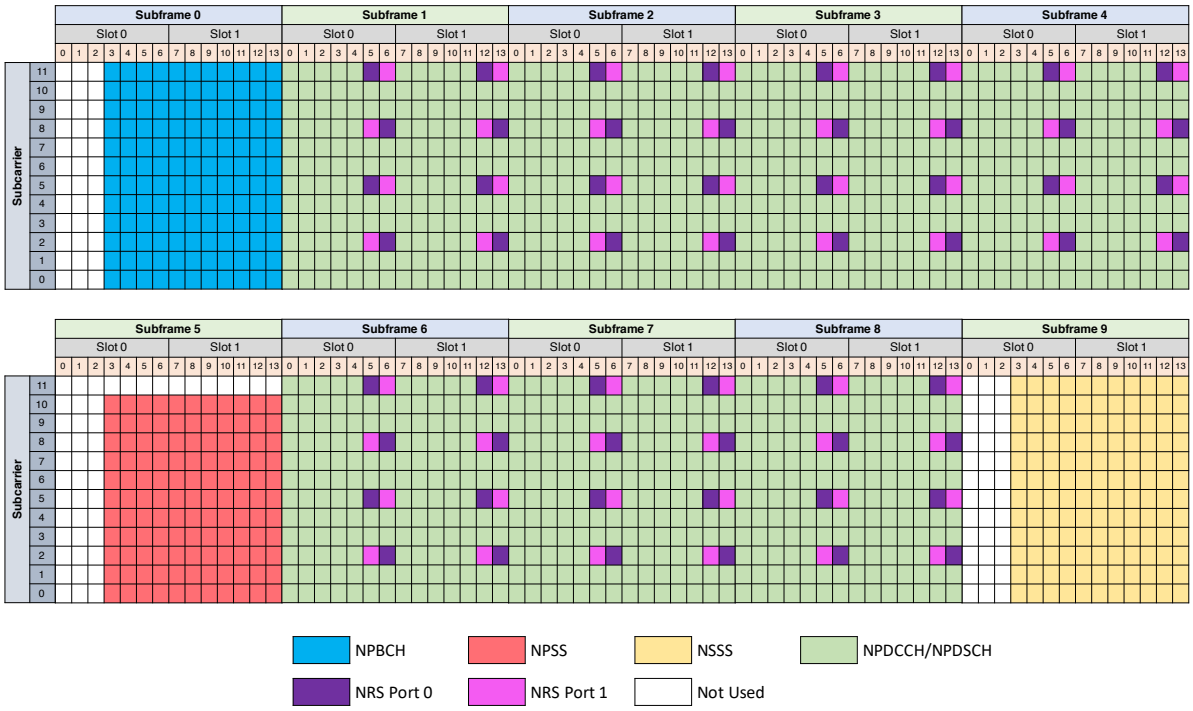


Figure 4.1: NB-IoT frame structure.

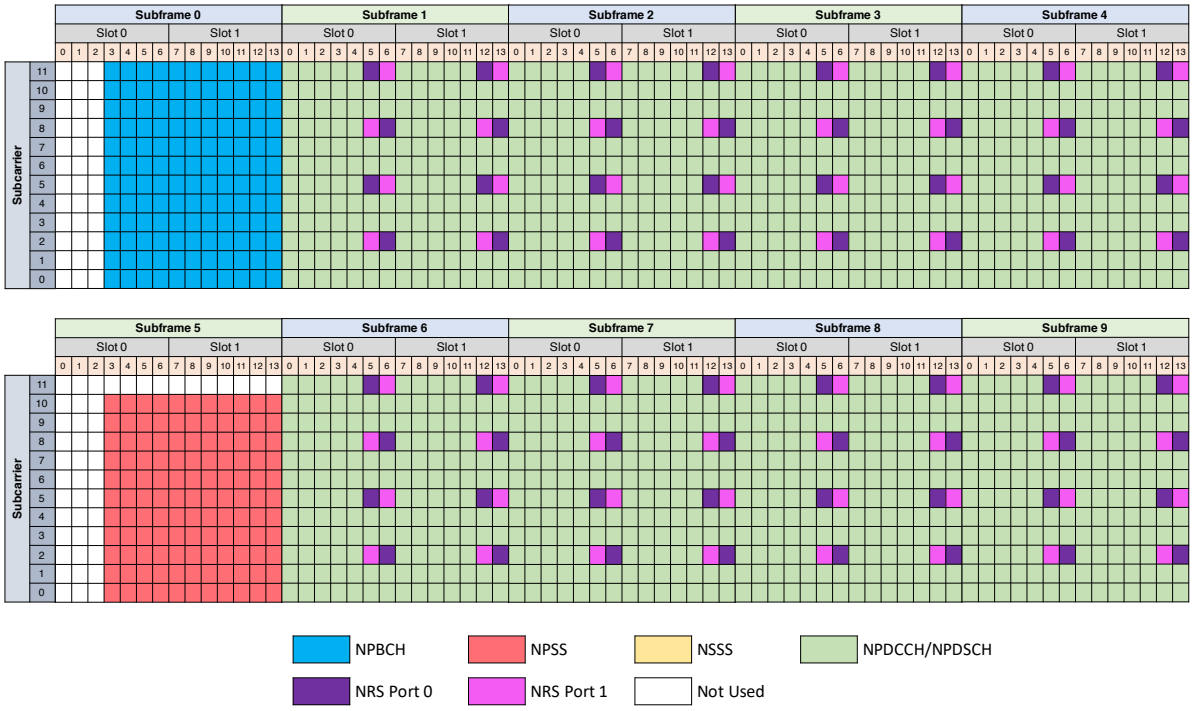
- **Uplink physical channels**. There are two physical channels specified for uplink (UL) transmissions, namely *Narrowband Physical Random Access Channel* (NPRACH) and *Narrowband Physical Uplink Shared Channel* (NPUSCH). NPRACH bears the random access preamble from UE, while NPUSCH carries all the UL packet data from the UE.

4.2.3 Random Access Procedure

The NB-IoT UE follows a series of procedures to establish *Radio Resource Control* (RRC) Connection and send data packets to application servers, as shown in Figure 4.3. The procedure is summarized as follows.



(a) Frame with even number.



(b) Frame with odd number.

Figure 4.2: NB-IoT subframe structure in either Standalone or Guardband deploy mode. Every 10 subframes make a radio frame.

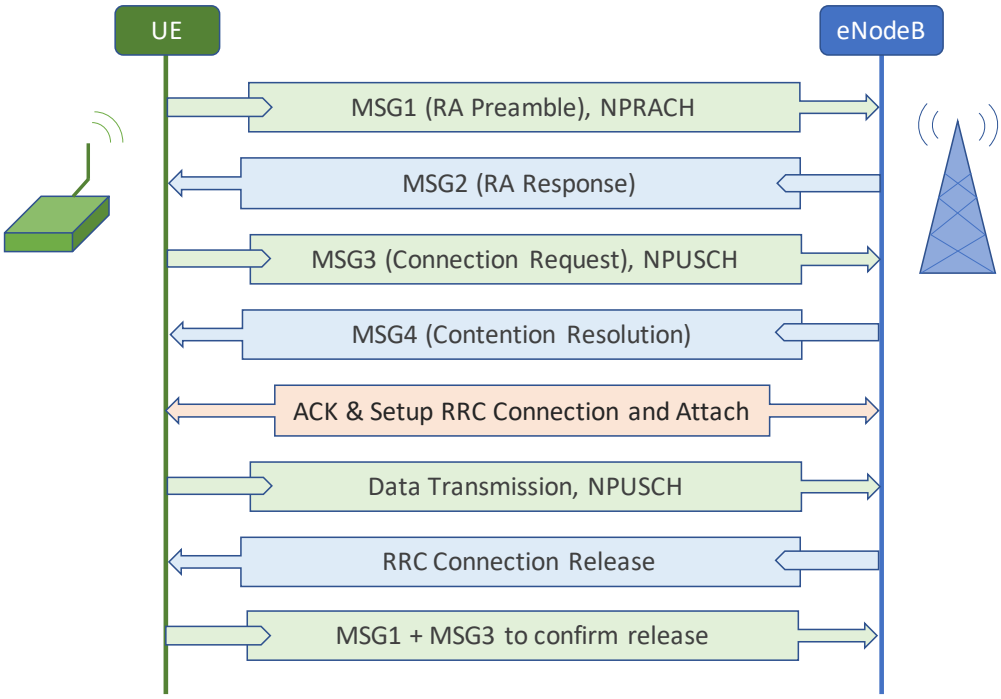


Figure 4.3: Signalings between UE and eNodeB base station during a UL packet transmission.

1. The UE initiates a contention-based *Random Access* (RA) by sending a *Random Access Preamble* (`MSG1`) to the eNodeB in NPRACH. The eNodeB replies with a *Random Access Response* (RAR), also known as `MSG2`, which allocates resources for the UE to send subsequent requests.

2. Next, the UE transmits the `MSG3` to the eNodeB, carrying the requested resources, in the NPUSCH using the resource allocated by the eNodeB. In the meantime, it starts a timer immediately. The UE considers itself failed if the timer expired.

3. After receiving `MSG3`, the eNodeB resolves contention from all the UE requests, allocates resource, and then notifies the UE by sending a Contention Resolution (`MSG4`). If the UE receives `MSG4` before CRT timeout, the RA procedure completes successfully and the UE sets up an RRC connection to the eNodeB. Otherwise, the UE needs to go back to step-1 and sends `MSG1` to re-initiate RA.

4. The UE then sends data packets to the eNodeB using the scheduled resource in NPUSCH.

5. After data transmission, the UE enters passive reception, called `Inactivity` state until the RRC connection is released, initiated by the eNodeB.

6. Finally, the UE sends the *RRC Connection Release Confirm* via `MSG1` and `MSG3` again to the eNodeB to mark the end of a packet transmission cycle.

We note that `MSG1` is transmitted in the Narrowband Random Access Channel (NPRACH). Both `MSG3` and the data packet transfer blocks are transmitted in the NPUSCH, which is one of the major contributing factors to total energy consumption.

4.2.4 Energy Management

UE working modes. To reduce power consumption, the UE is operated in four working modes, namely `Active`, `Idle`, extended Discontinuous Reception (`eDRX`), and Power Saving Mode (`PSM`). The UE is in `Active` mode when the radio is transmitting or receiving, consuming current up to tens or hundreds mA. In `Idle` mode, the radio is inactive but remains awake. In `eDRX` mode, the UE turns off its radio for a specific period negotiated with the eNodeB. The `eDRX` mode of NB-IoT inherits the `DRX` mode of LTE, but specifies additional higher interval configurations, targeting the IoT application scenarios. The `PSM` mode is the most energy-efficient mode, where all UE components are turned inactive despite a wake-up timer until the timer expires and issues an interruption to activate the UE. In this manner, the current in `PSM` can be reduced dramatically to only several μA.

In typical NB-IoT applications, the nodes hibernate with `PSM` for most of the time and transmit a small number of packets daily.

Enhanced coverage level (ECL). NB-IoT adopts an open loop power control, where the UE decides its Tx power based on a set of pre-defined parameters broadcasted by the eNodeB and the lively measured reference signal strength. NB-IoT defines up to three *Coverage Enhancement Levels* (ECLs): ECL0, ECL1, and ECL2 that are adopted in good, moderate, and bad channel conditions, respectively. Each ECL corresponds to a specific uplink ra-

dio profile, which specifies the number of repetitions of MSG1 and MSG3, the number of sub-carriers, etc. The selection of ECL is determined by eNodeB and may vary across different network operators and locations. Before uplink radio access, the UE characterizes its channel condition by measuring *Reference Signal Received Power* (RSRP). It then compares the RSRP with the thresholds specified by eNodeB to determine its ECL.

4.3 NB-Scope Design

We have developed NB-Scope, which is an open-source experimental platform designed to enable fine-grained power consumption analysis for NB-IoT. In this section, we describe the design of NB-Scope in detail.

4.3.1 System Overview

The design objective of NB-Scope is two-fold. First, it should enable in-depth diagnoses of the energy efficiency of an NB-IoT network. Second, it should support heterogeneous NB-IoT modules with minimal system modifications, since UE modules of different vendors may differ in pin assignment, command language, and signaling message format of radio access logs. To achieve the above objectives, NB-Scope features three key designs.

- **Layered hardware architecture.** NB-Scope layers the hardware system into three parts, including the mainboard, the shield board, and the UE module. The shield board is geared to host the UE module of a specific vendor. It serves as an abstraction layer that hides module heterogeneity from the mainboard. The mainboard integrates general components, such as power supply, sensors, SD card interfaces, and exposes a group of pins to support the plug-and-play of shield boards.
- **Software abstraction.** NB-Scope offers APIs that allow developers to access and control heterogeneous UE modules. The most important function of NB-Scope is that it hosts li-

braries to decode the radio access logs of different UE modules and reports the decoded events in a unified format.

- **Trace fusion of power consumption and radio access.** At run-time, NB-Scope collects and synchronizes power consumption trace and radio access logs. This is critical for correlating the current variation with the network state transition and further discovering the possible strategies to minimize energy waste. To our best knowledge, NB-Scope is the first fine-grain trace fusion toolset for NB-IoT network diagnostics.

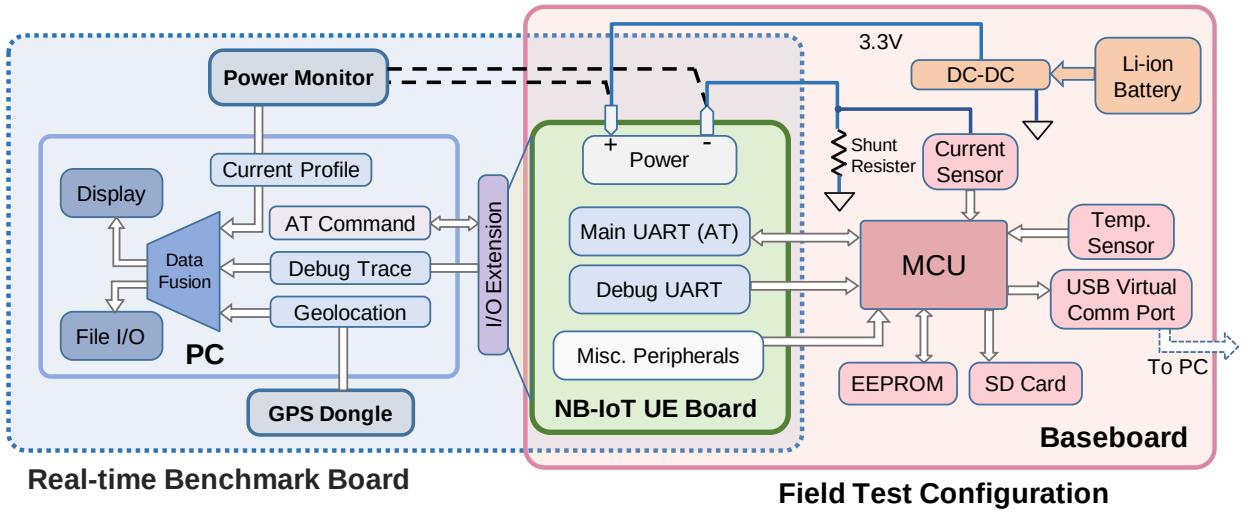


Figure 4.4: System architecture of NB-Scope.

As shown in Figure 4.4, NB-Scope supports two different measurement modes: real-time benchmark mode and field test mode. Specifically, in the field test mode, the system can work in a standalone manner, logging current and debug traces in its onboard storage. In real-time benchmark mode, the NB-IoT shield board can be connected to a computer that retrieves the logs from the I/O extension adapter and analyzes them together with power measurements from a power monitor.

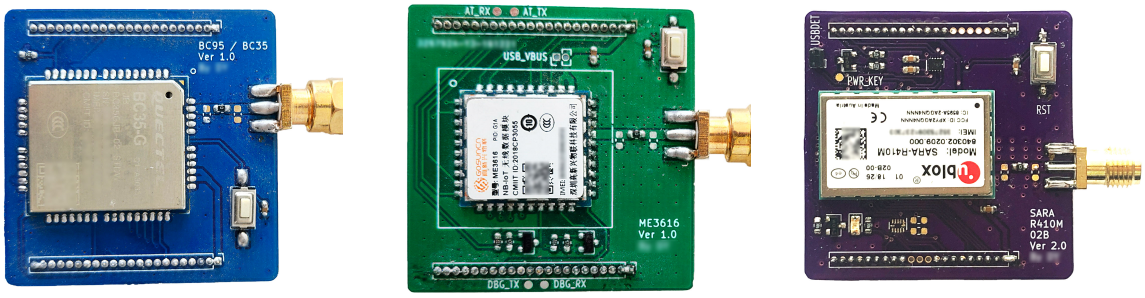
4.3.2 NB-Scope Hardware Design

NB-IoT module shield board. NB-Scope consists of a shield board that abstracts the interfaces of heterogeneous UE modules, providing a unified footprint assignment to the hosts. We select a list of popular commercial NB-IoT modules from Quectel, Gosuncn, and uBlox, and analyze their connectivity, as shown in Table 4.1. Then we identify the necessary pins to be interacted with: an intersection set of pins, including power supply, SIM card interface, main UART, debug UART, and RESET, which are essential to let a module work properly, and special purpose connectivity for different modules, such as PWR_ON key, USB interface, SPI, or I2C. Next, all the selected pins are aggregated into 2 rows of pin headers, such that all the modules share a unified "interface" to external hosts, which also enables straightforward PCB design for module integration. In this manner, we achieve the plug-and-play feature and interchangeability for heterogeneous modules. Moreover, thanks to the modular design, the shield board energy consumption is separated from other node components, providing a fair comparison between different NB-IoT modules. We have developed 7 types of shields, supporting up to 8 different NB-IoT modules, including Quectel BC28/BC35/BC26/BC66/BG96, Gosuncn ME3616, and uBlox SARA-R410M-02B. We show 5 types of the shield boards in Figure 4.5 due to space limit.

Table 4.1: List of modules that NB-Scope supports.

Module model	Manufacturer	Chip	Region [1]
SARA-R410M-02B	uBlox	Qualcomm MDM9206	Global
BC35	Quectel	HiSilicon Hi2110	Europe, Asia-Pacific
BC28	Quectel	HiSilicon Hi2115	Europe, Asia-Pacific
BC26	Quectel	MediaTek MT2625	China
BC66	Quectel	MediaTek MT2625	Global
BG36	Quectel	Qualcomm MDM9206	China
BG96	Quectel	Qualcomm MDM9206	Global
ME3616	Gosuncn	MediaTek MT2625	Global

[1] BC26 and BG36 share the same chipset with BC66 and BG96 respectively. The difference is that Quectel places a restriction on their available network operators so that they can be sold in different regions.



(a) BC35/BC95 shield (b) ME3616 shield (c) SARA-R410M-02B shield

(d) BC28 shield (e) BG96 shield

Figure 4.5: NB-IoT UE module shield boards.

Real-time benchmark mode. In the benchmark mode, NB-Scope allows one to interact with and analyze the energy performance of the NB-IoT modules in real-time. This is achieved by installing the module shield board on an IO extension board. It supports a wide range of measurement capabilities via connecting to a computer. For example, the developer can emit the AT command and observe the UE behavior immediately. In this case, the UART ports, including the main AT and debug log export functionalities, are routed to the USB-Serial adapters. We use a Monsoon HVPM [41] to measure the current consumed by the NB-IoT node. In this manner, the user can easily know when specific events occur and how they introduce the variation in the current.

Field test mode. We design a mainboard to host the UE module board, as shown in Figure 4.6, to enable real-world deployment. It is essentially a lightweight and low-power NB-IoT

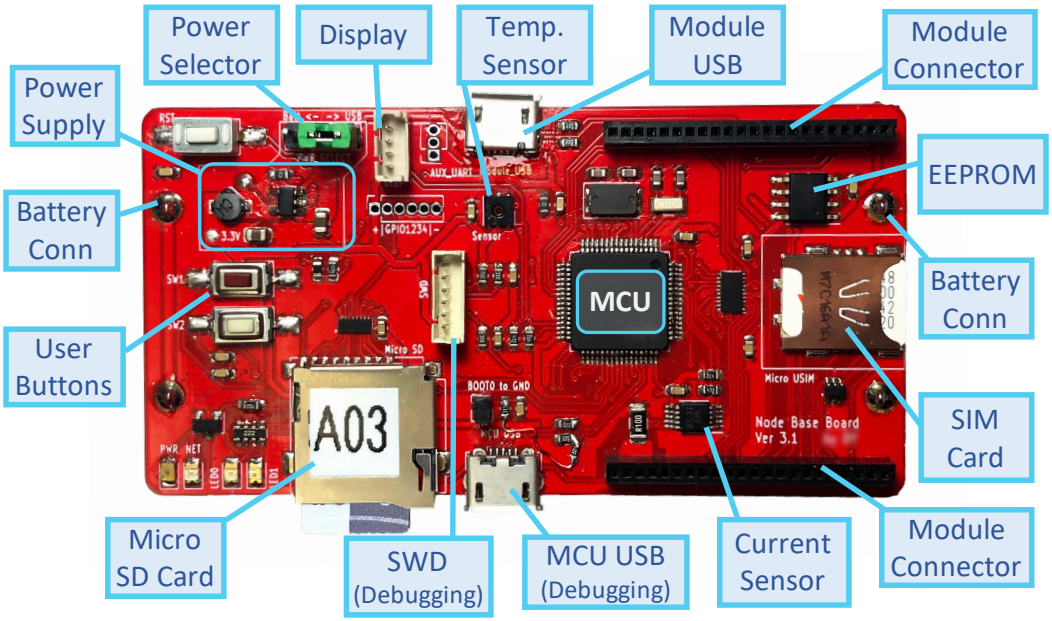


Figure 4.6: STM32-based mainboard for NB-IoT field test.

application node that logs the current and the debug logs into a micro SD card for offline analysis. The mainboard carries an 84 MHz STM32F103 series chip as the processing unit, a temperature and humidity sensor, a current sensor, an EEPROM, and a micro SD card.

Another important feature of the field measurement node is the capability of current sensing with a Texas Instruments INA226 chip, which is capable of up to 7,600 16-bit resolution samples per second, with an error rate of less than 2% at the tens or hundreds of mA magnitude. Moreover, the node is powered by either a single or a pair of 18650 Li-ion batteries, supporting the node measurement for up to 1,000 packets.

4.3.3 NB-Scope Software Design

Figure 4.7 shows the architecture of NB-Scope software stack, which consists of three pipelines for data collection, module control, and data processing, respectively.

Data collection and module control. The data collection component of NB-Scope receives debug logs and current measurement samples from the shield board, and then stores collected data in an SD card. Both current measurements and debug logs are timestamped so

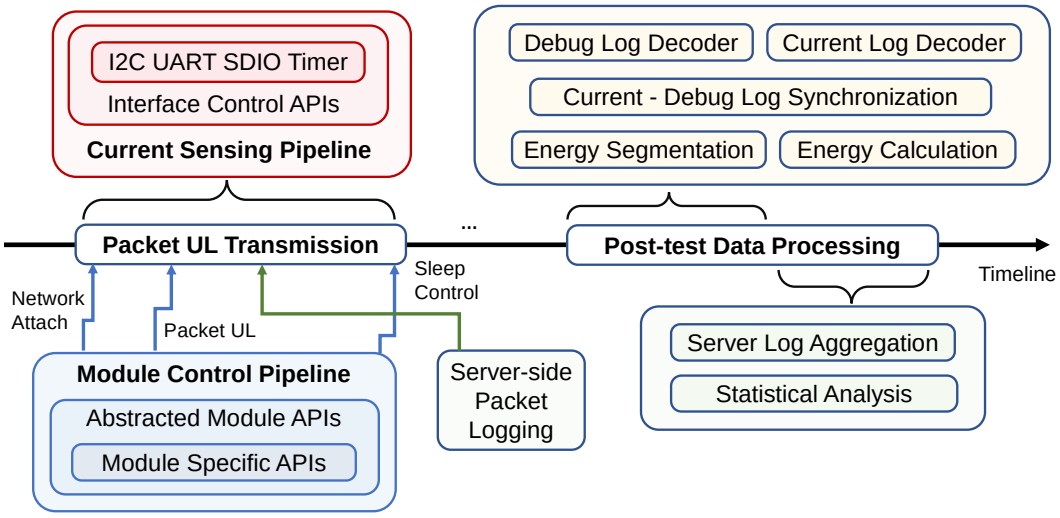


Figure 4.7: NB-Scope software architecture (field test mode). The debug log collection pipeline is similar to the current sensing pipeline, thus is not shown in the figure.

that they can be later aligned in the time domain to enable energy performance diagnosis of radio access.

The module controller of NB-Scope allows users to compose control logic of the NB-IoT module without knowing low-level details of chip commands, which may differ across different NB-IoT modules. It achieves this by abstracting basic functionalities of the control logic as atom functions, which are then exposed as unified APIs to hide module-specific implementations. For example, different modules may define different AT commands to turn the module into airplane mode. We implement low-level APIs for each module. As a result, the user can control all the modules via the high-level API `at_enable_airplane_mode()`, without being concerned with the implementation details. New modules can be easily added to the library by implementing the corresponding atom functions in the library. Using the APIs, we further develop a group of representative NB-IoT applications and services, such as environmental sensing, uplink packet sending, downlink packet parsing, current sensing, SD card R/W, etc.

Data processing pipeline. The data processing pipeline first decodes debug logs using a backend service, as exemplified in Figure 4.8, which handles and hides subtle seman-

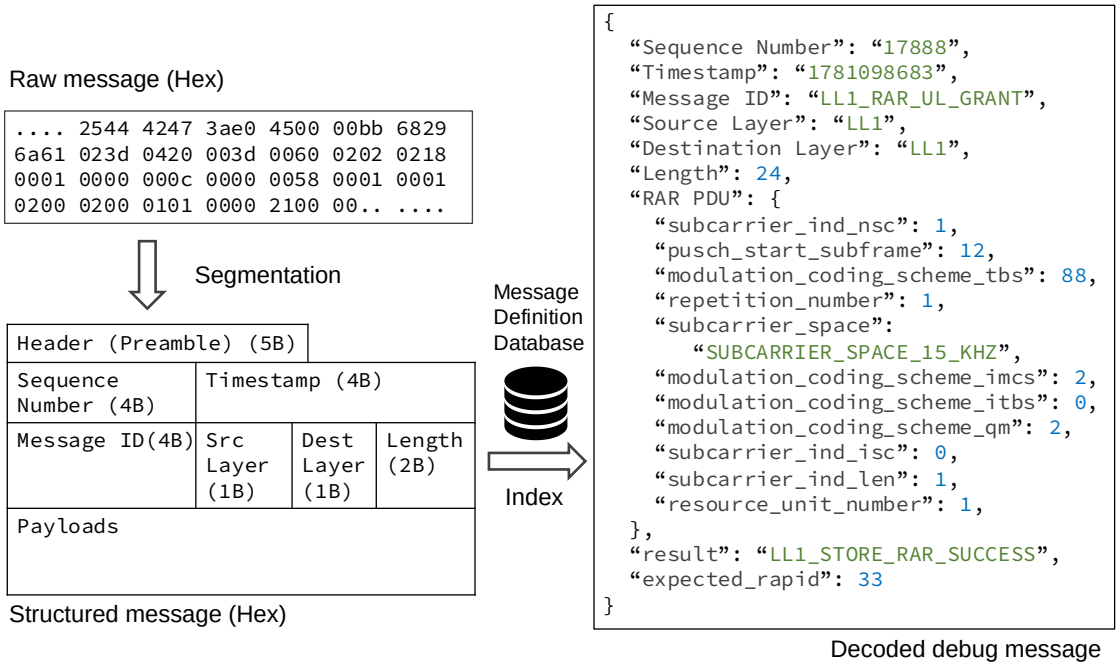


Figure 4.8: Message decoding example. The raw debug log data is first segmented into different fields by the byte length, and then is translated to human-readable texts according to the message definition database.

tic difference among the debug logs of different NB-IoT modules. Specifically, the task of debug log decoding is two-fold. First, the decoder needs to dissect a radio access cycle and determine the timing of each phase (such as contention, control message, data transmission, receiving, and idle). Second, the decoder needs to extract the configurations and performance of the NB-IoT network, such as transmission schedule, RSRP/SNR measurement, ECL selection, and block error rate, which allow for fine-grained diagnosis of their impacts on the energy of UE.

To understand energy consumption in different radio access phases, NB-Scope first splits continuous current measurements using a threshold-based method and then associates each part of the current with a radio access phase based on the timing information decoded from debug logs. A major challenge in this step is that some NB-IoT modules may not produce debug logs for radio access phases. To address this issue, we first utilize the synchronized current and debug traces from the UE with debug log output to train

an algorithm to determine the state transition, triggered by the current pulse width and amplitude in time sequence. In this algorithm, we mainly focus on the Tx states, including `MSG1`, `MSG3`, `ACK` and `Data`. As discussed in Section 4.2, the occurrences of the states during the packet uplink transmission follow a specific order. For example, once we find the `MSG3-ACK`, we can immediately declare that the next state is `Data`. One intuitive simple forward labeling algorithm, from `MSG1` to `Idle`, is to leverage the state transition rules together with the state features, e.g. current magnitude, pulse width. However, some states repeat (i.e. re-transmission) due to bad signal quality or failure in receiving the ACK from the eNodeB. Therefore, we utilize other passive states, such as `Idle` and `Inactvitity` as auxiliary anchors to determine the Tx states. Moreover, we propose a bi-directional state-machine-based labeling algorithm, which divides the uplink process into two parts by the start time of `Data` and leverages the state transition rules and features of both forward and backward directions to increase the labeling accuracy. By segmenting the current according to the states, we can obtain a detailed picture of the power consumption for each state for modules either with or without the debug trace capability. The results of the current labeling algorithm are discussed in Section 4.4.4.

4.4 NB-IoT Measurement Study

In this section, we conduct the measurement in the wild massively with NB-Scope and analyze the energy performance in fine granularity. At a glimpse, our measurement deployment lasts 3 months and consists of 30 NB-Scope nodes on NB-IoT networks from major operators in 3 regions, including two major cities in China and a mid-sized city in the US.

The road map of the measurement study first starts from planning the experiments to address the heterogeneity in location, network operators, and module vendors. Then, we analyze the UE energy consumption w.r.t to these factors and identify a substantially high UE power consumption imbalance. Next, to understand the causes of the high energy

variance, we analyze the ECL ratio in deployment and reveal its selection mechanism. Next, a fine-grained energy decomposition is conducted, by which we identify two heavy network control overheads: long tail due to large `Inactivity Timer` and the excessive RA message repetition. By in-depth study on these two factors, we show the opportunity of optimizing them to save UE power. Finally, the UE battery life is estimated with the measurement study results.

4.4.1 Field Measurement Methodology

NB-IoT measurement is fundamentally different from LTE cellular network measurement, where the latter could be conducted through mobile applications [50,52] installed on massive mobile devices. In contrast, an NB-IoT node must be physically deployed to measure network performance in place. As discussed in Section 4.3.2, we build a series of NB-IoT modules to address the vendor heterogeneity. In addition, we still face two challenges in conducting a large-scale field measurement of NB-IoT networks. First, NB-IoT enables a wide range of applications with high spatial diversity, from indoor to outdoor, from the rural area to the metropolis downtown area. Our results may include significant bias if the deployment of measurement nodes could not cover the wide spectrum of these applications. Second, there usually exist multiple network operators in the same area, offering different rates, services, and IoT backend platforms. Each network operator has its network configurations, which may affect the measurement results in terms of power consumption, network coverage, etc.

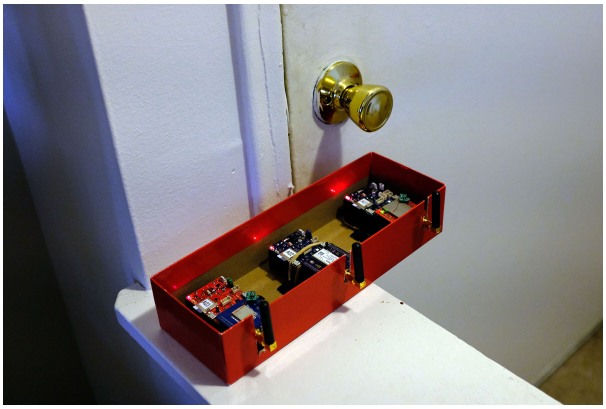
To mitigate the impact of biased location selection on our measurement results, we choose the measurement location by types of applications. Specifically, five common NB-IoT applications[2] are selected in this work: outdoor parking, indoor parking (ramp), smart door lock, smoke detection, and smart water/electricity meter. These applications represent a class of typical locations, ranging from outdoor open areas to residential/office

[2]We mainly focus on stationary applications in this work. Mobile applications, such as cold chain tracking [89] or shared bike service [74], are left for open problems.

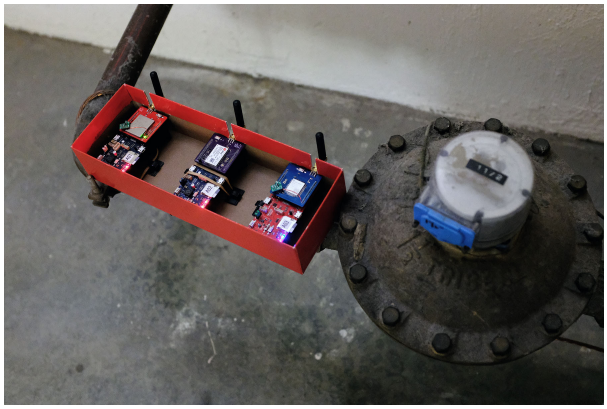
indoor environments and basement. We refer them to as "location profiles" in the rest of this chapter. Figure 4.9 exemplifies the location profiles of the measurement node deployment.

(a) Outdoor parking

(b) Indoor parking



(c) Smart lock



(d) Water meter

Figure 4.9: Actual node deployment of different location profiles.

Second, we conduct our field measurement in three regions: two major cities in China and a mid-sized city in the US. In each of the three cities in China, we measure the networks of two different major operators referred to as "CN-OP1" and "CN-OP2", while the network of only one operator, "US-OP1", is measured in the US. Because different operators may differ in the coverage, we choose multiple ($\geq$ 10) locations to deploy our nodes for each network operator, such that the variance of the UE performance due to environmental and operator-specific factors could be captured in the results. To evaluate

the performance of UEs from different vendors, we deploy at least three types of UEs for each operator at each spot.

In summary, our deployment covers 5 location profiles, each consisting of more than 10 measurement spots. At least 6 nodes are deployed at each measurement spot in China, corresponding to the Cartesian product of the 2 network operators and the 3 modules (BC26, BC28, and ME3616). We choose BC66, BG96, and SARA-R410M-02B NB-IoT modules in the US, running on the US-OP1 network.

Finally, we discuss the setting of UL communication in our deployment. Since the power drain for the `PSM` mode is highly predictable and negligible, we focus on measuring the behavior of UE when it is awake. In each round of UL transmission, the UE wakes up from PSM, searches and attaches to an NB-IoT network, carries out a UL packet transmission following the radio access procedure described in Section 4.2, and then goes back to PSM for 10 seconds. At each measurement location, the above procedure is repeated for 25 to 30 UL packets, usually lasting about 30 minutes. For NB-Scope nodes deployed around the same measurement spots, considering the short UL transmission time (1-3 seconds), the interference due to the node concurrent UL transmission is negligible. Even though two or more nodes may initiate random access simultaneously, the contention can be resolved by eNodeB (see Section 4.2) by scheduling different resource units for those nodes, which avoids possible mutual interference for the subsequent `MSG3` and data transmission. Due to the small number of UL packets and the low duty cycle of NB-Scope nodes, the impact of our measurement campaign on the NB-IoT network is negligible.

4.4.2 Measurement Result Analysis

Following the methodology described in Section 4.4.1, we built 30 NB-Scope nodes of 6 different types of modules[3], and then conduct a large-scale field measurement by moving them around over 1,200 locations during a three-month measurement campaign, collect-

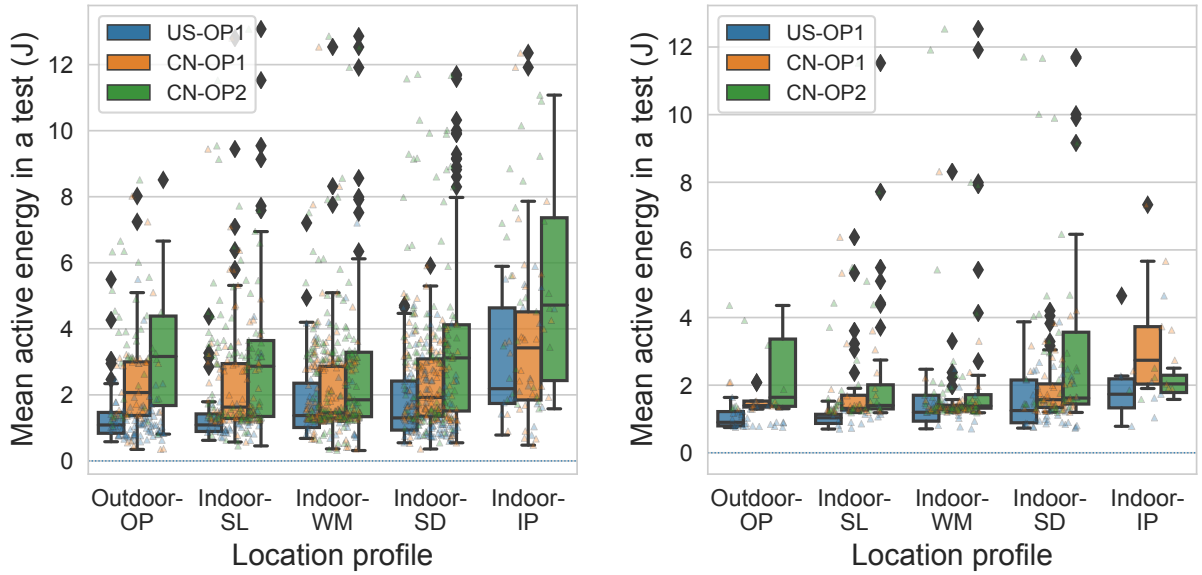
[3]There are 30 mainboards and 30 shield boards that are assembled. An NB-IoT node consists of a shield board and a mainboard plugged together.

ing more than 49.0 GB data including current traces and debug logs for more than 36,000 UL packets, where each current trace and debug log is recorded for a complete radio access cycle, from issuing a packet transfer AT command to entering `PSM`. We now present the statistical results from the measurement study. In particular, our results reveal that the UE battery life can be highly imbalanced in a real-world deployment. The causes of such high energy imbalance are analyzed in detail through Section 4.4.3 to Section 4.4.6.

4.4.2.1 Power consumption w.r.t location profiles

We first compare the UE power performance under different location profiles for different network operators. As shown in Figure 4.10a, as the deployment location changes from open areas to semi-indoor then to the basements, the mean active energy cost per UL packet transmission increases significantly. Specifically, with CN-OP2, **the indoor parking nodes consume about 2.7x more energy than the water metering nodes to transmit one packet**.

In addition to energy variance *across* different applications, we also notice high energy imbalance *within* each location profile. In particular, **for most indoor applications, the 75th percentile of node energy consumption can be up to 3 times higher than the 25th percentile**. In water meter and smoke detection profiles, **the ratio of the highest and lowest energy consumption of different nodes can be up to 75:1**. This may result in network partitions, because a portion of the nodes consumes significantly more energy than their peers, leading to highly imbalanced battery life. Moreover, we observe that the specific location of an indoor UE (e.g., the distance to the building walls) does not show a significant impact on energy consumption, as exemplified in Figure 4.11. For instance, some UEs in the basements yield low energy consumption. This is because, as indicated by our measurement, the operators usually deploy indoor micro-cells in the buildings, providing local NB-IoT access.



(a) All modules

(b) BC26 and BC66

Figure 4.10: Mean active energy per packet distribution by location profiles and network operators. The upper and lower error bar are at most 1.5x interquartile range away from the 75th and 25th percentile respectively. OP: outdoor parking, SL: smart lock, WM: water meter, SD: smoke detection, IP: indoor parking.

4.4.2.2 Power consumption w.r.t network operators

From Figure 4.10a, we observe that, for the same location profile, the network operators have a non-negligible impact on the packet energy. For example, nodes of US-OP1 yield the lowest energy consumption and variance in most location profiles. For example, the packet energy in smart lock location profiles is only 1 J on average. For a fair comparison between network operators in the two countries, we further study the energy performance of a pair of NB-IoT modules, namely Quectel BC26 and BC66, which are deployed in China and the US, respectively. We note that both BC26 and BC66 adopt the MediaTek MT2625 chipset, thus allowing us to exclude any measurement bias introduced by the heterogeneity of modules. In Figure 4.10b, we observe that, in most applications, BC66 deployed in the US has relative lower energy consumption and variance than BC26 deployed in China, which is consistent with the results shown in Figure 4.10a. We also observe less deviation

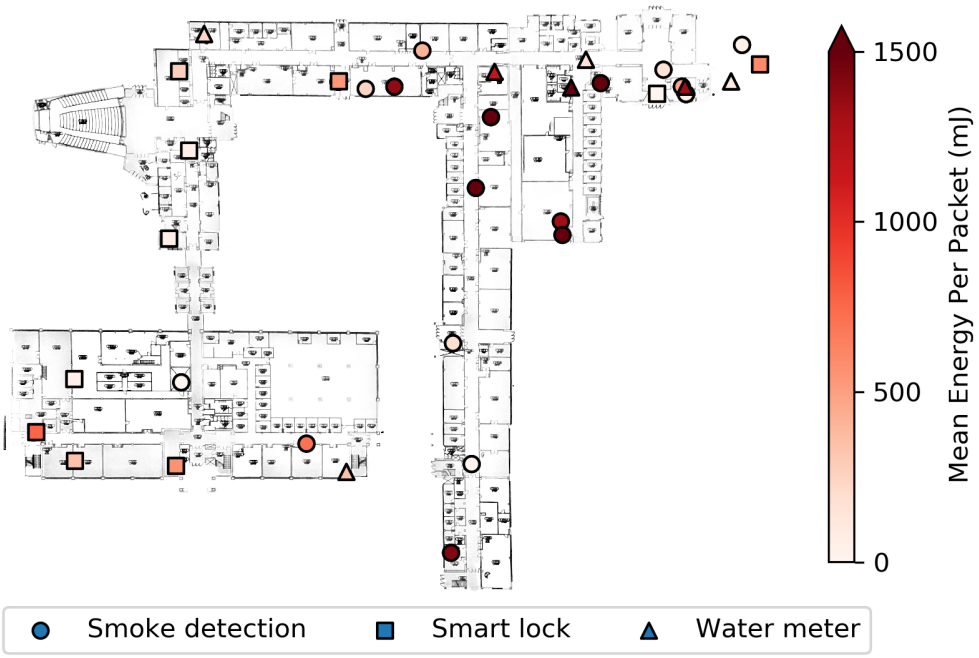


Figure 4.11: Average power consumption per packet transmission for indoor applications in a building. Note that some co-located points may be on different floors.

Table 4.2: eNodeB Configurations of different Network Operators. MSG1 repetition and ECL threshold are not available in the debug logs of NB-IoT modules deployed in the US.

Operator	US-OP1	CN-OP1	CN-OP2
MSG1 Repetitions in ECL0/1/2	N/A	2/8/32	2/8/32
MSG3 Repetitions in ECL0/1/2	1/2/8	1/2/32	1/2/32
ECL Threshold	N/A	-107, -117	-109, -119
`Inactivity Timer`	3s	20s	20s
Band	13	5	8

in the packet energy distribution in US-OP1, while the BC66 (in CN-OPs) spends significantly higher energy than average.

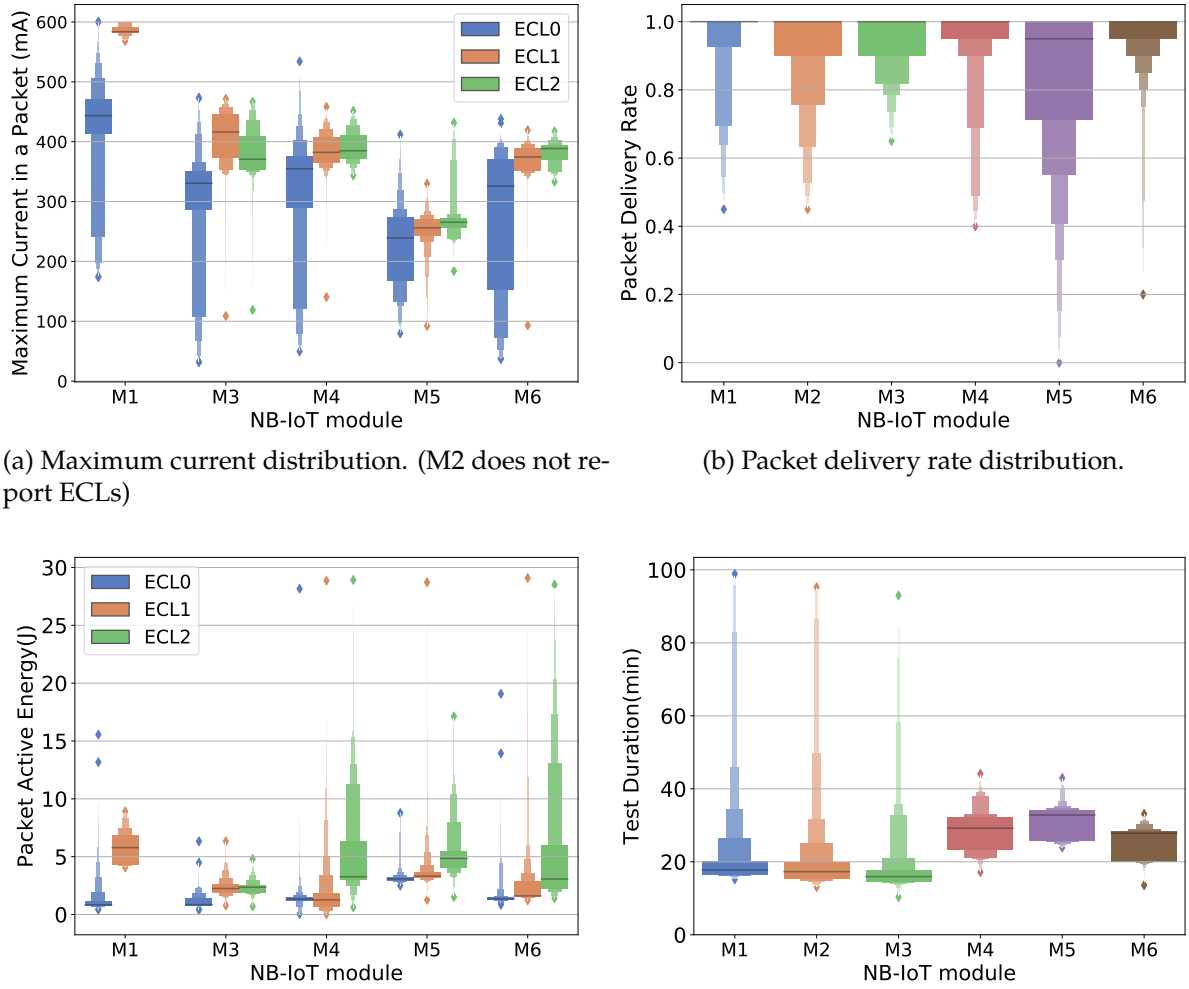
To further understand why node energy consumption varies significantly across different operators, we list the eNodeB configurations of different operators in Table 4.2. We notice that CN-OP1 and CN-OP2 employ more conservative configurations featured by prolonged `Inactivity Timer` and excessive `MSG3` repetitions, which improve the UL link

reliability but result in relative higher node energy consumption. We will show in Section 4.4.5 and 4.4.6 that `Inactivity Timer` and `MSG3` repetition have critical impacts on UE energy consumption, leaving a significant space for optimization. Although US-OP1 outperforms CN-OP1 and CN-OP2 in terms of energy performance, our efforts in identifying the energy loopholes are still important, because on one hand, not all network operators adopt the same configuration as US-OP1; on the other hand, an energy-efficient configuration may come at the cost of lower link connectivity under bad signal quality. So each network operator should set the configuration with a desirable trade-off between the network coverage and the node lifespan.

4.4.2.3 Comparison of NB-IoT modules

The NB-IoT modules employed in the field test demonstrate noticeable differences. We plot the performance of the NB-IoT module in the smoke sensing location profiles in Figure 4.12. We number these modules without showing their models to conceal the identities of vendors.

First, from Figure 4.12a, we can see that the maximum current in a UL packet transmission cycle varies significantly. For example, module M1 has a higher maximum current than all other modules, while M5 tends to place a small upper limit on the maximum current. We note that NB-Scope measures the power consumption of the hosted NB-IoT modules instead of the entire board. This allows us to exclude the impact of board design features. Therefore, the variations of power consumption between different modules reported in Figure 4.12a should be largely attributed to the different designs of NB-IoT modules. We note that **the different maximum currents of modules can cause a significant energy variance.** In a typical UL packet cycle, the total transmission period usually lasts 2.8 seconds. Suppose the maximum current difference between two modules is 100 mA, the energy difference during the transmission period can be 0.92 J (assuming 3.3 V voltage), which yields significant energy difference in the long term. We note that a lower



(a) Maximum current distribution. (M2 does not report ECLs)

(b) Packet delivery rate distribution.

(c) Packet active energy distribution. (M2 does not report ECLs)

(d) Time duration to upload 20 packets.

Figure 4.12: Performance of different models in smoke sensing location profile. M1-M3 are deployed in the US, while M4-M6 in China.

maximum current can lead to degradation in the packet delivery rate, as shown in Figure 4.12b. In such a case, the energy waste from the loss packets should not be overlooked. Second, we confirm again in Figure 4.12c that the nodes in ECL2 have significantly higher energy than in ECL0 and ECL1, for the majority of the NB-IoT modules, which means that the energy imbalance is primarily introduced by the ECL instead of the module. Finally, we can see from Figure 4.12d that almost all the modules can finish transmitting 20 packets (assume the sleep timer is 10 seconds for all modules) in about 20-30 minutes, meaning

that generally, the NB-IoT has a similar level of latency.

Moreover, we notice that some modules have difficulty in cell searching and network attaching. For example, one of the modules fails in attaching the network in 20 out of the 125 spots, while the other modules under the same network configuration succeeded.

The modules can vary in many other aspects, such as SNR, block error rate, ECL selection strategy, cell selection approach, and so on. While we cannot exhaust all these variances, our results above suggested that the diversity in the performance of different modules has an important impact on the development and deployment of large-scale NB-IoT systems.

4.4.2.4 Temporal variation of power consumption

We study the temporal variation of node energy consumption by deploying 3 different modules in 3 location profiles and then measure their energy per UL packet transmission in a period of 12.5 hours.

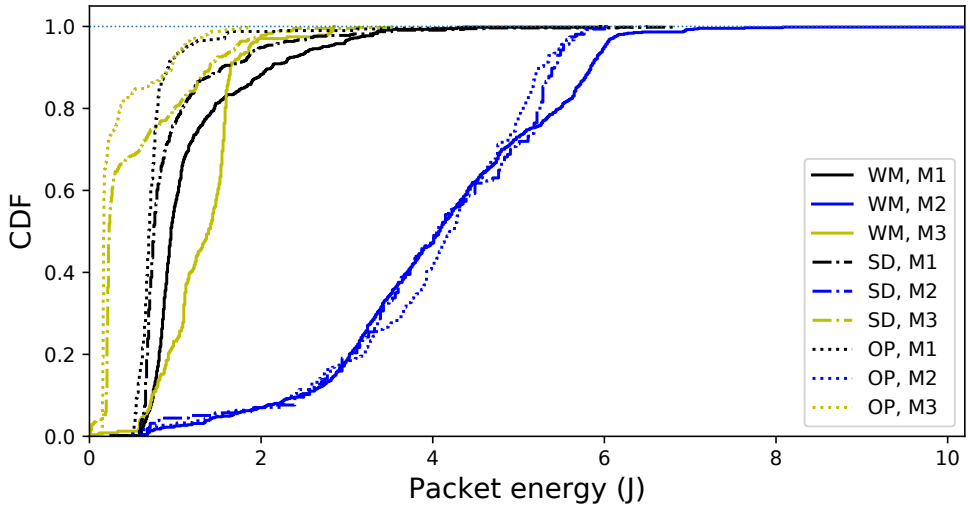


Figure 4.13: Distribution of the packet energy in a 12.5-hour period.

Figure 4.13 plots the distribution of energy consumption per UL packet. We observe that all nodes experience large energy variance over time, especially for nodes in the loca-

tion profile of the water meter, where the maximum per packet energy consumption can be 45 times higher than the minimum value. We found that the major reason is that the RSRP in the indoor location profiles has a significant variance. Even the nodes are stationary during measurement, the wireless quality is highly dynamic. Once the node chooses ECL2 to perform radio access, the energy can increase significantly. We will further analyze this in the later sections.

4.4.2.5 Power consumption v.s. distance to eNodeB

To study the impact of distance to eNodeB on UE energy consumption, we conduct a measurement in an outdoor area, where a UE is deployed at the line-of-sight (LoS) path of the eNodeB, and then moved away from the eNodeB at a step of 100 meters until the connection is lost. In each step, the UE is configured to transmit 15 packets.

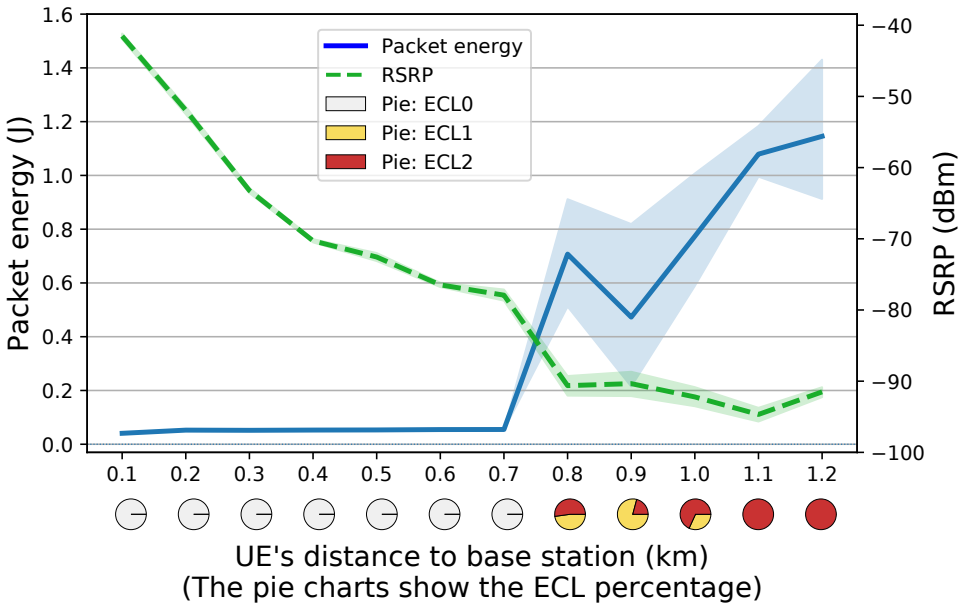


Figure 4.14: Packet energy w.r.t the distance between the UE and the eNodeB.

Figure 4.14 plots the energy per packet (`Inactivity` period excluded) and the RSRP of UE as a function of distance to eNodeB. As shown in the figure, the energy per packet demonstrates a rapid surge after the UE is moved to 0.8 km away. In particular, the average

per-packet energy at 1.2 km is about 28x higher than that at 0.1 km. We also observe that per-packet energy has a strong correlation with the ECL of UE. Specifically, during the measurement, we observe that the ECL of the UE remains in ECL0 in the range of 0.1-0.7 km consistently, and then drops to ECL1 and ECL2 after 0.8 km, which explains the drastic surge of per-packet energy between 0.7-0.8 km. We will study the impact of ECL on UE energy consumption in more depth in Section 4.4.3.

4.4.2.6 Measurement summary

First, by decomposing the energy consumption by location profiles, we see that different applications may have significantly different energy consumption. In some applications, such as indoor parking, there exists a significant energy imbalance among the nodes. Second, we notice that the network operators play an important role in energy consumption because not only the network configurations may cause different UE behaviors, but also the cell deployment can be one of the decisive factors for node signal strength. Third, different modules in the same spot can consume energy at different rates due to the hardware design and chipset implementation. Finally, the distance to the base station can also impact energy consumption. Particularly, we notice a sharp transition of energy consumption with respect to the node-cell distance, which is caused by the ECL mechanism in NB-IoT.

We will further discuss the impact of ECL and present the detailed energy breakdown to diagnose the causes for the UE energy behaviors in the rest of this section.

4.4.3 The Impact of ECL

The results in Section 4.4.2 show that heterogeneity of modules, location profiles, and network operators can affect the node energy consumption to different extents. As a result, the nodes have a significant energy imbalance, where the nodes under the same type of location profile drain energy at a distinct rate. From the energy v.s. distance experiment, we see that ECL can be one of the major factors that cause such an imbalance. Since the

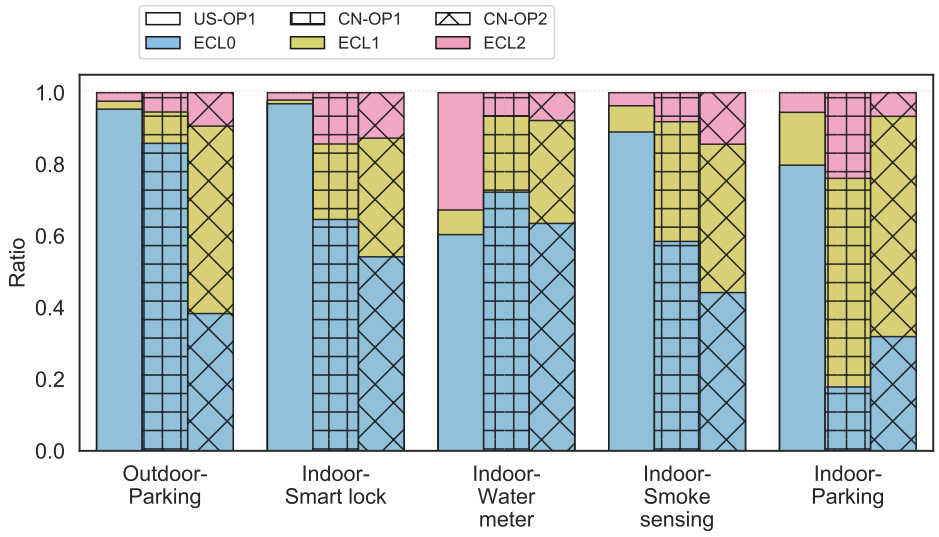


Figure 4.15: ECL ratio w.r.t location profiles and network operators.

NB-IoT network relies on the ECL to determine the Tx power, the UE drains different levels of energy under different ECLs. Therefore, we first diagnose the high energy consumption and variance of UEs shown in Section 4.4.2 and the role of ECL. To this end, we decompose the ECL distribution and correlate it with the energy profiles. The ECL histogram is plotted for each location profile in Figure 4.15. We observe that the UE selects ECL0 and ECL1 in a majority of the outdoor locations. However, for indoor parking, building corridors, and water meter deployment spots, ECL1 and ECL2 begin to dominate. This is expected because the eNodeB base station signal will be attenuated by the concrete walls before being received by the UE. A notable outlier is the indoor smart lock location profile with US-OP1. The reason is that, as indicated by our measurement, US-OP1 installs micro-cells in the building corridors, leading to a better signal strength around the doorway. For example, the building in Figure 4.11 is covered by up to 11 eNodeBs, while only 3 of which are the outdoor macro base stations. Therefore, the smart lock nodes usually receive a strong RSRP, leading to the dominant ECL0 selection. Overall, we note that although the ratio a node runs under ECL2 is relatively smaller than that running in ECL0 and ECL1 in all location profiles and operators, the energy consumption in ECL2 is still significant be-

cause UE in ECL2 suffers up to 4 times higher transmission power consumption than that in ECL0 (as shown in Figure 4.12). The problem is particularly acute in indoor scenarios, where the ratio of ECL2 is much higher than that of outdoor.

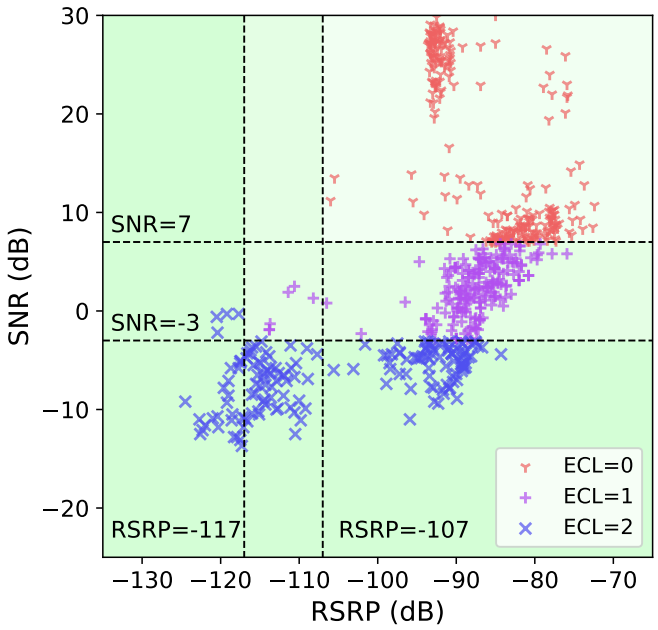
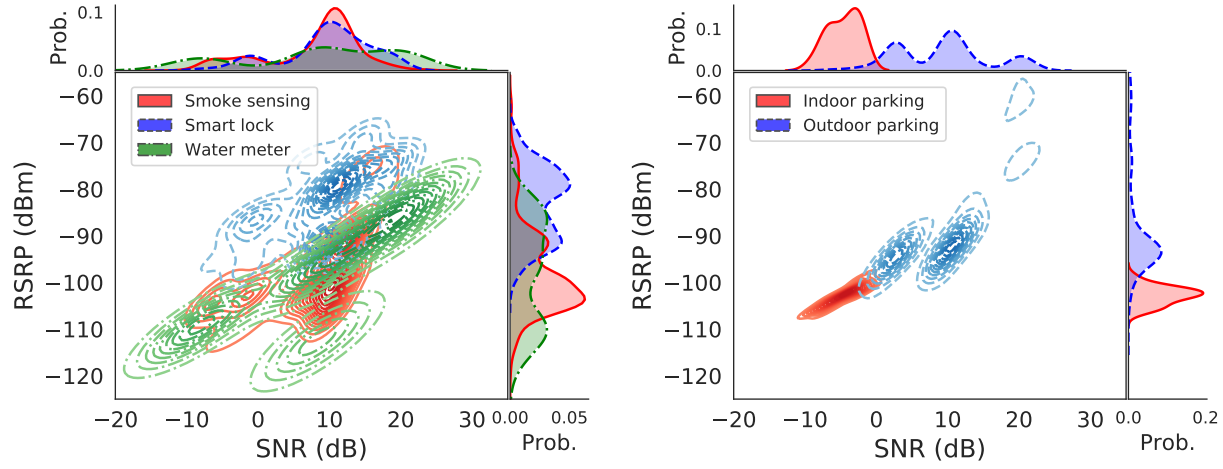


Figure 4.16: ECL selection v.s. RSRP and SNR measurement.



(a) Three location profiles: (1) smoke sensing; (2) smart lock; (3) water meter.

(b) Two location profiles: (1) indoor parking; (2) outdoor parking.

Figure 4.17: RSRP and SNR distribution w.r.t five types of location profiles.

Understanding the ECL selection mechanism is important. As mentioned in Section 4.2, the UE decides its Tx power adaptively by itself based on the channel RSRP. Since UE consumes much more power in ECL2 than ECL0 and ECL1, we are interested in the factors that affect the selection of ECL. According to the 3GPP specification, the UE decides its ECL by comparing the latest RSRP with the two thresholds given by the eNodeB. However, the actual ECL determination algorithm is not specified and the implementation is up to the NB-IoT modem manufacturers. For example, the debug traces indicate that the UE also relies on the SNR to determine the ECL, as shown in Figure 4.16. As we can see, **it adopts intuitive hard SNR and RSRP thresholds to decide the ECLs**, forming one rectangle and two L-shape regions on the SNR-RSRP 2-D space. Such a naive ECL determination algorithm is likely to introduce energy waste due to several reasons. First, the selected thresholds may not be optimal, such that a higher ECL is utilized to transmit the packet that could be potentially uploaded with less random access resources. Second, such a fixed mapping between the (RSRP, SNR) pair and ECLs cannot adapt to complex dynamic wireless environments due to the heterogeneity of signal propagation. Our measurements show that the (RSRP, SNR) pairs in outdoor locations, such as outdoor parking, concentrate around the top right corner of Figure 4.16, while the semi-indoor locations, such as smart lock, smoke detection, and smoke detection, have large RSRP and SNR variance, spanning across the figure from top right to bottom left, which explains their larger ECL1 and ECL2 percentages in Figure 4.15. Moreover, in other indoor locations, such as indoor parking, (RSRP, SNR) pairs focus on the bottom left area with lower variance than the semi-indoor locations, which leads to a higher chance of selecting ECL2. We note that higher variance of RSRP and SNR usually causes higher energy imbalance, but not necessarily higher maximum energy consumption.

Fortunately, the UE has another strategy in choosing the ECL called "ECL by next level", where the UE adopts a higher ECL *after* the random access failure using the lower one. This is a desirable feature at the first glimpse because it opts for a parameter set that

has a higher probability of succeeding the random access, such as increasing the Tx power or message repetitions. However, if the higher ECL is proven to be a better choice, then the power consumption spent on the previous attempts is wasted. Our measurement results show that there is about **5% probability on average that the UE chooses the ECL by RSRP measurement but fails in RA**. Therefore, the existing ECL selection strategy in the current UE can be potentially improved to avoid energy waste due to choosing an inappropriate ECL.

To further understand the ECL variance across location profiles, we plot the Gaussian kernel density estimation (KDE) contours of their SNR and RSRP in Figure 4.17. As shown in the figure, the locations of smoke sensing, smart lock, and water meter have large variation in RSRP and SNR, while the indoor and outdoor parking have more concentrated distributions. This is consistent with the results shown in Figure 4.15. We note that a large variation of signal quality metrics would result in severe energy imbalance among the nodes, resulting in potentially higher maintenance costs.

4.4.4 Energy Consumption Breakdown

One of the key features of the NB-IoT network is the promise of long battery life, up to 10 years. NB-IoT UE adopts several techniques to minimize the energy consumption, including coverage-level-based power control, `eDRX`, and `PSM`. Our results in Section 4.4.2 showed that the UEs suffer from high energy drain in signal-limited environments, such as indoor and underground facilities. Understanding how the energy is spent and identifying possible energy loopholes in the NB-IoT network is thus of great importance. In this section, we present our observation in breaking down the NB-IoT UE power consumption under different signal conditions.

We first present how the typical current varies with the time under different ECLs in Figure 4.18. The right column shows a complete packet UL Tx cycle. As we can see, there are four distinct current patterns in the current profile: (1) The period with the highest

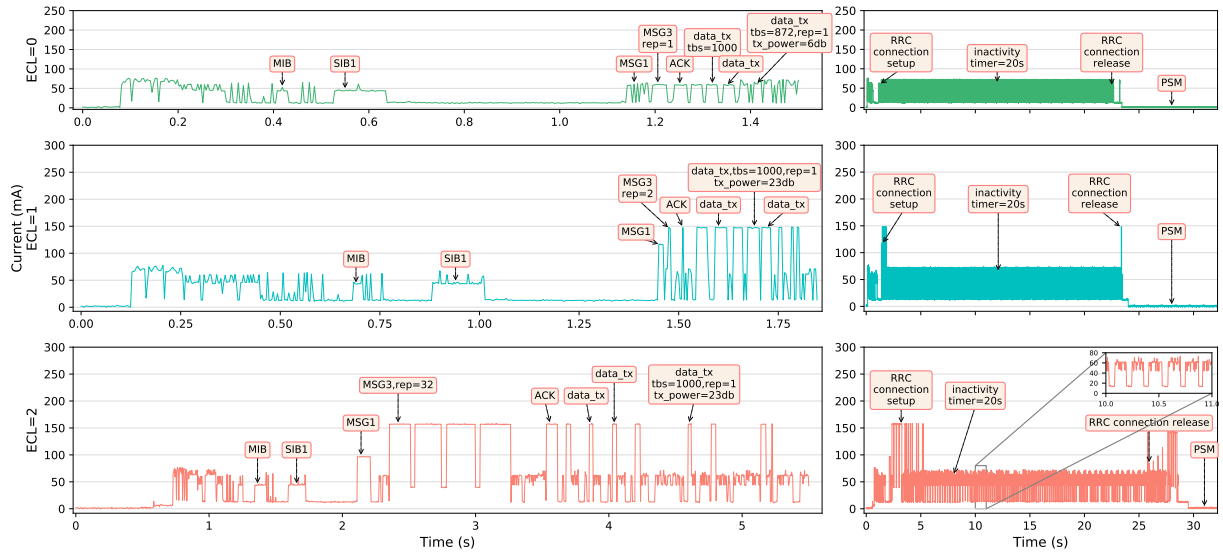


Figure 4.18: The typical power consumption profiles under different ECLs. The left column shows the UL packet Tx current profile for each ECL. The right column shows UE's power profile in a complete packet Tx cycle.

magnitude and pulse width, when both the Tx and Rx are turned on. (2) around 60 mA, distributed across the majority of the time, when only the Rx is turned on to receive DL messages. (3) around 20 mA, which is the base module power without either Tx or Rx. (4) less than 1 mA, where the module enters PSM mode or eDRX mode to preserve energy. The UE current switches between these current patterns frequently during the span of packet transmission, due to the numerous message exchanges and state transitions. However, it is hard, if not impossible, to determine what messages are delivered to the UE and what causes the variation of the current by relying only on the current profile. Therefore, we align the current and the debug logs in time, using the NB-Scope real-time benchmarking function, as described in Section 4.3.

As we can see in Figure 4.18, a complete packet cycle consists of the same procedures among the three ECLs. `MSG1`, `MSG3`, `ACK`, packet data, and `RRC` Connection Release Request are transmitted one by one, and the UE listens to the DL channels between these messages. The left column shows the zoom-in packet UL Tx current profile, which is the distinct part between different ECLs.

To begin with, the UE turns on its receiver to acquire the system information blocks and synchronize with the network, preparing for the random access. Then it follows the procedures discussed in Section 4.2 to establish the RRC connection for uploading the packet data. To transmit the packet with the same size, the UE in ECL0 consumes a lower current and less time, while in ECL1 the UE transmits the data with a slightly longer time and higher current. However, the ECL2 requires the UE to transmit the data at its maximum power capacity, with an extended time for repetition, which introduces significantly high power dissipation. It turns out that the transmission of MSG3 varies significantly among these three ECLs. We acknowledge from the debug log analysis, that the `MSG3` pulse width t_{MSG3} in `ms` is calculated by

$$t_{MSG3} = N_{REP} \times (N_{RU} \times 8)$$

where N_{RU} is the allocated number of resource units; N_{REP} is the number of scheduled repetitions of the resource unit. The network diagnostic traces for CN-OP1 and CN-OP2 show that, for ECL0, 1, 2, the N_{RU} are 1, 3, 4 respectively, while the N_{REP} are 1, 2, 32 respectively, meaning that `MSG3` current in ECL2 has significant higher pulse width than the other two coverage levels. Overall, there is a 4 dB coverage gain (1 dB when the repetition doubles) by increasing the repetition from 2 to 32, while the power consumption is 16x higher. The lack of a transition zone renders the power consumption between ECL1 and ECL2 deviate dramatically. Moreover, the node in ECL2 transmits at its maximum Tx capacity without power control. The wide pulse and the high current magnitude are the two reasons for the high energy consumption under ECL2 random access.

One can also notice from the figure that there is a long tail in the current profile, introduced by the passive DL channel listening. We refer to this period as an "Inactivity" state because its length is specified by a timer called `Inactivity Timer`, configured by the eNodeB. The RRC connection between the UE and the eNodeB is still active during this period, but the UE does not perform any uplink transmission. Our analysis of the debug traces indicates that the UE turns on the reception module regularly to listen to the down-

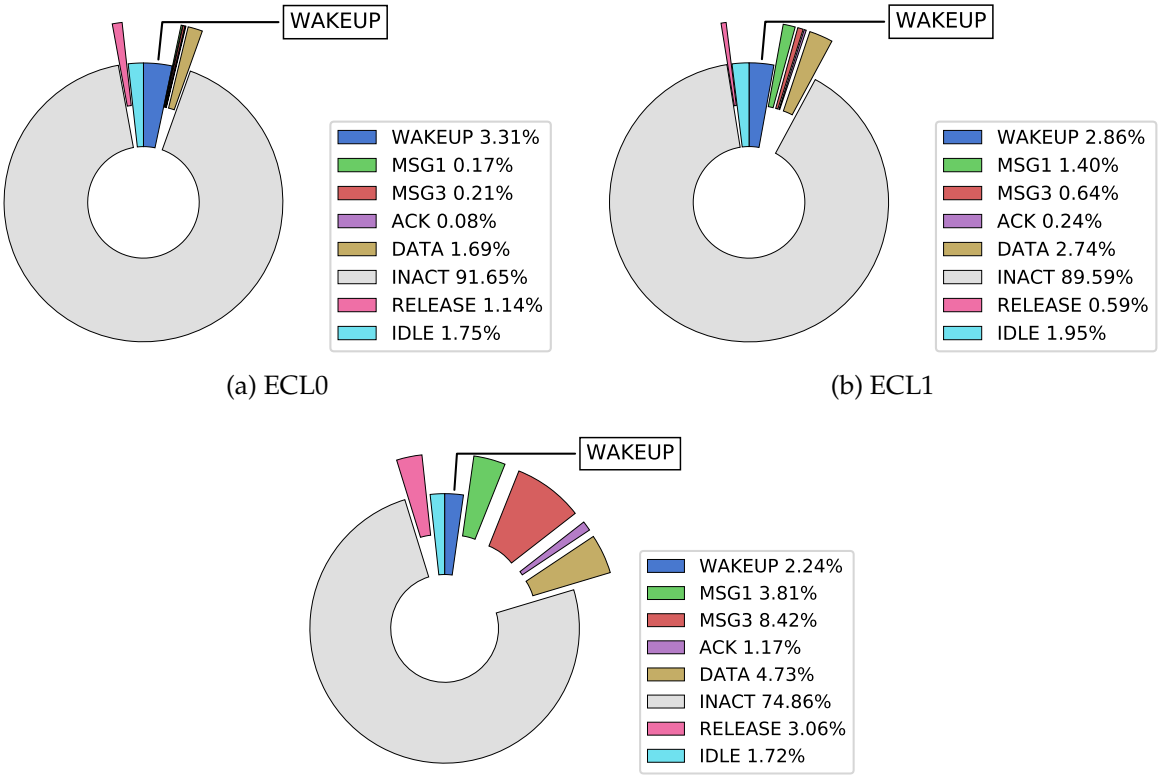
Table 4.3: Power consumption breakdown (mean and standard deviation) by radio access procedures in the UL cycle under different ECLs. Unit: mJ.

ECL	Wake-up	MSG1	MSG3	ACK	Data	Inactivity	Release	Idle	Total
ECL0	109.43	5.61	7.05	2.57	55.68	**3025.97**	37.57	57.79	3301.68
	(5.38)	(3.02)	(5.14)	(0.59)	(16.46)	(180.46)	(5.68)	(20.67)	(164.44)
ECL1	107.28	52.68	23.94	8.90	102.84	**3364.03**	22.10	73.19	3754.98
	(11.32)	(65.30)	(9.70)	(3.33)	(31.60)	(250.96)	(7.76)	(20.49)	(238.16)
ECL2	108.05	184.16	**407.05**	56.33	228.55	**3616.66**	147.62	83.06	**4831.47**
	(19.91)	(72.74)	(52.37)	(11.26)	(29.29)	(370.27)	(192.78)	(20.60)	(409.96)

link channels for the broadcasted network configuration, or wait for possible downlink packets. It may also perform cell searches. However, we could not find any evidence in the debug log of how the UE utilizes the information collected during this period. **This is surprising because the UE may waste significant energy during the long tail.** We also note that the `Inactivity` state is present in all ECLs.

After the long tail `Inactivity` period, the UE starts another random access procedure, including sending `MSG1` and `MSG3`, to inform the eNodeB that the UE is ready to release the RRC connection and enter `Idle` mode. These are the final two Tx events before a UE finishes a UL packet cycle. Since it follows the same process as the initial random access, the UE may choose the ECL2 resources to access, observed in many current logs, leading to extra power consumption.

To quantitatively characterize the partial energy contribution in the UL packet transmission cycle, we calculate the energy according to the time segmentation derived from the debug logs, as shown in Figure 4.19 (detailed in Table 4.3). We see that the `Inactivity` period can occupy up to 90% of the packet energy in ECL0, while the energy consumption of `MSG3` Tx in ECL2 is significantly high, compared to that of ECL0 and ECL1. The total energy of an ECL2 packet can be about 1.46x and 1.29x more than the ECL0 and ECL1 respectively. However, if we ignore the largest part, namely the `Inactivity` period, **the UE in ECL2 consumes 4.41x and 3.11x more energy than ECL0 and ECL1 respectively**, which would cause severe imbalance battery life among the NB-IoT nodes in the long term.



(a) ECL0

(b) ECL1

(c) ECL2

Figure 4.19: Averaged energy consumption breakdown by radio access procedures and ECLs. The wedges are arranged clockwise according to the legend. The exploded wedges require UL transmission.

4.4.5 Impact of the Inactivity Period

From the energy breakdown above, mitigating the `Inactivity` energy waste is important to prolong the UE battery life. `Inactivity Timer` is configured in eNodeB to indicate that the UE does not have any downlink and uplink traffic within this period. The objective is to reduce the UE energy consumption by preventing it from staying in the RRC connected status for too long. From that perspective, we can conclude that the current configuration in the eNodeB is suboptimal. For example, the two network operators in China set the timer value as 20 seconds, while the US-OP1 chooses 3 seconds, which is one of the key factors that contribute to the significant differences in packet energy in these two regions. However, the negative effect of the long tail can be mitigated by a feature called release

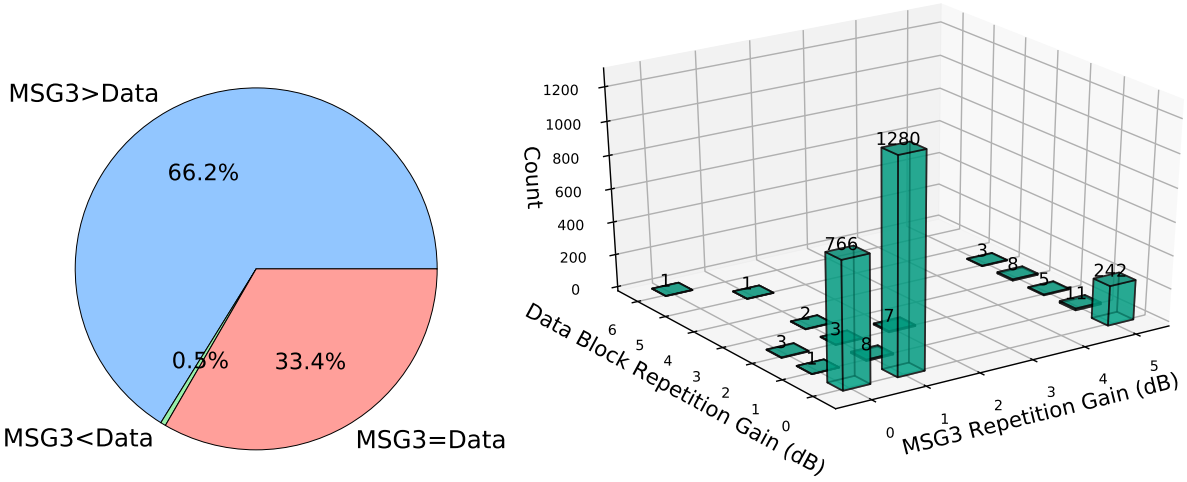
assistance indication (RAI) introduced in 3GPP Rel. 14 [68]. It is a short message sent by the UE, informing the eNodeB that there is no more UL data and it does not anticipate receiving further DL data, such that the RRC connection can be released earlier before the `Inactivity Timer` expires. Accordingly, the energy waste during `Inactivity` period can be avoided because the NB-IoT application developers can either skip the whole period immediately if there is no DL packet to receive or skip it right after the anticipated DL packets are delivered. However, optimizing the `Inactivity Timer` is still of great significance for two reasons. First, some application developers may not be aware of such a feature. Second, the NB-IoT network has been widely deployed since 2018, when the 3GPP Rel. 14 has not yet been implemented in the commercial NB-IoT modules. One-third of the modules used in our field tests do not include RAI control in their documentation. It is believed that there exist numerous NB-IoT nodes on the market that do not support RAI. For these legacy nodes, changing the `Inactivity Timer` on eNodeB would be a better solution without requiring developers to manually update the firmware of UE.

We discuss our evaluation of changing the `Inactivity Timer` value in Section 5.2.

4.4.6 Repetition of Random Access MSGs

`MSG1` and `MSG3` are the two messages transmitted by the UE in the random access to inform the eNodeB about its connection requests and the required uplink resources. As shown in Section 4.4.4, `MSG3` transmission dominates the total energy consumption of uplink transmission under ECL2. Unlike data packet transfer blocks, where every successful block delivery will be acknowledged by the NB-IoT eNodeB, the random access `MSG1` and `MSG3` transfer blocks are transmitted with the fixed number of repetition without stopping. Although the NB-IoT node enjoys an increased coverage gain (1 dB when the repetition doubles) from the repetition, the debug log analysis shows that the high `MSG3` repetition, which is a fixed number 32 in ECL2, is unnecessary, leading to significant energy waste. We compare the repetition of the data transfer block with that of the `MSG3` for more than

2,300 packets, as shown in Figure 4.20. The results indicate that more than 66% of the packets have more MSG3 repetitions than the data transfer block repetitions.



(a) Categorical statistics of repetition

(b) The count by repetition combination

Figure 4.20: The distribution of MSG3 repetitions v.s. data transfer block repetitions.

As plotted in Figure 4.20b, we notice that the data block repetition in most of the packets is *1*, regardless of the ECL. Moreover, as indicated in our measurement, the packet data delivery rate, which is 100% in the majority of the tests, does not suffer too much from the low data block repetition. If we assume that the wireless channel quality remains relatively stable during the short UL transmission period, the MSG3 repetition is significantly larger than needed, providing an opportunity to reduce energy waste. From the eNodeB's point of view, it has to wait until MSG3 completes the repetition to proceed to the next step, even if the message is already received. Since the MSG3 and the packet data share the NPUSCH, the unnecessary repetition of MSG3 may lead to network congestion or low throughput.

Additionally, although the exact number of MSG3 repetitions remains unknown due to the absence of debug logs, the pulse width of MSG3 transmission by nodes of US-OP1 is observed to be smaller (about 0.5x) than that of CN-OP1 and CN-OP2, which is another factor that contributes to lower packet energy of nodes on US-OP1 than on the other two operators as shown in Figure 4.10a.

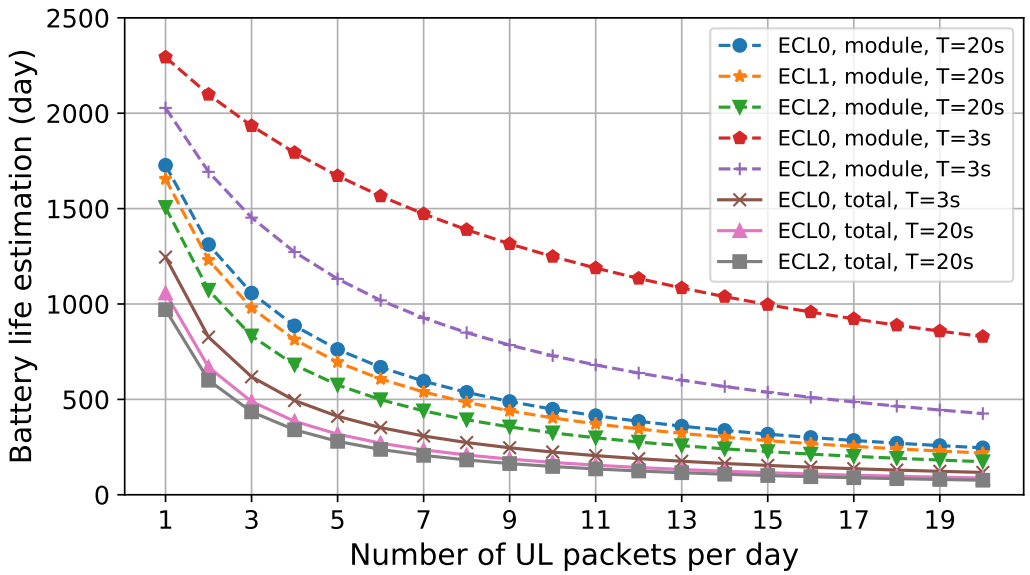


Figure 4.21: Battery life estimation under different conditions. "module" refers to only the NB-IoT module energy; "total" includes the energy consumption of both the module and other components on the board; "T" means `Inactivity Timer`.

4.4.7 UE Battery Life Estimation

Our fine-grained profiling of UE power consumption and the commodity eNodeB configurations allow us to estimate UE battery life in real-world deployments. The total packet energy can be calculated by summing up the energy of the radio access phases, while each phase energy can be tuned according to network parameters. We assume that the UE stays in `PSM` when it is not transmitting. The UE voltage and battery capacity are 3.3 V and 5 Wh, respectively. We assume that the UE consumes 50 mA extra current, a typical energy profile for an embedded system, to account for the power consumption of other system components, such as MCU and onboard sensors, during the packet cycle. We then calculate the energy during the packet transmission cycles. The battery life in days is estimated by dividing the battery capacity by the energy spent per day, depending on the data traffic.

Figure 4.21 shows the estimated battery life under different ECLs, `Inactive Timers`. First, the battery life decreases sharply with the number of packets transmitted. Second,

consistent with our observation in Section 4.4.3, the coverage level plays a critical role in the total energy consumption, and indoor UEs suffer significantly shorter life than their peers in good signal coverage. We now use an example to illustrate the impact of bad coverage levels. In a typical indoor parking monitoring system (which is one of the popular NB-IoT applications [72]), suppose each node monitoring a parking spot transmits 5 packets during a workday and works under ECL2, the battery life is only 270 days. However, this is an optimistic estimation, because the battery can be affected by various environmental factors. This would lead to frequent node replacement and high maintenance costs. As a result, it is of great importance to optimize UE power consumption. In particular, a small amount of saved packet energy can lead to tens or hundreds of days of prolonged node life. For example, considering the above scenario, if the `Inactivity Timer` is set to 3 seconds, the battery life extends to 440 days, which improves by about 63%.

4.5 Conclusion

In this chapter, we present NB-Scope – the first NB-IoT diagnostic hardware tool that supports fine-grained fusion of power and protocol traces, and a large-scale field measurement study consisting of 30 nodes deployed at over 1,200 locations in 3 regions during three months. Our studies show that NB-IoT nodes yield significantly imbalanced energy consumption in the wild, due to poor network coverage level, long-tail power profile, and excessive control message repetitions. Based on our estimation, the 10-year battery life expectation is difficult if not impossible to achieve due to the heterogeneous factors that shorten the battery life. Moreover, the hardware/software architecture of NB-Scope is largely independent of the evolution of NB-IoT. Therefore, it provides a powerful tool to instrument large-scale NB-IoT networks and will facilitate the understanding and optimization of NB-IoT during its evolution.

Our study in this chapter focuses on the energy consumption of NB-IoT UE. By combining the debug log with packet-level measurements, our tool NB-Scope can be used to

instrument large-scale NB-IoT networks to study various factors related to network performance such as reliability, latency, interference, etc. We leave these to open problems for the research community.

CHAPTER 5

NB-IOT NETWORK ENERGY OPTIMIZATION AND BEYOND

5.1 Introduction

Narrowband Internet of Things (NB-IoT) provides new infrastructure for the low-power wide area network, targeting massive connection, deep coverage, and long battery life. To achieve the 10-year battery life promise, NB-IoT adopts multiple energy-saving techniques, including extended Discontinuous Reception (eDRX), Power Saving Mode (PSM), and Enhanced Coverage Levels (ECL). However, the effectiveness of the energy preserving methods and the message signaling between the nodes and the base station was not well-understood.

To understand the NB-IoT power consumption insights in the wild, in previous Chapter 4, we propose the design of the toolset, NB-Scope, for NB-IoT network benchmark and diagnosis. By large-scale measurements in the field, we present the energy consumption profiles of the deployed NB-IoT networks by addressing the influence of various heterogeneous factors, such as location profiles, network operator, module chipset design, and distance between the node and the base station. We reveal that the energy drain is significantly imbalanced across these factors. This leads to short battery life and frequent network partition, where some nodes may die out in a shorter time than their siblings. As a consequence, the IoT application may suffer from high maintenance costs in battery replacement, or data loss issues. By decomposing the energy consumption with respect to data packet procedures, we show that the ECL mechanism, the long-tail timer configuration, and the random access procedure can be blamed for the large energy consumption. Fortunately, we find that the room for optimizing energy consumption from the NB-IoT protocol is huge.

In this chapter, we first propose our experimental platform for studying the NB-IoT

energy optimization, then we propose three energy optimization techniques. Particularly, we optimize the `Inactivity Timer` and the `MSG3` repetition count.Next, our evaluation indicates that, by combining these energy optimizations, it is possible to save up to 66.4% of the average power consumption. Then we propose three additional optimizations, however, evaluation via experiment is not yet possible due to the limitation in the hardware features. Finally, we propose an updated version of NB-Scope hardware to support both NB-IoT and LoRa modules, which opens new research opportunities for further exploiting the emerging wireless infrastructures.

5.2 Methodology

Our measurements show that inappropriate configurations of eNodeB base station, such as prolonged `Inactive Timer` and excessive `MSG3` repetition count, cause a significant portion of energy drain at UE. In current NB-IoT specification, `Inactive Timer` and `MSG`s repetition count are determined and enforced by eNodeB as fixed values. In this section, we explore the possibility of optimizing `Inactive Timer` and `MSG3` repetition by eNodeB. Our results provide important guidelines for network operators to optimize the power consumption of UE devices.

Specifically, an optimal `MSG3` repetition should assure reliable upload of `MSG3` while avoiding unnecessary retransmissions. An optimal `Inactive Timer` should minimize the waiting time of UE after completing packet upload. Meanwhile, it should avoid downlink packet loss, which occurs when the UE enters PSM or eDRX before the arrival of downlink packets. Ideally, the `Inactive Timer` should be configured based on downlink latency, which is determined by network round trip time (RTT) and the application server processing delay.

Commercial eNodeBs are closed systems, which cannot be accessed for experiments. To address this issue, we build an NB-IoT eNodeB testbed using software radios. We implement the eNodeB by using Amarisoft LTE100 as baseband core, and employing a USRP

Figure 5.1: The SDR eNodeB implementation, with Amarisoft LTE100 and USRP N210.

N210 with an SBX-40 as RF frontend, as shown in Figure 5.1. To emulate a commodity eNodeB, we configure our eNodeB using the same MSGs repetition and `Inactivity Timer` as the commodity one, running at Band 5 (EARFCN=2530). The detailed configurations are listed in Table 5.1. Our eNodeB achieves coverage of 1 km in open areas, which is comparable to the coverage of commodity eNodeB. Then, we deploy 9 BC35 UEs in the network, such that their average RSRPs are monotonically decreasing. Specifically, 2, 3, and 4 nodes are set to work under ECL0/1/2 respectively.

In our experiment, the RTT is defined as the delay from issuing the packet uplink AT command to the downlink packet delivery. The server replies with a DL packet immediately once it receives a UL packet. In total, there are 400 packets transmitted to a server on the local Cloud.

To explore the impact of MSG3 repetition, we first try to decrease it in ECL2 from 32 to 16 in eNodeB, then further down to 8. Each UE in the network is programmed to transmit 50 UL packets per reconfiguration. If the MSG3 is delivered to the eNodeB, the UE receives MSG4 and proceeds to data transmission. Otherwise, it retries random access from MSG1 again. Thus, we can quantify the effectiveness of the repetition configuration by calculating the probability of excessive MSG3 schedules in the nodes in the ECL2 coverage area.

Table 5.1: List of eNodeB default configuration

Parameter	Value		
	ECL0	ECL1	ECL2
NB-IoT mode	Standalone		
Band	5		
Tx power (dBm)	15		
EARFCN	2530		
Inactivity Timer (s)	20		
`MSG1` Repetition	2	8	32
`MSG3` Repetition	1	2	32
ECL RSRP threshold (dB)	N/A	-97	-107

5.2.1 Inactivity Timer Optimization Evaluation

Figure 5.2 shows the distribution of the RTT under different ECLs. As we can see, the UEs in ECL2 typically have larger packet RTT than UEs of ECL1 and ECL0. Since the RTT between the eNodeB and the application server is similar across these ECLs, the large RTT in ECL2 is primarily due to the delay in the downlink transmission. Compared to the 20-second timer value in CN-OP1 and CN-OP2, a timer value around 8 seconds can save up to *60%* of the energy consumption during the `Inactivity` period while 98% of the ECL0 and ECL1 nodes and 81% of the ECL2 nodes can receive the DL packet before the `Inactivity Timer` expires. Considering the portion of ECL2 nodes relatively smaller, a more aggressive strategy is to further reduce the timer to 5 seconds, while the ECL2 nodes can release the RRC connection earlier and enable eDRX to listen to the DL paging signal. The drawback of such an approach is the slight increase in the RRC connection requests in the network. However, it is worth considering an extra 10% of the `Inactivity` period energy can be saved. Our experiment aims to find a lower bound for the timer configuration, as a proof of concept. We note that the timer value in commercial eNodeBs should also consider various factors that may induce extra delays, such as node density,

downlink traffic, and the RRC connection request traffic.

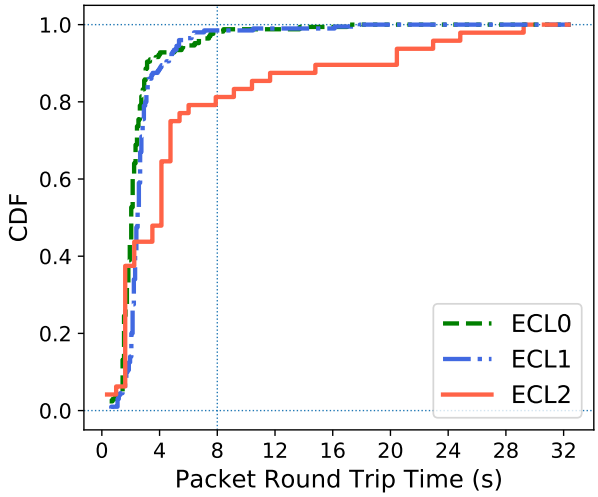


Figure 5.2: Round trip time CDF in the evaluation experiment.

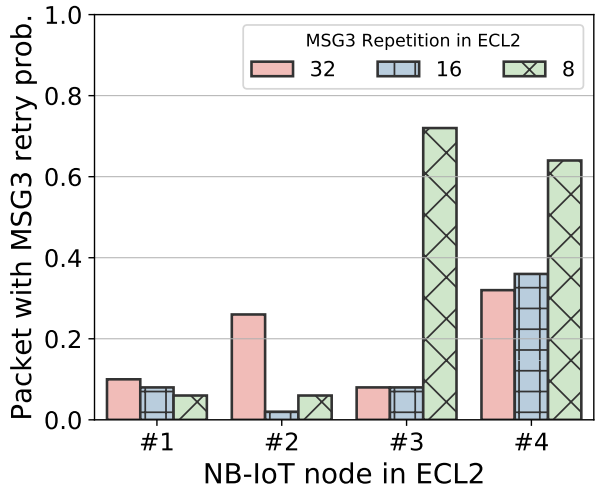


Figure 5.3: Prob. of packets with MSG3 re-transmission w.r.t ECL2 MSG3 repetition.

5.2.2 MSG3 Repetition Count Optimization Evaluation

Our goal is to investigate the minimum number of repetitions that can still ensure a high probability of one-time MSG3 delivery. The MSG3 retransmission probability is shown in Figure 5.3. All the nodes in the figure are designated to work in ECL2, and their average

Table 5.2: Mean energy consumption of the MSG3 period for different ECL2 MSG3 repetitions.

Node	Rep. = 8	Rep. = 16	Rep. = 32
#1	170.93 (1.04)	350.68 (1.46)	807.68 (11.63)
#2	170.60 (1.10)	346.44 (1.33)	680.63 (2.59)
#3	173.40 (0.91)	347.98 (3.21)	838.31 (26.04)
#4	169.70 (7.32)	354.13 (5.09)	697.69 (3.41)

signal strength is decreasing from left to right. As we can see, when the nodes are in a relatively good signal condition in ECL2, such as Node #1 and #2, different MSG3 repetitions yield a similar MSG3 retransmission rate, which means the reduced repetition count can still ensure delivery of MSG3 at the first attempt. We measure the energy consumption of the MSG3 transmission to be 171.1/349.8/756.1 mJ for 8/16/32 MSG3 repetition counts respectively on the test nodes. As a result, about 77% of the MSG3 Tx energy can be saved if a UE succeeds in RA by using 8 repetition counts of MSG3. Therefore, our experimental results demonstrate the feasibility of reducing the energy by using a lower MSG3 repetition without hurting the random access success rate. However, when the signal continues to attenuate, the MSG3 repetition number of 8 yields substantially higher random access retry probability, as shown by Node #3 and #4, resulting in energy wastage.

By combining the two optimizations above, a UE in ECL2 can potentially save up to **66.4%** of the packet energy consumption.

5.3 New Directions in Energy Optimization

Based on our measurement results, we now discuss several important design aspects that can be considered by future NB-IoT specifications and chipsets for optimizing energy consumption.

5.3.1 Per-UE Inactivity Timer

Our results indicated that a cell-wise `Inactivity Timer` can lead to significant energy waste for some UEs. A better solution would be to allow each UE to set the timer by negotiation with eNodeB. A UE without DL traffic can release the RRC connection immediately after the UL, saving a significant amount of energy. For example, a UE in ECL0 can save up to 90% packet energy by skipping the entire `Inactivity` period when it does not expect any incoming DL packets. Moreover, a UE can update the timer based on its measurement of network RTT or signal strength, which can achieve a desirable balance between the energy consumption and the RRC connection re-establishment rate.

5.3.2 ECL Adaptation of UE

Our results showed that the simple threshold-based ECL selection policy plays a critical role in the significant energy waste and imbalance of UE. We now discuss several ways to improve the ECL decision algorithm. First, other metrics could be integrated with RSRP and SNR to determine the ECL selection, including the block error rate which directly measures the block transmission status. The UE may also refer to its ECL decision history to explore better ECL choices. If the UE fails in using a lower ECL at certain channel quality in the past, it may consider using a higher ECL in a similar channel condition, avoiding the energy waste in the low ECL attempts. Finally, instead of responding to the UL packet AT command immediately, the UE may postpone the random access for delay-tolerant applications, waiting for a time window with better channel quality. We note that the implementation of these policies would require the support of NB-IoT chipsets, e.g., allowing developers to access and override ECL decisions.

5.3.3 Fine-grained ECLs

Our results showed that ECL2 yields significantly more energy consumption than other ECLs, due to excessive control message repetitions and unnecessary random access retries. To achieve a more desirable trade-off between energy consumption and random access efficiency, we suggest including extra ECLs and finer-grained RSRP thresholds and RA resource allocation between the ECLs. For instance, a UE in a relatively good signal condition may choose a new ECL between the legacy ECL1 and ECL2, which entails a lower repetition count of `MSG3` to save power. We note that adding new ECLs may introduce extra signaling between the eNodeB base station and UE.

5.3.4 Collaborative Energy Saving

The evolution of edge computing provides new opportunities for energy optimization. Specifically, the nodes deployed in a wide area can upload their signal strength and block error rate to the edge computing server. If the node supports mobility, then the signaling can be associated with the geolocation in real-time. Then, by building the spatio-temporal quality of service (QoS) profile from crowdsourcing, the edge computing server is able to recommend the nodes for their next random access timestamp as well as the ECL for random access. In this manner, the power-saving timer can be dynamically configured, avoiding crowded channels, signal blind spots, or periods with low signal strength. We note that this is difficult, if not impossible, if there are only hundreds or fewer nodes in the area. Instead, when tens of thousands of nodes coexist in the area, the megatrends underlying the LPWAN and the global knowledge of the massive connectivity will enable the novel patterns for energy optimization.

5.4 Beyond NB-Scope

In this section, we present our upgraded NB-Scope design, which is referred to as *NB-Scope V2*. It supports Long Range (LoRa) communication and in-situ diagnostic message

decoding while inherits all the major functionalities from its predecessor.

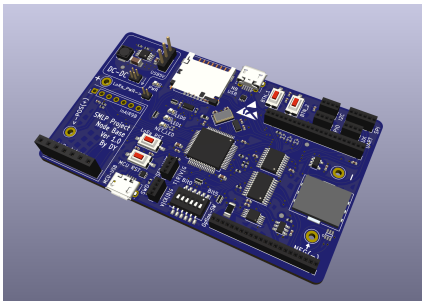
5.4.1 In-device Signaling Decoding

One of the notable upgrades in NB-Scope V2 software design is the in-device debugging message decoding. The original NB-Scope supports streaming the debugging messages from the NB-IoT module to the SD card. However, understanding the real-time UE behavior in the field-test mode was not supported by the node. Retrospecting and reconstructing the UE's behavior are required to further analyze the network status.

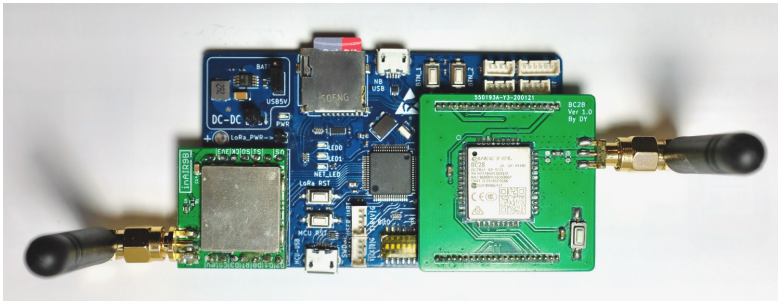
Fortunately, the in-device message decoding is implemented in the latest version of NB-Scope. To achieve this, first of all, we recompile and compress the debugging message definition database using hexadecimal indices to allow for a storage-constrained embedded system. Next, we implement the message decoder finite state machine (FSM) in the MCU. Finally, by combining the message decoder and the message database, we achieve real-time in-device debug message decoding, such that the device is aware of the comprehensive network status without the need of querying the NB-IoT module using the low efficient AT commands. Enabled by this functionality, the node can not only explore real-time dynamic control strategies to improve the communication efficiency but also able to feed back the network status in the packet payload to the application servers for real-time collaboration.

5.4.2 Towards the Coexistence of NB-IoT and LoRa

The NB-IoT network adopts a star topology, where the nodes only communicate with the base station but not aware of their neighbor NB-IoT nodes. To address this, we upgrade NB-Scope to version 2, as shown in Figure 5.4. One of the major hardware updates of NB-Scope is to support the commercial inAIR9B LoRa module while backward compatible with all of our NB-IoT shield boards.



(a) NB-Scope V2 mainboard.



(b) NB-Scope V2 node, with the inAIR9B module and the NB-IoT shield board plugged together.

Figure 5.4: NB-Scope V2 hardware design.

By communicating with the neighbor NB-IoT nodes, the V2 nodes can establish a mesh network, which opens up new research opportunities. For example, we can achieve collaborative sensing, where multiple nodes finish the tasks by distributed consensus. It is also possible to save much energy if the nodes in the ECL2 condition using the help of their neighbor nodes with good signal conditions to route the uplink data packet to the server. As shown in Section 4.4.4, the nodes in ECL2 consume up to 4.41x higher energy than the nodes in ECL0. Therefore, such a multi-hop routing strategy is able to save energy significantly, and the energy imbalance can be substantially mitigated by using optimal routing protocols.

The coexistence of NB-IoT and LoRa powered by NB-Scope V2 provides a new paradigm for the LPWAN research community.

5.5 Conclusion

In this chapter, we proposed two energy optimizations regarding the `Inactivity Timer` and the `MSG3` Repetition, both of which preserve the energy significantly. Specifically, up to 66.4% of energy can be saved for nodes under critical signal conditions. Then, we propose 4 different energy optimizations under the NB-IoT framework, providing new research directions to the community. Next, we present our upgrade to the NB-Scope design, which now supports in-device real-time diagnostic messages decoding and NB-IoT-LoRa dual-

mode wireless communication. Our efforts open new paradigms for LPWAN research and applications.

CHAPTER 6

CONCLUSION

The wireless technologies have been evolved speedily, whose infrastructure is built and delivered speedily. The fast growth of wireless networks generates not only opportunities for new applications, but also issues in high energy consumption, unexpected latency, and potential privacy breach.

In this dissertation, we propose two novel cyber-physical systems to demonstrate the possibility of enabling new applications by leveraging the emerging wireless charging infrastructures, and benchmark the energy performance of end nodes in the recently debuted NB-IoT networks, respectively.

First, we present QID, the first system that can identify a Qi-compliant device during wireless charging in real-time using wireless charging fingerprints. QID employs a 2-dimensional motion unit to emulate a variety of multi-coil designs of Qi, which allows for fine-grained device fingerprinting. With the novel mobile coil design and a set of novel features, such as control error packet interval distribution and control error value distribution, QID achieves about 90% device recognition accuracy and almost 100% device brand recognition accuracy. QID demonstrates the possibility of leveraging wireless charging stations to identify mobile devices and track the users. With the prevalence of public wireless charging stations, our results also have important implications for mobile user privacy.

Second, we develop a novel benchmarking ecosystem, called *NB-Scope*, to study the energy performance of the Narrowband Internet-of-things (NB-IoT) network. NB-Scope adopts a hierarchical design, resolving the heterogeneity in network operators, node module vendors, and location profiles, to allow for the fusion of fine-grained diagnostic traces and current measurement. The small form factors of the NB-Scope field test configuration enable the agile and flexible deployment, which supports concurrent multi-spot measure-

ment, making a fair comparison between the nodes with different settings. We then conduct a large-scale field measurement study consisting of 30 nodes deployed at over 1,200 locations in 3 regions during a period of three months. Our in-depth analysis of the collected 49 GB traces showed that NB-IoT nodes yield significantly imbalanced energy consumption in the wild, up to a ratio of 75:1, which may lead to short battery lifetime and frequent network partition. By extensive data analysis, we identify several key factors including diverse network coverage levels, long-tail power profile, and excessive control message repetitions, that lead to high variance in the energy performance. Based on our data, we conclude that the NB-IoT UEs are not able to achieve 10-year projected battery life, even in the most optimistic scenario, which is noteworthy for the NB-IoT application developers.

Third, we explore the optimization of NB-IoT base station settings on a software-defined eNodeB testbed and suggest several important design aspects that can be considered by future NB-IoT specifications and chipsets. Overall, our work for the first time addressing the energy performance of the NB-IoT network, providing insights for the NB-IoT research and development communities. As our efforts to keep NB-Scope evolved, we propose its successor, with both the hardware and software upgraded. By supporting in-device message decoding and LoRa-NB-IoT coexistence, NB-Scope V2 provides new opportunities for LPWAN research and applications.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Ieee standard for low-rate wireless networks. *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pages 1–709, April 2016.

[2] Omid Abari, Deepak Vasisht, Dina Katabi, and Anantha Chandrakasan. Caraoke: An e-toll transponder network for smart cities. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 297–310. ACM, 2015.

[3] Carlos A Astudillo, Fernando HS Pereira, and Nelson LS da Fonseca. Probabilistic retransmissions for the random access procedure in cellular iot networks. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2019.

[4] Atmel Corporation. *SAM G53N SMART ARM-based Flash MCU Datasheet*, 11240f edition, July 2015. 32-bit ARM Cortex-M4 RISC processor.

[5] Amin Azari, Guowang Miao, Cedomir Stefanovic, and Petar Popovski. Latency-energy tradeoff based on channel scheduling and repetitions in nb-iot systems. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2018.

[6] Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. Ieee, 2000.

[7] Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, pages 280–293. ACM, 2009.

[8] SIG Bluetooth. Bluetooth 4.2 core specification. *Bluetooth SIG*, 2009.

[9] Hristo Bojinov, Yan Michalevsky, Gabi Nakibly, and Dan Boneh. Mobile device identification via sensor fingerprinting. *arXiv preprint arXiv:1408.1416*, 2014.

[10] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[11] Taoufik Bouguera, Jean-François Diouris, Jean-Jacques Chaillout, Randa Jaouadi, and Guillaume Andrieux. Energy consumption model for sensor nodes based on lora and lorawan. *Sensors*, 18(7):2104, 2018.

[12] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[13] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 116–127. ACM, 2008.

[14] Alistair Charlton. Wirelessly charge anywhere with these qi-enabled tables, lamps, speakers and accessories. https://www.gearbrain.com/qi-wireless-charging-tables-lamps-2528825500.html. Accessed: 2018-07-25.

[15] Shichao Chen, Gang Xiong, Jia Xu, Shuangshuang Han, Fei-Yue Wang, and Kun Wang. The smart street lighting system based on nb-iot. In *2018 Chinese Automation Congress (CAC)*, pages 1196–1200. IEEE, 2018.

[16] Xiaomeng Chen, Ning Ding, Abhilash Jindal, Y Charlie Hu, Maruti Gupta, and Rath Vannithamby. Smartphone energy drain in the wild: Analysis and implications. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):151–164, 2015.

[17] Yu-Shin Chou and Jing-Sin Liu. A robotic indoor 3d mapping system using a 2d laser range finder mounted on a rotating four-bar linkage of a mobile platform. *International Journal of Advanced Robotic Systems*, 10(1):45, 2013.

[18] Wireless Power Consortium. Magnetic resonance and magnetic induction - what is the best choice for my application? https://tinyurl.com/wireless-charging-choices, 2017. Accessed: 2019-02-13.

[19] Wireless Power Consortium. Qi wireless charging goes mainstream. http://www.air-charge.com/news/21/19/Qi-wireless-charging-goes-mainstream, 2017. Accessed: 2019-01-09.

[20] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[21] Marius Cristea and Bogdan Groza. Fingerprinting smartphones remotely via icmp timestamps. *Communications Letters, IEEE*, 17(6):1081–1083, 2013.

[22] Boris Danev and Srdjan Capkun. Transient-based identification of wireless sensor nodes. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 25–36. IEEE Computer Society, 2009.

[23] Anupam Das, Nikita Borisov, and Matthew Caesar. Do you hear what i hear? fingerprinting smart devices through embedded acoustic components. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 441–452, 2014.

[24] Anupam Das, Nikita Borisov, and Matthew Caesar. Tracking mobile web users through motion sensors: Attacks and defenses. In *NDSS*, 2016.

[25] Ning Ding, Daniel Wagner, Xiaomeng Chen, Abhinav Pathak, Y Charlie Hu, and Andrew Rice. Characterizing and modeling the impact of wireless signal strength on smartphone battery drain. In *ACM SIGMETRICS Performance Evaluation Review*, pages 29–40. ACM, 2013.

[26] Sarun Duangsuwan, Aekarong Takarn, Rachan Nujankaew, and Punyawi Jamjareegulgarn. A study of air pollution smart sensors lpwan via nb-iot for thailand smart cities 4.0. In *2018 10th International Conference on Knowledge and Smart Technology (KST)*, pages 206–209. IEEE, 2018.

[27] Werner Dubitzky, Martin Granzow, and Daniel P Berrar. *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media, 2007.

[28] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. Empowering low-power wide area networks in urban settings. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 309–321. ACM, 2017.

[29] Ericsson. The ericsson mobility report. `https://www.ericsson.com/en/blog/2019/10/what-is-NB-IoT`, 2019.

[30] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[31] Jessica Fridrich. Digital image forensics. *IEEE Signal Processing Magazine*, 26(2):26–37, 2009.

[32] Jon Gjengset, Jie Xiong, Graeme McPhillips, and Kyle Jamieson. Phaser: Enabling phased array signal processing on commodity wifi access points. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 153–164, 2014.

[33] Juergen Graefenstein, Amos Albert, Peter Biber, and Andreas Schilling. Wireless node localization based on rssi using a rotating antenna on a mobile robot. In *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*, pages 253–259. IEEE, 2009.

[34] GSACOM. Global narrowband iot – lte-m networks – march 2019. `https://gsacom.com/paper/global-narrowband-iot-lte-m-networks-march-2019/`, 2019. Accessed: 2020-03-19.

[35] GSMA. Mobile iot network launches. `https://www.gsma.com/iot/mobile-iot-commercial-launches/`, 2020. Accessed: 2020-02-19.

[36] Nils Y Hammerla and Thomas Plötz. Let's (not) stick together: pairwise similarity biases cross-validation in activity recognition. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pages 1041–1051. ACM, 2015.

[37] Canlong He, Mingxia Shen, LongShen Liu, C Okinda, Ji Yang, Hong Shi, et al. Design and realization of a greenhouse temperature intelligent control system based on nb-iot. *Journal of South China Agricultural University*, 39(2):117–124, 2018.

[38] Jun Huang, Wahhab Albazrqaoe, and Guoliang Xing. Blueid: a practical system for bluetooth device identification. In *INFOCOM, 2014 Proceedings IEEE*, pages 2849–2857. IEEE, 2014.

[39] Junxian Huang, Feng Qian, Alexandre Gerber, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. A close examination of performance and power characteristics of 4g lte networks. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 225–238. ACM, 2012.

[40] Faheem Ijaz, Hee Kwon Yang, Arbab Waheed Ahmad, and Chankil Lee. Indoor positioning: A review of indoor ultrasonic positioning systems. In *Advanced Communication Technology (ICACT), 2013 15th International Conference on*, pages 1146–1150. IEEE, 2013.

[41] Monsoon Solutions Inc. High voltage power monitor. `https://www.msoon.com/high-voltage-power-monitor`, 2019. Accessed: 2019-09-09.

[42] Infineon Technologies. *Application brochure - Wireless charging for consumer*, 3 2018. Rev 4.0.

[43] Han Seung Jang, Hu Jin, Bang Chul Jung, and Tony QS Quek. Versatile access control for massive iot: Throughput, latency, and energy efficiency. *IEEE Transactions on Mobile Computing*, 2019.

[44] Tadayoshi Kohno, Andre Broido, and Kimberly C Claffy. Remote physical device fingerprinting. *Dependable and Secure Computing, IEEE Transactions on*, 2(2):93–108, 2005.

[45] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 269–282. ACM, 2015.

[46] Surapon Kraijak and Panwit Tuwanut. A survey on internet of things architecture, protocols, possible applications, security, privacy, real-world implementation and future trends. In *2015 IEEE 16th International Conference on Communication Technology (ICCT)*, pages 26–31. IEEE, 2015.

[47] Sivanand Krishnan, Pankaj Sharma, Zhang Guoping, and Ong Hwee Woon. A uwb based localization system for indoor robot navigation. In *2007 IEEE International Conference on Ultra-Wideband*, pages 77–82. IEEE, 2007.

[48] Mads Lauridsen, Huan Nguyen, Benny Vejlgaard, István Z Kovács, Preben Mogensen, and Mads Sorensen. Coverage comparison of gprs, nb-iot, lora, and sigfox in a 7800 km^2 area. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE, 2017.

[49] Yilong Li, Yun Cheng, Xiucheng Li, Yu Wang, Guoliang Xing, and Xiaofan Jiang. Qiloc–a qi-wireless based platform for robust user-initiated indoor location services:

Demo abstract. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, BuildSys'14, pages 184–185, New York, NY, USA, 2014. ACM.

[50] Yuanjie Li, Haotian Deng, Chunyi Peng, Zengwen Yuan, Guan-Hua Tu, Jiayao Li, and Songwu Lu. icellular: Device-customized cellular network access on commodity smartphones. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 643–656, 2016.

[51] Yuanjie Li, Chunyi Peng, Zengwen Yuan, Jiayao Li, Haotian Deng, and Tao Wang. Mobileinsight: Extracting and analyzing cellular network information on smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, MobiCom '16, pages 202–215, New York, NY, USA, 2016. ACM.

[52] Yuanjie Li, Jiaqi Xu, Chunyi Peng, and Songwu Lu. A first look at unstable mobility management in cellular networks. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, pages 15–20, 2016.

[53] Yuke Li, Xiang Cheng, Yang Cao, Dexin Wang, and Liuqing Yang. Smart choice for the smart grid: Narrowband internet of things (nb-iot). *IEEE Internet of Things Journal*, 5(3):1505–1515, 2017.

[54] Jansen C Liando, Amalinda Gamage, Agustinus W Tengourtius, and Mo Li. Known and unknown facts of lora: Experiences from a large-scale measurement study. *ACM Transactions on Sensor Networks (TOSN)*, 15(2):16, 2019.

[55] Kaikai Liu, Xinxin Liu, and Xiaolin Li. Guoguo: Enabling fine-grained indoor localization via smartphone. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 235–248, 2013.

[56] Xiao Lu, Dusit Niyato, Ping Wang, Dong In Kim, and Zhu Han. Wireless charger networking for mobile devices: Fundamentals, standards, and applications. *IEEE Wireless Communications*, 22(2):126–135, 2015.

[57] Marko Malajner, Peter Planinsic, and Dusan Gleich. Angle of arrival estimation using rssi and omnidirectional rotatable antennas. *IEEE Sensors Journal*, 12(6):1950–1957, 2012.

[58] IHS Markit. Half a billion smartphones and other devices with wireless power technology shipped in 2017, ihs markit says. `https://tinyurl.com/qi-half-billion`, 2018. Accessed: 2018-05-19.

[59] Tom M Mitchell. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45:995, 1997.

[60] Ashkan Nikravesh, Hongyi Yao, Shichang Xu, David Choffnes, and Z.Morley Mao. Mobilyzer: An open platform for controllable mobile network measurements. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '15, pages 389–404, New York, NY, USA, 2015. ACM.

[61] NXP Semiconductors. *Freescale Wireless Charging ICs, MWCT1000CFM, MWCT1200CFM, MWCT1101CLH*, 5 2014. REV 1.

[62] NXP Semiconductors. *MWPR1516: Higher integration receiver controller MCU for wireless power transfer application*, 1 2015. Rev. 2.00.

[63] Panasonic Corporation. *AN32258A: Intergrated Wireless Power Supply Receiver, Qi (Wireless Power Consortium) Compliant*, 10 2014. Rev. 2.00.

[64] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[65] Marco Pennacchioni, Maria-Gabriella Di Benedette, Tommaso Pecorella, Camillo Carlini, and Pietro Obino. Nb-iot system deployment for smart metering: Evaluation of coverage and capacity performances. In *2017 AEIT International Annual Conference*, pages 1–6. IEEE, 2017.

[66] Drew Prindle. Impress your guests (and top off their phones) with this diy wireless charging table. `https://www.digitaltrends.com/how-to/diy-wireless-charging-table/`. Accessed: 2018-07-25.

[67] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM, 2000.

[68] Rohde&Schwarz. Power saving methods for lte-m and nb-iot devices. `https://www.rohde-schwarz.com/lt/solutions/test-and-measurement/wireless-communication/iot-m2m/whitepaper-power-saving-lte-m-nb-iot-register_251417.html`, 2019. Accessed: 2020-02-16.

[69] ROHM Semiconductor. *BD57011AGWL: A stand-alone integrated IC for wireless power receiver*, 1 2018. Rev. 003.

[70] Ali Sehati and Majid Ghaderi. Online energy management in iot applications. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1286–1294. IEEE, 2018.

[71] Junyang Shen and Andreas F Molisch. Passive location estimation using toa measurements. In *2011 IEEE International Conference on Ultra-Wideband (ICUWB)*, pages 253–257. IEEE, 2011.

[72] Jiong Shi, Liping Jin, Jun Li, and Zhaoxi Fang. A smart parking system based on nb-iot and third-party payment platform. In *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, pages 1–5. IEEE, 2017.

[73] Hyojeong Shin, Yohan Chon, Yungeun Kim, and Hojung Cha. Mri: Model-based radio interpolation for indoor war-walking. *IEEE Transactions on Mobile Computing*, 14(6):1231–1244, 2015.

[74] Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang. A survey on lpwa technology: Lora and nb-iot. *Ict Express*, 3(1):14–21, 2017.

[75] Texas Instruments. *bq51013B Highly Integrated Wireless Receiver Qi (WPC v1.2) Compliant Power Supply*, 3 2018. REVISED MARCH 2018.

[76] Emiliano Trevisani and Andrea Vitaletti. Cell-id location technique, limits and benefits: an experimental study. In *Sixth IEEE workshop on mobile computing systems and applications*, pages 51–60. IEEE, 2004.

[77] Arvin Wen Tsui Tsui, Wei-Cheng Lin, Wei-Ju Chen, Polly Huang, and Hao-Hua Chu. Accuracy performance analysis between war driving and war walking in metropolitan wi-fi localization. *IEEE Transactions on Mobile Computing*, 9(11):1551–1562, 2010.

[78] Diego Valsesia, Giulio Coluccia, Tiziano Bianchi, and Enrico Magli. Compressed fingerprint matching and camera identification via random projections. *IEEE Transactions on Information Forensics and Security*, 10(7):1472–1485, 2015.

[79] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-level localization with a single wifi access point. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 165–178, 2016.

[80] Jianxin Wang, Junpan Su, and Ruyuan Hua. Design of a smart independent smoke sense system based on nb-iot technology. In *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, pages 397–400. IEEE, 2019.

[81] Jue Wang and Dina Katabi. Dude, where's my card? rfid positioning that works with multipath and non-line of sight. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 51–62, 2013.

[82] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.

[83] Wireless Power Consortium. *The Qi Wireless Power Transfer System Power Class 0 Specification*, 1.2.3 edition, February 2017. Parts 1 and 2: Interface Definitions.

[84] Wireless Power Consortium. *The Qi Wireless Power Transfer System Power Class 0 Specification*, 1.2.3 edition, February 2017. Part 4: Reference Designs.

[85] Di Wu, Dmitri I Arkhipov, Yuan Zhang, Chi Harold Liu, and Amelia C Regan. Online war-driving by compressive sensing. *IEEE Transactions on Mobile Computing*, 14(11):2349–2362, 2015.

[86] Jie Xiong and Kyle Jamieson. Arraytrack: a fine-grained indoor location system. Usenix, 2013.

[87] Faheem Zafari and Ioannis Papapanagiotou. Enhancing ibeacon based micro-location with particle filtering. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2015.

[88] Jiexin Zhang, Alastair R Beresford, and Ian Sheret. Sensorid: Sensor calibration fingerprinting for smartphones. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 638–655. IEEE, 2019.

[89] Ning Zhang and Yingjie Liu. Nb-iot drives intelligent cold chain for best application. In *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 1–4. IEEE, 2019.