Automated Monitoring for Data Center Infrastructure

Automated Monitoring for Data Center Infrastructure

By Mehdi Jafarizadeh

A Thesis Submitted to the School of Graduate Studies in the Partial Fulfillment of the Requirements for the Degree Ph.D. Thesis

McMaster University

Ph.D. Thesis (2021)

Hamilton, Ontario (Computer Science)

TITLE: Automated Monitoring for Data Center Infrastructure

AUTHOR: Mehdi Jafarizadeh (McMaster University),

M.Sc. in Computer Science, Amirkabir University, Tehran, Iran

SUPERVISOR: Dr. Rong Zheng

NUMBER OF PAGES: xvii, 130

Abstract

Environmental monitoring using wireless sensors plays a key role in detecting hotspots or over-cooling conditions in a data center (DC). Despite a myriad of Data Center Wireless Sensor Network (DCWSN) solutions in literature, their adoption in DCs is scarce due to four challenges: low reliability, short battery lifetime, lack of adaptability, and labour intensive deployment. The main objective of this research is to address these challenges in our specifically designed hierarchical DCWSN, called Low Energy Monitoring Network (LEMoNet).

LEMoNet is a two-tier protocol, which features Bluetooth Low Energy (BLE) for sensors communication in the first tier. It leverages multi-gateway packet reception in its second tier to mitigate the unreliability of BLE. The protocol has been experimentally validated in a small DC and evaluated by simulations in a midsize DC. However, since the main application of DCWSNs is in colocation and large DCs, an affordable and fast approach is still required to assess LEMoNet in large scale. As the first contribution, we develop an analytical model to characterize its scalability and energy efficiency in a given network topology. The accuracy of the model is validated through extensive event-driven simulations. Evaluation results show that LEMoNet can achieve high reliability in a network of 4800 nodes at a duty cycle of 15s.

To achieve the network adaptability, we introduce and design SoftBLE, a Software Defined Networking (SDN) based framework that provides controllability to the network. It takes advantages of advanced control knobs recently available in BLE

protocol stacks. SoftBLE is complemented by two orchestration algorithms to optimize gateway and sensor parameters based on run-time measurements. Evaluation results from both an experimental testbed and a large-scale simulation study show that using SoftBLE, sensors consume 70% less power in data collection compared to those in baseline approaches while achieving the Packet Reception Rate (PRR) no less than 99.9%.

One of its main steps of DCWSN commissioning is sensor localization, which is labour-intensive if is driven manually. To streamline the process, we devise a novel approach for automated sensor mapping. Since Radio Frequency (RF) alone is not a reliable data source for sensor localization in harsh and multi-path rich environments such as a DCs, we investigate using non-RF alternatives. Thermal Piloting is a classification model to correlate temperature sensor measurements with the expected thermal values at their locations. It achieves an average localization error of 0.64 meters in a modular DC testbed. The idea is further improved by a multi-modal approach that incorporates pairwise Received Signal Strength (RSS) measurements of RF signals. The problem is formulated as Weighted Graph Matching (WGM) between an analytical graph and an experimental graph. A parallel algorithm is proposed to find heuristic solutions to this NP-hard problem, which is 30% more accurate than the baselines. The evaluation in a modular DC testbed shows that the localization errors using multi-modality are less than one-third of that of using thermal data alone.

Acknowledgements

First of all, I would like to thank McMaster University, Computing and Software (CAS) department and Computing Infrastructure Research Center (CIRC) for providing the required facilities for this research. Next, I want to express my gratitude to my supervisor, Prof. Rong Zheng, who advised me professionally during all the steps of this research. Beside, I highly appreciate the time and comments of my committee members, Dr Douglas Down, and Dr Dongmei Zhao. I also should mention that accomplishment of this research project was impossible without the help of my colleagues, and without the full support of my wife.

Contents

Abstract iii

Acknowledgements v

Acronyms xv

Declaration of Authorship xvii

1 Introduction 1

1.1 Environmental Monitoring in Data Centers 1

1.2 Hierarchical IoT for Environmental Monitoring 5

1.3 Challenges of Implementing a DCWSN 7

1.4 Contributions 8

1.5 Organization 10

2 Background - Bluetooth Low Energy 12

2.1 BLE Protocol Stack 13

2.2 Advertising Packet Structure of BLE 14

2.3 BLE Legacy Advertising 16

2.4 Reliability of Advertisement in BLE 17

3 Related Work **20**

3.1 Data Center Wireless Sensor Networks 20

3.1.1 Protocol Design 21

3.1.2 Performance Evaluation Techniques 22

3.2 Network Controllability 24

3.2.1 SDN in IoT Edge 24

3.2.2 Parameter Configuration of BLE Networks 27

3.3 Automated Sensor Mapping 28

3.3.1 Localization in IoT 28

3.3.2 Weighted Graph Matching 30

4 Analysis of Low Energy Monitoring Networks **33**

4.1 Introduction 33

4.2 Overview of LEMoNet 34

4.2.1 Multi-Gateway Packet Reception 34

4.2.2 NCL vs SCL nodes 36

4.3 Performance Modeling of LEMoNet 37

4.3.1 Simplified Assumptions 37

4.3.2 System of Equations 39

4.4 Performance Evaluation 44

4.4.1 Validation Study 44

4.4.2 Parameter Assessment 45

4.5 Conclusion 48

5 SoftBLE: an SDN Framework for Network Adaptability **49**

5.1 Introduction 49

5.2 SoftBLE Design 50
5.3 Control Plane 53
5.3.1 Provisioning Timeline 55
5.3.2 Information Building 56
5.3.3 Control Knobs 56
5.4 Gateway Orchestration 57
5.5 Sensor Orchestration 60
5.5.1 Problem Formulation 60
5.5.2 Estimating PRR and Power Consumption 62
5.5.3 Optimal Parameter Selection 66
5.6 Performance Evaluation 67
5.6.1 Experimental Validation 67
5.6.2 Simulation Study 70
5.7 Conclusion 78

6 Automated Sensor Mapping for DCWSN Commissioning 79
6.1 Introduction 79
6.2 thermal piloting 81
6.2.1 System Architecture 81
6.2.2 Problem Formulation 82
6.2.3 Data Model 84
6.2.4 Localization Process 85
6.3 Multimodal Sensor Localization 86
6.3.1 Problem Formulation 87
6.3.2 SWAP-ing based Algorithm for Labeled WGM 91

6.3.3 Localization Process 97
6.4 Experimental Setup 98
6.4.1 Testbed Data Center 99
6.4.2 Pairwise Data Collection 100
6.4.3 Pointwise Data Collection 102
6.4.4 CFD Simulation 103
6.5 Performance Evaluation 104
6.5.1 Ground Truth 104
6.5.2 Accuracy of Thermal Estimation 104
6.5.3 Performance of the Proposed WGM Algorithm 106
6.5.4 Single Modality versus Multimodality 109
6.6 Conclusion 112

7 Conclusion and Future Work **113**

List of Figures

1.1 Thermal map of a sample data center (DC) with raised floor cooling. 2

1.2 Two models for Data Center Infrastructure Management System (DCIM). [20] 4

1.3 Hierarchical Internet of Things (IoT) architecture for dense monitoring in a sample DC. 6

1.4 The relation between the challenges of developing a Data Center Wireless Sensor Network (DCWSN) and our contributions for addressing them. 9

2.1 Bluetooth Low Energy (BLE) protocol stack. 13

2.2 Broadcasting BLE packet structure. 15

2.3 The time line of advertising *ADV_SCAN_IND* packets to an active scanner. 17

2.4 Bit Error Rate (BER) model in BLE and the corresponding Packet Error Rate (PER) for a 32-bytes packet. The noise level is set to 110dBm. 18

3.1 The role of coordinator in Zigbee and Network Server in LoRaWAN. 26

4.1 Comparing Packet Reception Rate (PRR) per gateway versus aggregated PRR by simulating a sample DC with 60 racks, monitored with 360 sensors. Blue filled circles are the gateways. 35
4.2 Energy consumption of measurement and advertising in LEMoNet using Normal Connectionless (NCL) mode (top row) versus Scannable Connectionless (SCL) mode (bottom row). 36
4.3 Block diagram of the analyical model for Low Energy Monitoring Network (LEMoNet). The white rectangular boxes are nonlinear equations and the shaded boxes are the outputs. 40
4.4 Simultaneous interference in a sample multi-gateway reception scenario with three gateways. 42
4.5 Comparison between the performance evaluation results [50] of simulation and the analytical model of LEMoNet. The network includes of 360 sensors deployed in a DC of 60 racks. The error bars represent the highest and the lowest values among the sensors. 44
4.6 Layout of the parameter assessment DCWSN with 1440 sensors. . . 46
4.7 Impacts of different parameters on LEMoNet performance in large scale DCWSNs. 47

5.1 The components of Software Defined Networking (SDN) framework in a SoftBLE. 51
5.2 The timeline of packet transmission in the control and forwarding plane of SoftBLE. 54
5.3 A scene of the experiment setup in our lab. 68

5.4 Experimental comparison between LEMoNet and SoftBLE. 48 sensors and 2 gateways have been deployed in a 11m-by-8m space. Duty cycles of all sensors are set to 3s. 68

5.5 The effect of TX power and channel assignment of a randomly selected sensor node. All other sensors' parameters remain fixed. Results include channel assignments of a single channel (•), two channels (+), and three channels ($\star$) at different TX power levels. . 72

5.6 The output of running orchestration algorithms on the scaled scenario including 2500 sensors advertising on 5s duty cycles, and 121 gateways. The sensors are symbolized based on whether they assigned to one channel (•), two channels (+), or three channels ($\star$). Gateways are shown with $\triangle$, colored based on their scanning channel (red: 37, purple:38, green:39). 73

5.7 Performance of SoftBLE in a network with 2500 nodes. 74

5.8 Effects of the number of sensors on the performance of SoftBLE. The sensors are deployed randomly in a $50m \times 50m$ area. Duty cycles of all sensors are set to 5s. 76

5.9 Effects of duty cycle on the performance of SoftBLE. 600 sensors are deployed randomly in a $50m \times 50m$ area. 77

6.1 Components of thermal piloting 81

6.2 Flowchart of the thermal piloting process 85

6.3 The proposed multimodal sensor localization framework. Sensors are mapped to designated locations by weighted matching between analytical and experimental graphs. 86

6.4 Calculation diagram of cost difference (Δ) matrix. 92
6.5 Flowchart of the thermal piloting process 98
6.6 Experiment testbed, a modular DC with five racks and two cooling units. In the first test, 25 sensors were monitoring the front of the servers. 99
6.7 The random switching strategy to collect Received Signal Strength (RSS) measurements between pairs of sensors. Sensors alternate between scanning and advertising periods. 100
6.8 Data model for the pointwise data. $T^z_{x,y}$ is the temperature of measured by sensor x in the steady state of scenario y at run z. 101
6.9 Experimental and simulation datasets; Cold side thermal maps from designated scenarios in different tests. 102
6.10 Sensor placement in tests 2 to 5. The locations are indexed in row-major order from upper most left. 105
6.11 Correlation between experiments and Computational Fluid Dynamics (CFD) simulations 106
6.12 Caption for LOF 107
6.13 Comparison between the objective values and the localization errors from the baseline methods versus those from our method in all tests 108
6.14 The effect of using the pointwise inputs along with pairwise inputs on the localization error in different tests 109
6.15 The final localization results and the graph matching objective values for different tests. 110

List of Tables

3.1 Specification of wireless protocols on the representative modules [50] 21

4.1 The parameters and constants of the system 39

4.2 The computation time of simulation versus analytical model for evaluating a network of 360 sensors 45

5.1 Control Knobs of SoftBLE devices 57

5.2 The parameters SoftBLE analytical model 63

6.1 Running time of different algorithms 109

Acronyms

DC data center

BLE Bluetooth Low Energy

DCIM Data Center Infrastructure Management System

DCWSN Data Center Wireless Sensor Network

WSN Wireless Sensor Network

WGM Weighted Graph Matching

PRR Packet Reception Rate

SNR Signal to Noise Ratio

RSS Received Signal Strength

BER Bit Error Rate

SCL Scannable Connectionless

NCL Normal Connectionless

PER Packet Error Rate

LEMoNet Low Energy Monitoring Network

SDN Software Defined Networking

RF Radio Frequency

RoI Return of Investment

PUE Power Usage Effectiveness

CRAC Computer Room Air Conditioner

IoT Internet of Things

IIoT Industrial IoT

ITE IT equipment

GFSK Gaussian Frequency Shift Keying

GAP Generic Access Profile

GATT Generic Attribute Profile

HCI Host-Control Interface

RPNR Random Private Non-Resolvable

CFD Computational Fluid Dynamics

Declaration of Authorship

I, M. Jafarizadeh, declare that this thesis titled, "Automated Monitoring for Data Center Infrastructure" and the work presented in all the chapters are my own.

Chapter 1

Introduction

1.1 Environmental Monitoring in Data Centers

Data centers are becoming more critical and more complex these days, which makes the need for automated infrastructure management to deal with their forthcoming operational considerations [20]. Besides traditional considerations like security and up-time, power efficiency has become a main concern in a data center (DC) as well [22]. According to the latest reports of U.S Environmental Protection Agency (EPA), 70 billion kWh equals to 1.8% of the whole electricity power of the U.S. was consumed in DCs by 2014 [82]. Up to 50% of this power is consumed for cooling of IT and communication equipment in a wasteful way [22].

The most common air cooling architectures of DCs are based on raised-floor and overhead cooling units. In both architectures, server racks are lined up in alternating rows with cold air intakes facing one way and hot air exhausts facing the other (Figure 1.1). These architectures often cause the cold air to be distributed among the servers inefficiently, regardless of their need. Detecting and estimating

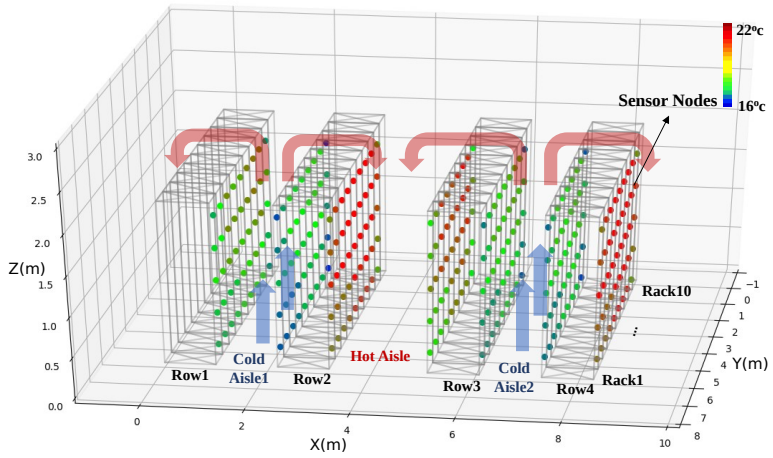


FIGURE 1.1: Thermal map of a sample DC with raised floor cooling.

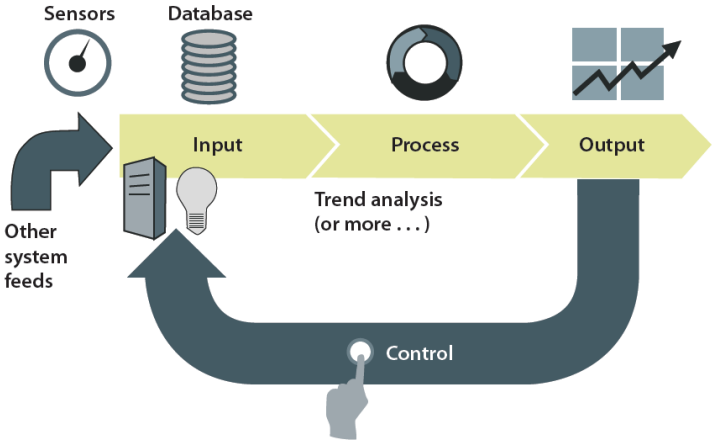
over-cooled regions or hot-spots enables the controllers and operators of Computer Room Air Conditioner (CRAC) units to take efficient thermodynamic actions. It can also support load balancers to make judicious decisions about which servers are suitable for running the upcoming tasks. Beside power efficiency, the technical bulletin [8] lists other benefits of environmental monitoring as:

- Cooling performance visualization through software
- Humidification requirements
- Floor tile tuning
- Historical data trending
- Preventative maintenance prediction
- Real-time Power Usage Effectiveness (PUE) calculation

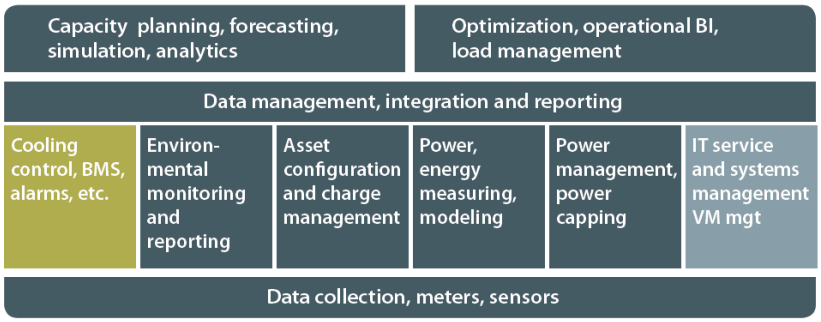
Inefficient regions in a DC are detectable based on high temporal and spatial fidelity maps of temperature, relative humidity, airflow, and other environmental parameters. The party that is responsible for providing such information varies in different DC types. This work targets *colocation DCs* (also called "Colocation" or "CoLo"), a specific type of DCs in which operators provide the required infrastructure and space for third party IT equipment (ITE) according to some rental agreements. In a colo, ITE owners are responsible for the maintenance of their own servers, while its operators need to maintain power, cooling, and the other required infrastructures to ensure proper operations of the servers. Typically, colo DC operators have no access to on-board sensors and no control over the utilization of ITE. Instead, they rely on Data Center Infrastructure Management System (DCIM) tools for monitoring the infrastructure status and planning resources [20].

A DCIM is a family of software tools that enables DC operators to provision and monitor cooling and power infrastructure in their sites. A typical DCIM provides asset management, predictive analysis, floor planning, thermal maps of ambient air temperature, and infrastructure control features along with other management capabilities through an interactive graphical user interface. Figure 1.2 shows Gartner's component diagram and 451 group's functional diagram of a typical DCIM. In Gartner's model, sensors are the first primary component that sends raw data to the other processes. Also, data collection in 451 group's model is the basis to support all other DCIM functionalities. Both models emphasize the importance of sensors.

On-board sensors (e.g., CPU temperature sensors, sensors in cooling systems)



(A) Gartner model



(B) 451 group models

FIGURE 1.2: Two models for DCIM. [20]

can be a valuable source of data for DCIM; however, these sensors alone do not provide sufficient spatial resolution to fully characterize DC thermal dynamics and component level hotspots, which may lead to early equipment failure, low energy efficiency or degraded server performance. A complimentary solution is to use extra out-of-band environmental monitoring sensors. The sensors must be non-intrusive to the servers due to both technical and non-technical reasons:

- Non-interference policy of most colo DC hosts

- Reducing the dependence to the always-ON servers

- Ease of sensor replacement, etc.

Data Center Wireless Sensor Network (DCWSN) is a cost-effective enabling technology for non-intrusive environmental monitoring in DCs. Wireless sensing saves up to approximately 75% of the labour and material costs spent for wiring [9]. This research focuses on the enhancement of the data collection process in DCIM by proposing a zero-configuration, low power, and reliable DCWSN. A part of this work also takes advantage of the capability of DCIM in controlling cooling units during commissioning the network.

1.2 Hierarchical IoT for Environmental Monitoring

Internet of Things (IoT), allows large-scale environmental monitoring by connecting low-cost sensors via low-power wireless technologies [1]. It has been applied in wide range of applications, from water quality assessment in agriculture [74], bridge displacement monitoring [39], smart power [25], and malfunction detection in industrial plants [94], to efficient thermal management in smart buildings [26] and DCs [50]. Most of these monitoring applications require dense and/or on-demand installation of thousands of wireless sensors in target environments. In DCWSNs sensors are typically installed in the front and back of every server (Figure 1.3). So, in a medium to large-scale DC, that contains tens of rows, and tens of racks per row the density of sensors can be as high as 100 per meter cube. To support this high density and to ensure data reliability, hierarchical network architecture with a wired backbone is desirable. Due to limited wireless bandwidth,

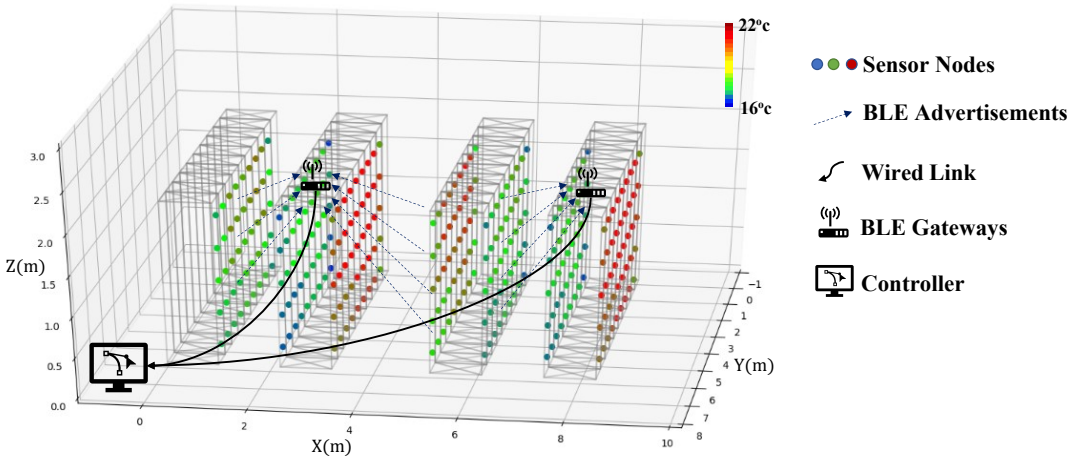


FIGURE 1.3: Hierarchical IoT architecture for dense monitoring in a sample DC.

in absence of efficient cooperation or a wired backbone in a hierarchical topology, it is well known that individual sensor's goodput in a dense network with a single sink decreases with the number of sensors n according to $\Theta\left(\frac{1}{n}\right)$ [99]. Hierarchical organization have led to some recent successful solutions to DC monitoring [21].

In this research, we advocate the deployment of a 2-tier network for DC monitoring. Multiple backbone nodes serve as the gateways between the two tiers. They gather data from sensors in the first tier and forward them through a reliable transport protocol such as TCP over wires or over a single-hop or mesh WiFi network to a central controller in the second tier (Figure 1.3). Battery-powered sensors communicate with the gateways through Bluetooth Low Energy (BLE) or LoRA for better power conservation, low deployment costs, and wide availability [50].

1.3 Challenges of Implementing a DCWSN

Despite the early successes of experimental DCWSNs for DCIM [e.g. 9, 24, 52], their adoption in DCs is sparse. From extensive discussions with our industrial partner, we believe that *low reliability*, *short battery lifetime*, *lack of adaptability*, and *burden of commissioning* are among the main roadblocks that hinder wider adoption of such solutions. These four challenges also persist in our 2-tier BLE based architecture as well.

Low reliability Legacy advertising in BLE is power efficient and fast but error-prone. It does not have any mechanism to prevent or recover from packet losses, such as collision detection, error correction code, or acknowledgment. To utilize BLE advertising in the first tier of DCWSN, improvements are required to enhance its reliability.

Short battery lifetime To reduce costs and consequently accelerate Return of Investment (RoI) a light-weight and power-efficient communication protocol is needed so that sensors can last for years without battery replacement. Although BLE PHY has the lowest power consumption among short-range technologies for personal area networks, extra power consumption due to protocol overheads can still be substantial. For example, advertisements need to be transmitted three times over the advertising channels. Transmissions using fixed transmission power level reduces energy utilization. Furthermore, maximum allowable advertise interval restricts power saving in low data rate applications.

Lack of adaptability Controllability is the main requirement to access network devices and make them programmable and adaptable. In the first tier of hierarchical architecture, data is transferred in one direction from sensors to gateways through broadcasting BLE advertisements. An alternative mechanism is needed to deliver commands to the sensors. Furthermore, proper network parameters selection algorithms are needed be developed for adapting the network to the application requirements.

Labor-intensive sensor mapping From our on-site experiences with two DCs, commissioning of the DCWSN is a labor-intensive task in practice, with much time spent on recording sensor IDs and their locations in the DC (e.g., by 4-tuple (aisle, rack, server, front/back)). This problem, called *sensor mapping* can be solved by automated localization. Despite of a myriad of wireless sensor localization solutions in literature, the prevalence of dense metal racks hosting IT equipment as well as pipes and wires in cooling and power subsystems make DCs multipath rich environments. As a result, RF-based localization solutions are not adequate for DCs, and a novel approach is needed to automate sensor mapping in the commissioning stage.

1.4 Contributions

At the core of our DCWSN solution is a 2-tier network called Low Energy Monitoring Network (LEMoNet). It uses BLE in the first tier and a reliable transport protocol such as TCP over WiFi or a wired connection in the second tier. My contributions to development of LEMoNet is reliability enhancement using *Scannable* mode for advertising, and performance evaluation by simulation.

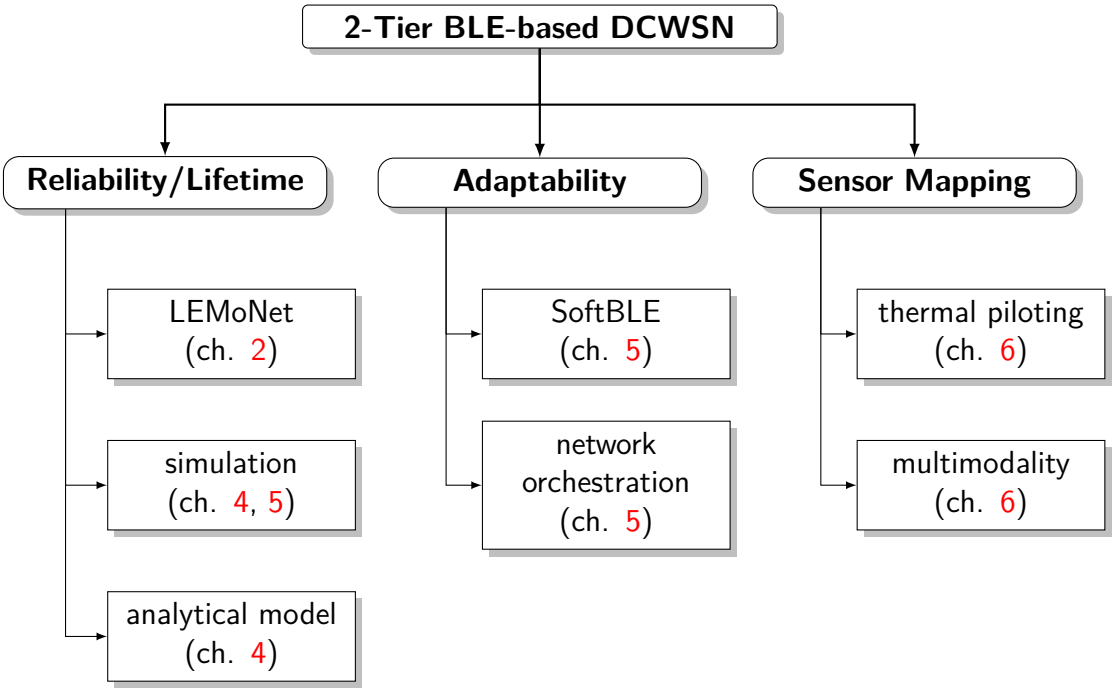


FIGURE 1.4: The relation between the challenges of developing a DCWSN and our contributions for addressing them.

The challenges of developing a DCWSN are addressed through three solutions: *performance modeling*, *network controllability*, and *fast sensor mapping*. The rest of my contributions are classified accordingly. Figure 1.4 shows the relation of each contribution with the challenges.

Performance Modeling and Evaluation at Scale We extended a network simulator to evaluate the performance of LEMoNet. Beside simulation, we also developed an analytical model to accelerate the evaluation process. For analyzing the collision probability in multi-gateway reception, a union bound approach is proposed to tackle dependencies of the gateways. The approach is verified originally in LoRaWAN [66] (another hierarchical IoT network), and then is applied to LEMoNet.

Network Controllability To address the challenge of network adaptability, we provide network controllability through a central controller by designing a framework, named SoftBLE. The framework is based on the concept of Software Defined Networking (SDN) with an out of band control plane. SoftBLE leverages recent agilities in BLE5 devices. It also comprehends two orchestration algorithms, the first one is for channel assignment to the gateways, and the second one is for selecting optimal advertising parameter for the sensors. Experimental results show that the framework can save up to 70% of the transmission power while preserving the Packet Reception Rate (PRR) above the threshold of 99.9% in almost all the sensors.

Fast Sensor Mapping To alleviate the commissioning burden, we propose *thermal piloting*, which in cooperation with DCIM can automate sensor mapping and turn it into a zero-configuration task. It maps the sensors to a given set of locations using the very beginning measurements of temperature sensors. The performance of thermal piloting is subsequently improved by incorporating pairwise Radio Frequency (RF) signals between the sensors. The network is modeled as two weighted graphs, analytical and experimental. A novel optimization-based algorithm is proposed for matching the graphs that are more accurate than selected baseline approaches. Multimodality (using both thermal and RF data) shows three times higher localization accuracy then single modal approaches.

1.5 Organization

The rest of the thesis is organized as follows. Chapter 2 reviews key aspects of the BLE protocol. Chapter 3 summarizes related work. Chapter 4 explains in detail

the base protocol, LEMoNet, and its analytical model. Chapter 5 presents the proposed SDN framework for network interoperability. Chapter 6 discusses our solutions to automated sensor mapping. Lastly, chapter 7 concludes the research.

Chapter 2

Background - Bluetooth Low Energy

Since the Bluetooth SIG introduced Low Energy (LE) features in Bluetooth v4 in 2010, Bluetooth Low Energy (BLE) has become one of the most popular low-power IoT edge technologies. By *Legacy Advertising*, BLE allows arbitrary number of concurrent advertisers in an area without the need of coordination. However, since only three advertising channels (channel 37 – 39) are available, when a large number of advertisers are in close proximity, collisions can be prohibitively frequent leading to low Packet Reception Rate (PRR) and low neighbor discovery rates [71].

In this section we first review the protocol stack, packet structure, and legacy advertising timeline of BLE, aiming to target the concepts that are required to explain the details of our main solutions. At the end, we discuss how poor performance of BLE legacy advertising can affect the PRR in a sensor network.

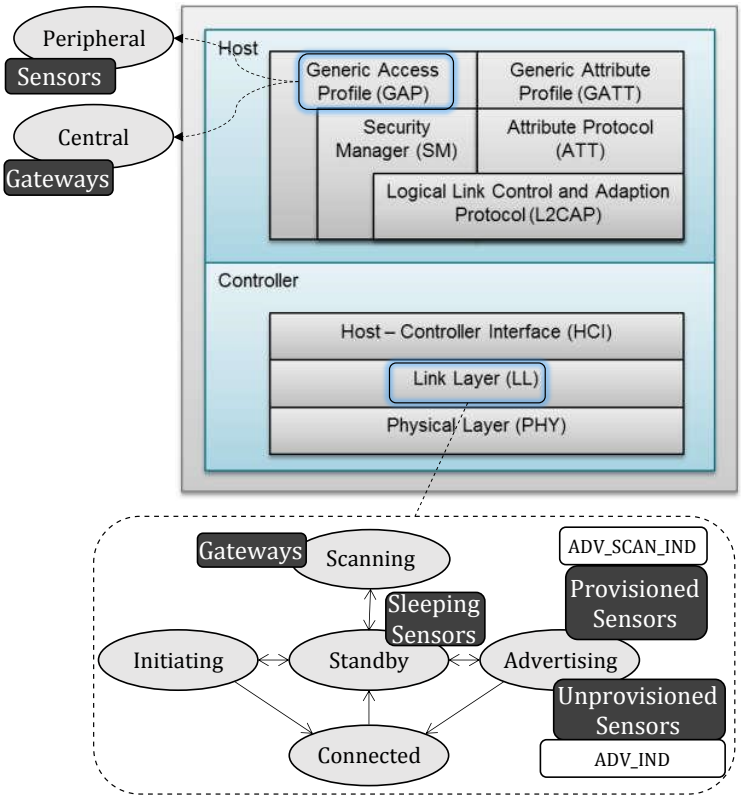


FIGURE 2.1: BLE protocol stack.

2.1 BLE Protocol Stack

As shown in Figure 2.1, BLE protocol stack is comprised of two major divisions: *Host* at the top and *Controller* underneath. The host layer includes components like Generic Access Profile (GAP) and Generic Attribute Profile (GATT). Those components are mainly responsible for organizing the profiles and defining the role of a BLE device. Based on its GAP profiles, any BLE device prior to connection establishment is assigned a role that is either peripheral or central. Peripheral devices, such as sensors, advertise their data, while central devices, such as gateways,

scan for the advertisers. The host layer is connected to the controller through Host-Control Interface (HCI). In the lower half of the stack, the controller provides interoperability between HCI and radio hardwares by implementing a physical layer a link layer.

In the *link layer*, a BLE transmitter can transit in five different states (Figure 2.1-LL): Standby, Advertising, Scanning, Initiating, and Connected. Among them, three are utilized in our base solution: sensors periodically switch between *Advertising* and *Standby*, while gateways are always in the *Scanning* state. We omit connection related states in the forwarding plane to keep the protocol lightweight and power-efficient.

BLE *physical layer* has the same data rate (1Mbps) and the same modulation (Gaussian Frequency Shift Keying (GFSK)) as conventional Bluetooth at the basic rate. Also, like its predecessor, BLE's Radio Frequency (RF) channels are in the ISM 2.4-GHz band. But, unlike 79 channels in Bluetooth, BLE uses 40 channels, 3 for advertising, and 37 for connections.

2.2 Advertising Packet Structure of BLE

As shown in Figure 2.2, a broadcasting BLE packet contains 5 bytes of a fixed header (1 byte *preamble*, 4 bytes *Access Address*), followed by 8-39 bytes of protocol data unit (PDU), that is controllable by the host. The packet is then appended with 3-byte CRC that is handled by the physical layer. Each PDU contains a two-byte header including its type code, followed by 6 bytes of advertising address

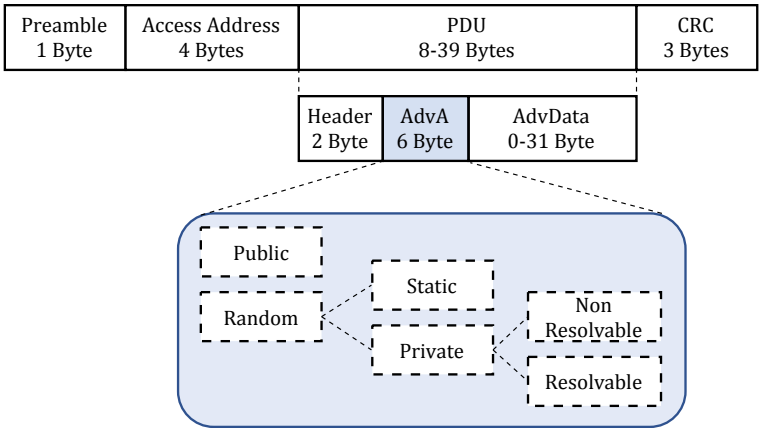


FIGURE 2.2: Broadcasting BLE packet structure.

and 0 to 31 bytes of data. In our solutions, an advertising packet PDU is of one of the following three types:

ADV_IND (code 0000b) declares that the advertiser accepts connection requests.

ADV_SCAN_IND (code 0110b) declares that the advertiser accepts scan requests.

ADV_NONCONN_IND (code 0010b) declares that the advertiser is just broadcasting and not listening for any connection or scan requests.

An active scanner responds to advertisers with the following packet types accordingly:

CONNECT_REQ (code 0101b) is a connection request identifier in response to *ADV_IND* advertisers.

SCAN_REQ (code 0011b) is a scan request identifier in response to *ADV_SCAN_IND*

advertisers. The PDU of this packet is fixed and only includes 6 bytes for the responded advertiser's address.

2.3 BLE Legacy Advertising

Legacy advertising in BLE is a scheduled and regular process. At the end of each *Advertise Interval* (T_{ai}) after a relatively small random delay (T_{rd}), advertisers broadcast their data on the primary advertising channels, namely, channels 37 – 39 by default. Recent TI and Nordic BLE devices allow selective advertising on an arbitrary subset of the three primary channels, and the information is stored in a *Advertising Channel Map*. In addition to the *Advertise Interval*, and *Advertising Channel Map*, other parameters that can be configures on a BLE advertiser at run-time include *BLE address*, and *TX power level*. Every BLE advertiser has a 6-octet address, which is exposed in BLE packet PDUs. By default, this field represents the MAC address of the device. However, depending on application requirements, it can be set to three other types of addresses, including *Random Static(RS)*, *Random Private Resolvable (RPR)*, or *Random Private Non-Resolvable (RPNR)*. The last case is an application-defined number, which can be set to an arbitrary value. TX power can be set to one of 13 predefined levels: {-21,-18,-15,-12,-9,-6,-3,0,1,2,3,4,5}dBm. The default TX power is 0dBm in most devices.

A BLE scanner listens to each channel in its *Scanning Channel Map* for a length of time defined by *Scan Window* (T_{sw}), and at the end of the *Scan Intervals*(T_{si}) switches to the next channel in the map. *Scanning Channel Map* by default includes all three primary channels, but similar to *Advertising Channel Map* in advertisers, it can be configured to any arbitrary subset of them. If the

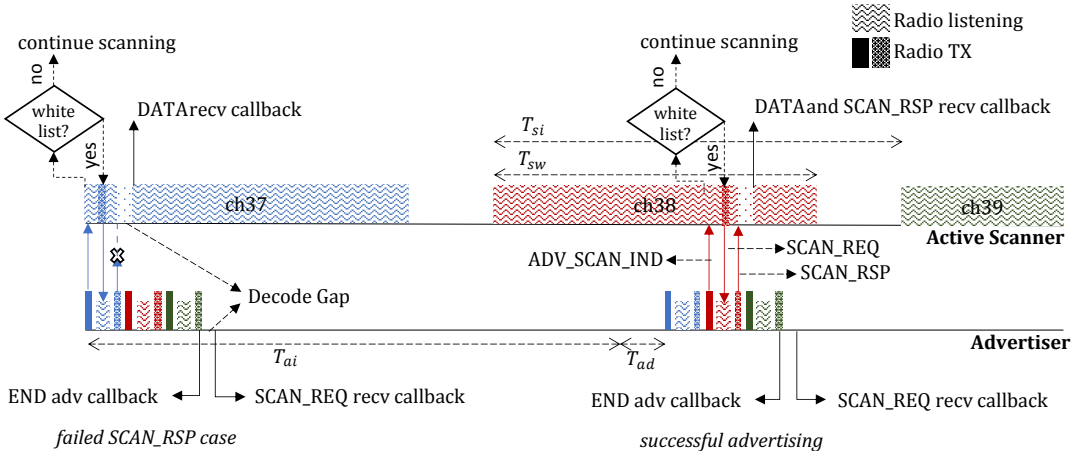


FIGURE 2.3: The time line of advertising *ADV_SCAN_IND* packets to an active scanner.

length of T_{sw} and T_{si} are equal, channel switching happens immediately with a relatively short gap. A scanner can be either active or passive. If it is passive, it only monitors and discovers advertisers in its neighborhood. An active scanner, on the other hand, responds to the *ADV_IND* or *ADV_SCAN_IND* advertisements with *CONNECT_REQ* and *SCAN_REQ* packets, respectively. In both cases, the discovery of an advertiser and the possible *SCAN_RSP* are reported to the host using a callback function. A BLE scanner can be configured to respond only to advertisers whose addresses are in its *whitelist*. Figure 2.3 shows the timeline of packet transmissions between an *ADV_SCAN_IND* advertiser, and an active scanner.

2.4 Reliability of Advertisement in BLE

BER of BLE packets can be estimated with the GFSK BER model [54] as,

$$ber = \frac{N_h - 1}{N_h} P_{e(N_h-1)} + \frac{1}{N_h} P_{e1}, \tag{2.1}$$

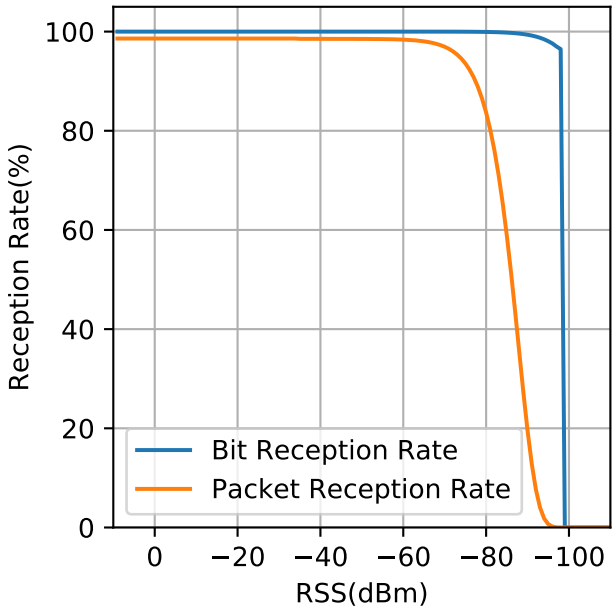


FIGURE 2.4: Bit Error Rate (BER) model in BLE and the corresponding Packet Error Rate (PER) for a 32-bytes packet. The noise level is set to 110dBm.

where P_{e1} is the error probability of the first bit in GFSK frequency hop, $P_{e(N_h-1)}$ is the error probability of the rest of the bits in that hop, and N_h is the number of bits in the hop. P_{e1} and $P_{e(N_h-1)}$ are calculated using covariance matrices as a function of the received Signal to Noise Ratio (SNR), the details of which can be found in [54]. The resulted BLE BER model along with the error rate of a sample BLE packet with 32 bytes length are shown in Figure 2.4. It can be seen that the reception rate of a BLE bit or packet drops sharply when Received Signal Strength (RSS) goes below -80dBm. (Noise level is assumed $-110dBm$).

The high bit error rate along with the lack of collision detection or acknowledgement mechanisms cause BLE to be an unreliable communication protocol for a network of multiple advertisers. Perez-Diaz *et al.* [71] analytically prove that

the rate of neighbor discovery in a network of more than 200 devices is reduced to 80% for an advertising interval of 500ms compared to the case with 20 device. To mitigate the poor performance of legacy advertising in BLE, improvements have been suggested in the later versions of the protocol, including the BLE mesh standard, and extended and periodic advertising in BLE5. However, they are not effective in improving neighbor discovery rates in dense networks. The authors of [71] repeated the same analysis with newer BLE5 enabled chips and have almost the same findings [37]. In a further study, Shan and Roh [81] show that extended advertising in BLE5 does not lead to a significant improvement in the network discovery rate. Furthermore, these features have not yet been widely implemented by vendors. Therefore, our solution depends on BLE legacy advertising, which is universally supported by BLE4 through BLE5.

Neighbor discovery rate is directly correlated with the success rate of advertisements, i.e., low neighbor discovery rate implies low PRR for the sensors. These findings motivated us to improve the performance of first tier of our network by designing acknowledgement mechanism and a central controller for parameter optimization.

Chapter 3

Related Work

This chapter discusses the related work to the proposed solutions from three perspectives. First, the previous attempts to develop practical Data Center Wireless Sensor Networks (DCWSNs) are reviewed. Next, we summarize existing work on Software Defined Networking (SDN) in Internet of Things (IoT), and controllability in Bluetooth Low Energy (BLE) networks. Third, methods for sensor localization in IoT are summarized and their practicality in a data centers (DCs) are investigated. Lastly, the solutions to Weighted Graph Matching (WGM) are presented.

3.1 Data Center Wireless Sensor Networks

The benefits of environmental monitoring have been verified in several research reports. For instance, it has been shown in [9] that the introduction of a DCWSN and subsequent changes made based on sensor measurements result in an overall energy reduction of 17% and a reduction of Power Usage Effectiveness (PUE)

Table 3.1: Specification of wireless protocols on the representative modules [50]

	WiFi 802.11b	**Zigbee**	**BLE**
Module	ESP8266	TI cc2520	TI cc2640r2f
Data rate	11 Mbps	250 kbps	1, 2 Mbps
TX power	17 dBm	5 dBm	5 dBm
Sensitivity	-91 dBm	-98 dBm	-97 dBm
TX current	170 mA	33.6 mA	9.1 mA
RX current	56 mA	18.5 mA	5.9 mA

from 1.94 to 1.51. The payback time of deploying such a Wireless Sensor Network (WSN) is 3.4 years. Similar favorable results have been reported elsewhere [24]. CapNet [77] is a real-time DCWSN for power capping for DC management. It uses distributed event detection to eliminate the overhead of regularly polling all nodes in the network. In another work, Chen et al. [17] demonstrated that using the real-time sensor data to calibrate a Computational Fluid Dynamics (CFD) model, temperature evolution of servers with highly dynamic workloads can be forecast at an average error of 0.52^{o}C, within a duration up to 10 minutes. Due to their promises, a lot of efforts have been devoted to the development of prototypes for DCWSNs. This section reviews these efforts from two perspectives: communication protocol design, and evaluation techniques.

3.1.1 Protocol Design

802.15.4 is one of the most popular wireless protocols for DCWSN. Researchers at Microsoft Research designed RACNet, a large-scale sensor network for high-fidelity data center environmental monitoring [52]. It uses a token-passing mechanism to provide network-wide arbitration for data collection that has been experimentally

demonstrated to yield data that is 99% or higher. Sensors in RACNet are daisy-chained to master nodes, which are in turn connected via a mesh over 802.15.4 protocol. The wiring overhead for sensors will increase the labor cost for installation and hamper changes in the network. Chen et al. [17] deployed temperature and airflow sensors to monitor inlet and outlet server temperatures and Computer Room Air Conditioner (CRAC) units. All sensors run TinyOS and form a single-hop network using 802.15.4. The cluster area sensor network (CASN) [38] comprises TelosB sensor nodes running TinyOS attached to compute servers or workstations. It verifies the server's physical presence through wireless cluster-wide command dissemination, and thus enhances the security of DC management. In CapNet [77], sensor nodes use a single IEEE 802.15.4 channel for communication inside a cluster, where the transmission schedule is slotted and coordinated by a power-capping manager.

Unlike the aforementioned prototypes that are based on Zigbee or WiFi, our choice for short range communications is BLE, which outperform them in power consumption as shown in Table 3.1. This out-performance is desirable since even small improvements in sensor nodes power consumption can extend the network lifetime of DCWSN in the order of months or years.

3.1.2 Performance Evaluation Techniques

To our knowledge, almost all existing DCWSN solutions are evaluated only in experimental DCs. In [58], a network of 588 SynapSense sensors are installed in a 1,200m^2 DC with 16 temperature and 16 humidity sensors on a CRAC unit, and 420 tempreture sensors on top, middle, and bottom of the racks. Chen et al. [17]

deployed 35 temperature and 4 airflow velocity sensors to monitor five racks in a DC row. For the evaluation of RACNet [52], 174 wireless Genomotes are deployed in a 1,100m^2 production site. Each Genomote collects data from three to four daisy-chained wired sensors every 30 seconds and forward them to the controller using up to four channels in 802.15.4. CapNet [76] was tested with an experiment over 81 TelosB motes (1 for the manager, 80 for the servers) in a Microsoft DC in Redmond, WA. Our base protocol, Low Energy Monitoring Network (LEMoNet), was also evaluated first in an experimental testbed [50]. 58 sensors along with 3 gateways were deployed on the racks of a high-performance computing facility. Although testbed experiments are the necessary first step to verify any DCWSN solution, they are often limited in scale and time duration. Conducting long-term experiments in large scale DCs is difficult and costly. Alternatives are simulation and analytical methods.

Simulation is the most common evaluation technique for WSN and IoT protocols [18]. There are variety of network simulators for different IoT research domains. Some simulators like CupCarbon [60] (used for node mobility in smart cities) or Cooja [68] (used for simulating Contici OS based devices) target a specific domain, while other simulators like OMNET++ [91] and NS3 [36] are generic and multi-purpose. Despite the difference in application domain, all aforementioned simulators are Discrete Event Simulator (DES), which means that every event is queued and processed serially. We also have leveraged a DES for scalability validation of LEMoNet. For this purpose, we have extended a current implementation of BLE in OMNET++ and simulated a network of 1440 sensors and 12 gateways [50]. Comparing to experimental testbed, simulation is more affordable for scalability

analysis, however, one single round of simulation in a DES can take up to several days or even weeks to complete.

Besides experimental testbed and simulation, analytical modeling is another performance evaluation technique that has been widely developed for studying well-known MAC layer protocols such as WiFi, BLE [44], and Zigbee [48]. There have been many efforts in studying WSNs analytically as well. These efforts began since the early days of WSN with studies like [19] where Chiasserini and Garetto analyzed a duty-cycled WSN with sleeping nodes. More recently Nageswari et al. [65] analyzed clustered WSNs using energy-efficient N-policy (EENP) models. or in [73], the end-to-end routing delay in WSNs is modeled by Ramesh and Kannan. Though modeling techniques for general WSNs can be extended to DCWSNs, the unique design of LEMoNet such as multi-gateway reception warrants the development of a new analytical model.

3.2 Network Controllability

This section reviews two categories of related work for network controllability. The first category includes the recent SDN developments in IoT edge networks, and the second category consists of the studies on BLE parameter tuning.

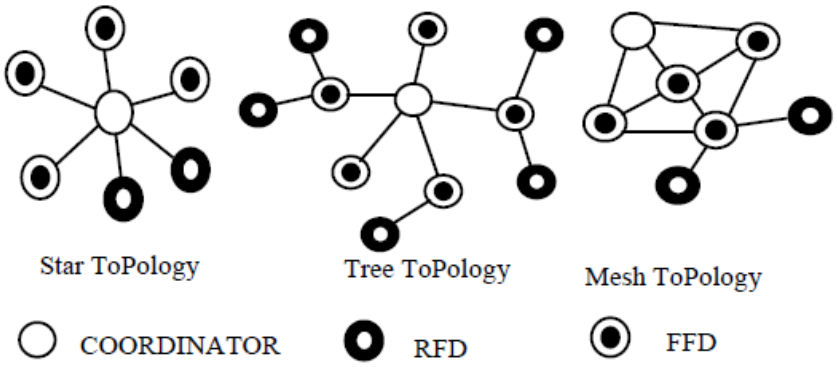
3.2.1 SDN in IoT Edge

In the last decade, SDN controllers have become popular in IoT edge to enhance the efficiency of data transmission protocols[11]. Zheng *et, al.* [98] formulate the task scheduling of sensor nodes as a mixed-integer linear programming (MILP)

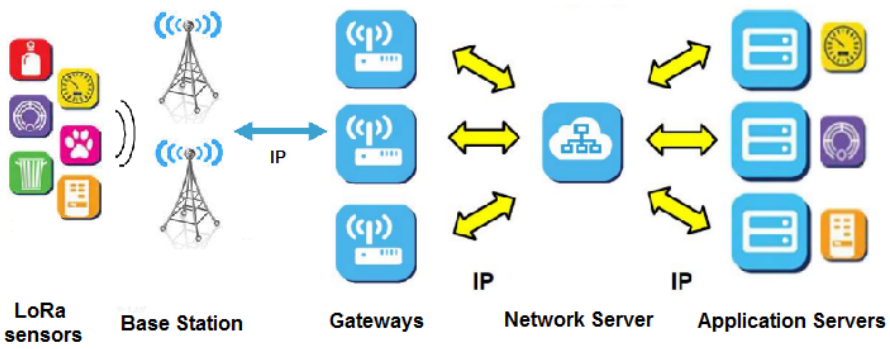
problem, with the objective of minimizing the energy consumption of sensor nodes in a multi-task software-defined sensor network. Li *et al.* [51] utilize path difference degree (PDD) in an SDN-based Industrial IoT (IIoT) network to find the optimum flow path with respect to time delay and goodput. In [93], the authors propose an SDN based framework to prioritize IIoT tasks based on their real-time performance to decide whether a task should be offloaded to a fog server or a cloud server. All these approaches are generic and evaluated by simulations, based on abstract physical connectivity models. They fail to account for the intricacy of wireless standards such as Zigbee, LoRaWAN, or BLE and demonstrate the effectiveness of their approaches in real-world testbeds.

Although Zigbee and LoRaWAN do not explicitly incorporate SDN, they utilize coordinators and network servers as central controllers (Figure 3.1). A Zigbee Coordinator is responsible for bootstrapping its network by selecting a Personal Area Network (PAN) identifier, and an operating channel for 802.15.4 based devices. Taking advantage of such capabilities, several work introduces SDN improvements for 802.15.4 based networks, including Atomic-SDN [6], SDN-based topology management [10], or WISE-SDN [30]. In LoRaWAN, network servers have a wider range of responsibilities, from security features such as authentication and encryption to routing and data rate adaptation. In [64], the authors propose to deploy a distributed version of SDN controllers at the edge servers to reduce the load in the core network that connects LoRa access networks.

In contrast to Zigbee and LoRaWAN, BLE does not feature any built-in module as coordinator or controller. The concepts of central controller and SDN have been introduced in two recent works on BLE networks. In an early work, Uddin



(A) Zigbee network architecture [96]
FFD: Full Functional Device (Router)
RFD: Reduced Functional Device (End Device)



(B) LoraWAN network architecture [23]

FIGURE 3.1: The role of coordinator in Zigbee and Network Server in LoRaWAN.

et al. [89] use an SDN-based architecture to scale up a network by adding a programmable BLE service switch (BLESS) in between the connected devices. However, the establishment and termination of BLE connections in this work impose significant delay and extra power consumption to the network. A SDN framework is also proposed in [63] to reduce network congestion in BLE mesh networks. However, its evaluation is only limited to a network of 12 nodes. As mentioned in Section 1, BLE mesh has not been widely supported by BLE chip manufacturers, and thus is of limited relevance. Unlike these two recent studies, LEMoNet utilizes

legacy advertising to avoid connection overhead and can be implemented on most BLE commercial-off-the-shelf devices.

3.2.2 Parameter Configuration of BLE Networks

After designing an SDN based framework for SoftBLE, the next problem is how to find and set proper parameters for BLE devices. There have been several recent studies dedicated for parameter assessment of BLE networks. For instance, Luo *et al.* study the effect of legacy advertising parameters on energy consumption [55] and the neighbor discovery latency [57] of network nodes. They find that the scanning window time of BLE scanners should be set to its maximum possible value (10.24s) for the maximum efficiency. Inspired by the findings of these studies, we adopt this setting in our work.

Other researches have taken one step further and tried to optimize the parameters of BLE networks. Song *et al.* [85] propose a mechanism for parameter negotiation between BLE devices to reduce neighbor discovery latency. The feasibility of their solution is validated for scenarios with no more than 100 advertisers. Other work considers the optimization of BLE scanning intervals [16], advertising interval [56], or both [53]. Comparing to these works that deal only with interval settings, the proposed solution for controllability in Chapter 5, named SoftBLE, considers more control parameters, such as TX power, advertising address, and recently added features such as scanning and advertising channel maps. Furthermore, unlike above studies, SoftBLE enables tuning of the parameters based on the statistical information observed from the target network.

3.3 Automated Sensor Mapping

In this section, we review two lines of related work for automated sensor mapping. In the first line, unimodal methods for localizing IoT devices are discussed. In the second line, we review existing algorithms for the WGM problem, which is at the core of the proposed multi-modal localization solution in Chapter 6.

3.3.1 Localization in IoT

From the methodology perspective, our solution is similar to a group of localization techniques that are based on optimization. These methods are classified based on five criteria [62]: single-hop vs multi-hop, range-free vs range-based, anchor-free vs anchor-based, static vs dynamic, and centralized vs distributed. The proposed multimodal method in chapter 6 is a single-hop, semi-range based, anchor-free, static and centralized approach.

Range-free methods are mainly depend on hop counts [70]. In contrast, for range-based methods, different features of Radio Frequency (RF) signals such as Time of Arrival (TOA) [33, 31], Time Difference of Arrival (TDOA) [40], Received Signal Strength Indicator (RSSI) [61], Angle of Arrival (AOA) [95], or a hybrid of them [88, 95] are utilized to calculate the distances between nodes and localize them. For DC monitoring applications, sensors are installed in high density with an average distance between them as small as 0.5 meter. TOA and TDOA methods are infeasible with narrow-band radios such as BLE at such a scale. Besides, the existence of racks, pipes, ducts and other metal infrastructures in DCs make it a harsh environment for radio propagation. Inferring absolute distances

based on Received Signal Strength (RSS) tends to be highly inaccurate. The proposed method utilizes RSS information similar to some range-based approaches. However, it mitigates the limitations due to complex wireless channels in DCs by incorporating measurements from temperature sensors.

Sensor localization algorithms in IoT can also be classified into anchor-free and anchor-based approaches. Anchor-free methods do not rely on any extra infrastructure [62]. However, in the grid topology of DCWSNs, it is impossible to discriminate the true result in absolute coordinate from the rigid transformations of the result in relative coordinate. Anchor-based methods [7, 43, 92, 49, 84] solves this challenge using the location of a subset of sensors or some stationary anchors. For this purpose, they transform the result from relative coordinate to absolute coordinate by graph transformation techniques such as Procrustes analysis [7] or Helmert transformation [43]. The localization result in relative coordinate can be extracted with or without the help of anchors. In the later case, the relative node positions are calculated by dimensionality reduction of a graph, whose nodes correspond to sensors and edge weights correspond to the measured distances between the sensors. The dimensionality of the graph is reduced to 2D or 3D space mostly using manifold learning methods such as MDS [72], LLE [75], ISOMAP[92], PLS-ISOMAP [49], and LLE-ISOMAP [84].

Similar to manifold learning based methods, our proposed multimodal approach in Chapter 6 also takes distances between the sensor nodes as an input. Though, our method does not rely on anchors to transform the relative result into absolute coordinate. Instead, it leverages an additional data source to tackle the rigid transformations issue. To ensure uniqueness of the result in relative coordinate,

the proposed multimodal approach imposes pointwise constraints to the problem using location-dependent temperature sensor measurements.

Furthermore, there is a key difference in the problem definition of automated sensor mapping in DCs and the sensor localization problem in WSNs. In the former case, sensors can only be at a set of known locations, while in the latter case, the search space is continuous and sensors can be localized anywhere. In other words, instead of searching among continuous locations in a target area, the goal is to find mapping between the candidate locations and sensor nodes. Therefore, unlike the manifold learning methods like ISOMAP where the input is distances of the nodes in one graph, the of proposed multimodal approach is formulated as WGM of two input graphs. One graph is constructed based on the measurements same as ISOMAP, but the additional one is analytically estimated based on the distances of the given potential sensor locations.

3.3.2 Weighted Graph Matching

Given two graphs G_M and G_D, the goal of graph matching is to find a one-to-one mapping between their vertices such that a certain distance (dissimilarity) measure is minimized. Different distance measures have been investigated in literature including isomorphism for exact matching, and Graph Edit Distance (GED) for inexact matching [28]. WGM is a specific type of graph matching problems, whose objective is minimizing the corresponding edge weights differences between the two input graphs. If the vertices in the graphs are also weighted, the objective function includes both pointwise and pairwise distances. In this case, the problem is called labeled WGM, which can be formulated as Quadratic Assignment Problem (QAP)

[97]. There are three main categories of WGM algorithms: those based on relaxation to continuous optimization, spectral methods and direct search methods. An extensive 10-year survey of WGM algorithms is provided in [28].

To relax inexact graph matching to a continuous optimization problem, different techniques can be applied including *Binary Linear Programming* [45], *Graduate Assignment* [15] and *Expectation Maximization* [79]. The main source of errors of such methods arises from relaxation. The closest discrete solution to the global minimum point in a continuous space is often not even a local minimum in the original objective. Almohamad, et al. [2] initially developed the idea of converting WGM to a convex function in continuous domain. The idea was extended later to a path following algorithm by Zaslavskiy et al. [97] by gradually pushing the minimum of the convex function to the edges of a concave function. Despite the improvements in this convex-concave optimization and later through other enhancements like Factorized Graph Matching [101], discretization errors cannot be completely eliminated.

In contrast, spectral methods rely on the fact that the eigenvectors of the adjacency matrix of a graph remains unchanged when the order of its vertices are permuted. The first and most representative spectral algorithm was proposed by Umeyama [90]. In this method, the absolute value of the eigenvectors of two given graphs are aligned by the Hungerian algorithm in $O(N^3)$ time. Since then, many studies have been dedicated to improve the spectral solutions [13]. However, spectral methods are highly sensitive to noise in the input graphs [12]. Moreover, when the graphs allow rigid transformations, some eigenvalues are zero and the number of non-identical eigenvectors will be less than the number of vertices in

the graph. In this case, the problem becomes under-determined where multiple solutions exist.

The third category of graph matching methods are based on direct searches of discrete search spaces [28]. Most methods consider GED based objectives or exact graph matching. Such algorithms do not suffer from suboptimal discretization as in the relaxation methods and are more resilient to noise compared to spectral methods. However, when the number of evaluations is large, their time complexity increases drastically comparing to the previous two categories of methods.

As will be discussed Chapter 6, in formulating sensor mapping as a labeled WGM problem, we have to handle noisy measurements, imprecise model predictions, non-rigidity of sensor placements and potentially high computation complexity in similarity evaluation. A new WGM method is warranted to deal with these challenges.

Chapter 4

Analysis of Low Energy Monitoring Networks

4.1 Introduction

As mentioned in Chapter 1, a hierarchical Bluetooth Low Energy (BLE)-based network, named Low Energy Monitoring Network (LEMoNet), is the basis for this research. Details of the network protocol are reviewed in the next session. LEMoNet has been evaluated in a high-performance data center (DC) with 60 sensors and studied thoroughly by simulations in a scenario of 360 sensors. However, further evaluations of its scalability for large-scale enterprise and cloud DCs face challenges in either prohibitive costs in deployment or excessive computation time. [1]

As a fast and low-cost alternative solution, we develop an analytical model

[1] a single round of discreet event simulation for a network with more than one thousand sensors takes days or weeks to finish in OMNET++.

that can quantify the performance of a network with 5000 sensors in less than one minute on a regular desktop PC. The proposed model takes network parameters as inputs and solves a system of nonlinear equations to compute the Packet Reception Rate (PRR) and the average power consumption of each sensor. In contrast to existing models for wireless sensor networks that typically assume random placement of sensor nodes, the proposed model can handle regular or arbitrary network topologies. For small-scale networks, the predictions of the model is corroborated by simulations. For large-scale networks, analytical results demonstrate that with only a slight increase in the battery power consumption of sensors, LEMoNet is able to achieve 99.9% PRR.

In the rest of the chapter, we first review the details of LEMoNet architecture and protocol, and then discuss the details of its analytical modeling. This chapter is concluded by the evaluation results from the model.

4.2 Overview of LEMoNet

LEMoNet is a hierarchical and duty-cycled protocol, proposed for Data Center Wireless Sensor Network (DCWSN)s. The architecture and design of the protocol are reviewed in this section. Further details can be found in [50].

4.2.1 Multi-Gateway Packet Reception

LEMoNet consists of two tiers, where sensors advertise their measurements to nearby gateways via BLE in the first tier, and the gateways forward them to a controller in the second tier. The second tier utilizes reliable transport protocols

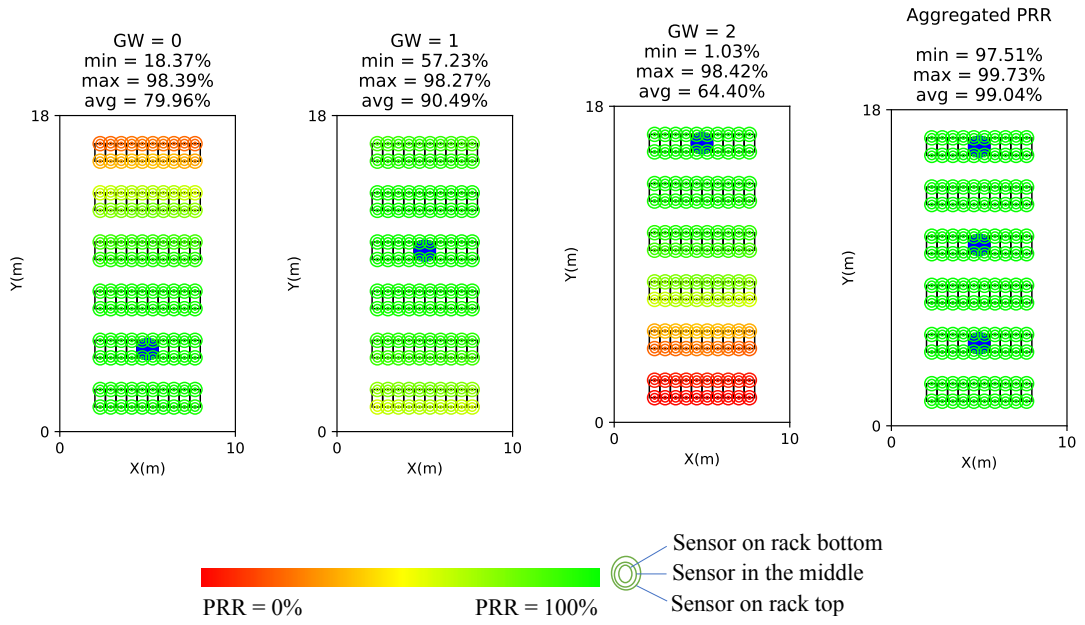


FIGURE 4.1: Comparing PRR per gateway versus aggregated PRR by simulating a sample DC with 60 racks, monitored with 360 sensors. Blue filled circles are the gateways.

such as TCP over Ethernet or WiFi (mesh). Therefore, as long as any of the gateways receives a sensor measurement, it is guaranteed to be delivered correctly to the controller. LEMoNet is designed such that every sensor is reachable by more than one gateway, which increases the chance of correct packet reception exponentially. Multi-gateway packet reception can mitigate the unreliable BLE advertising service in the first tier.

To understand the effect of multi-gateway packet reception, a simulation has been conducted. The simulation scenario is a DC with 6 rows and 10 racks per row that is monitored by 360 sensors and 3 gateways. Each rack is monitored by 6 sensors, 3 sensors installed in front and 3 at the back of the rack. In Figure 4.1, the PRR of the sensors in each gateway is compared to the aggregated one in the

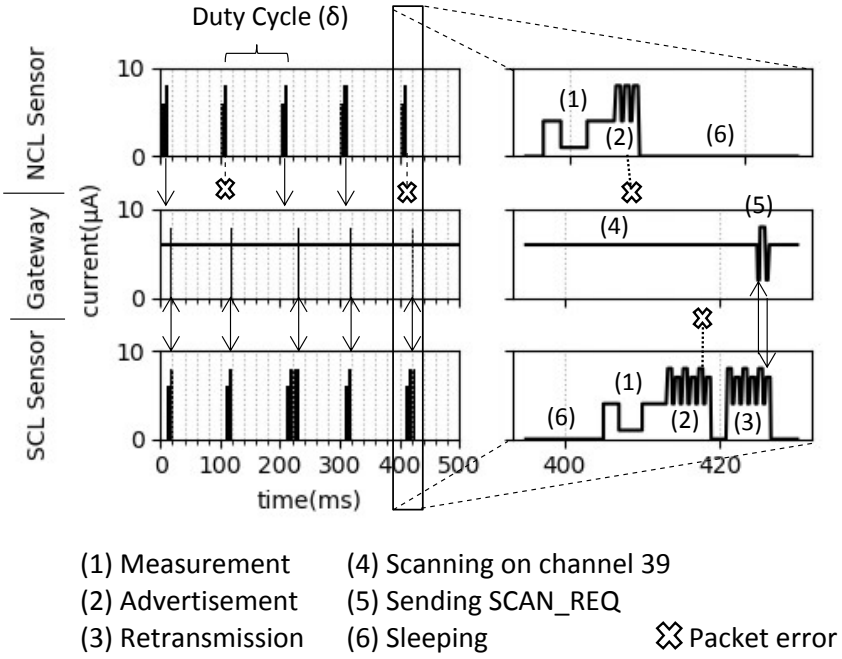


FIGURE 4.2: Energy consumption of measurement and advertising in LEMoNet using Normal Connectionless (NCL) mode (top row) versus Scannable Connectionless (SCL) mode (bottom row).

controller. It can be observed that multi-gateway packet reception can increase the maximum PRR from around 98.3% on individual gateways to 99.7% when aggregated.

4.2.2 NCL vs SCL nodes

Activities in LEMoNet are scheduled according to a preconfigured duty cycle. At the beginning of every cycle, sensors collect measurements and depends on the level of urgency choose between two modes to advertise them: *NCL* or *SCL*. If the measured data is inside a pre-defined normal range, sensors use the NCL mode and the data is advertised only once through a ADV_NONCONN_IND packet. The NCL mode is power efficient but does not guarantee packet delivery. If the measured

data exceeds the normal range or if the payload is urgent, sensors switche to the SCL (scannable) mode. In this mode, a sensor uses an ADV_SCAN_IND packet, and after each advertisement waits to be acknowledged by a SCAN_REQ. It repeats the transmission of advertisement packets until any of the gateways in range acknowledges it. Figure 4.2 compares the activity timeline and the power consumption of a sensor in the NCL and SCL modes. By introducing the SCAN_REQ listening period and possible re-transmissions, a sensor trades off extra power consumption for better reliability.

4.3 Performance Modeling of LEMoNet

The performance of LEMoNet is modeled by a system of non-linear equations. For tractability, approximations are introduced by taking some simplified assumptions. The equations are then solved numerically until the result converges.

4.3.1 Simplified Assumptions

The analysis makes several simplified assumptions regarding Radio Frequency (RF) signal propagation, packet reception under interference, and independence of transmission times for better tractability.

Log Distance Path Loss Model With 0dBm for transmission (TX) power of all the sensors, the average Received Signal Strength (RSS) from a sensors s at gateway g at distance d_s^g is estimated based on the log-distance path loss model [86],

as:

$$rss_s^g = -10\alpha \log_{10}(d_s^g) + \beta, \tag{4.1}$$

where α is the path loss exponent and β is a constant depending on the transmitter and receiver antenna gains, and the central frequency of the transmitted waveform.

Protocol Model for Interference The system utilizes the *Protocol Model* introduced in [32] to determine *Interference Ranges* for every pair of sensors and gateways. In the protocol model, any concurrent transmission in the interference range will cause a packet loss regardless of the RSS of the colliding node. Interference range is a disk area whose diameter is defined based on the TX power of the sensors and the sensitivity of the gateway.

Base on the definition of interference ranges we define *Interferer Counter*(I^χ), as the number of the nodes that can cause a collision in any of the gateways in a given set χ. In the protocol model, I^χ is formally defined as:

$$I^\chi = \sum_{g \in \chi} \sum_{s \in \mathcal{S}} \begin{cases} 1 & \text{if } rss_s^g > P_{sen} \\ 0 & \text{otherwise} \end{cases}. \tag{4.2}$$

Interferer counter will later be used for estimating the packet collision probability of each sensor node.

Independent Packet Arrival Time Due to the random delay before advertisements (T_{rd}), data transmission of the sensors are assumed to be independent at

Table 4.1: The parameters and constants of the system

Name	Description	Default
	Dimensional Parameters	
s_x	Rows	6
s_y	Rack per row	10
s_z	Sensor per rack	6
g_x	Gateway per row	1
g_y	Row per gateway (when $g_x = 1$)	3
$\mathcal{S}$	Set of all sensors	
$\mathcal{G}$	Set of all gateways	
N	$\|\mathcal{S}\| = s_x * s_y * s_z$	360
M	$\|\mathcal{G}\| = s_x * s_y * s_x/g_y$	2
	TX Parameters	
α	Path loss exponent	4.7
β	Path loss constant	40dB
	Data Rate Parameters	
δ	Duty cycle	15s
$\|PDU\|$	Bit length of the PDU	26*8
r_{scl}	The rate of SC mode sensors	0.1
R	Maximum re-transmissions	10 (3 in SoftBLE)
	Constants	
S_{rack}	Standard rack size (X*Y*Z)	2*1*0.6m
$\mathcal{N}$	Background noise	-110dBm
Th_{col}	Collision threshold	5dB
P_{sen}	Sensitivity of BLE receivers	-120dB
μ	BLE bit rate	1Mbps
e_{nc}	Energy for one NCL advertisement	75μJ
e_{sc}	Energy for one SCL advertisement	105μJ
e_{ret}	Energy for one SCL re-transmission	51μJ
$\|HEADER\|$	Bit length of advertising packet header	16*8

each duty cycle. Furthermore, the back-off mechanism in scanning (on gateways) [37] along with clock drifts avoids repeated collisions.

4.3.2 System of Equations

The performance of the sensor nodes in LEMoNet is modeled as a system of non-linear equations. The properties of the network are defined by three sets

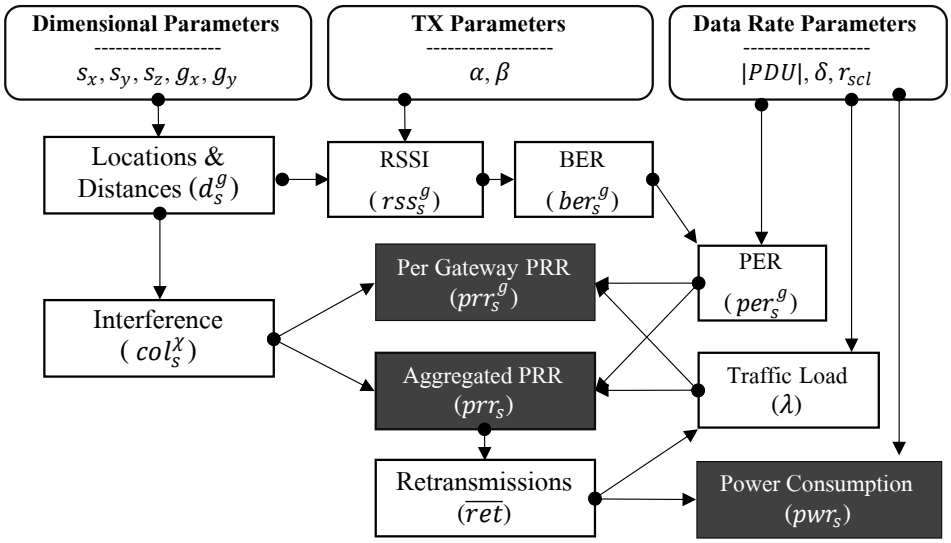


FIGURE 4.3: Block diagram of the analyical model for LEMoNet. The white rectangular boxes are nonlinear equations and the shaded boxes are the outputs.

of inputs: dimension, transmission, and data rate parameters. The definition of these parameters and their default values are listed in Table 4.1. The model takes these parameters and yields three outputs: the PRR of each gateway, the PRR of each sensor, and the average power consumption of all sensors. The block diagram of the relations between the input parameters and the outputs are represented in Figure 4.3.

Distances (d_s^g), RSS (rss_s^g), Bit Error Rate (BER) (per_s^g), and Packet Error Rate (PER) (per_s^g) Using the dimensional parameters as input, the location of the sensors and the gateways can easily be extracted based on the geometry of an standard DC. Since the geometrical equations for their calculations are trivial we skip the details. The locations are used to calculate the distances of every pair of sensor $s \in \mathcal{S}$ and gateway $g \in \mathcal{G}$ as d_s^g. Given the distances,

rss_s^g is calculated based on the log-distance path loss model (4.1). Consequently, the corresponding BER, denoted by ber_s^g, is estimated using Signal to Noise Ratio (SNR) ($rss_s^g - \mathcal{N}$) and the BLE BER model (2.1). The BER is then utilized to calculate PER as,

$$per_s^g = 1 - (1 - ber_s^g)^{(|PDU|+|HEADER|)} \tag{4.3}$$

PRR per Gateway (prr_s^g) and Aggregated PRR (prr_s) Since advertisements from a sensor can be received by more than one gateway, a packet is correctly delivered if any of the gateways in the sensor's interference range receives the packet successfully. Therefore, the expected aggregated PRR of a single transmission from sensor s that is covered by m gateways, denoted by prr_s, can be determined from the union bound of PRRs at the potential receiver gateways as:

$$prr_s = \sum_{i=1}^{m} prr_s^{\{i\}} - \sum_{i<j=1}^{m} prr_s^{\{i,j\}} + \sum_{i<j<k=1}^{m} prr_s^{\{i,j,k\}} - \cdots + (-1)^{m-1} prr_s^{\{1..m\}}. \tag{4.4}$$

where prr_s^χ is the probability of packet reception in all the gateways in set $\chi \subseteq \mathcal{G}$. Under the assumption of independent bit error among gateways, prr_s^χ can be calculated as a function of PER and *collision* probabilities:

$$prr_s^\chi = (1 - col^\chi) \times \prod_{g \in \chi} (1 - per_s^g)\,. \tag{4.5}$$

Collisions in Multiple Gateways (col^χ) Unlike BER, collisions at neighboring gateways are dependent on each other. Consider an example in Figure 4.4. In the example, the interference ranges of a target sensor, which is covered by three

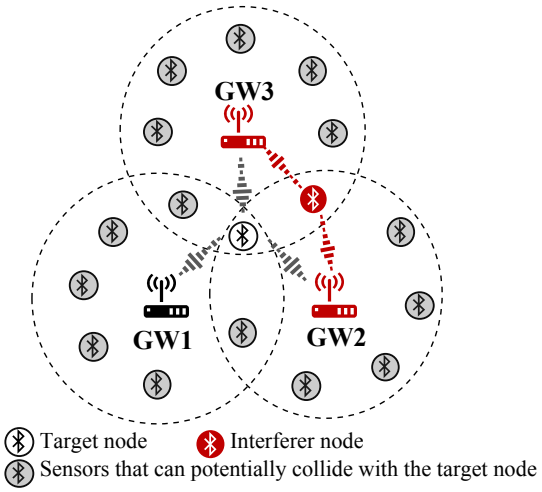


FIGURE 4.4: Simultaneous interference in a sample multi-gateway reception scenario with three gateways.

gateways, overlap with one another. As a result, the collision in one gateway that is caused by a sensor in the intersection area will cause a collision in a neighboring gateway too. We have studied the effect of multi-gateway packet reception for LoRaWAN, where the nodes are uniformly distributed, and the gateways form a circle in [66]. But a DCWSN does not generally follow either of the two assumptions. Instead, the locations of sensors and gateways are arbitrary but all are known. As a result, we can count the number of sensors in the range of all the gateways in χ using the principle of inclusion-exclusion. For instance, for $\chi = \{g_1, g_2, g_3\}$, we have:

$$I^{\chi} = \sum_{i=1}^{|\chi|} |I^{\{g_i\}}| - \sum_{i<j=1}^{|\chi|} |I^{\{g_i\}} \cap I^{\{g_j\}}| + |I^{\{g_1\}} \cap I^{\{g_2\}} \cap I^{\{g_3\}}| \tag{4.6}$$

Using ALOHA based collision analysis [87], the joint collision probability in χ can be approximated as:

$$col^{\chi} = e^{\left(-2I^{\chi} \cdot \lambda \cdot \overline{ret}\right)} \tag{4.7}$$

Retransmissions($\overline{ret}$) and Traffic Load (λ) Nodes operating in the SCL mode attempt to send each lost packet up to R times. The expected number of *re-transmissions* of node s is given:

$$\overline{ret} = \sum_{k=1}^{R} (1 - prr_s)^k prr_s \tag{4.8}$$

These number of *re-transmissions* increase the *traffic load* in the network as,

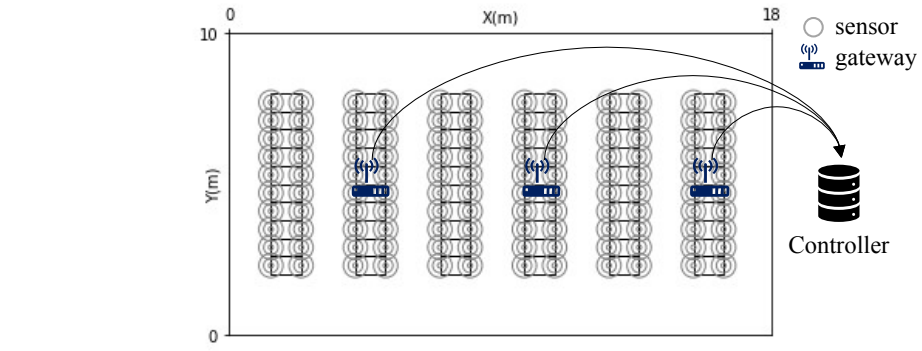
$$\lambda = \frac{(|PDU| + |HEADER|) \cdot (1 + r_{scl} \cdot \overline{ret})}{\delta \cdot \mu} \tag{4.9}$$

Power Consumption($\overline{pwr}$) Lastly, the average power consumption can be computed by the weighted average of the power consumption of NCL nodes and SCL nodes, as,

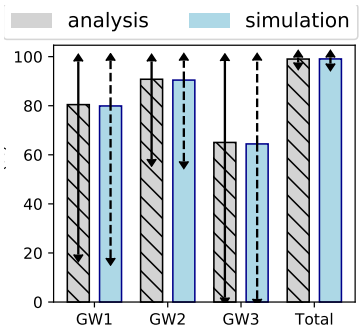
$$pwr_s = \frac{1}{\delta}\Big(r_{scl}(e_{sc} + \overline{ret} \cdot e_{ret}) + (1 - r_{scl})e_{ncl}\Big) \tag{4.10}$$

The average power consumption depends on the energy consumption of each transmission, the percentage of SCL nodes, and the expected number of retransmissions.

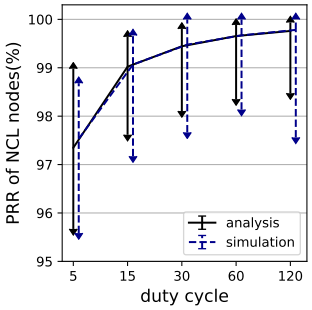
Solving the System of Equations As shown in Figure 4.3, more *traffic load* results in higher *interference* and accordingly less *aggregated PRR*, which in turn leads to more *re-transmissions*. Thus, equations (4.4), (4.8), and (4.9) need to be solved iteratively until prr_s and $\overline{ret}$ converges.



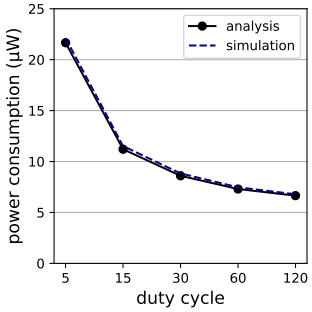
(A) Network layout



(B) Per gateway PRR



(C) NCL nodes PRR



(D) Power consumption

FIGURE 4.5: Comparison between the performance evaluation results [50] of simulation and the analytical model of LEMoNet. The network includes of 360 sensors deployed in a DC of 60 racks. The error bars represent the highest and the lowest values among the sensors.

4.4 Performance Evaluation

We first validate the analytical model and then use it to study the impacts of different parameters on LEMoNet in large-scale DCs.

4.4.1 Validation Study

The fidelity of the analytical model is first evaluated by comparison against the simulation results of a medium-scale DC with 6 rows and 60 racks. As shown in Figure 4.5a, each rack is assumed to be monitored with six sensors, three installed

TABLE 4.2: The computation time of simulation versus analytical model for evaluating a network of 360 sensors

Method	**Computation Time**
Simulation	>5h
Analytical Model	<1m

in front and three on back. Also three gateways are installed on top of the middle rack of every other row, forwarding the sensor measurements to a central controller. This scenario is simulated in OMNET++, and also analysed through the proposed model.

The results in Figure 4.5 demonstrate a very good agreement in their estimated values for per gateway PRR, aggregated PRR, and nodal power consumption of sensors. Even though the simulation and analytical methods use different interference models (*physical model* in the simulations versus *protocol model*, in the analysis) their outputs are almost identical. However, the analytical approach has a considerable advantage over simulation in computation time. As shown in Table 4.2, the analytical model could calculate the network performance in less than a minute while simulation took hours to be accomplished.

4.4.2 Parameter Assessment

Three key parameters affect the power consumption and PRR in LEMoNet, namely, node density, duty cycle, and the percentage of sensors in the SCL mode. We next study the effects of these parameters using the proposed analytical model in a large scale DCWSN. In this scenario, as shown in Figure 4.6, 1440 sensors are deployed on 240 racks in a DC with 24 rows. Figure 4.7 summarizes the outputs of the analytic model with varying node density, duty cycle and percentage of SCL

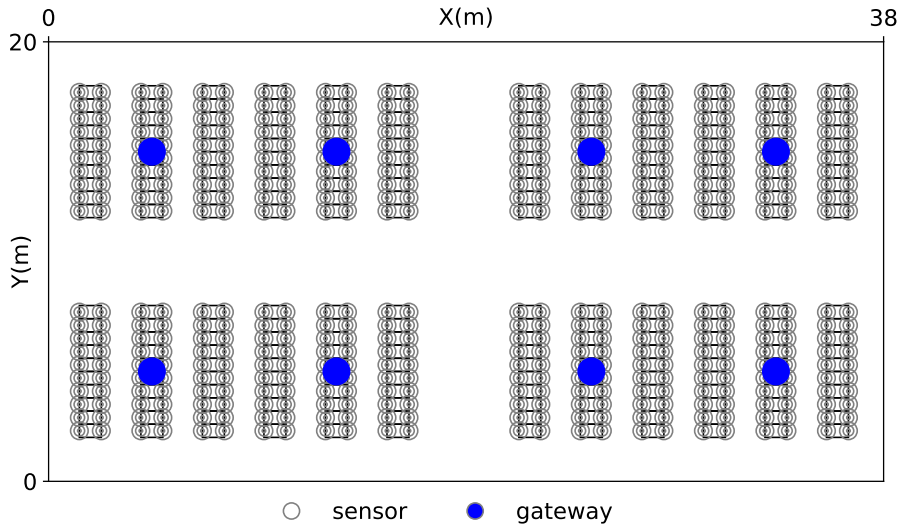
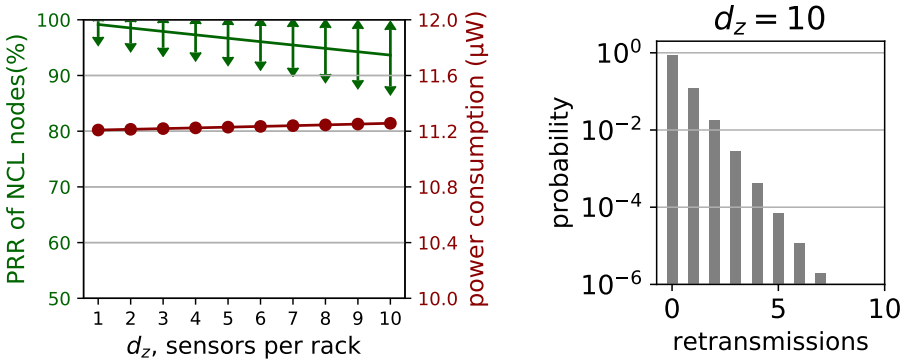


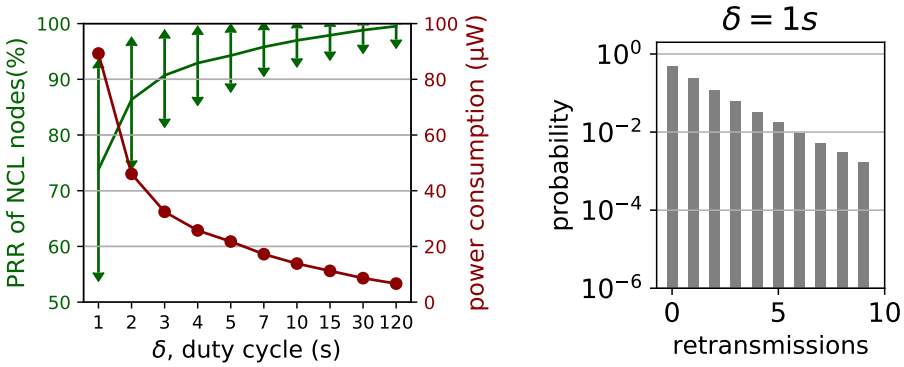
FIGURE 4.6: Layout of the parameter assessment DCWSN with 1440 sensors.

nodes in the network. Figure 4.7a illustrates how the increase in node densities linearly decreases the average PRR in NCL nodes. Adding 10 times more sensors (i.e., around 4500 sensors in our case) to the network increases the packet collision probability, and consequently, the PRR of NCL nodes drops by 6%. Meanwhile, since collisions result in extra re-transmissions among SCL nodes, the average power consumption increases by at most 1% in the worst case.

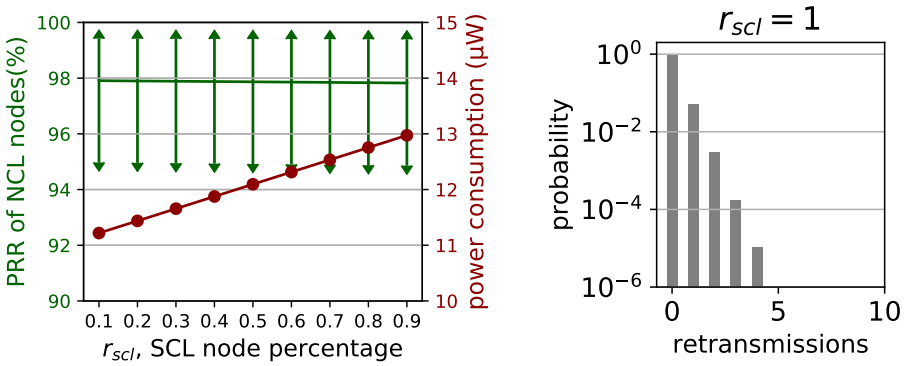
Duty cycles of sensors are expected to have a significant impact on the PRR and power consumption in the network. Figure 4.7b shows that when $\delta > 5s$, the NCL PRR remains above 90%. For example, with a duty cycle of the 30s, the PRR is around 98%. This is consistent with the experimental results reported in [50]. Nevertheless, it also reveals a significant decline in PRR and an increase in the average power consumption when the duty cycle is below 5s. At one-second duty cycles, the average PRR of NCL nodes can go down to 75% with power consumption at $95\mu W$.



(A) The effect of node densities, duty cycle=15s



(B) The effect of duty cycles, # of nodes=1440



(C) The effect of SCL node percentage, duty cycle=15s, # of nodes=1440

FIGURE 4.7: Impacts of different parameters on LEMoNet performance in large scale DCWSNs.

The third parameter to be investigated is the r_{scl}, namely the percentage of nodes in SCL mode. As shown in Figure 4.7c, when the number of SCL nodes increases, the PRR of NCL nodes remains the same and the average power consumption grows linearly. Interestingly, even in the case that all sensors are in the SCL mode ($r_{scl} = 1$), the power consumption is only $2\mu W$ higher than that when $r_{scl} = 0.1$. This result in conjunction with the fact that PRR of SCL nodes never falls below 99.999% even in the worst-case scenarios (histograms on the right side of Figures 4.7a, 4.7b and 4.7c), implies that when the duty cycle is low to medium, it is beneficial to put all nodes to the SCL mode to improve PRR at a marginal increase of power consumption.

4.5 Conclusion

This chapter presented an analytical model to study the performance of LEMoNet in a large scale DC. The model has been verified using event-driven simulations and used to study the effects of different parameters on the protocol performance. It was illustrated that there is no significant increase in power consumption when all the sensors are set in reliable (SCL) mode when duty cycle of the sensors is set to 15s.

Chapter 5

SoftBLE: an SDN Framework for Network Adaptability

5.1 Introduction

Bluetooth Low Energy (BLE) is a parametric protocol. Beside common configurable parameters such as advertising intervals and power levels, major BLE vendors (e.g., TI and Nordic) recently enable more agility in their protocol stacks by introducing tunable advertising and scanning channels. Some research works have studied the effects of these parameters on neighbor discovery [55, 57, 71], or tried to improve the overall network performance by tuning one of them [85, 16, 56]. Unfortunately, to the best of our knowledge, there has not yet been a generic framework that facilitates run-time optimization of BLE advertising parameters based on network conditions, traffic loads, and application-defined performance requirements. To bridge this gap, we propose SoftBLE, a Software Defined Networking (SDN) based framework. SDN is attractive for BLE-based monitoring

networks since it allows optimizing network parameters via a central controller and avoids the complexity of reaching a consensus in distributed systems.

There is no one-size-fits-all fixed BLE parameter settings for all BLE based Internet of Things (IoT) applications or deployment environments. It is important to be able to *configure and adapt* BLE parameters according to application requirements both during deployment and at operational time. The key advantage of SoftBLE lies in its ability to adapt the configurations of sensors and gateways based on current network conditions through orchestration algorithms. The algorithms aim to find the best set of control knobs such that the average power consumption is minimized when the PRRs of all sensors are above a certain threshold. For tractability, we break it down into two sub-problems. First, we determine scanning channel mapping on gateways so that the gateways assigned to the same channel are as far as possible. Second, we solve for the optimal TX power and channel assignment of sensors by modeling their effects on Packet Reception Rate (PRR).

In the rest of the chapter, we first review the details of SoftBLE framework's control plane and then discuss the details of two orchestration algorithms. This chapter will be concluded by evaluating SoftBLE in an experimental testbed as well as through large-scale simulations.

5.2 SoftBLE Design

Network traffics in SDN are made up of network flows. Each flow is defined in RFC 3697 as a sequence of packets with specific source and destination(s). In

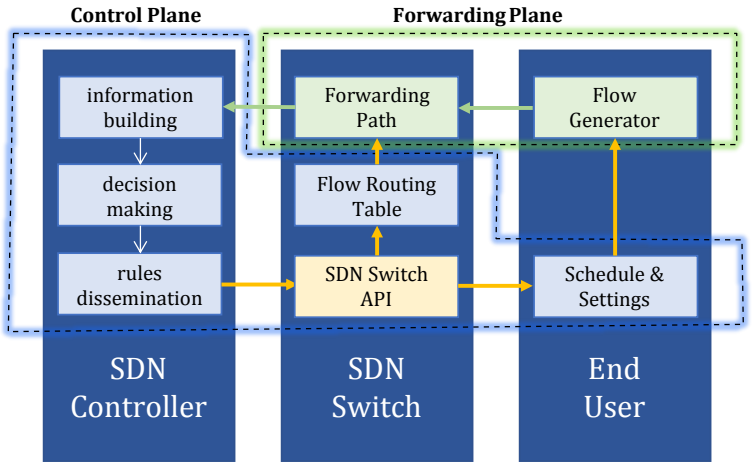


FIGURE 5.1: The components of SDN framework in a SoftBLE.

the forwarding plane, an end-user generates flows based on its predefined schedule and settings, and SDN switches redirect the flows based on their flow routing tables. These tables are defined by the controller, which may collocate with the switches or run in a separate device (called *detached*). In either case, the process of orchestration consists of three parts: information building, decision making, and rule dissemination.

Information building aims to gather flow statistics from SDN switches. Decision making generates updates for flow table entries and end-user settings using orchestration algorithms. Rules dissemination is the process of broadcasting the updates via control command packets to switches and edge devices.

SoftBLE, as shown in Figure 5.1, is based on the detached controller design with the following SDN elements:

End Users are the *sensor nodes*. They are in *Unprovisioned* state initially and only advertise *Provision Requests*. When a sufficient number of requests

from a sensor are collected, its advertising parameters will be orchestrated by the controller and disseminated through the gateways. Upon successful delivery and configuration of the parameters on a sensor, its state is changed to *Provisioned*. A sensor starts sensing and advertising its measurements only when it is provisioned.

Flows are the sensor measurements generated at the beginning of each duty cycle. Flow settings consist of the BLE parameters assigned to every flow, including the Advertising Channel Map and TX power of a sensor. These parameters are defined during sensor provisioning in the control plane.

SDN Switches are the *BLE gateways*. Each gateway is assigned to a group of sensors by the controller. Its flow routing table contains a BLE scanning whitelist, which includes the Random Private Non-Resolvable (RPNR) addresses of sensors assigned to the gateway. The scanning channel map of a gateway is also stored in the flow table.

SDN Controller is a central computer connected to the gateways via a reliable network such as Ethernet.

The forwarding plane in SoftBLE is the same as the reliable -Scannable Connectionless (SCL)- mode in Low Energy Monitoring Network (LEMoNet). Provisioned sensors wake up at the beginning of each duty cycle, read their measurements, and advertise their measured data in *ADV_SCAN_IND* packets. The advertisements are acknowledged by *SCAN_REQ* packets upon successful receptions; otherwise, they will be re-transmitted at most R times. Other than advertising time, the

sensors are in Standby mode. All legacy advertising intervals are fixed in the forwarding plane of SoftBLE. Advertising interval of every sensor, T_{ai}, is set to its minimum value to reduce latency and scanning interval of every gateway, T_{si}, is set to its maximum value to reduce gap blind times. Note that sensor duty cycles are application-dependent, and are independent of T_{si}. When its duty cycle is more than T_{si}, a sensor will remain in Standby mode and skip the respective advertisement(s).

5.3 Control Plane

The basic responsibility of the control plane is to provision newly installed sensor devices in batches or to re-provision any sensor that was temporarily disconnected. Batch provisioning in the control plane runs on top of the forwarding plane and consists of three steps:

Information building: The controller extracts the Received Signal Strength (RSS) of Provision Requests, advertised in ADV_IND packets from new sensors to gateways. The information is stored in an observation matrix.

Gateway Orchestration: The controller determines which advertising channels gateways scan. (Section 5.4)

Sensors orchestration determines and configures the TX power levels of all sensors in the network, their advertising channel map, and consequently the whitelists of gateways (Section 5.5).

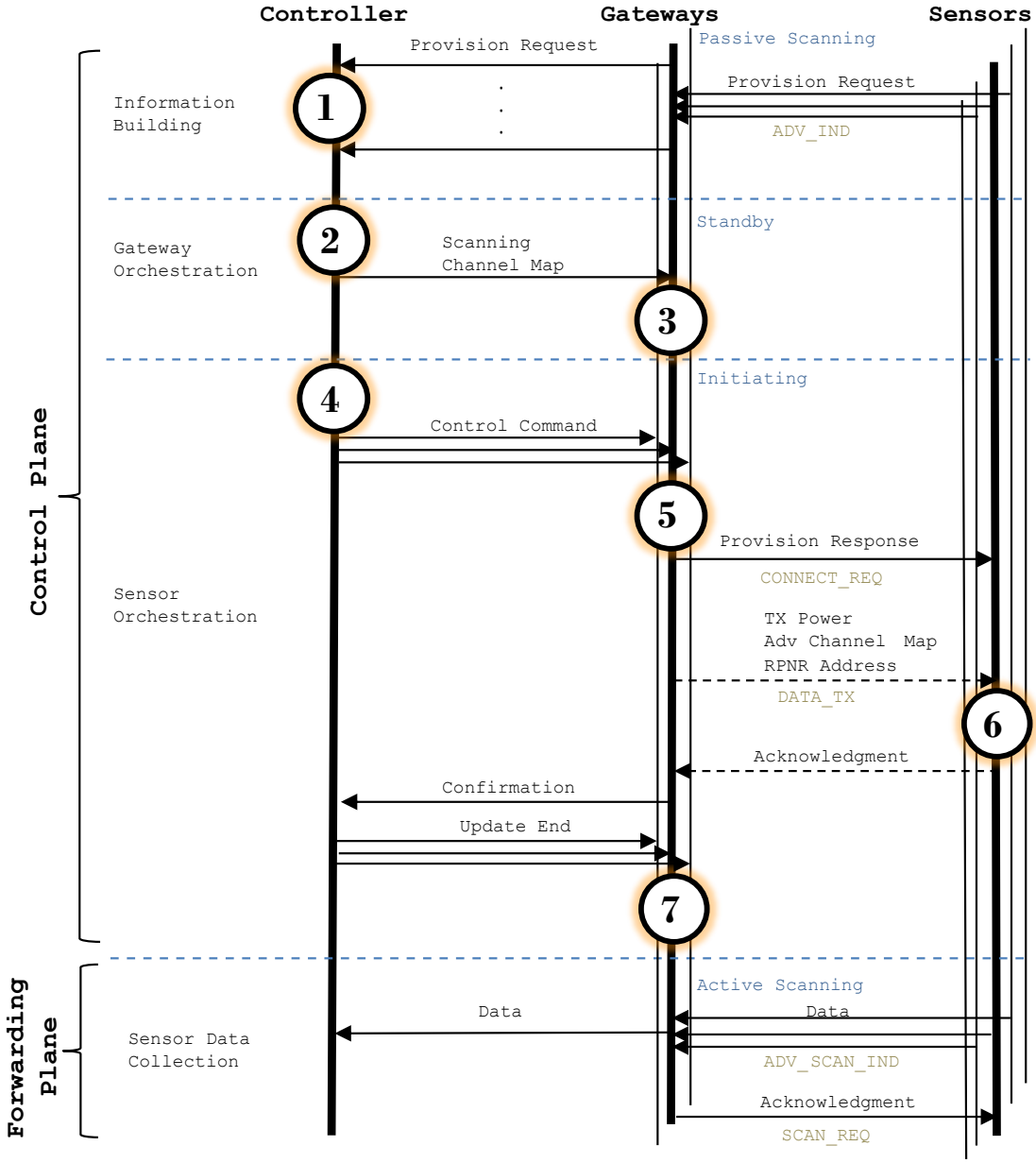


Figure 5.2: The timeline of packet transmission in the control and forwarding plane of SoftBLE.

The results of orchestrations are disseminated to all gateways reliably, and to each sensor through a connection from its closest gateway. Multiple connections can be established from all gateways at the same time, which allows parallel and fast parameter dissemination. Furthermore, to reduce the number of entries in a gateway's whitelist, all sensors assigned to the same set of gateways are clustered and are given the same RPNR address. Thus, only one address is inserted in a gateway's whitelist per cluster.

5.3.1 Provisioning Timeline

The detailed message exchanges in different stages of SoftBLE are shown in Figure 5.2. The timeline of the provisioning process has 7 steps:

1. Observation matrix is constructed based on RSS of the collected provision requests.

2. A scanning channel is assigned to each gateway using Algorithm 1 (graph coloring)

3. Gateways set up their scanning channels

4. Advertising channels and TX power are assigned to sensors using Algorithm 2.

5. Gateways establish connections with sensors and disseminate updated parameters

6. Provisioned sensors update their advertising parameters

7. Gateways substitute the MAC address of each sensor in their whitelist with the new RPNR address of the cluster that the sensor belongs to.

5.3.2 Information Building

The information building stage of the control plane builds up many-to-many relations between the provisioned sensors and the gateways. Each gateway can be assigned to several sensors, and a sensor can advertise to more than one gateway. A sensor-gateway observation matrix ($O_{M\times N}$) represents this relation. The matrix is extracted from the RSS of provision requests as:

$$O = \begin{bmatrix} o_{11} & \dots & o_{1N} \\ \vdots & o_{ij} & \vdots \\ o_{M1} & \dots & o_{MN} \end{bmatrix}, o_{ij} = \begin{cases} 1 & \text{if } \max\left(rss_j^i\right) > P_{sen} \\ 0 & \text{if } \max\left(rss_j^i\right) < P_{sen} \end{cases}_{\forall i\in\{1,..,N\},j\in\{1,..,M\}}. \tag{5.1}$$

Unlike the analytical model of LEMoNet, where the RSS values are estimated based on distance, in (5.1), rss_s^g vector is extracted from the provision requests received from sensor s in gateway g ($\emptyset$ if no observation). Matrix O is input to the two sub-problems of sensors and gateways orchestration, the details of which are discussed in the next two sessions.

5.3.3 Control Knobs

Five parameters can be tuned on the BLE devices, three on sensors and two on gateways. These parameters are called the control knobs of the framework. The sensor control knobs include:

TABLE 5.1: Control Knobs of SoftBLE devices

Name	Description	Default
$C_s^{\mathcal{S}}$	Channel map of sensors s	{37,38,39}
TX_s	TX power level of sensor s	0dBm
$AdvA_s$	Advertise address of sensor s	public
$C_g^{\mathcal{G}}$	Channel map of gateway g	37
WL_g	Whitelist of gateway g	$\emptyset$

1. A sensor's advertising channel can be configured to subsets of $\{37, 38, 39\}$

2. TX power levels can be set to any value in {-21, -18, -15, -12, -9, -6, -3, 0, 1, 2, 3, 4, 5}dBm.

3. A sensor's sdvertising address can be configured to its Public or a Random Non-Resolvable Private address

For the gateways, the control knobs are:

1. Scanning channels can be set to 37, 38, or 39

2. Whitelist can be filled by up to 8 sensor addresses

Table 5.1 lists the five control knobs together with their default values.

5.4 Gateway Orchestration

In SoftBLE, each gateway only scans a single channel. The gateway orchestration aims is how to assign each gateway to one of the three legacy channels such that neighboring gateways have the least interfering sensors. This problem is equivalent to *Weighted Improper 3-coloring* of a graph [4] in graph theory and *Minimum*

Algorithm 1: Channel assignment to gateways

input : observation matrix ($O_{M\times N}$)
output: assigned gateways channels ($C^{\mathcal{G}}$)
`// Constructing` $A^{\mathcal{G}}_{M\times M}$
for $i \leftarrow 1$ **to** M **do**
  for $j \leftarrow 1$ **to** M **do**
    $A^{\mathcal{G}}_{i,j} \leftarrow \left\|\vec{O}_i \wedge \vec{O}_j\right\|$;
`// Coloring of the graph with adjacency` $A^{\mathcal{G}}_{M\times M}$
Initialize SCS and $C^{\mathcal{G}}$ to $\{0\}_M$
  ▷ *SCS*: *Sum of the Covered Sensors*;
for $i \leftarrow 1$ **to** M **do**
  $candidGW \leftarrow \arg\max(SCS)$;
  Initialize CC to $\{0,0,0\}$ ▷ *Channel Covered*;
  for $j \leftarrow 1$ **to** M **do**
    if $C^{\mathcal{G}}_j > 0$ **then**
      $CC_{(C^{\mathcal{G}}_j - 36)} \overset{+}{\leftarrow} A^{\mathcal{G}}_{candidGW,j}$
  $C^{G}_{candidGW} \leftarrow \arg\min(CC) + 36$;
  for $k \leftarrow 1$ **to** M **do**
    if $C^{G}_k = 0$ **then**
      $SCS_{candidGW} \overset{+}{\leftarrow} A^{G}_{candidGW,k}$
    else
      $A^{G}_{candidGW,k} = 0$

Interference Frequency Assignment Problem (MI-FAP) [3] in wireless communication. The variations of MI-FAP that deal with channel assignment in cellular networks, such as the one in [34], are the closest to our problem. The problem was initially formulated as Integer Linear Programming in [3] has been proven to be NP-Hard in [102]. Heuristic solutions, including greedy search [3], tree search [59], or branch and cut algorithm [27] have been suggested. In this research, we propose a max-min optimization heuristic to solve it.

Let $\mathcal{G}$ and $\mathcal{S}$ denote the set of all gateways and sensors, respectively. Two

gateways u, v are connected by an edge (u, v) if there exist common sensors within their communication ranges. Let $E = \{(u, v)|u, v \in \mathcal{G}, \exists s \in \mathcal{S}, o_{u,s} = o_{v,s} = 1\}$, where o denotes the elements of observation matrix O, defined in (5.1). Edge (u, v) is associated with weight $w_{uv} = \sum_{s \in S} o_{u,s} \cdot o_{v,s}$, or equivalently, the number of common sensors. Given $G(\mathcal{G}, E, w)$ and the set of channels $C = \{37, 38, 30\}$, the objective of gateway channel assignment is to minimize the number of overlapping sensors among gateways in the same channel. Formally,

$$\begin{aligned} \min \quad & \sum_{(u,v) \in E} w_{uv} \mathbb{I}(x_u = x_v) \\ s.t. \quad & x_u \in \{37, 38, 39\}, \forall u \in V, \end{aligned} \tag{5.2}$$

where $\mathbb{I}(\cdot)$ is an indicator function and x_u is the channel assigned to gateway u.

When the chromatic number of G is greater than three, G cannot be 3-colored such that neighboring gateways are always assigned different channels. To minimize the objective function, we propose a max-min heuristic algorithm that iterates over all gateways. In each iteration, two steps are taken:

1. The gateway that has the maximum number of common sensors with already assigned gateways in the three channels is designated as *candidate*.

2. A channel that has the least number of common sensors between the candidate gateway and already assigned gateways is set as the channel of the candidate gateway.

The details of the heuristics are presented in Algorithm 1.

5.5 Sensor Orchestration

A sensor can choose to advertise in any of the three primary advertising channels as long as there exists a gateway in its vicinity scanning the channel. Increasing the number of advertising channels of a sensor may on one hand increase its PRR since its advertisement can be received by multiple gateways. On the other hand, doing so on all sensors may contribute to high traffic loads and consequently, reduced PRR. TX power levels have a similar effect in that increased TX power can reduce packet error rates but may lead to higher contention. Additionally, an increased TX power level leads to higher power consumption.

5.5.1 Problem Formulation

Let $E[PRR]_s$ and $E[PWR]_s$ be the expected PRR and power consumption of sensor s. The objective of sensor orchestration problem is to minimize the average $E[PWR]$ of all the sensors, such that $E[PRR]$ of every sensor remains higher than the application defined PRR threshold. It can thus be formulated as,

$$
\begin{aligned}
&\min_{\mathcal{C}^{\mathcal{S}},TX} \sum_{s\in\mathcal{S}} E[PWR]_s \\
&\quad s.t. \quad E[PRR]_s \geq T, \forall s \in \mathcal{S}, \\
&\qquad E[PWR]_s = f^s_{PWR}(\mathcal{C}^{\mathcal{S}}_s, TX_s, E[PRR]_s), \forall s \in \mathcal{S}, \\
&\qquad E[PRR]_s = f^s_{PRR}(\mathcal{C}^{\mathcal{S}}, TX), \forall s \in \mathcal{S}, \\
&\qquad TX = \{TX_s | s \in \mathcal{S}\}, \mathcal{C}^{\mathcal{S}} = \{\mathcal{C}^{\mathcal{S}}_s | s \in \mathcal{S}\} \\
&\qquad TX_s \in \{-21, -18, -15, -12, -9, -6, -3, 0, 1, 2, 3, 4, 5\}, \forall s \in \mathcal{S}, \\
&\qquad \mathcal{C}^{\mathcal{S}}_s \subseteq \{37, 38, 39\}, \forall s \in \mathcal{S}
\end{aligned}
\tag{5.3}
$$

The definitions of the parameters in above formulation are listed in Tables 5.2 and 5.1.

Two control knobs are involved in the process of minimization: advertising channel map ($\mathcal{C}_s^{\mathcal{S}}$), and TX power ($TX_s$). The analytical form of $E[PWR]_s$, denoted by f^s_{PWR}, depends only on the knobs of sensor s. But the analytical form of $E[PRR]_s$ (denoted by f^s_{PRR}) is dependant on the knobs of all the other sensors. It is because the collision probability on the neighboring sensors are not independent. As a result, the size of the search space can grow exponentially with the number of sensors.

To avoid the combinatorial explosion as the number of sensors grows (with 9 power levels and 7 combinations of advertisement channels per sensor), an approximation is warranted. We make the following three simplifications so that sensors can be analyzed independently:

1. The TX power of all other sensors remain at the values they use during information building (maximum possible option)

2. All other sensors advertise at all three primary channels.

3. Protocol model for interference is adopted

The simplifications result in an upper bound estimation for collision probability. They, thus, yield an underestimation of $E[PRR]_s$, which helps the satisfaction of the lower bound condition for PRR, though at the cost of higher TX power.

The above assumptions also facilitate approximating $\mathcal{C}$ and TX in f^s_{PRR} by

the observation matrix (O), defined earlier in (5.1). Subsequently, the problem formulation in (5.3) is equivalent to the minimization of pwr_s on each sensor $s \in \mathcal{S}$ individually, as:

$$\begin{aligned}
&\min_{\mathcal{C}_s^{\mathcal{S}}, TX_s} E[PWR]_s \\
&\quad s.t. \quad E[PRR]_s \geq T, \\
&\qquad E[PWR]_s = f^s_{PWR}(\mathcal{C}_s^{\mathcal{S}}, TX_s, E[PRR]_s), E[PRR]_s = f^s_{PRR}(\mathcal{C}_s^{\mathcal{S}}, TX_s, O), \\
&\qquad TX_s \in \{-21, -18, -15, -12, -9, -6, -3, 0, 1, 2, 3, 4, 5\}, \mathcal{C}_s^{\mathcal{S}} \subseteq \{37, 38, 39\}
\end{aligned} \tag{5.4}$$

5.5.2 Estimating PRR and Power Consumption

A similar analytical approach that is used to evaluate LEMoNet in Section 4.3, can be utilized to estimate the PRR and the mean power consumption of sensors in SoftBLE as well. As listed in Table 5.2, the analytical model of SoftBLE takes almost the same input parameters as that of LEMoNet. However, four fundamental differences between the two protocols affect some of the equations in the system model:

1. In SoftBLE unlike LEMoNet, sensors are assigned to the gateways by the means of whitelisting, i.e., only an assigned gateway will respond with Scan_REQ to an advertiser.

2. The RSSs between sensors and gateways are extracted directly from the provision requests in SoftBLE and do not need to be estimated.

Table 5.2: The parameters SoftBLE analytical model

Name	**Description**	**Default**
$\mathcal{G}$	Set of gateways	
$\mathcal{S}$	Set of sensors	
N	Number of sensors	$\|\mathcal{S}\|$
M	Number of gateways	$\|\mathcal{G}\|$
δ	Duty cycle	3s
T	Application defined PRR threshold	99.9%
R	Maximum re-transmissions	3
μ	BLE bit rate	1Mbps
$\|PDU\|$	Length of an advertisement's PDU in bits	16*8
$\|HEADER\|$	Length of advertising packet header	16*8
$\|DATA\|$	Bit length of *SCANNABLE_ADV_IND* packet	|HEADER|+|PDU|
$\|SR\|$	Bit length of *SCAN_REQ* packet	|HEADER|
$\|SS\|$	Bit length of *SCAN_ESP* packet	|HEADER|
$\mathcal{N}$	Noise level	-110dBm
V	Supply voltage	3v
P_{sen}	Sensitivity of BLE receivers	-91dBm
P_{tx}	Transmission power of sensor s when TX_s=0dBM	(V*8.9)mW
P_{rx}	Listening power consumption	(V*5.9)mW
P_{ifs}	Power consumption of intra-frame spacing	(V*3.8)mW

3. There is no Normal Connectionless (NCL) mode sensor in SoftBLE, and all sensors are in SCL mode.

4. The TX power of sensors in SoftBLE is variable.

Consequently, the following equations are modified to address the above differences.

Expected PRR ($E[PRR]_s$) Since the number of gateways that are assigned to a sensor is at most 3 (one for each channel) in SoftBLE, (4.4) is rewritten as:

$$\begin{aligned} \hat{prr}_s =& c_s^{37} prr_s^{\{g_s^{37}\}} + c_s^{38} prr_s^{\{g_s^{38}\}} + c_s^{39} prr_s^{\{g_s^{39}\}} \\ & - c_s^{37} c_s^{38} prr_s^{\{g_s^{37}, g_s^{38}\}} - c_s^{37} c_s^{39} prr_s^{\{g_s^{37}, g_s^{39}\}} - c_s^{38} c_s^{39} prr_s^{\{g_s^{38}, g_s^{39}\}} \\ & + c_s^{37} c_s^{38} c_s^{39} prr_s^{\{g_s^{37}, g_s^{38}, g_s^{39}\}}, \end{aligned} \tag{5.5}$$

where g_s^i is the gateway that listens on channel $i \in \{37, 38, 39\}$ and has the highest RSS from sensor s. It is *null* if there is no such a gateway, and the correspond term is removed. prr_s^{χ} is the probability of packet reception in all the gateways in set $\chi \subseteq \mathcal{G}$, and c_s^i is one if $i \in \mathcal{C}_s^{\mathcal{S}}$, otherwise zero.

Accordingly, the expected PRR for the packets of sensor s is given by:

$$E[PRR]_s = 1 - (1 - \hat{prr}_s)^R. \tag{5.6}$$

Per Gateway PRR (prr_s^{χ}) Since all the sensors are in the SCL mode in SoftBLE, PRR in (4.5) is modified to cover both data and SCAN_REQ transmissions as:

$$prr_s^{\chi} = (1 - col^{\chi}) \times \prod_{g \in \chi} \left(1 - (ber_s^g)^{(|DATA|)}\right) \left(1 - (ber_s^g)^{|SR|}\right), \tag{5.7}$$

where ber_s^g, which denotes the Bit Error Rate (BER) of advertising packets from sensor s at gateway g, can be calculated using equation (2.1), and col^{χ}, which denotes the probability of collisions for packets from sensor s at all the gateways $g \in \chi$, can be estimated by (4.7). Apparently, if χ is empty, prr_i^{χ} will be zero.

Interference Counter (I^{χ}) The number of potentially colliding sensors can directly be determined from the intersections of the rows in observation matrix O corresponding to the gateways in χ. For instance, if $\chi = \{g_1, g_2, g_3\}$, (4.6) is updated as:

$$I^{\chi} = \sum_{i=1}^{3} \left\| \vec{O_{g_i}} \right\| - \sum_{i<j=1}^{3} \left\| \vec{O_{g_i}} \wedge \vec{O_{g_j}} \right\| + \left\| \vec{O_{g_1}} \wedge \vec{O_{g_2}} \wedge \vec{O_{g_3}} \right\|, \tag{5.8}$$

where $\vec{O_{g_i}}$ denotes the row g_i in O.

Traffic Load (λ) With all nodes in SCL mode the calculation of traffic load is changed as well. The traffic load (λ) is the sum of the transmission times of *ADV_SCAN_IND*, response time of *SCAN_REQ*, the reception time of *SCAN_RSP*, and the decode gap, as:

$$\lambda = \frac{1}{\delta} \left(\frac{|DATA| + |SR| + |SS|}{\mu} + gap \right), \tag{5.9}$$

Expected Number of Retransmissions ($\overline{ret}$) In analyzing the PRR of sensor s, the PRRs of the other sensors are substituted by a lower bound, $1 - \sqrt[R]{T}$ (recall that T is the minimally required PRR). Doing so can greatly reduce the computation complexity of estimating col^{χ}. Thus, the worst case $\overline{ret}$ can be estimated as:

$$\overline{ret} = \left[\sum_{i=1}^{R-1} \left(\sqrt[R]{T} \right)^{i-1} \left(1 - \sqrt[R]{T} \right) \right] + \left(\sqrt[R]{T} \right)^{R-1}, \tag{5.10}$$

Expected Power Consumption ($E[PWR]_s$) In SoftBLE, the transmission power consumption depends on the transmission power (P_{tx}) as well. Thus, (4.10) which estimates the power for a PDU delivery from sensor s is updated as:

$$\begin{aligned} E[PWR]_s =& \sum_{r=1}^{R} \frac{1}{\delta} \cdot E_s^{adv} \cdot E[PRR]_s (1 - E[PRR]_s)^{(r-1)} \\ E_s^{adv} =& P_s^{tx} \cdot \left\| C_s^{\mathcal{S}} \right\| \cdot \left(\frac{|DATA|}{\mu} + P_{ifs} \right) \\ &+ P_{rx} \cdot \left(\frac{|SR|}{\mu} \right) + P_{ifs} + P_s^{tx} \cdot \left(\frac{|SS|}{\mu} \right) \\ P_s^{tx} =& P_{tx} \cdot 10^{(TX_s/10)}, \end{aligned} \tag{5.11}$$

where E_s^{adv} is the energy consumption of a single advertisement for sensor s in Joule.

5.5.3 Optimal Parameter Selection

To this end, we are in the position to present the sketch for sensor channel assignment and power control in Algorithm 2. This algorithm has two nested loops, the outer one is for 7 possible combinations of advertising channels ($2^3 - 1$), and the inner one is for 13 different TX power levels. In each iteration, it estimates the expected power consumption ($E[PWR]$) and expected PRR ($E[PRR]$) for each setting. Among settings that meet PRR threshold, the one with the lowest $E[PWR]$ is selected for the sensor. Algorithm 2 runs for each sensor node separately according to the simplified assumptions in the problem formulation. Since for each sensor it takes a constant number of calculations to investigate all possible combinations of control knobs, the complexity of Algorithm 2 is $O(c)$.

Algorithm 2: TX power optimization on the sensors

input : sensor ID (s), observation matrix (O), RSS of provision requests received from s ($r\vec{s}s_s$)
output: assigned advertising channel map ($C_s^{\mathcal{S}}$) and TX power ($TX_s^{\mathcal{S}}$) to sensor s

$PTX \leftarrow \{-21, -18, -15, -12, -9, -6, -3, 0, 1, 2, 3, 4, 5\}$
bestC $\leftarrow \{37, 38, 39\}$;
bestP $\leftarrow 5$;
bestPWR $\leftarrow \infty$;
for $C \leftarrow$ *Subsets of {37,38,39}* **do**
    for $p \leftarrow$ *1* **to** *13* **do**
        $TX \leftarrow PTX[p]$;
        Estimate $E[PRR_s]$ based on $C, TX, r\vec{s}s_s, O$ using (5.5);
        Estimate $E[PWR_s]$ based on $E[PRR_s]$ using (5.11);
        if $E[PRR_s] > T$ ***and*** $E[PWR_s] <$ *bestPWR* **then**
            bestC $\leftarrow C$;
            bestP $\leftarrow PTX[p]$;
            bestPWR $\leftarrow E[PWR_s]$;
$C_s^{\mathcal{S}} \leftarrow$ bestC;
$TX_s \leftarrow$ bestP;

5.6 Performance Evaluation

In this section, we first evaluate the performance of SoftBLE using a BLE sensor testbed. A large-scale simulation study is then conducted to investigate its scalability. The baseline protocol for comparisons is LEMoNet in two setups, all the sensors are in NCL mode or all are in SCL mode. LEMoNet is the representative of the protocols with fixed parameter settings for the sensors and the gateways.

5.6.1 Experimental Validation

48 BLE sensor nodes and two gateway devices have been deployed in a 11m-by-8m laboratory (Figure 5.3). Both sensors and gateways are equipped with TI cc2640r2 MCU chips for BLE communication. The gateway devices are implemented on

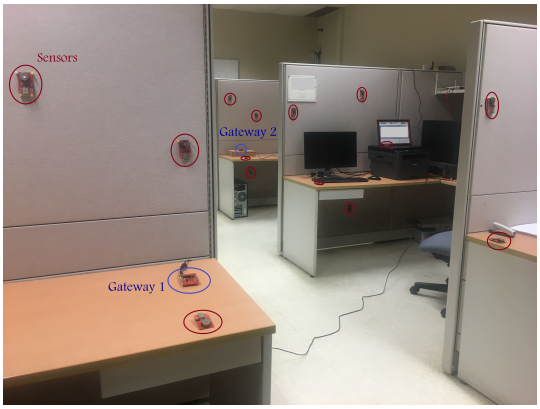
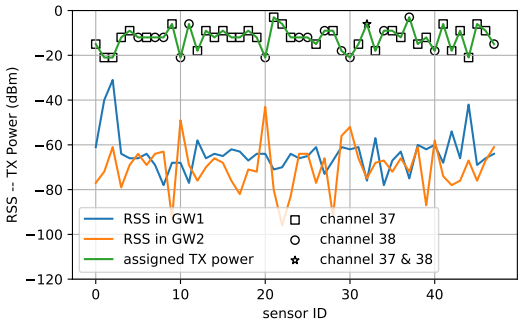
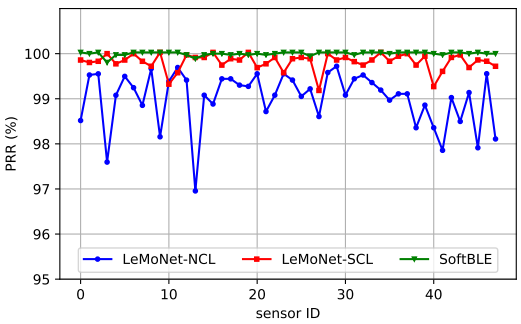


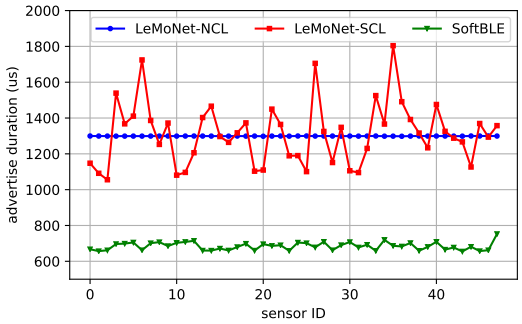
FIGURE 5.3: A scene of the experiment setup in our lab.



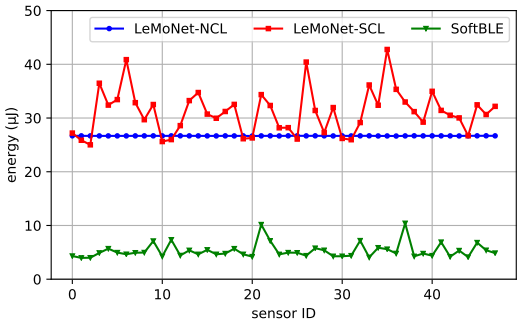
(A) RSS at gateways, sensor channel maps and TX power levels



(B) Packet Reception Rate



(C) Advertise duration per duty cycle



(D) Mean energy consumption for TX per duty cycle

FIGURE 5.4: Experimental comparison between LEMoNet and SoftBLE. 48 sensors and 2 gateways have been deployed in a 11m-by-8m space. Duty cycles of all sensors are set to 3s.

LAUNCHXL-CC2640R2 development kits and communicate to a SDN controller running on a desktop PC via Ethernet. The parameter settings in the experiments are summarized in Table 4.1.

Each experiment runs for three hours. Utilizing the software development kit (SDK) by TI, we are able to extract the energy consumption of each sensor and the time periods that sensors spend in advertising through *BEGIN_ADV* and *END_ADV*. Figure 5.4a shows the results of sensor orchestration and the average RSS values at the two gateways from each sensor. Gateway 1 and 2 are assigned Channel 37 and Channel 38, respectively. The majority of sensors are assigned to a single channel associated the gateway with a higher RSS except for sensor 32, which has low RSS to both gateways. To ensure a high PRR, it advertises its measurements on both channels 37 and 38. In this scenario, all sensors operating in channel 37 (38) belong to one cluster and are assigned the same RPNR address. Sensor 32 is assigned a third RPNR address and is included in the whitelists of both gateways.

Figure 5.4b shows PRRs under different schemes. We see that both SoftBLE and LEMoNet-SCL can achieve high PRRs due to the use of SCAN_REQ messages as acknowledgment and possible re-transmissions. SoftBLE outperforms both LEMoNet-SCL and LEMoNet-NCL and meets the required PRR threshold of 99.9% for 47 out of 48 sensor nodes. The PRR of sensor 4 is 99.80, which is slightly below the threshold. The small discrepancy can be attributed to the simplified assumptions in PRR modeling in Section 5.5.

We observe from Figure 5.4c, LEMoNet-NCL sensors roughly spend a constant

amount time in advertising in each duty cycle. This is due to the predefined length of legacy advertising events. Among SCL mode sensors, the advertisement duration varies because of the additional time to receive SCAN_REQ. Furthermore, the total number of advertisement messages is unpredictable in each duty cycle, depending on which channel a gateway responds with a SCAN_REQ. In contrast, in SoftBLE, the advertising duration is more than halved since the gateways listen on a single channel and most sensors only need to advertise on one channel in each duty cycle. The reduced advertising duration combined with lower TX power levels leads to around 70% less mean energy consumption in SoftBLE than LeMoNeT-SCL and LeMoNeT-NCL nodes as shown in Figure 5.4d.

5.6.2 Simulation Study

To study the scalability of the proposed SDN framework and the effects of different parameters, we have implemented BLE and SoftBLE in OMNET++ [91], an event-driven network simulator.

Simulation Setup and Performance Metrics

Simulations have been conducted in two scenarios:

Performance at Scale In the first scenario, 2500 sensor nodes are deployed in a $10000m^2$ area, and 121 gateways are distributed among them to collect advertised measurements. The placements are regular, where sensors and gateways form 50×50 and 11×11 grids, respectively.

Parameter Study In the second scenario, sensors are deployed randomly in a $2500m^2$ area and covered by 36 gateways arranged in a 6×6 grid. In the

simulations, we fix the number of gateways and vary the number of sensor nodes (N) and duty cycles (δ) to study their impacts on the performance.

The remaining fixed parameters can be found in Table 4.1. The results of each scenario are the averages of 5 runs, each lasting for 10,000s and with a different random seed. In addition to PRR and the power consumption of sensors, we also evaluate sensor utilization defined as,

$$\mathrm{U} = \frac{\text{total amount of application data received (bit)}}{\text{total transmitted bits}}.$$

The denominator includes advertisement packets, retransmissions as well as SCAN_REQs during sensor data collection.

Performance at Scale

Figure 5.5 illustrates the effects of channel assignment and TX power levels on the PRR and power consumption of a randomly selected sensor. In the experiment, the parameters of all other sensors are fixed. We observe a non-trivial relation between the control knobs and performance metrics of interest. The final configuration determined by Algorithm 2 is indicated by a green circle, which clearly has the lowest power consumption among settings satisfying the PRR threshold.

Figure 5.6 gives a snapshot of gateway and sensor channel assignments as well as sensor TX power levels according to Algorithm 1. Sensors advertise with an average TX power at $-5.11dBm$ (compared to default $0dBm$) and use an average of 2.76 channels for advertisement (compared to the default number of 3). Sensors in the middle of the area mostly advertise on all three channels with lower TX

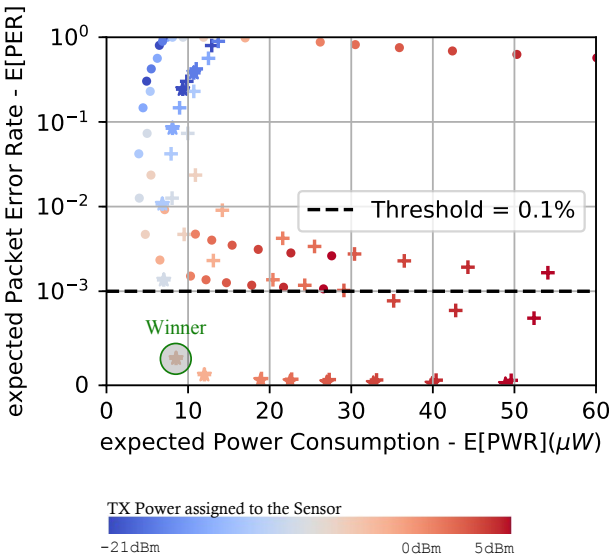


FIGURE 5.5: The effect of TX power and channel assignment of a randomly selected sensor node. All other sensors' parameters remain fixed. Results include channel assignments of a single channel (•), two channels (+), and three channels ($\star$) at different TX power levels.

power. In contrast, sensors at the corners or along the boundaries of the area need to advertise with higher TX powers but on fewer channels on average since they can only reach one or two gateways. Furthermore, a small collection of neighboring gateways are assigned the same channel. This is because only three primary advertising channels are available for assignments.

Figure 5.7 compares the performance of SoftBLE and LEMoNet in the large-scale network. It is seen from Figure 5.7a that in LEMoNet-NCL, around 10% of the nodes have more than 4% packet loss. This result is expected since BLE advertising is error-prone, and there is no mechanism in LEMoNet-NCL to detect and recover from packet losses or collisions. In contrast, both LEMoNet-SCL and SoftBLE can deliver almost all the data packets correctly because of the

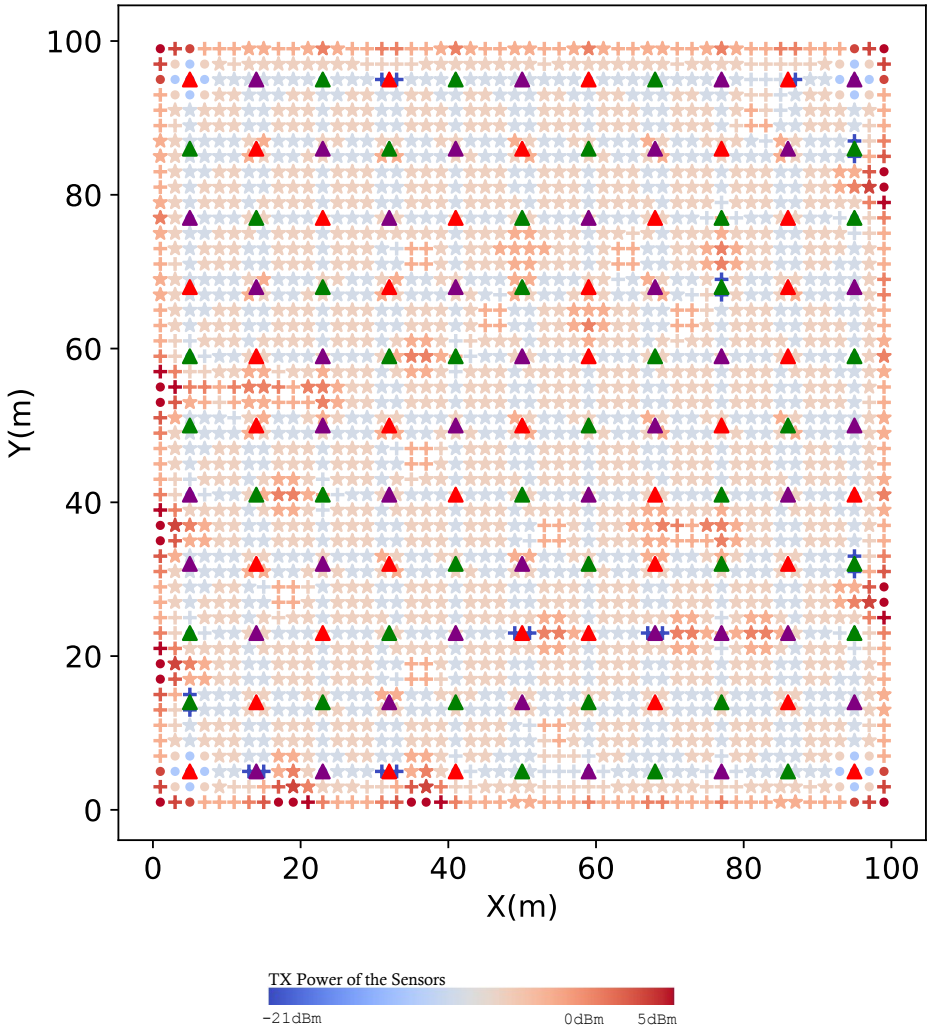
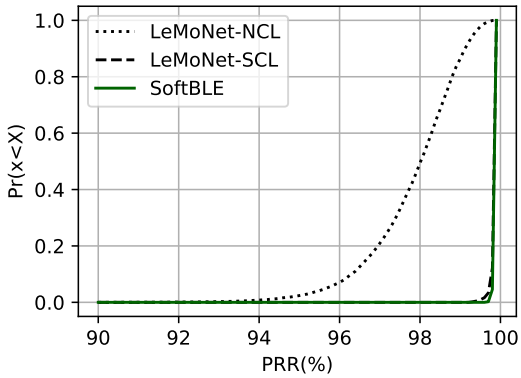
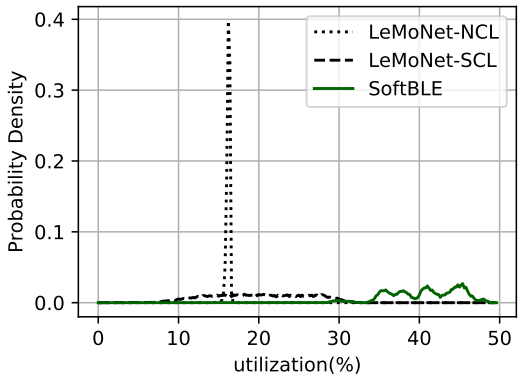


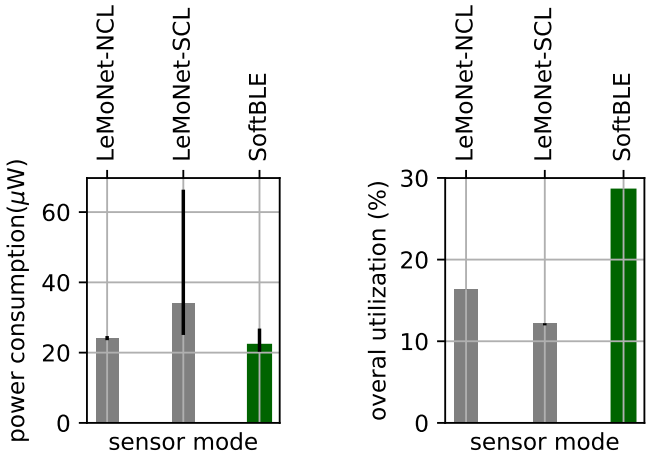
Figure 5.6: The output of running orchestration algorithms on the scaled scenario including 2500 sensors advertising on 5s duty cycles, and 121 gateways. The sensors are symbolized based on whether they assigned to one channel (•), two channels (+), or three channels ($\star$). Gateways are shown with $\triangle$, colored based on their scanning channel (red: 37, purple:38, green:39).



(A) CDF of mean sensors PRR



(B) PDF of sensors utilizations



(C) Mean power consumption of sensors

(D) Network Utilization

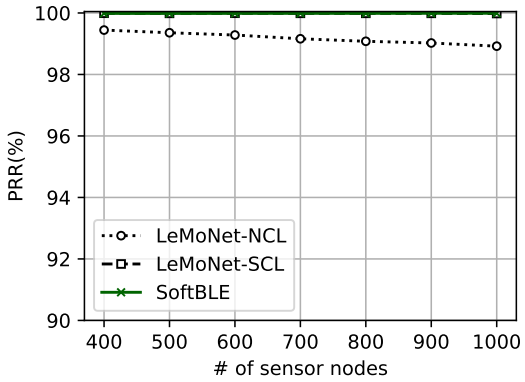
FIGURE 5.7: Performance of SoftBLE in a network with 2500 nodes.

use of acknowledgment and re-transmissions. However, Figure 5.7c shows that SoftBLE sensors exceed the PRR threshold of 99.9% while consuming 50% less power than LEMoNet-SCL ones. They even have less power consumption than the unacknowledged LEMoNet-NCL sensors. Furthermore, as shown in Figure 5.7b, all sensors in SoftBLE have more than 30% utilization compared to less than 17% among LEMoNet-NCL sensors, and 10 – 30% among LEMoNet-SCL sensors. In LEMoNet-SCL, sensor utilization varies between 10% to 30% because of the variable number of bits transmitted in each duty cycle. Lastly, network utilization, which is compute as the ratio of the amount of application data bits and total transmitted bits by both sensors and gateways, is almost 2 times higher in SoftBLE than its closest baseline (LEMoNet-NCL) as shown in Figure 5.7d.

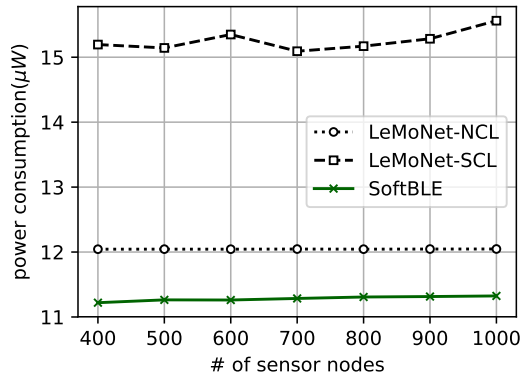
Impacts of Network Size and Duty Cycle

Figures 5.8a – 5.8c show the effect of the number of sensors (N) on network performance. As expected, as the number of sensors increases, medium contention increases, and thus the PRRs of LEMoNet-NCL sensors decrease. The power consumption of SoftBLE sensors increases and their utilization drops only slightly with increasing network sizes. This is due to higher TX power and additional channels assigned to sensors to mitigate increased collisions. SoftBLE is able to maintain a consistent PRR above the required threshold regardless of the number of nodes. Thanks to sensor and gateway provisioning, packet collisions are rare under these settings with SoftBLE, resulting in both high utilization and low power consumption.

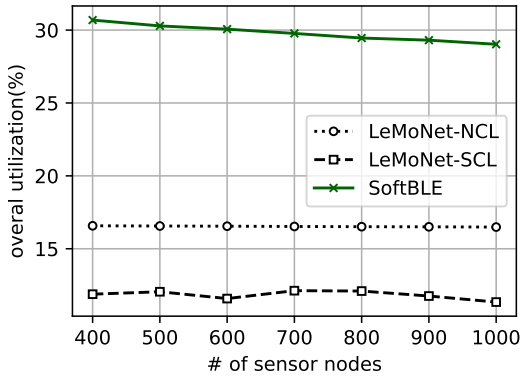
Figure 5.9a – 5.9c show the effects of sensor duty cycle. Reducing duty cycles (or



(A) Packet Reception Rate

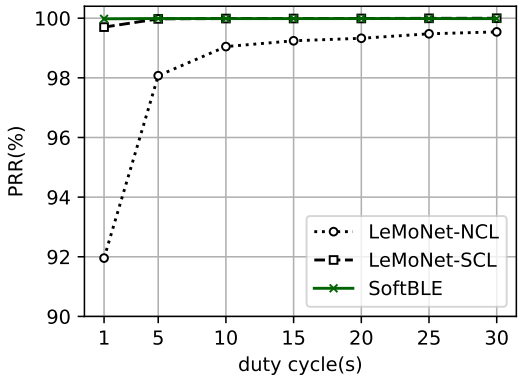


(B) Mean Power Consumption

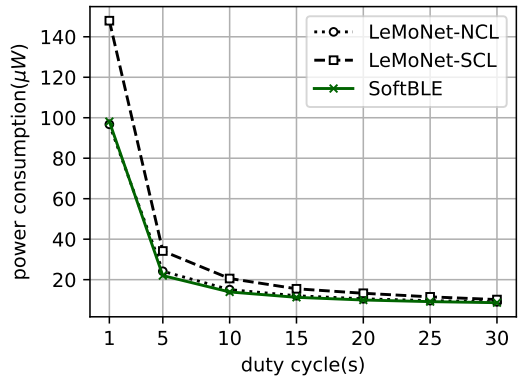


(C) Utilization

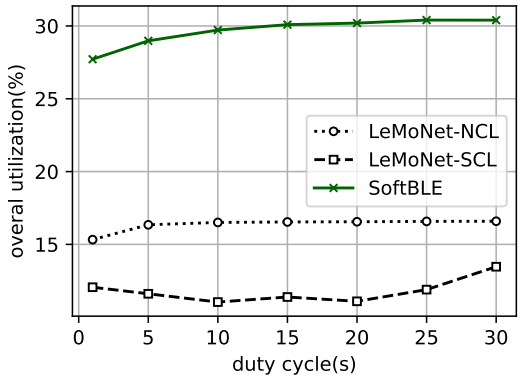
Figure 5.8: Effects of the number of sensors on the performance of SoftBLE. The sensors are deployed randomly in a $50m \times 50m$ area. Duty cycles of all sensors are set to 5s.



(A) Packet Reception Rate



(B) Mean Power Consumption



(C) Utilization

Figure 5.9: Effects of duty cycle on the performance of SoftBLE. 600 sensors are deployed randomly in a $50m \times 50m$ area.

increasing application data rates) increases collision and packet loss probabilities among LEMoNet-NCL sensors. At 1s duty cyle, LEMoNet-SCL sensors also suffer from low PRRs. In contrast, SoftBLE behaves consistently over a wide range of duty cycles with PRR $>$ 99.9%. The power consumption of all three approaches goes down when the duty cycle increases since sensors spend longer periods in the StandBy mode. Among all three approaches, SoftBLE has the lowest power consumption.

The high PRRs in SoftBLE come at the expense of reduced utilization at higher traffic loads in the network. As shown in Figure 5.9c, the utilization of SoftBLE gradually decreases from 31% at 30s duty cycle to 27% at 1s duty cycles. This is because, at a higher traffic load, sensors may need to retransmit packets and do so over more advertising channels.

5.7 Conclusion

Recent improvements in BLE legacy advertising makes the advertisers and scanners more configurable. SoftBLE leverages this agility to provide adaptability to a 2-tier BLE based network by an SDN framework. It is shown that SoftBLE considerably reduces the average power consumption of sensor nodes while meeting the application-defined performance requirements. Our proposed framework enables long lifetime Industrial IoT (IIoT) network deployments for large-scale and dense monitoring applications.

Chapter 6

Automated Sensor Mapping for DCWSN Commissioning

6.1 Introduction

Commissioning thousands of wireless devices in distributed settings is labor-intensive if done manually. For instance in data center (DC) monitoring applications, prior to deployment, technicians need to assign unique identifiers to individual devices and hard code them in sensor software. After deployment, the server-rack locations and the identifier of deployed sensors shall be recorded and entered into a Data Center Infrastructure Management System (DCIM) system. Such a manual process is both time-consuming and error-prone. Moreover, sensor locations may change due to misplacement, removal, and addition of new equipment.

To streamline the deployment process, it is imperative to automatically map sensor identifiers to unique locations among the set of known locations (according to the deployment plan). We call this problem *sensor mapping*, a special case

of sensor localization. Although several classes of algorithms have been proposed for localization [83], most of them depend on a single modality, namely Radio Frequency (RF). Inside a DC, the presence of IT equipment (ITE), metal racks, cable trays, and wires creates a multi-path rich environment and hinders reliable localization based on only RF signals.

Furthermore, methods based on pairwise ranging (or distance estimation from received signal strength) may violate global rigidity conditions in sparse or regular deployments [67]. As a result, the inferred locations are not unique and are subject to "local flips" or "symmetric flips" (two possible types of rigid transformations in our problem). In order to distinguish the correct placement from flipped ones, the common approach is to resolve the ambiguity by employing anchor nodes at known [69] or inferred locations [35]. Unfortunately, adding anchor nodes increases the commissioning cost.

In this chapter, we first propose a novel alternative to pure RF based approaches, named *thermal piloting*, that utilize the measurements of sensors as the only input for localization. The next section describes of this method in detail. *thermal piloting* is completely dependent on thermal data, which makes it prone to errors caused by inaccurate thermodynamic estimations. To address this issue, Section 6.3 proposes multimodal sensor mapping using pairwise Received Signal Strength (RSS) between the sensors as an additional data source. The new problem is formulated as Weighted Graph Matching (WGM) and solved by a novel parallel algorithm. The chapter is concluded by evaluating thermal piloting and comparing it with multimodal sensor mapping approach in a modular DC.

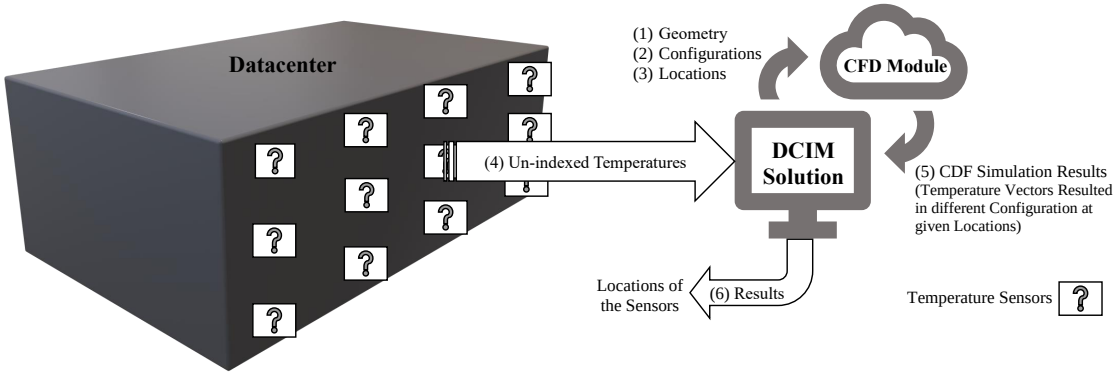


FIGURE 6.1: Components of thermal piloting

6.2 thermal piloting

6.2.1 System Architecture

As illustrated in Figure 6.1, there are three main components in thermal piloting:

1. **Sensors** measure the thermal condition of a target DC.

2. **A DCIM module** collects sensor measurements. It also stores the geometry of the target DC such as the number of the racks, the location of the servers, and the dimensions of the DC environment.

3. **A Computational Fluid Dynamics (CFD) module** takes the geometry of the DC, the Computer Room Air Conditioner (CRAC) unit configuration, and returns predicted temperatures and airflows at defined locations. Due to its high computation complexity, CFD simulations are often conducted in high performance DCs remotely.

Also shown in Figure 6.1 is data flow between DCIM and the sensors on one side and CFD on the other side during localization.

We assume that sensors are calibrated and the communication protocol between the sensors and the DCIM backend is reliable. Additionally, the DC geometry is accurate and CFD simulations are of high fidelity. High fidelity CFD models are still in the realm of extensive research and are outside the scope of this study.

6.2.2 Problem Formulation

Consider a total of N sensor locations and C configurations. Define a round of C CFD simulations as simulations with each one of the C configurations once. After S rounds of simulations with random perturbation and server utilization, we obtain a training data-set of $S \times N$ data samples $\mathcal{D} = \{x_i, y_i, i = 0, 1, \ldots, S \times N - 1\}$, where x_i's are C dimensional feature vectors, $y_i \in \{1, 2, \ldots, N\}$'s are their labels (sensor locations). In the inference phase, we obtain thermal measurements in C configurations from N sensors deployed at N locations. Let o_j be the C-dimension vector from sensor $j, j \in \{1, 2, \ldots, N\}$. Given $\mathcal{D}$ and o_j's, the goal of thermal piloting is *to determine a bijective mapping (permutation)* $\pi : \{1, 2, \ldots, N\} \rightarrow \{1, 2, \ldots, N\}$ *from locations to sensor indices*. Formally, we formulate it as the following *Maximum Likelihood Estimation (MLE)* problem:

$$\underset{\pi}{\text{maximize}} \quad L\left(o_{\pi(1)}, o_{\pi(2)}, \ldots, o_{\pi(N)} | \mathcal{D}\right) \tag{6.1}$$

where L is the likelihood function. The population of all possible permutations include $O(n!)$ samples with equal probabilities. So the input of MLE problem in (6.1) is a permutation group with uniform probability distribution function (PDF). Every instance of the minimum weight problem in a permutation group can be reduced to this problem, and the later problem is known to be NP-hard [14].

Therefore, the MLE problem in 6.1 is NP-hard and we seek to relax it so that polynomial complexity solutions can be found. In particular, we make the simplifying assumption that the underlying models that generate the data at sensor locations are independent. In other words,

$$L\left(o_{\pi(1)}, o_{\pi(2)}, \ldots, o_{\pi(N)}|\mathcal{D}\right) = \prod_j L\left(o_{\pi_j}|\mathcal{M}_j\right),$$

where $\mathcal{M}_j$ is the model for the observations at location j. Let Π be an $N \times N$ permutation matrix, namely, Π satisfies,

$$\begin{aligned} &\mathbf{\Pi}_{ij} \in \{0,1\},\ i,j \in \{1,2,\ldots,N\} \\ &\mathbf{\Pi}_{ij} = 1,\ \forall i = \pi(j), i \in \{1,2,\ldots,N\} \\ &\sum_{j=1}^{N} \mathbf{\Pi}_{ij} = \sum_{j=1}^{N} \mathbf{\Pi}_{ji} = 1, i \in \{1,2,\ldots,N\}. \end{aligned} \tag{6.2}$$

Under the independence assumption, the optimization problem in (6.1) is equivalent to

$$\begin{aligned} \underset{\Pi}{\text{maximize}} \quad & \sum_{i,j} \Pi_{i,j} \times \log L(o_i|\mathcal{M}_j) \\ s.t., \quad & \text{Constraints (6.2)}, \end{aligned} \tag{6.3}$$

which is the well-known maximum weighted bipartite matching (MWBM) problem, where the vertices correspond to the sensor indices on one side and locations on the other, and the weight between index i to location j is given by $\log L(o_i|\mathcal{M}_j)$. MWBM in a complete bipartite graph of N vertices can be solved in $O(N^4)$ time using the Hungarian algorithm or in $O(N^2 \log N + N^3)$ time with the Dijkstra algorithm and Fibonacci heap [78].

6.2.3 Data Model

To solve (6.3), what remains to be determined is the data model for likelihood $L(o|\mathcal{M}_j)$. For this purpose, we can use different models such as Gaussian Mixture Model (GMM) or log-linear model. GMM models the likelihood of data given class labels as a linear superposition of Gaussian distributions. Log-linear models, on the other hand, are a class of machine learning models with the defining characteristic that the log unnormalized probability is a linear combination of a set of features (or equivalently, the inner product of feature vector and model parameter vector) [47]. After trying both mentioned models, log-linear model is selected as the most accurate one. The likelihood in this model is formally defined as:

$$L(o|\mathcal{M}_j) = e^{\theta_j \cdot o} Z_{\theta_j}^{-1}, \tag{6.4}$$

where $Z_{\theta_j}^{-1}$ is a normalizing factor. Here, we estimate the parameter θ_j as the empirical mean of the training data at location j, namely,

$$\theta_j = \sum_{\{x_i, y_i\} \in \mathcal{D} | y_i = j} x_i. \tag{6.5}$$

By taking log of both sides of the likelihood function, we have the the log-linear model as:

$$\log L(o_i|\mathcal{M}_j) = c_j \theta_j \cdot o_j,$$

where $c_j = -\log Z_{\theta_j}$.

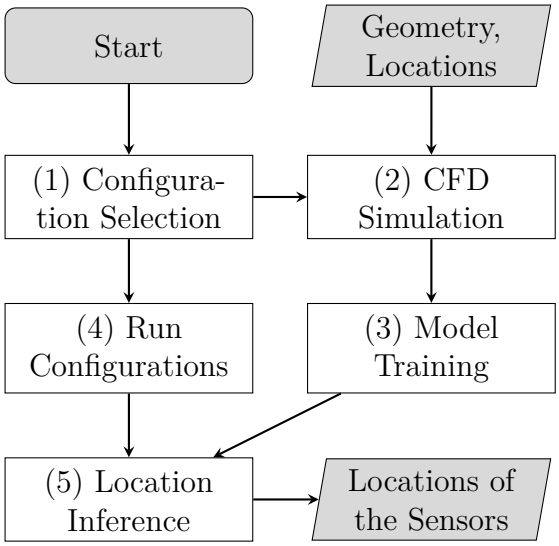


Figure 6.2: Flowchart of the thermal piloting process

6.2.4 Localization Process

Our proposed thermal piloting approach consists of six steps (Figure 6.2). In the first step, the most discriminative set of cooling configurations are selected. This can be done either through domain knowledge, e.g., turning on a single fan is likely to create large temperature variations at nearby locations in the DC, or can be done computationally. In the latter case, one can simulate a variety of different configurations and score them based on their chi-squared test results with the location labels. Alternatively, feature selection approaches [42] can be adopted by treating each configuration as a feature component. In our experimental DC with two in-row cooling units and five fans, the chi-squared test results in a set of five isomorphic configurations: one fan is on and the others are off.

In the second step, DCIM uploads the geometry of the target DC, along with the list of candidate locations and the set of CRAH configurations to the CFD

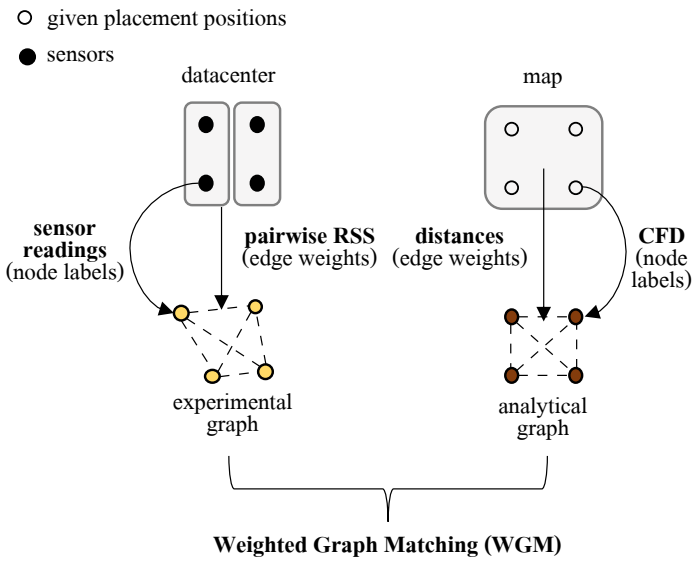


FIGURE 6.3: The proposed multimodal sensor localization framework. Sensors are mapped to designated locations by weighted matching between analytical and experimental graphs.

server. Steady-state CFD simulations return simulated thermal values at the desired locations. Multiple simulations can be executed for each configuration under different server utilization. These values form feature vectors in training data to generate machine learning models in Step 4. During the commission phase, the configurations are applied to the target DC one by one. Actual sensor measurements are collected by the DCIM for each configuration when the system reaches a steady-state. The machine learning model is then applied to the experimental data to infer the location of each sensor in Step 5.

6.3 Multimodal Sensor Localization

In this Session, we consider *pairwise* RSS between sensors as an extra data source in addition to *pointwise* temperature measurements. It leads to reformulating

sensor mapping as a labeled WGM problem. As shown in Figure 6.3, the two graphs in WGM are instantiated from real-world measurements and predictions from analytical models, respectively. The objective is to find the best mapping such that the two agree the most in a probabilistic sense. To solve the NP-hard labeled WGM problem, we develop a novel search-based heuristic and evaluate it through experimental data collected from a real-world modular DC testbed. We find that the heuristic is superior to state-of-the-art methods for labeled WGM in accuracy with comparable complexity.

6.3.1 Problem Formulation

In this section, we first introduce notations and then provide our data model and problem formulation.

Notations

Pairwise Similarity: Consider a total of N sensors at N possible locations $l_1, l_2, \ldots, l_N$ from a deployment plan. Let D be an $N \times N$ symmetric matrix representing the Euclidean distances between the N known locations, where d_{ij} is the Euclidean distance between sensor locations i and j. Same as formulation of thermal piloting, a mapping (permutation) function $\pi : \{1, 2, \ldots, N\} \rightarrow \{1, 2, \ldots, N\}$ defines the bijective relation from the locations to sensor indices, and $\mathbf{o}_i^m, m = 1, 2, \ldots, M$ be a d-dimension vector corresponding to the mth pointwise observation of sensor i. Conversely, π^{-1} maps sensor indices to locations $l_1, l_2, \ldots, l_N$. Let $\mathcal{P}$ be the set of all possible permutation matrices where $\mathbf{\Pi} \in \mathcal{P}$ is defined in (6.2).

Pointwise Dissimilarity: Denote by $\mathbf{A}^k, k = 1, 2, \ldots, K$, the kth pairwise measurement matrix. Its element in the ith row and jth column $\mathbf{A}^k_{ij}$ is the kth pairwise measurement between sensors i and j[1] for $i, j = 1, 2, \ldots, N$. Denote by $\mathcal{M}_\theta$ and $\mathcal{M}_\gamma$ the data models for pairwise and pointwise measurements parameterized by θ and γ, respectively.

Data Model

For pairwise RSS measurements, we adopt the well-known log-normal path loss model [86]. The RSS of the transmitted signal from a sensor at location i to j is given by,

$$a_{ij} = P_{\pi(i)} - PL_0 + G_{\pi(j)} + 10\beta \log_{10} \frac{d_{ij}}{d_0} + \chi,$$

where χ is white Gaussian noise with standard deviation of σ. We assume that the noise on one sensor is uncorrelated with other model parameters and independent from the the other sensors. $P_{\pi(i)}$ is the transmitted power level of the sensor at location i, $G_{\pi(j)}$ is the receiving antenna gain at the sensor at location j, β is the path loss exponent, and PL_0 is the path loss at reference distance d_0. Under the assumption that the transmission power levels and antenna gains are the same at all nodes, it can be further simplified as,

$$a_{ij} = c_1 - 10\beta \log_{10} d_{ij} + \chi,$$

where c_1 is a constant. Let $\mathbf{A} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{A}^k$. Under the assumption that $a_{ij}, i, j = 1, 2, \ldots, N$ are independent, we thus have

[1] Missing elements will be imputed based on domain knowledge.

$$\log P(\mathbf{A}|\mathcal{M}_\theta, \mathbf{\Pi}) = c_2 - \frac{K}{\sigma_1^2} \left\| \mathbf{A} + 10\beta \log_{10}(\mathbf{\Pi} D \mathbf{\Pi}^T) - c_1 \mathbf{J}_N \right\|_F^2, \tag{6.6}$$

where $\mathbf{J}_N$ is a $N \times N$ matrix of all ones, $||\cdot||_F$ is the Frobenius norm defined as the square root of sum of the absolute squares of its elements. In (6.6), $\theta = \{\sigma_1, \beta, c_1, c_2\}$.

For pointwise measurements, we model the thermal measurements at location i under C cooling conditions using a multivariate Gaussian distribution. Let $j = \pi^{-1}(i)$. The log-likelihood of mth observation $\mathbf{o}_i^m$ of sensor i is given by,

$$\log P(\mathbf{o}_i^m|\mathcal{M}_{\gamma_j}) = c_3 - \frac{1}{2}\log|\mathbf{\Sigma}_j| - \frac{1}{2}\left((\mathbf{o}_i^m - \mu_j)^T \mathbf{\Sigma}_j^{-1} (\mathbf{o}_i^m - \mu_j)\right), \tag{6.7}$$

where c_3 is a normalizing factor, μ_j and $\mathbf{\Sigma}_j$ are the mean and covariance matrix of measurements at location j, $\gamma_j = \{\mathbf{\Sigma}_j, \mu_j\}$. Let $\mathbf{o}_i = \frac{1}{M}\sum_{m=1}^{M} \mathbf{o}_i^m$. Denote by $\mathbf{O}$ ($\mathbf{M}$) a $C \times N$ matrix with $\mathbf{o}_i$ (μ_i) as the ith column. Under the assumption of independent measurements and identity covariance matrices $\mathbf{\Sigma}_j = \sigma_2^2 \mathbf{I}, \forall j$, we have

$$\log P\left(\mathbf{O}|\mathcal{M}_\gamma, \mathbf{\Pi}\right) = -\sigma_2^{-2} tr\left((\mathbf{O} - \mathbf{M}\mathbf{\Pi}^T)^T (\mathbf{O} - \mathbf{M}\mathbf{\Pi}^T)\right) + c_4, \tag{6.8}$$

where c_4 is a normalizing constant.

Maximizing Joint Likelihood of Pairwise and Pointwise Measurements

The *sensor mapping problem* can be formulated as finding a permutation matrix that maximizes the weighted log-likelihood, namely,

$$\begin{aligned} \max \quad & \alpha \underbrace{\log P\left(\mathbf{A}|\mathcal{M}_\theta, \mathbf{\Pi}\right)}_{pairwise} + (1-\alpha) \underbrace{\log P\left(\mathbf{O}|\mathcal{M}_\gamma, \mathbf{\Pi}\right)}_{pointwise}, \\ s.t. \quad & \mathbf{\Pi} \in \mathcal{P} \end{aligned} \tag{6.9}$$

where α is a pre-configured parameter that indicates the relative importance of pairwise and pointwise measurements.

Using the models for pairwise and pointwise measurements in (6.6) and (6.8), we can further simplify the optimization problem as follows.

Theorem 6.3.1. *Under the models for pairwise and pointwise measurements in* (6.6) *and* (6.8)*, the sensor mapping problem is equivalent to:*

$$\begin{aligned} \min \quad & \alpha' \left\| (\mathbf{A} - c_1 \mathbf{J}_N) - \mathbf{\Pi} \log_{10} \mathbf{D} \mathbf{\Pi}^T \right\|_F^2 + (1-\alpha') tr(\mathbf{L}\mathbf{\Pi}^T) \\ s.t. \quad & \mathbf{\Pi} \in \mathcal{P}, \end{aligned} \tag{6.10}$$

where $\mathbf{L} = \mathbf{O}^T\mathbf{M}$.

Proof. Let $\mathbf{A}_H = \mathbf{A} - c_1\mathbf{J}_N$ and $\mathbf{A}_G = \log_{10}\mathbf{D}$. The LHS of (6.9) can be written as,

$$\begin{aligned} & c_2 - \tfrac{K}{\sigma_1^2}\left(\|\mathbf{A}_H\|_F^2 + \|10\beta\mathbf{\Pi}\mathbf{A}_G\mathbf{\Pi}^T\|_F^2 - 2tr(10\beta\mathbf{A}_H^T\mathbf{\Pi}\mathbf{A}_G\mathbf{\Pi}^T)\right) \\ & = \underbrace{c_2 - \frac{K}{\sigma_1^2}\left(\|\mathbf{A}_H\|_F^2 + \|10\beta\mathbf{A}_G\|_F^2\right)}_{\text{independent of } \mathbf{\Pi}} - \tfrac{2K}{\sigma_1^2} tr(10\beta\mathbf{A}_H^T\mathbf{\Pi}\mathbf{A}_G\mathbf{\Pi}^T) \\ & = c_2' - \tfrac{10K\beta}{\sigma_1^2}\left(\|\mathbf{A}_H\|_F^2 + \|\mathbf{\Pi}\mathbf{A}_G\mathbf{\Pi}^T\|_F^2 - 2tr(\mathbf{A}_H^T\mathbf{\Pi}\mathbf{A}_G\mathbf{\Pi}^T)\right) \\ & = c_2' - \tfrac{10K\beta}{\sigma_1^2}\left\|\mathbf{A}_H - \mathbf{\Pi}\mathbf{A}_G\mathbf{\Pi}^T\right\| \end{aligned} \tag{6.11}$$

The second equality is due to the fact that $\mathbf{\Pi}$ is a permutation matrix.

Similarly, the RHS of (6.9) can be written as,

$$\underbrace{c_4 - \frac{1}{\sigma_2^2} tr(\mathbf{O}^T\mathbf{O} + \mathbf{M}^T\mathbf{M})}_{\text{independent of } \mathbf{\Pi}} + \frac{1}{\sigma_2^2} 2tr(\mathbf{O}^T\mathbf{M}\mathbf{\Pi}^T).$$

Therefore, with proper re-scaling, we have the optimization problem in the statement. □

In the proof, if we treat $\mathbf{A}_H$ and $\mathbf{A}_G$ as adjacency matrices of graphs H and G, and $\mathbf{L}$ as the (vector) labels on the vertices. The sensor mapping problem is equivalent to the labeled WGM problem, which is known to be NP-hard [80]. Therefore, it is important to devise an efficient heuristic solution.

A important implication of Theorem 6.3.1 is that the optimization objective is independent of β. Furthermore, c_1 is a function of given wavelength and known antenna gains. Thus, we only need to estimate the model parameter μ for the log-likelihood functions.

6.3.2 SWAP-ing based Algorithm for Labeled WGM

The main idea is similar to gradient descent, namely, to decrease the objective function iteratively. However, instead of re-evaluating the objective function in each iteration, we update it using lightweight vector operations that can be executed in parallel. The search space of the optimization problem consists of sequences of sensor indices in the range of 1 to N in $\mathbb{Z}^n$. We define the particle movement using the swap operation.

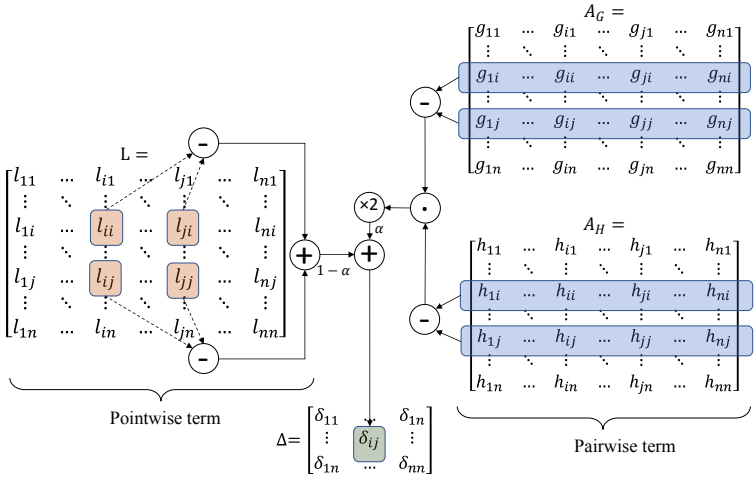


FIGURE 6.4: Calculation diagram of cost difference (Δ) matrix.

The operation of swapping the ith and jth elements in a sequence $S \in \mathbb{Z}^n$, denoted by $\rtimes$, is mathematically defined as the following function:

$$
\begin{aligned}
S' &= SWAP(S, i, j) = \tbinom{i}{j} \rtimes S \in \mathbb{Z}^n, \\
S'_k &= \begin{cases} S_i & \text{if } k = j \\ S_j & \text{if } k = i \\ S_k & \text{otherwise} \end{cases} \quad , \forall k \in \{1, .., N\}.
\end{aligned}
\tag{6.12}
$$

Next, we first show how to calculate the difference in the objective function with a single swap operation efficiently, and then present the details of the proposed steepest descent based algorithm to LWGM.

Cost Difference Matrix (Δ)

Between iterations, we need to incrementally update the objective function. This can be done by calculating a $N \times N$ symmetric matrix defined as follows:

$$\Delta^S = \begin{bmatrix} 0 & \dots & \delta_{1n} \\ \vdots & \ddots & \vdots \\ \delta_{n1} & \dots & 0 \end{bmatrix}, \delta_{ij} = \mathcal{O}(\substack{i \\ j} \rtimes S) - \mathcal{O}(S), \tag{6.13}$$

where $\mathcal{O}$ is the objective function in (6.10).

Starting from an initial seed S_0, calculating Δ^{S_0} involves two terms: pairwise, and pointwise. Since `swap` modifies two rows of $\mathbf{\Pi}$, only two rows in $\mathbf{A}_G$ and $\mathbf{A}_H$ are included in the pairwise term. Similarly, the pointwise term only depends on four elements of $\mathbf{L}$. Accordingly, the equation (6.13) can be expanded as,

$$\delta_{ij} = 2\alpha' \left(\mathbf{A}_G^{i,:} - \mathbf{A}_G^{j,:}\right)^T \left(\mathbf{A}_H^{i,:} - \mathbf{A}_H^{j,:}\right) + (1-\alpha')\left(L_{ii} + L_{jj} - L_{ij} - L_{ji}\right) \tag{6.14}$$

Figure 6.4 shows the schematic view of the above expansion. Calculating Δ^{S_0} using (6.14) takes $O(N^3)$ time. However, subsequent updates of Δ can be done with much lower complexity. This is because most of the elements in Δ can be updated based upon the Cartesian Product of the columns i and j in $\mathbf{A}_G$ and $\mathbf{A}_H$, except for the row or column corresponding to the indices of the latest `swap`. As a result, the complexity of subsequent updates is $O(N^2)$. Details of updating Δ after the initial step are presented in Algorithm 3.

Algorithm 3: Updating Δ after Swapping i and j

input : Δ^{old}, i, j
output: Δ^{new}

$\vec{c}_G \leftarrow \mathbf{A}_G^{:,i} - \mathbf{A}_G^{:,j}$;
$\vec{c}_H \leftarrow \mathbf{A}_H^{:,i} - \mathbf{A}_H^{:,j}$;
$\vec{r}_G \leftarrow \mathbf{A}_G^{i,:} - \mathbf{A}_G^{j,:}$;
$\vec{r}_H \leftarrow \mathbf{A}_H^{i,:} - \mathbf{A}_H^{j,:}$;
$\Delta^{new} \leftarrow \Delta^{old} - 2(\vec{c}_G - \vec{c}_H) \bigotimes (\vec{r}_G - \vec{r}_H)$;
for $k \leftarrow 1$ **to** N **do**
  $\delta_{k,i}^{new}, \delta_{i,k}^{new}, \delta_{k,j}^{new}, \delta_{j,k}^{new} \leftarrow$(6.14)
return Δ^{new}

Steepest Descent Search

To explore the search space by `swap`, we utilize multi-start Steepest Descent Search. It starts with a random seed (sequence) and moves iteratively until the objective function cannot be further reduced. In each iteration, the sequence is replaced by the neighbor with the minimum objective value. To evaluate quickly all neighbors that are reachable in one `swap`, we compute the respective difference cost matrix following Algorithm 3. Once the neighbor with the maximum decrement is identified, Δ is updated accordingly for subsequent iterations. The exploration steps of each node is summarized in lines 6 to 10 of algorithm 4. To increase the chance to reach a global minimum, we run this method from multiple starting points and return the best result among them.

As it can be seen in line 5 of the algorithm 4, the exploration terminates when either of two conditions is met. First, the algorithm terminates when there exists no more swap that can reduce the object value. Second, after running the algorithm for at least N swaps, no further swap is performed if the maximum decrements in the objective value in one swap is no more than a threshold *Threshold*.

Algorithm 4: Steepest Descent Search Method for Weighted Graph Matching

input : The number of nodes (n), The number of starting points (C_p)
output: The point with the minimum objective value among the explored ones

for $k \leftarrow 1$ **to** C_p **do**
  Start with a random permutation P^k;
  Initialize $\mathcal{O}_P^k$ with (6.10) and Δ_P^k with (6.14);
  $i \leftarrow 1$;
  while ($i <= n$ ***and*** $\min(\Delta) > 0$) ***or*** ($i > n$ ***and*** $\min(\Delta) > threshold$) **do**
    $\mathcal{O}_P^k \leftarrow \mathcal{O}_P^k + \min\left(\Delta_P^k\right)$;
    $index_1, index_2 \leftarrow \arg\min_{i,j}\left(\Delta_P^k\right)$;
    $P^k \leftarrow SWAP(P^k, index_1, index_2)$;
    $\Delta_P^k \leftarrow UpdateDelta(\Delta_P^k, index_1, index_2)$;
    $i \leftarrow i + 1$;
  Append $(P, \mathcal{O}_P)^k$ to $\mathcal{L}$ ▷ The list of points results;
return $\min_{\mathcal{O}_P}(\mathcal{L})$;

The reason for running the algorithm for at least N swaps is due to the fact as stated in Proposition 6.3.2.

Proposition 6.3.2. *Given two arbitrary sequences $S_1, S_2 \in \mathbb{Z}^N$, it is possible to convert S_1 to S_2 within N `swap` operations.*

Proof. In general, any source sequence $S = \langle s_0 s_1 ... s_{n-1} \rangle$ is convertible to any destination sequences Q by swapping the members of the source with the member

of destination that holds the corresponding value in indexing order:

$$
\begin{aligned}
1) \quad & {}^{0}_{index(s_0,Q)}\rtimes S && \rightarrow X^0 \\
2) \quad & {}^{1}_{index(s_1,Q)}\rtimes X^0 && \rightarrow X^1 \\
& ... \\
i+1) \quad & {}^{i}_{index(S_i,Q)}\rtimes X^i && \rightarrow X^{i+1} \\
& ... \\
n-2) \quad & {}^{n-3}_{index(S_{n-3},Q)}\rtimes X^{n-3} && \rightarrow X^{n-2} \\
n-1) \quad & {}^{n-2}_{n-1}\rtimes X^{n-2} && \rightarrow Q
\end{aligned}
$$

where $index(x, Y)$ shows which member of Y holds the value x. For example, the first SWAP to be transform into the sample destination sequence $Q = \langle q_0...q_i(= s_o)...q_{n-1}\rangle$ is $(0 \leftrightarrow i)$. The total number of above swaps is $n-1$. □

In other words, by enforcing at least N swaps if the first condition is not met, we allow the algorithm to "sufficiently" explore the search space so that it is unlikely to be trapped in local optimal prematurely. After n iterations, the termination condition is changed to a positive threshold on the minimum of Δ. This threshold sets an upper bound on the iteration counts and ensures convergence in polynomial time.

Complexity

Recall that the complexity of computing the initial Δ^{S_0} is $O(N^3)$ while updating Δ in subsequent iterations takes $O(N^2)$. The worst case number of iterations is given by $O(N+C_r/Threshold)$, where C_r is a constant that represents the difference

between the maximum and the minimum possible objective values in the search space. Let C_p be the number of starting points in the multi-start steepest descent. The overall complexity of the proposed algorithm is thus,

$$\underbrace{O(C_p N^3)}_{Initialization} + \underbrace{O\left(C_p(N + C_r/Threshold)N^2\right)}_{Exploration} \equiv O(N^3) \tag{6.15}$$

The algorithm can be easily parallelized for further acceleration for the following reasons:

- All the members of Δ can be initialized independently in parallel threads.
- The Cartesian product in updating Δ can be computed as a block by vector processors such as GPUs.
- In each iteration, every particle can move independently based on all the information from the previous round.

The only remaining serial operations are the iterations of exploration, which can be done in $O(N)$.

6.3.3 Localization Process

The complete sensor mapping procedure takes four sets of data as input:

1. Sensor locations
2. Pair-wise RSSI measurements between the sensors

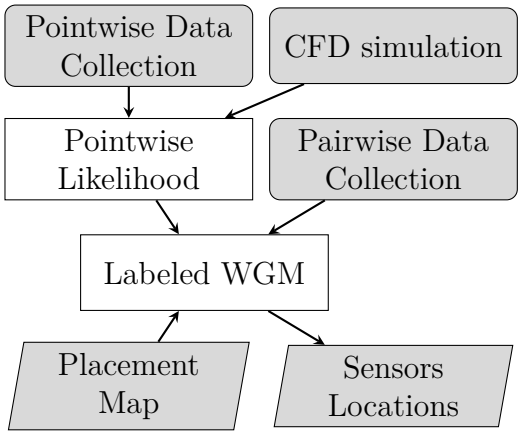


Figure 6.5: Flowchart of the thermal piloting process

3. Temperature readings of the sensors in selected scenarios

4. CFD simulation results of the selected scenarios to build the pointwise likelihood model.

It returns the mapping between the locations and the sensors. Figure 6.5 shows the steps of the procedure. Collected measurement data and the CFD simulation results provide the pointwise similarity of experimental and analytical graphs. Meanwhile, the collected RSS data between the sensors in accordance with the placement map lead to the pairwise dissimilarities of the two graphs. Given the pointwise and pairwise distances, running the proposed heuristic algorithm on two graphs gives the estimated mapping between them.

6.4 Experimental Setup

Although the proposed methodology is conceptually applicable to all DCs with DCIM, we provide a concrete implementation based on modular DCs, where racks, IT servers, and cooling units are isolated in an enclosure. In this type of DCs, air

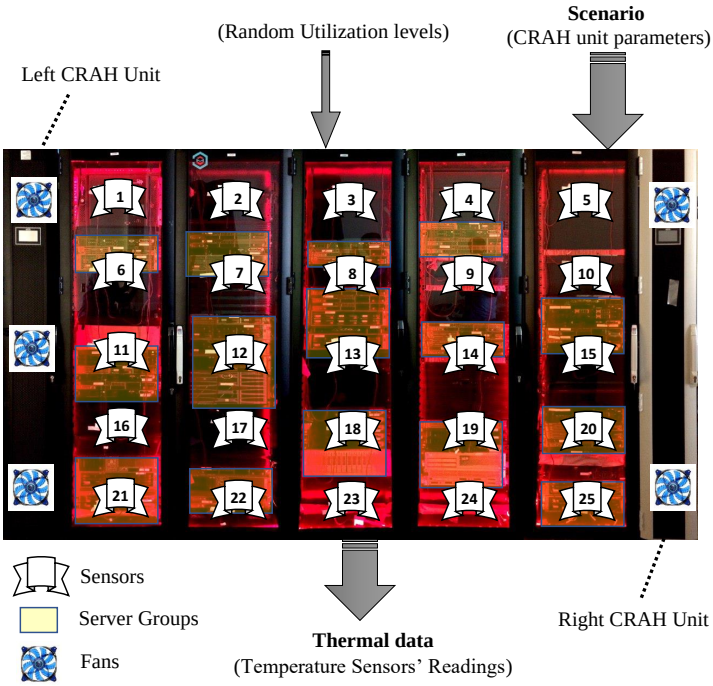


FIGURE 6.6: Experiment testbed, a modular DC with five racks and two cooling units. In the first test, 25 sensors were monitoring the front of the servers.

handlers are often located on the sides with racks and servers in the middle. The cold aisle is isolated from the hot aisle with little leakage between them except for through the fans of air conditioners.

6.4.1 Testbed Data Center

As illustrated in Figure 6.6, our testbed modular DC consists of five side-by-side Rittal TS IT enclosures, each of dimension $2m \times 0.6m \times 1m$ ($H \times W \times D$). Two in-row Rittal liquid cooling units have been installed on two ends. The left cooling unit has three fans installed at the top, middle, and bottom fan slots. The right cooling unit has two fans installed on the top and bottom fan slots.

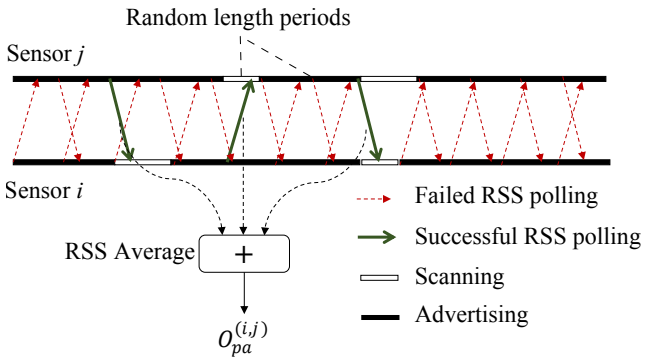


FIGURE 6.7: The random switching strategy to collect RSS measurements between pairs of sensors. Sensors alternate between scanning and advertising periods.

In testbed, the racks host groups of servers at different heights. The remaining empty space in the racks is covered by blanking panels to prevent cold air leakage or hot air re-circulation. The sensors monitor and capture the thermal conditions of the cold aisle every 30 seconds. The reason for placing sensors only on the cold aisle sides is two-fold: First, hot aisle temperatures are more sensitive to IT workloads. Second, cold aisle temperatures are more indicative of cooling system performance. ASHRAE guidelines specify the acceptable operating temperature range to be from 18 ^oC to 27 ^oC[5].

6.4.2 Pairwise Data Collection

Pairwise RSS measurements are taken and then transmitted to a server via a Bluetooth Low Energy (BLE) gateway. Each sensor collects at least 30 RSS samples received from the other sensors. Afterward, pointwise data collection is launched. In the evaluation, the pairwise objective weighted (α') is set to 0.02.

The wireless sensors are featured with TI2640 transceivers and communicate

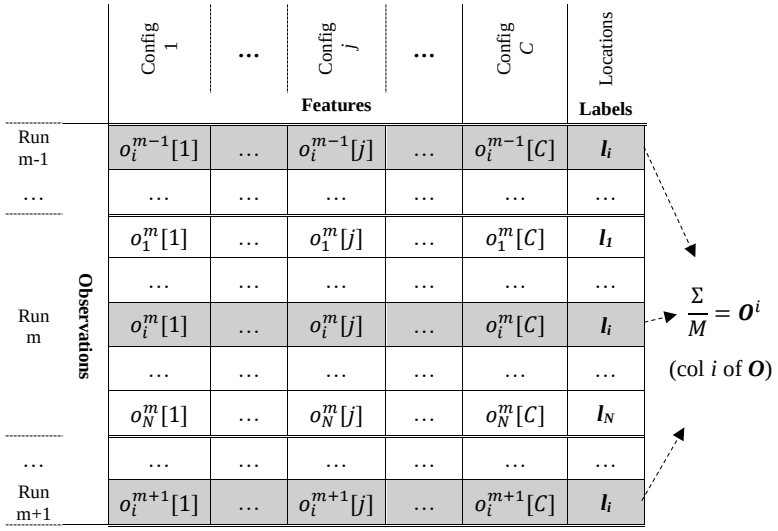
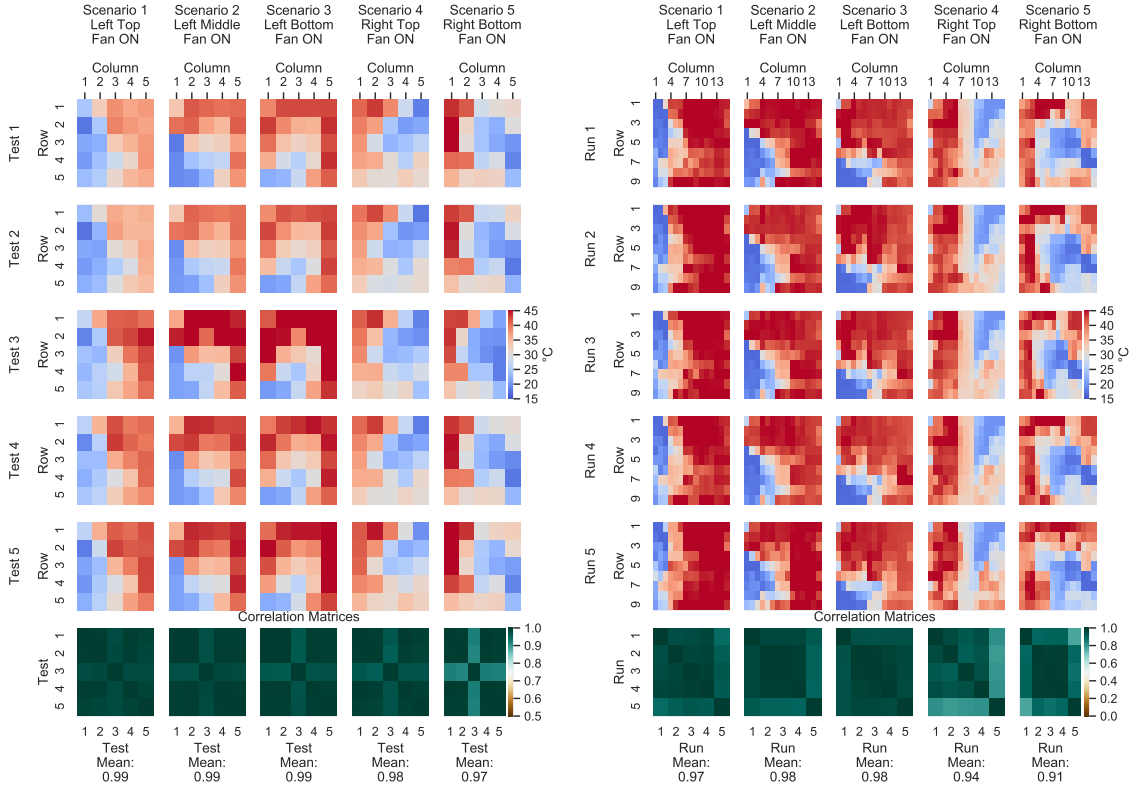


FIGURE 6.8: Data model for the pointwise data. $T^z_{x,y}$ is the temperature of measured by sensor x in the steady state of scenario y at run z.

with one another via the BLE protocol. To measure pairwise RSS, sensors need to take turns to transmit broadcast messages and extract RSS from received signals from others. An energy-efficient schedule for RSS pairwise measurements can be devised based on optimal block design [100]. Here, we implement a simple randomized strategy for ease of implementation. In the approach, each sensor transmits multiple advertisement messages for a random duration (called advertising periods) and then switches to a listening mode to receive broadcasts from the neighbor sensors for some time (called scanning periods). This is repeated several runs on each sensor. As long as the advertising period is longer than the listening period, with random starting times, it can be proven that there exists overlapping between the advertising and listening periods of any two neighbor sensors [100].

Pairwise RSS measurements are sent to a server for further processing. In particular, we take an average of K measurements and use them to instantiate



(A) Experiment (B) Simulation

FIGURE 6.9: Experimental and simulation datasets; Cold side thermal maps from designated scenarios in different tests.

$\mathbf{A}_H = \mathbf{A} - c_1 \mathbf{J}_N$, where c_1 can be estimated from sensor calibration.

6.4.3 Pointwise Data Collection

Pointwise data is comprised of the measurements of the temperature sensors in selected scenarios. Each scenario corresponds to a unique cool unit configuration such as a selected number of fans to be active, setpoint temperatures, airflow rates, etc. At a given sensor location i, temperature measurements in the scenarios form an observation (feature) vector. We found in [41], according to Chi-squared tests, the most discriminative scenarios are those with one cooling unit ON with 100%

rpm and all the others OFF. Therefore, given three fans in the modular DC, each observation vector is of dimension 5.

In data collection, we run each scenario multiple times with different utilization levels in the servers. until steady states are reached and then collect temperature measurements from the sensors. Figure 6.8 illustrates the process. For each server, the utilization changes in random periods between 10 to 600s, to a random value between 0 to 100%. Figure 6.9a shows the thermal maps in different scenarios (columns) and different runs (rows). The last row in the figure presents the correlation matrices between different scenarios (with different server utilization) for the same scenario.

6.4.4 CFD Simulation

To estimate the parameter μ in the pointwise model in (6.8), three-dimensional CFD simulation is performed in ANSYS Fluent using realizable $k-\epsilon$ turbulent model. [29]. CFD studies fluid behavior through numerical modeling that involves solving equations based on physical laws, including mass conservation, momentum balance, and energy equations[46]. With predefined boundary conditions, the CFD modeling simulates the complex fluid and thermal behaviors and shows the velocity, pressure, and temperature distribution throughout the entire modeling regions. CFD is widely used in the field of aerospace, combustion, civil and environmental engineering, biomedical modeling, etc. In DC applications, CFD is often used during the design stage for capacity and floor planning. Similar to collecting sensor measurements, steady-state CFD simulations are executed for each selected scenario over multiple runs with different server utilizations. The results

will be used to estimate the parameters in γ using maximum likelihood estimation.

Due to the dynamic changes of server loads, the reported temperatures at the monitoring points are the average temperatures over the last 100 iterations. The results of the predicted values under five scenarios are illustrated in Figure 6.9b. To this end, we present the procedure to collect both pointwise and pairwise measurement data for a modular DC as well as to estimate the parameters in the data models described in Section 6.3.1.

6.5 Performance Evaluation

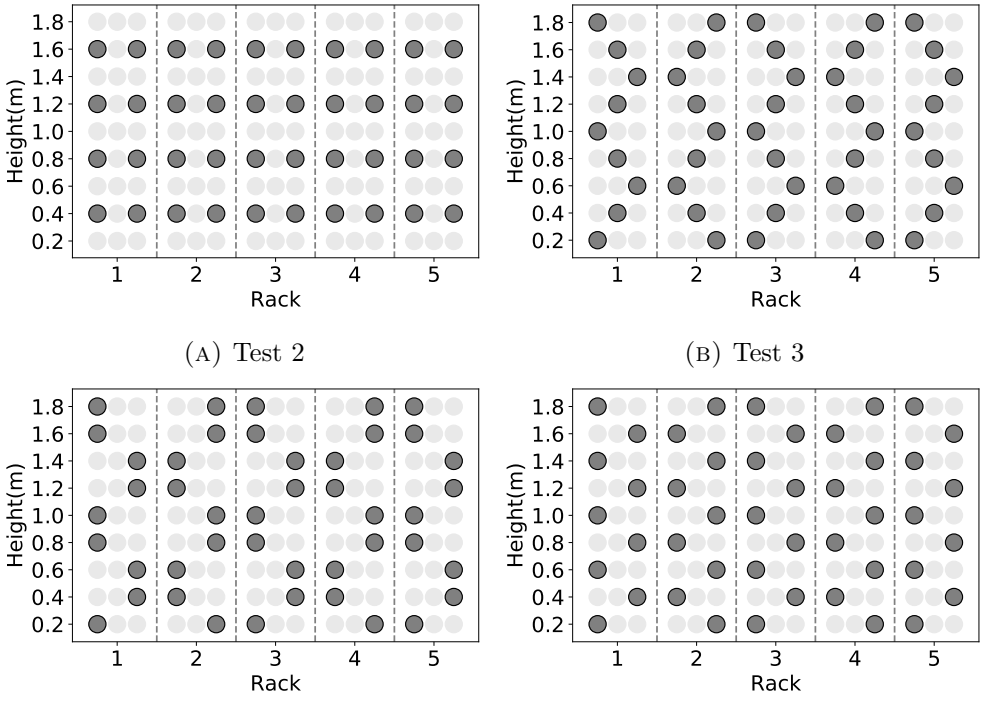
To evaluate the proposed solution, 25 to 45 sensors have been installed on the racks of a modular DC in five different test cases. The evaluation aims to verify the validity of the proposed method and the integrity of data models.

6.5.1 Ground Truth

In the first test, 25 sensors are placed in a 5×5 grid, while in the remaining four tests more sensors are installed with different patterns: 45 sensors for tests 2, 3, and 4, and 40 sensors for test 5. Figure 6.6 and Figure 6.10 show the sensor placement in test 1, and in the other tests, respectively.

6.5.2 Accuracy of Thermal Estimation

In this section, we evaluate the fidelity of CFD simulations by comparing measurement results from predicted values from CFD. Comparing Figure 6.9a and Figure 6.9b, we observe similar trends in the thermal maps for the same scenario.



(a) Test 2

(b) Test 3

(c) Test 4

(d) Test 5

Figure 6.10: Sensor placement in tests 2 to 5. The locations are indexed in row-major order from upper most left.

However, there exists discrepancy in the temperature values reported from the experiments and CFD simulations especially pronounced in scenarios 3 and 5. To gauge the discrepancy between experiments and CFD simulations, we use PCC again. Let $x^e_{i,j,k}$ and $x^s_{i,j,k}$ be temperature values at ith location of the jth scenario in the kth scenario from experiments and simulations, respectively. The correlation efficient for jth scenario and kth scenario $Corr_{j,k}(e,s)$ is given by,

$$\frac{\sum_i (x^e_{i,j,k} - \sum_i x^e_{i,j,k})(x^s_{i,j,k} - \sum_i x^s_{i,j,k})}{\sqrt{\sum_i (x^e_{i,j,k} - \sum_i x^e_{i,j,k})^2}\sqrt{\sum_i (x^s_{i,j,k} - \sum_i x^s_{i,j,k})^2}} \tag{6.16}$$

In other words, $Corr_{j,k}(e,s)$ captures the *spatial* similarity. The results are

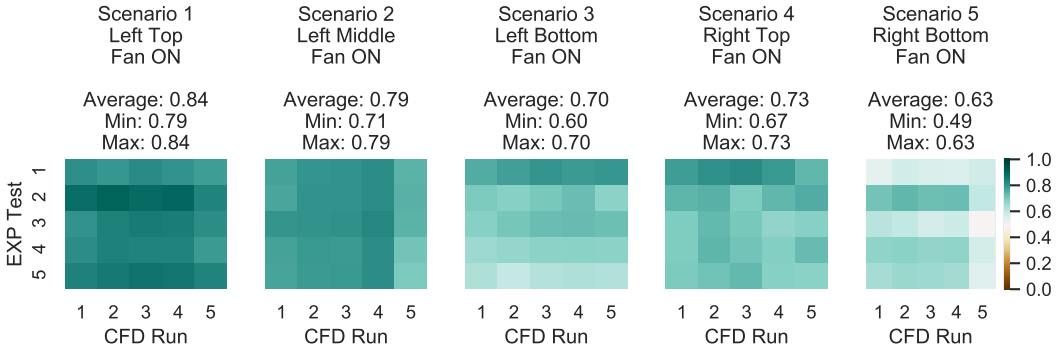


FIGURE 6.11: Correlation between experiments and CFD simulations

illustrated in Figure 6.11. The large discrepancy in scenarios 3 and 5 may be attributed to the uncertain characterization of the boundary conditions on the right side of the modular DC that was not considered in CFD simulations. High-fidelity CFD modeling is an active area of research and is outside the scope of this paper.

6.5.3 Performance of the Proposed WGM Algorithm

To compare the performance of the proposed solution, we have implemented two other well-known methods for WGM as baselines:

1. The first one is PATH [97], a representative optimization-based algorithm. It formulates the problem as convex-concave optimization and solves it by following a path from the solution of the convex problem to an estimation of the concave one.

2. The second baseline algorithm is Umeyama [90], a spectral algorithm. It runs bipartite matching between the eigenvectors of the adjacency matrices of two input graphs.

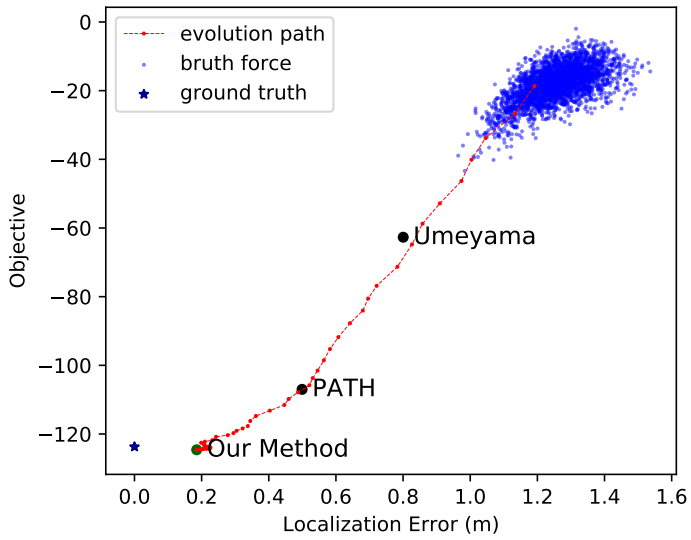


FIGURE 6.12: The objective value and the localization error from the baseline methods versus our method in test 2 [2]

Figure 6.12 shows the localization errors and objective values achieved by different approaches in test 2. The solid dots are the final solution upon convergence. We can see that the proposed method can indeed achieve a lower localization error and lower objective value. We also plot in blue the results from 3000 random permutations in a brute force method. Clearly, brute force is unlikely to find a good solution in a limited number of permutations since the search space is large (e.g., of size 40!). Also shown in Figure 6.12 in a red trajectory are the intermediate results of the proposed method in each iteration. Interestingly, by comparing the result obtained by our method and the ground truth, we find that though the localization error of our method is larger, the objective value is in fact smaller. On one hand, this indicates that the proposed method can indeed approach very low objective values. On the other hand, the data models used are approximations of

[1]The objective values of all the plots in this paper have -200 unit offset for better representation

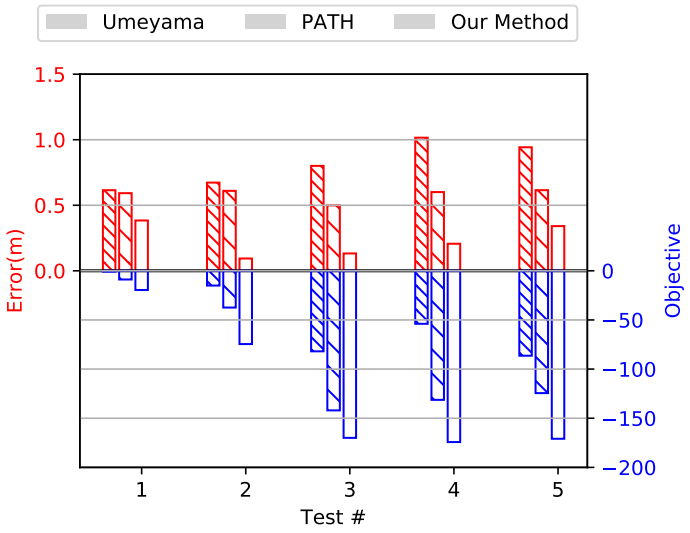


FIGURE 6.13: Comparison between the objective values and the localization errors from the baseline methods versus those from our method in all tests

the real processes, and thus the objective function is only a close surrogate of the true objective.

By comparison, a spectral method such as Umeyama is not robust against noise. Furthermore, when sensor placements are regular and the resulting graphs are rigidly transformable by mirroring, flipping, or rotating, the eigendecomposition of their adjacency matrices contain zero eigenvalues. The corresponding graph matching problem becomes under-determined.

The main idea behind PATH and many continuous optimization methods for WGM is to solve the problem for Doubly Stochastic matrices and find the closest permutation ($\mathbf{\Pi} \in \mathcal{P}$) to the solution. Although PATH tries to gradually reach the integer solution instead of in one shot, it is still prone to be trapped in a permutation that is far from optimal when the difference between the real solution

TABLE 6.1: Running time of different algorithms

nodes	algorithm		
	PATH	Umeyama	Ours
25	$< 1s$	$< 1s$	6.5s
40	$< 1s$	1s	25s
45	$< 1s$	2s	36s

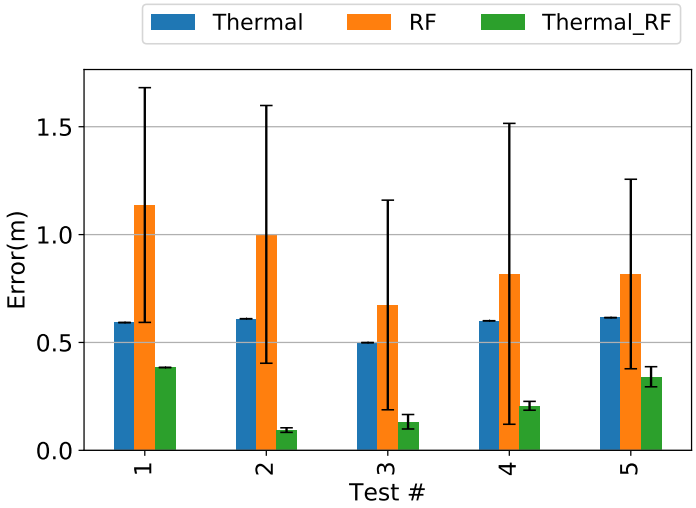


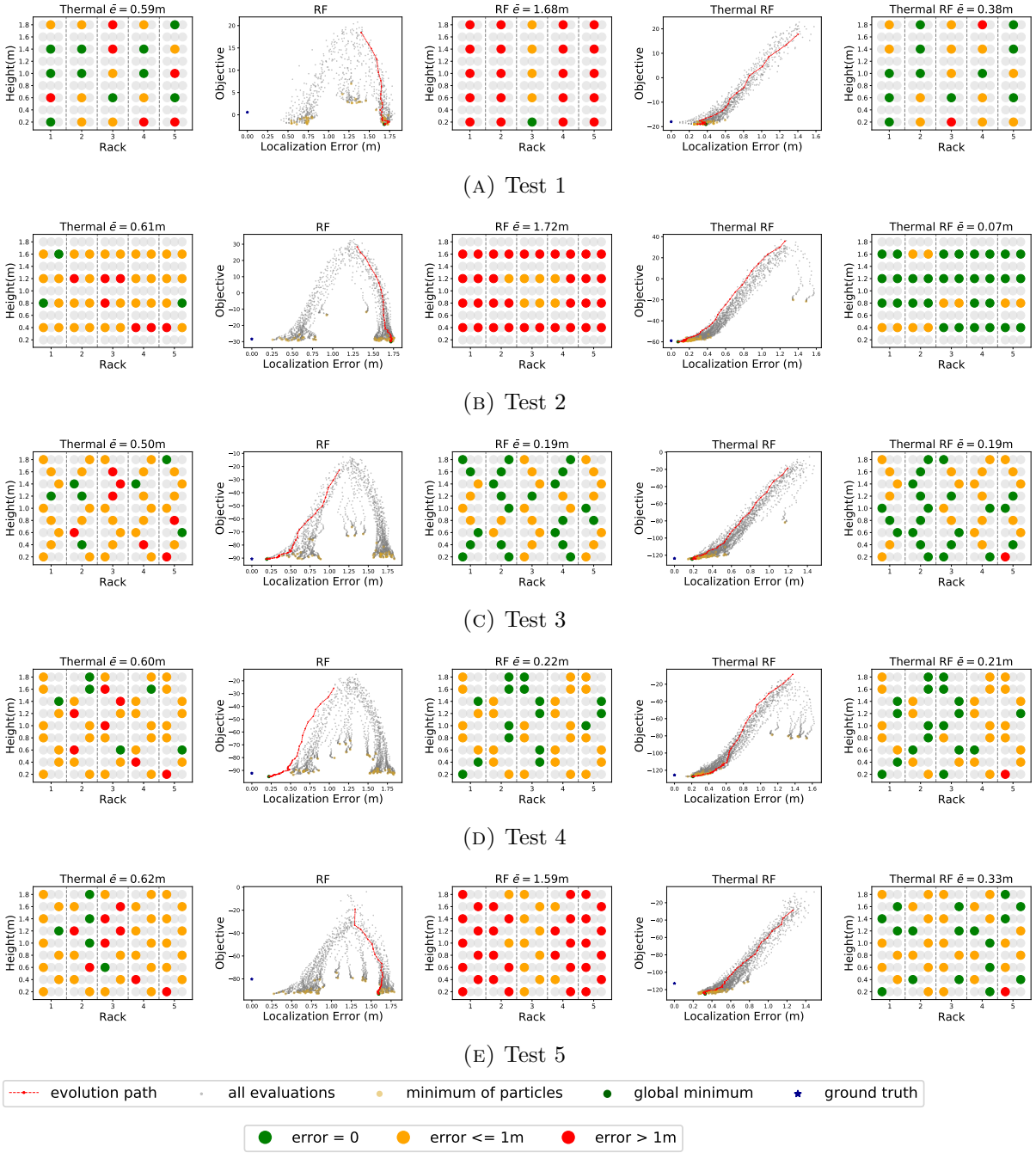
FIGURE 6.14: The effect of using the pointwise inputs along with pairwise inputs on the localization error in different tests

and the integer solution is too much.

Table 6.1 summarizes the running time of different algorithms in a X64-based PC with Intel(R) Core(TM) i7-4790 CPU on Python 3. We can see that the proposed method is slower than the baseline. As discussed in Section 12, the method can be easily parallelized to reduce the running time.

6.5.4 Single Modality versus Multimodality

We investigate the effects of utilizing multimodality measurements in sensor mapping. Comparisons are made among three approaches, thermal piloting using



(A) Test 1

(B) Test 2

(C) Test 3

(D) Test 4

(E) Test 5

Figure 6.15: The final localization results and the graph matching objective values for different tests.

thermal-only (pointwise) measurements, using only RF (pairwise) measurements, and multimodality using combined thermal and RF measurements. In the case of thermal-only, maximum bipartite matching is applied to solve for the optimal solution [41].

Figure 6.14 shows the range of the localization errors from 150 runs. We observe that the multimodality approach outperforms the two single-modality ones. Furthermore, the error bars show that the method using pairwise RF measurements suffer from large variants. However, they are complementary to the thermal measurements.

The main reason for high error variances with RF measurements only is that when the placement of the nodes are regular and the graphs have rigid transformations, it is not possible to distinguish from flipped or mirrored version the ground truth placement. To illustrate this problem, the detailed output of one sample run is represented in Figure 6.15, including their error-vs-objective plots along with their final results. The evolution paths of the particles in RF-only approaches are bifurcated into two groups of minimums. The left group contains those close to the ground truth, and the right branch leads to the rigid transformations of the ground truth. The particle may trap in both sides since there is no identifier in pairwise input to distinguish the branches. Pointwise thermal data can be seen like a pin that protects our WGM solution from trapping in the local minimums of the rigid transformations. Figure 6.14 and 6.15 also show that the performance of thermal only always falls behind cases that utilize additional RF data. The pairwise RF data helps to overcome the error-prone thermal data and recovers the nodes that are mislocalized relatively far from the ground truth.

6.6 Conclusion

In this chapter, two novel approaches for automated sensor mapping were proposed to accelerate Data Center Wireless Sensor Network (DCWSN) commissioning. The first approach was thermal piloting, in which temperature sensors are localized using their own measurements. As demonstrated in experimental results the localization of thermal piloting is promising, though highly dependent on the accuracy of the CFD simulation.

In the next step, we improved the accuracy of the thermal piloting by incorporating pairwise RF measurements. The problem has been reformulated as WGM between an analytical and an experimental graph. Evaluation results showed that considerations of multimodal data can lead to 50% less localization error than single modal approaches on average. Furthermore, a novel swapping based method was proposed for WGM, which is 30% more accurate than the current solutions.

Chapter 7

Conclusion and Future Work

The main outcome of this research is a reliable, low-power, adaptable, and easy to commission Bluetooth Low Energy (BLE) base network architecture for Data Center Wireless Sensor Networks (DCWSNs). Toward this goal, we designed a hierarchical network, named Low Energy Monitoring Network (LEMoNet). Its performance is analyzed by an analytical model and is further improved by a Software Defined Networking (SDN) based framework, named SoftBLE. Deployments can be accelerated by two automated sensor mapping approaches, named thermal piloting and multimodal sensor localization.

It can be inferred from analytical results of LEMoNet that the protocol can support a network of 4800 sensors at 15s duty cycles with 0.95% Packet Reception Rate (PRR). With the introduction of SoftBLE, PRR can be guaranteed at 99.9% while the 70% of the energy is saved. Furthermore, the sensors were automatically mapped to their installed locations with acceptable accuracy using their own measurements. In an experimental modular DC, 60 cm and 20 cm RMSE for locations

are achieved by thermal piloting and the proposed multi-modality sensor mapping approach respectively.

Although the performances of the above solutions have been validated mostly in data center (DC) environments, all the suggested frameworks are applicable in other applications as well. Both the proposed 2-tier BLE based network, and SoftBLE framework can be deployed in any Industrial IoT (IIoT) for monitoring applications with fixed sensors such as in Structural Health Monitoring (SHM) or soil quality monitoring in agriculture. Also, the proposed method for sensor mapping is extendable to localize wireless sensors in other environments such as BLE-enabled light fixtures in smart buildings.

Furthermore, the algorithms and methods can be extended to solve other problems in different contexts. For example, the multi-gateway packet reception analysis can be applied to similar hierarchical networks, such as what was done for analyzing LoRaWAN [66]. The simulation module of legacy advertising in BLE can be used to evaluate arbitrary topologies of BLE based networks. Finally, the proposed method for Weighted Graph Matching (WGM) problem in multi-modal sensor mapping, can be implemented in vectorized processing units and solve Quadratic Assignment Problem (QAP) in object tracking or in Natural Language Processing (NLP).

Bibliography

[1] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 2015.

[2] H.A. Almohamad and Salih O. Duffuaa. A linear programming approach for the weighted graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):522–525, 1993.

[3] Kiyomichi Araki. Fundamental problems of nation-wide mobile radio telephone system. *Review of the Electrical Communications Laboratories*, 16(5-6):357, 1968.

[4] Julio Araujo, Jean-Claude Bermond, Frédéric Giroire, Frédéric Havet, Dorian Mazauric, and Remigiusz Modrzejewski. Weighted improper colouring. *Journal of Discrete Algorithms*, 16:53–66, 2012.

[5] ASHRAE. Data center power equipment thermal guidelines and best practices, June 2016.

[6] Michael Baddeley, Usman Raza, Aleksandar Stanoev, George Oikonomou, Reza Nejabati, Mahesh Sooriyabandara, and Dimitra Simeonidou. Atomic-SDN: Is synchronous flooding the solution to software-defined networking in IoT? *IEEE Access*, 7:96019–96034, 2019.

[7] Adarsh Prasad Behera, Abhishek Singh, Shekhar Verma, and Manish Kumar. Manifold learning with localized procrustes analysis based WSN localization. *IEEE Sensors Letters*, 4(10):1–4, 2020.

[8] GC Bell. Wireless sensors improve data center energy efficiency. *US Dept. Energy Lawrence Berkeley Nat. Lab*, 2010.

[9] Alex Benik and B Ventures. The sorry state of server utilization and the impending post-hypervisor era. *Gigaom Res*, 2013.

[10] Samaresh Bera, Sudip Misra, Sanku Kumar Roy, and Mohammad S Obaidat. Soft-WSN: Software-defined WSN management system for IoT applications. *IEEE Systems Journal*, 12(3):2074–2081, 2016.

[11] Samaresh Bera, Sudip Misra, and Athanasios V Vasilakos. Software-defined networking for internet of things: A survey. *IEEE Internet of Things Journal*, 4(6):1994–2008, 2017.

[12] Horst Bunke and Kaspar Riesen. Towards the unification of structural and statistical pattern recognition. *Pattern Recognition Letters*, 33(7):811–825, 2012.

[13] Terry Caelli and Serhiy Kosinov. An eigenspace projection clustering method for inexact graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):515–519, 2004.

[14] Peter J Cameron and Taoyang Wu. The complexity of the weight problem for permutation groups. *Electronic Notes in Discrete Mathematics*, 28:109–116, 2007.

[15] Ming-Ching Chang and Benjamin B Kimia. Measuring 3D shape similarity by graph-based matching of the medial scaffolds. *Computer Vision and Image Understanding*, 115(5):707–720, 2011.

[16] Bo-Ren Chen, Shin-Ming Cheng, and Jia-Jhun Lin. Energy-efficient BLE device discovery for Internet of Things. In *2017 Fifth International Symposium on Computing and Networking (CANDAR)*, pages 75–79. IEEE, 2017.

[17] Jinzhu Chen, Rui Tan, Yu Wang, Guoliang Xing, Xiaorui Wang, Xiaodong Wang, Bill Punch, and Dirk Colbry. A high-fidelity temperature distribution forecasting system for data centers. In *Real-Time Systems Symposium (RTSS), 2012 IEEE 33rd*, pages 215–224. IEEE, 2012.

[18] Maxim Chernyshev, Zubair Baig, Oladayo Bello, and Sherali Zeadally. Internet of things (IoT): Research, simulators, and testbeds. *IEEE Internet of Things Journal*, 5(3):1637–1647, 2017.

[19] C-F Chiasserini and Michele Garetto. An analytical model for wireless sensor networks with sleeping nodes. *IEEE Transactions on Mobile Computing*, 5(12):1706–1718, 2006.

[20] Dave Cole. Data center infrastructure management. *Data Center Knowledge*, 2012.

[21] Gaoxiang Cong, Jianxiong Wan, Tianyang Hua, Jie Zhou, and Hongxun Niu. A data center thermal monitoring system based on LoRa. In *5th International Conference on Computational Intelligence and Applications (ICCIA)*, pages 70–75. IEEE, 2020.

[22] Miyuru Dayarathna, Yonggang Wen, and Rui Fan. Data center energy consumption modeling: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):732–794, 2015.

[23] Jonathan de Carvalho Silva, Joel JPC Rodrigues, Antonio M Alberti, Petar Solic, and Andre LL Aquino. LoRaWAN—a low power wan protocol for Internet of Things: A review and opportunities. In *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, pages 1–6. IEEE, 2017.

[24] Pierre Delforge and Josh Whitney. Data center efficiency assessment-scaling up energy efficiency across the data center industry: Evaluating key drivers and barriers. *Natural Resources Defense Council*, 2014.

[25] A.Y. Devadhanishini, R.K. Malasri, N. Nandinipriya, V. Subashini, and P.G. Padma Gowri. Smart power monitoring system using IoT. In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pages 813–816. IEEE, 2019.

[26] Bing Dong, Vishnu Prakash, Fan Feng, and Zheng O'Neill. A review of smart building sensing system for better indoor environment control. *Energy and Buildings*, 199:29–46, 2019.

[27] Matteo Fischetti, Chiara Lepschy, Giuseppe Minerva, Giorgio Romanin-Jacur, and Ema Toto. Frequency assignment in mobile radio systems using branch-and-cut techniques. *European Journal of Operational Research*, 123(2):241–255, 2000.

[28] Pasquale Foggia, Gennaro Percannella, and Mario Vento. Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence*, 28(01):1450001, 2014.

[29] Yogesh Fulpagare and Atul Bhargav. Advances in data center thermal management. *Renew. Sustain. Energy Rev.*, 43:981–996, 2015.

[30] Laura Galluccio, Sebastiano Milardo, Giacomo Morabito, and Sergio Palazzo. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for wireless sensor networks. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 513–521. IEEE, 2015.

[31] Seungryeol Go and Jong-Wha Chong. Improved TOA-Based localization method with BS selection scheme for wireless sensor networks. *Etri Journal*, 37(4):707–716, 2015.

[32] Piyush Gupta and Panganmala R Kumar. The capacity of wireless networks. *IEEE Transactions on information theory*, 46(2):388–404, 2000.

[33] Ismail Guvenc and Chia-Chin Chong. A survey on TOA based wireless localization and NLOS mitigation techniques. *IEEE Communications Surveys & Tutorials*, 11(3), 2009.

[34] Samuel W Halpern. Reuse partitioning in cellular systems. In *33rd IEEE Vehicular Technology Conference*, volume 33, pages 322–327. IEEE, 1983.

[35] Guangjie Han, Jinfang Jiang, Chenyu Zhang, Trung Q Duong, Mohsen Guizani, and George K Karagiannidis. A survey on mobile anchor node assisted localization in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 18(3):2220–2243, 2016.

[36] Thomas R Henderson, Mathieu Lacage, George F Riley, Craig Dowell, and Joseph Kopena. Network simulations with the ns-3 simulator. *SIGCOMM demonstration*, 14(14):527, 2008.

[37] Ángela Hernández-Solana, David Perez-Diaz-de Cerio, Antonio Valdovinos, and Jose Luis Valenzuela. Proposal and evaluation of BLE discovery process based on new features of Bluetooth 5.0. *Sensors*, 17(9):1988, 2017.

[38] Ke Hong, Shuo Yang, Zhiqiang Ma, and Lin Gu. A synergy of the wireless sensor network and the data center system. In *Mobile Ad-Hoc and Sensor Systems (MASS), 2013 IEEE 10th International Conference on*, pages 263–271. IEEE, 2013.

[39] Shitong Hou and Gang Wu. A low-cost IoT-based wireless sensor system for bridge displacement monitoring. *Smart Materials and Structures*, 28(8):085047, 2019.

[40] Xiong Hui, Chen Zhiyuan, Yang Beiya, and Ni Rongpei. TDOA localization algorithm with compensation of clock offset for wireless sensor networks. *China Communications*, 12(10):193–201, 2015.

[41] Mehdi Jafarizadeh, Peiying J Tsai, and Rong Zheng. Thermal piloting: A novel approach for sensor localization in data center monitoring. In

15th International Conference on Distributed Computing in Sensor Systems (DCOSS), pages 17–24. IEEE, 2019.

[42] Anil K Jain, Robert PW Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[43] Neeraj Jain, Shekhar Verma, and Manish Kumar. Adaptive locally linear embedding for node localization in sensor networks. *IEEE Sensors Journal*, 17(9):2949–2956, 2017.

[44] Wha Sook Jeon, Made Harta Dwijaksara, and Dong Geun Jeong. Performance analysis of neighbor discovery process in bluetooth low-energy networks. *IEEE Transactions on Vehicular Technology*, 66(2):1865–1871, 2016.

[45] Derek Justice and Alfred Hero. A binary linear programming formulation of the graph edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1200–1214, 2006.

[46] Takeo Kajishima and Kunihiko Taira. *Computational Fluid Dynamics*. Number 1. Springer International Publishing, Cham, 2017.

[47] David Knoke, Peter J Burke, and Peter Burke. *Log-linear models*, volume 20. Sage, 1980.

[48] Mikko Kohvakka, Mauri Kuorilehto, Marko Hännikäinen, and Timo D Hämäläinen. Performance analysis of IEEE 802.15.4 and ZigBee for large-scale wireless sensor network applications. In *Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks*, pages 48–57, 2006.

[49] Bing Li, Yigang He, Fengming Guo, and Lei Zuo. A novel localization algorithm based on ISOMAP and partial least squares for wireless sensor networks. *IEEE Transactions on Instrumentation and Measurement*, 62(2):304–314, 2012.

[50] Chenhe Li, Jun Li, Mehdi Jafarizadeh, Ghada Badawy, and Rong Zheng. LEMoNet; Low energy wireless sensor network design for data center monitoring. In *IFIP Networking Conference (IFIP Networking)*, pages 1–9. IEEE, 2019.

[51] Xiaomin Li, Di Li, Jiafu Wan, Chengliang Liu, and Muhammad Imran. Adaptive transmission optimization in SDN-based industrial Internet of Things with edge computing. *IEEE Internet of Things Journal*, 5(3):1351–1360, 2018.

[52] Chieh-Jan Mike Liang, Jie Liu, Liqian Luo, Andreas Terzis, and Feng Zhao. RACNET: A high-fidelity data center sensing network. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 15–28. ACM, 2009.

[53] Andreina Liendo, Dominique Morche, Roberto Guizzetti, and Franck Rousseau. BLE parameter optimization for IoT applications. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.

[54] Huaping Liu, Vinod Venkatesan, Curt Nilsen, Ron Kyker, and Mario E Magana. Performance of frequency hopped noncoherent GFSK in correlated rayleigh fading channels. In *Communications, 2003. ICC'03. IEEE International Conference on*, volume 4, pages 2779–2783. IEEE, 2003.

[55] Bingqing Luo, Jincheng Gao, and Zhixin Sun. Energy modeling of neighbor discovery in Bluetooth Low Energy Networks. *Sensors*, 19(22):4997, 2019.

[56] Bingqing Luo, Feng Xiang, Zhixin Sun, and Yudong Yao. BLE neighbor discovery parameter configuration for IoT applications. *IEEE Access*, 7:54097–54105, 2019.

[57] Bingqing Luo, Jia Xu, and Zhixin Sun. Neighbor discovery latency in Bluetooth Low Energy networks. *Wireless Networks*, 26(3):1773–1780, 2020.

[58] Rod Mahdavi and William Tschudi. Wireless sensor network for improving the energy efficiency of data centers. Technical report, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2012.

[59] Vittorio Maniezzo and Roberto Montemanni. *An exact algorithm for the min-interference frequency assignment problem*. Univ. di Bologna, Department of Computer Science, 2000.

[60] Kamal Mehdi, Massinissa Lounis, Ahcène Bounceur, and Tahar Kechadi. Cupcarbon: A multi-agent and discrete event wireless sensor network design and simulation tool. In *7th International ICST Conference on Simulation Tools and Techniques*, pages 126–131. Institute for Computer Science, Social Informatics and Telecommunications ..., 2014.

[61] Hetal P Mistry and Nital H Mistry. RSSI based localization scheme in wireless sensor networks: a survey. In *Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on*, pages 647–652. IEEE, 2015.

[62] S. S. Mohar, S. Goyal, and R. Kaur. A survey of localization in wireless sensor network using optimization techniques. In *4th International Conference on Computing Communication and Automation (ICCCA)*, pages 1–6, 2018.

[63] Yuri Murillo, Alessandro Chiumento, Brecht Reynders, and Sofie Pollin. An all-wireless SDN framework for BLE mesh. *ACM Transactions on Internet of Things*, 1(4):1–30, 2020.

[64] Mohammed Saleh Ali Muthanna, Ping Wang, Min Wei, Abdelhamied A Ateya, and Ammar Muthanna. Toward an ultra-low latency and energy efficient lorawan. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pages 233–242. Springer, 2019.

[65] D Nageswari, R Maheswar, and GR Kanagachidambaresan. Performance analysis of cluster based homogeneous sensor network using energy efficient N-policy (EENP) model. *Cluster Computing*, 22(5):12243–12250, 2019.

[66] Minming Ni, Mehdi Jafarizadeh, and Rong Zheng. On the effect of multi-packet reception on redundant gateways in lorawans. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[67] Sewoong Oh, Andrea Montanari, and Amin Karbasi. Sensor network localization from local connectivity: Performance analysis for the MDS-MAP algorithm. In *IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo)*, pages 1–5. IEEE, 2010.

[68] Fredrik Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. Cross-level sensor network simulation with cooja. In *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, pages 641–648. IEEE, 2006.

[69] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 51(8):2137–2148, 2003.

[70] Anup Kumar Paul and Takuro Sato. Localization in wireless sensor networks: A survey on algorithms, measurement techniques, applications and challenges. *Journal of Sensor and Actuator Networks*, 6(4):24, 2017.

[71] David Perez-Diaz de Cerio, Ángela Hernández, Jose Luis Valenzuela, and Antonio Valdovinos. Analytical and experimental performance evaluation of BLE neighbor discovery process including non-idealities of real chipsets. *Sensors*, 17(3):499, 2017.

[72] Mohammed Ziaur Rahman and Lindsay Kleeman. Paired measurement localization: A robust approach for wireless localization. *IEEE Transactions on Mobile Computing*, 8(8):1087–1102, 2008.

[73] K Ramesh and V Kannan. End-to-end delay analyses via LER in wireless sensor networks. In *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, pages 187–198. Springer, 2020.

[74] P Rekha, K Sumathi, S Samyuktha, A Saranya, G Tharunya, and R Prabha. Sensor based waste water monitoring for agriculture using IoT. In *2020*

6th International Conference on Advanced Computing and Communication Systems (ICACCS), pages 436–439. IEEE, 2020.

[75] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[76] Abusayeed Saifullah, Sriram Sankar, Jie Liu, Chenyang Lu, Ranveer Chandra, and Bodhi Priyantha. CapNet: A real-time wireless management network for data center power capping. In *Real-Time Systems Symposium (RTSS), 2014 IEEE*, pages 334–345. IEEE, 2014.

[77] Abusayeed Saifullah, Sriram Sankar, Jie Liu, Chenyang Lu, Ranveer Chandra, and Bodhi Priyantha. CapNet: Exploiting wireless sensor networks for data center power capping. *ACM Transactions on Sensor Networks (TOSN)*, 15(1):6, 2018.

[78] HA Baier Saip and Claudio Leonárdo Lucchesi. Matching algorithms for bipartite graph. *Relatorio Tecnico*, 700(03), 1993.

[79] Gerard Sanromà, René Alquézar, and Francesc Serratosa. A new graph matching method for point-set correspondence using the EM algorithm and softassign. *Computer Vision and Image Understanding*, 116(2):292–304, 2012.

[80] Christian Schellewald, Stefan Roth, and Christoph Schnörr. Evaluation of convex optimization techniques for the weighted graph-matching problem in computer vision. In *Joint Pattern Recognition Symposium*, pages 361–368. Springer, 2001.

[81] Gaoyang Shan and Byeong-hee Roh. Performance model for advanced neighbor discovery process in Bluetooth Low Energy 5.0-enabled IoT networks. *IEEE Transactions on Industrial Electronics*, 2020.

[82] Arman Shehabi, Sarah Smith, Dale Sartor, Richard Brown, Magnus Herrlin, Jonathan Koomey, Eric Masanet, Nathaniel Horner, Inês Azevedo, and William Lintner. United states data center energy usage report, 2016.

[83] Rathin Chandra Shit, Suraj Sharma, Deepak Puthal, and Albert Y Zomaya. Location of Things (LoT): A review and taxonomy of sensors localization in IoT infrastructure. *IEEE Communications Surveys & Tutorials*, 20(3):2028–2061, 2018.

[84] Krishna Pratap Singh, Rajat Bhai, Vishrut Mishra, Pradeep Nagar, and Jyoti Kasinayal. Localization in wireless sensor network using LLE-ISOMAP algorithm. In *IEEE Region 10 Conference*, pages 393–397. IEEE, 2017.

[85] Seung Whan Song, Youn Sang Lee, Fatima Imdad, Muhammad Tabish Niaz, and Hyung Seok Kim. Efficient advertiser discovery in Bluetooth Low Energy devices. *Energies*, 12(9):1707, 2019.

[86] William Stallings. *Wireless communications & networks*. Pearson Education India, 2009.

[87] Andrew S Tanenbaum et al. Computer Networks, 1996.

[88] Slavisa Tomic, Marko Beko, and Rui Dinis. 3-d target localization in wireless sensor networks using RSS and AOA measurements. *IEEE Transactions on Vehicular Technology*, 66(4):3197–3210, 2017.

[89] Mostafa Uddin, Sarit Mukherjee, Hyunseok Chang, and TV Lakshman. SDN-based service automation for IoT. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, pages 1–10. IEEE, 2017.

[90] Shinji Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.

[91] András Varga and Rudolf Hornig. An overview of the OMNET++ simulation environment. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, page 60. ICST (Institute for Computer Sciences, Social-Informatics and ..., 2008.

[92] Chengqun Wang, Jiming Chen, Youxian Sun, and Xuemin Shen. Wireless sensor networks localization with ISOMAP. In *2009 IEEE International Conference on Communications*, pages 1–5. IEEE, 2009.

[93] Juan Wang and Di Li. Adaptive computing optimization in software-defined network-based industrial Internet of Things with fog computing. *Sensors*, 18(8):2509, 2018.

[94] X Rosalind Wang, Joseph T Lizier, Oliver Obst, Mikhail Prokopenko, and Peter Wang. Spatiotemporal anomaly detection in gas monitoring sensor networks. In *European Conference on Wireless Sensor Networks*, pages 90–105. Springer, 2008.

[95] Yue Wang and KC Ho. Unified near-field and far-field localization for AOA and hybrid AOA-TDOA positionings. *IEEE Transactions on Wireless Communications*, 17(2):1242–1254, 2018.

[96] Chengbo Yu, Yanzhe Cui, Lian Zhang, and Shuqiang Yang. ZigBee wireless sensor network in environmental monitoring applications. In *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–5. IEEE, 2009.

[97] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2227–2242, 2008.

[98] Deze Zeng, Peng Li, Song Guo, Toshiaki Miyazaki, Jiankun Hu, and Yong Xiang. Energy minimization in multi-task software-defined sensor networks. *IEEE Transactions on Computers*, 64(11):3128–3139, 2015.

[99] Rong Zheng and Richard Barton. Toward optimal data aggregation in random wireless sensor networks. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pages 249–257. IEEE, 2007.

[100] Rong Zheng, Jennifer C Hou, and Lui Sha. Asynchronous wakeup for ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pages 35–45, 2003.

[101] Feng Zhou and Fernando De la Torre. Factorized graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1774–1789, 2015.

[102] J Arthur Zoeliner and C Lyle Beall. A breakthrough in spectrum conserving frequency assignment technology. *IEEE Transactions on Electromagnetic Compatibility*, (3):313–319, 1977.