

LEMoNet: Low Energy Wireless Sensor Network
Design for Data Center Monitoring

LEMONET: LOW ENERGY WIRELESS SENSOR NETWORK
DESIGN FOR DATA CENTER MONITORING

BY

CHENHE LI, B.Eng., Mechatronics Engineering
McMaster University, Hamilton, Canada

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTING AND SOFTWARE

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

© Copyright by Chenhe Li, 2018

All Rights Reserved

Master of Science (2018)
(Computing and Software)

McMaster University
Hamilton, Ontario, Canada

TITLE: LEMoNet: Low Energy Wireless Sensor Network Design
for Data Center Monitoring

AUTHOR: Chenhe Li
B.Eng., Mechatronics Engineering
McMaster University, Hamilton, Canada

SUPERVISOR: Dr. Rong Zheng

NUMBER OF PAGES: xii, 56

Abstract

Today's data centers (DCs) consume up to 3% of the energy produced worldwide, much of which is wasted due to over-cooling and underutilization of IT equipment. This wastage in part stems from the lack of real-time visibility of fine-grained thermal distribution in DCs. Wireless sensing is an ideal candidate for DC monitoring as it is cost-effective, facility-friendly, and can be easily re-purposed. In this thesis, we develop LEMoNet, a novel low-energy wireless sensor network design for monitoring co-location DCs. It employs a two-tier network architecture and a multi-mode data exchange protocol to balance the trade-offs between low power consumption and high data reliability. We have evaluated the performance of LEMoNet by deploying custom-designed sensor and gateway nodes in a SHARCNET DC at A.N. Bourns Science Building. We show experimentally that LEMoNet achieves an average data yield of over 98%. Under normal operations with one temperature and one humidity reading every thirty seconds, the battery lifetime of LEMoNet sensor nodes is projected to be 14.9 years on a single lithium coin battery.

Acknowledgements

I would like to thank my supervisor, Dr. Rong Zheng, for training me. She is incredibly dedicated to her research and developing new ideas. Whenever students require aid, she is sure to help. As well, she also helped improve my technical writing skills.

The advice of principal research engineer Dr. Ghada Badawy has been continuously valuable to my work. Whenever I am lost, she provides a fresh perspective.

I would like to thank my thesis committee members, Dr. Douglas Down and Dr. Emil Sekerinski, for all of their guidance; your discussion, ideas, and feedback have been absolutely invaluable.

I would like to thank Cinnos Mission Critical Incorporated for supporting me.

I would like to offer my thanks to Jun Li, my research partner. He is an outstanding hardware designer with an admirable strong sense of responsibility.

I am grateful for the assistance provided by my lab members Qiang Xu, Muhammad Hammuda, Mehdi Jafarizadeh, and Hosein Moazamigoodarzi. They gave me crucial supports with their strong technical skills.

Finally and most importantly, I am sincerely grateful to my family, for all the support they have provided me throughout the years. Without them, I would not be able to come this far.

Abbreviations

AC	Alternating Current
ADV	Advertising
ARQ	Automatic repeat request
BER	Bit Error Rate
BLE	Bluetooth Low Energy
CASN	Cluster Area Sensor Network
CRAC	Computer Room Air-Conditioning
CRC	Cyclic Redundancy Check
DC	Data Center
FEC	Forward Error Correction
GAP	Generic Access Profile
GATT	Generic Attribute Profile
HCI	Host Controller Interface
IND	Indicator
ITE	Information Technology Equipment
LSB	Least Significant Bit
M2M	Machine-to-Machine

MAC	Media Access Control
MQTT	Message Queuing Telemetry Transport
NCL	Normal Connectionless
NTP	Network Time Protocol
PDU	Protocol Data Unit
PUE	Power usage effectiveness
RC	Reliable Connection
RH	relative humidity
RPI3	Raspberry Pi 3
RSS	Received Signal Strength
RX	Receive
SCL	Scannable Connectionless
SHARCNET	The Shared Hierarchical Academic Research Computing Network
SNR	Signal Noise and Interference Ratio
SPI	Serial Peripheral Interface Bus
TI	Texas Instruments
TX	Transmit
UART	universal asynchronous receiver-transmitter
USB	Universal Serial Bus
WRAP	Wireless Reliable Acquisition Protocol
WSN	Wireless Sensor Network

Contents

Abstract	iii
Acknowledgements	iv
Abbreviations	v
1 Introduction	1
2 Background and Related Work	5
2.1 Related Work	5
2.2 Bluetooth Low Energy	7
2.3 Message Queue Telemetry Transport (MQTT)	11
2.3.1 Publish/Subscribe Pattern	12
2.3.2 Message Filtering	13
2.4 Error Detection and Correction	14
2.4.1 CRC Checksum	14
2.4.2 Forward Error-Correction	15
3 DESIGN of LEMoNet	16
3.1 Design Overview	16

3.1.1	Sensor to/from Gateway	17
3.1.2	Gateway to Controller	19
3.1.3	Sensor Operations	19
3.2	LEMoNet Protocol	20
3.2.1	LEMoNet Messages	20
3.2.2	Operation Modes	22
3.2.3	Multi-packet Reception	24
4	Implementation	27
4.1	Hardware	27
4.1.1	Sensor Node	28
4.1.2	Gateway’s BLE Shield	29
4.2	Communication between Gateway and Controller	29
5	Evaluation	31
5.1	Power Consumption Profiling	31
5.2	Testbed Evaluation	34
5.2.1	Effects of Multi-Packet Reception	35
5.2.2	Effects of Sampling Period	36
5.2.3	Number of Transmissions in the SCL mode	37
6	Insights from Deployment	39
6.1	Impacts of Cold Aisle Layouts in the SHARCNET DC	42
6.2	Measurement Study in the MCX DC	42
6.2.1	Effects of a single cooling unit	47
6.2.2	Effects of server utilization	48

6.2.3	Effects of blanking panel	48
7	Conclusion	50

List of Figures

2.1	BLE Protocol Stack	8
2.2	GAP State Diagram	9
2.3	BLE Link Layer State	10
2.4	MQTT Architecture	12
2.5	Publish/Subscribe Communication	12
2.6	CRC-24 Generator ($x^{24} + x^{10} + x^9 + x^6 + x^4 + x^3 + x + 1$)	15
3.1	LEMoNet system architecture	17
3.2	Sensor to/from Gateway communication	18
3.3	Sensor State Diagram	20
3.4	LEMoNet Advertising Packet Format	21
3.5	Data Transmission Timeline	23
3.6	Recovery from Packet Losses and Corruption	26
4.1	LEMoNet BLE shield and Sensor	28
5.1	Current measurements of a LEMoNet sensor node in different modes. The supply voltage is 3V	33
5.2	Placement of 58 sensors and 3 gateways in SHARCNET DC	35

5.3	Packet reception rate. GW1_recv, GW2_recv, GW3_recv and C_recv are the percentages of received packets passing CRC at the three gateways and at the controller. GW1_rec, GW2_rec, GW3_rec and C_rec are the percentage of packets recovered.	36
5.4	Effects of sampling interval on PRR in the NCL mode	37
5.5	The number of transmissions before receiving a scan request	38
6.1	MCX DC layout. The cooling units are at both sides of the DC. . . .	39
6.2	Cold aisle layouts of an SHARCNET DC	41
6.3	Inlet temperature readings of 9 racks at SHARCNET DC	43
6.4	Inlet temperature readings reported by deployed sensors and servers. The vertical red lines indicate the starts and ends of evaluation periods with different configurations (Table 6.3).	45
6.5	The CCDF of the correlation coefficients between the inlet temperature of the server and the deployed sensor in the same zone at MCX DC. .	46
6.6	Inlet temperature timelines at top of rack 4,5 during time interval 7 of the correlation experiment. The set point was 20 °C. Cooling unit next to rack 5 was off. The utilization of all servers are 70%.	47

List of Tables

2.1	Advertising Packet Types	10
5.1	Current Usage Profile under Different Working Mode	32
5.2	Current Usage in the NCL mode	34
6.1	The Number of Servers Providing Thermal Measurements in MCX DC	40
6.2	The Number of Servers Provides Inlet Temperature at Different Zone of MCX DC	44
6.3	Settings at Different Time Interval During the Correlation Experiment at MCX DC	44
6.4	Temperature at Different Zones of the MCX DC When Servers are Idle. Set point for both cooling units are 20 °C.	48
6.5	Temperature at Different Zones of the MCX DC When Servers are at 70% Utilization. Set point for both cooling units are 20 °C.	48
6.6	Rack 5's Inlet Temperature With/Without Blanking Panel at Middle up of Rack 5. The Set Point is 20 °C and servers are at 70% utilization.	49

Chapter 1

Introduction

The computing needs of our society, majority of which are fulfilled in data centers (DCs), double every five years. Despite increasing productivity, DCs are burdened with high capital and operating costs. These costs arise from widespread industry practice of over-design and over-provisioning as the server utilization is typically between 12% – 15% for on-premises DCs and 65% for cloud DCs [1]. The over-design stems from the fact that DC owners and operators have no real-time visibility of the performance and operational conditions of their DCs, thus becoming overly cautious. Therefore, a significant portion of the capital invested in establishing a DC is wasted. For instance, high capacity cooling systems are commonplace to mitigate occasional “hot-spots” during peak demand periods but enhancing air-circulation can fix this problem at a fraction of the cost [2]. Traditionally, DC monitoring systems measure only thermal metrics, such as the Information Technology Equipment (ITE) rack inlet temperature and the flow rate of air through computer room air-conditioning (CRAC) units. These variables are measured with sensors mounted at key locations. Alone, the sensors do not provide sufficient spatial resolution to fully describe the

thermal performance of a DC. This may lead to early equipment failure, low energy efficiency and degraded server performance [3].

Wireless sensing is an attractive and cost-effective approach to infrastructure monitoring by eliminating wiring, which is approximately 75% of labour and material costs [4]. This solution also increases the flexibility in sensor placement and allows seamless, automated system reconfiguration. As a result, Wireless Sensor Networks (WSNs) have been investigated as a low-cost solution for finer grained thermal mapping and thermal forecast in DCs [5, 6, 7, 8].

In this thesis, we target co-location DCs, where equipment, rack space, and network bandwidth are leased to customers. DC operators have no access to on-board sensors in customers' ITE and are not allowed to connect any monitoring device to them directly. In considering the needs for long battery lifetime and high scalability, we argue for rethinking WSN design to support low duty cycle monitoring in DCs. In particular, we recognize that in traditional multi-hop WSNs, significant energy is consumed as the result of protocol overheads, such as topology formation and maintenance, time synchronization, medium access control, and for reliable data transfer. Much of the overhead is due to the *symmetry* among transmitting and receiving devices. In contrast, for DC monitoring, heterogeneous devices with different power sources, wireless technologies, and compute and storage capabilities can be utilized. We propose a two-tier network architecture, where the bottom tier includes wireless sensor nodes and more powerful gateway nodes connected via Bluetooth low energy (BLE); and the top tier consists of gateways and a controller connected via WiFi. The sensor nodes are battery operated and can be deployed and replaced quickly. A DC can have hundreds to thousands of such nodes installed at any given time. Gateways

are AC powered and collect readings from multiple sensor nodes. BLE is chosen as it offers high data rates (compared to Zigbee) and low power consumption for short-range communications. More importantly, BLE supports broadcast communication without connection setup. The gateways and the controller communicate using WiFi since each gateway needs to transfer readings from multiple sensor nodes to the controller at a higher data rate. A mesh topology can be formed if gateways are beyond one hop away from the controller. To minimize sensor power consumption, normal sensing data is sent in a best effort manner without acknowledgment and sensors can be immediately put to sleep upon transmission of data. High reliability is achieved by aggregating duplicated packets received by multiple gateways and by performing error correction at the controller site. This shifts the burden of complexity from sensors to the controller. At the component level, we customize the board design of sensor nodes to minimize power consumption in different operation modes. As the result, under normal operations, with a single temperature and humidity reading every thirty seconds, the battery lifetime of the sensor nodes is projected to be 14.9 years using a single lithium coin battery.

We have evaluated the proposed system in SHARCNET DC. A total of 58 sensors are deployed next to the front and back of seven server racks. Experiment results show that LEMoNet can achieve an average data yield over 98% at a very low energy cost.

The rest of this thesis is organized as follows. Chapter 2 reviews related work on DC monitoring. In Chapter 3, we present an overview of LEMoNet design and describe details of LEMoNet protocol. In Chapter 4, we discuss implementation,

followed by performance evaluation in Chapter 5. In Chapter 6, we discuss the experiences and insights gained from our deployment in SHARCNET DC and a modular experimental DC. Finally, we conclude the thesis in Chapter 7.

Chapter 2

Background and Related Work

2.1 Related Work

WSN technologies have been investigated as a low-cost candidate solution to finer-grained thermal mapping and accurate hotspot detection in DCs since they require no additional network and facility infrastructure in an already complicated IT environment.

Previous report [5] studies energy saving measures utilizing information collected from a 588-node WSN deployed in a 13,005ft² area. This solution results in an overall energy reduction of 17% and PUE reduced from 1.94 to 1.51. The payback time of deploying this WSN is 3.4 years. In [6], researchers at Microsoft Research designed RACNet, a large-scale sensor network for high-fidelity DC environmental monitoring. RACNet consists of wireless master nodes powered by USB ports on servers or wall power. The wireless master node and several wired sensors form a daisy chain to cover one side of a rack, collecting data at different heights. The master nodes are equipped with IEEE 802.15.4 radios to form a mesh network for data collection. To

improve data yield, a Wireless Reliable Acquisition Protocol (WRAP) was designed that uses token passing for network-wide arbitration. Daisy chaining multiple sensors to a master node allows powering the sensors and ensures reliable gathering of sensor data. However, the wired connection is restrictive for sensor placement, and placing extra wires along the rack is generally considered undesirable by DC operators.

Rodriguez et al.[7] observed significant temperature variation in different locations in DC using a small-size network consisting of 10 nodes. In [9], Chen et al. deployed temperature and airflow sensors to monitor inlet and outlet server temperatures and CRAC units. All sensors run TinyOS and form a single-hop network using 802.15.4. Research demonstrated the benefit of using the real-time sensor data to calibrate a computational fluid dynamic model. The temperature evolution of servers with highly dynamic workloads can be forecast within ten minutes, with an average error of 0.52°C. Cluster area sensor network (CASN) [8] is comprised of TeloB sensor nodes running TinyOS and attached to compute servers or workstations. This system verifies servers physical presence through wireless cluster-wide command dissemination and thus enhancing the security of DC management. CapNet [10] is a real-time WSN protocol for power capping for DC management. It uses distributed event detection to eliminate the overhead of regularly polling all nodes in the network. In CapNet, 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.

The methods described here use IEEE 802.15.4 enabled sensor nodes. In this thesis, we develop Bluetooth low energy (BLE) enabled sensor nodes for low power consumption, higher data rate (compared to 802.15.4) and reliable connectionless

transmission. In [11] the authors compare the usage of four wireless technologies (WiFi, classical Bluetooth, Zigbee, and BLE) for a construction noise monitoring application. The authors conclude that BLE consumes the least power while providing sufficient data rates enabling dynamic sensing at high sampling rates.

The design of the proposed protocol bares similarity with LoRaWAN, a low-power wide area network for IoT applications [12]. In LoRaWAN, uplink communication from sensor devices to LoRA concentrator/gateways uses broadcast and can be received by multiple gateways. The effective throughput of end devices is limited by the physical data rate of 27kbps using LoRA PHY and the maximum duty cycle of 1% (in EU 868). The data rate offered by LoRaWAN is unsuitable for DC monitoring where sensors are deployed densely.

2.2 Bluetooth Low Energy

BLE operates in the spectrum range from 2.4 to 2.4835 GHz, which is further divided into 40 channels. Three channels are reserved for advertising with minimal overlapping with commonly used WiFi channels. The remaining 37 channels are data channels. A BLE device operates in four different roles. A *peripheral device* is used for advertising, which is connectable and operates as a slave. A *central device* scans for advertisers and initiates connections, operating as a master in one or more connections. A *broadcaster* is a non-connectable advertiser, while an *observer* scans for advertising packets without initiating connections. BLE advertising devices do not utilize carrier sensing multiple access, and thus advertising packets are transmitted immediately when data become available. A peripheral device or a broadcaster can

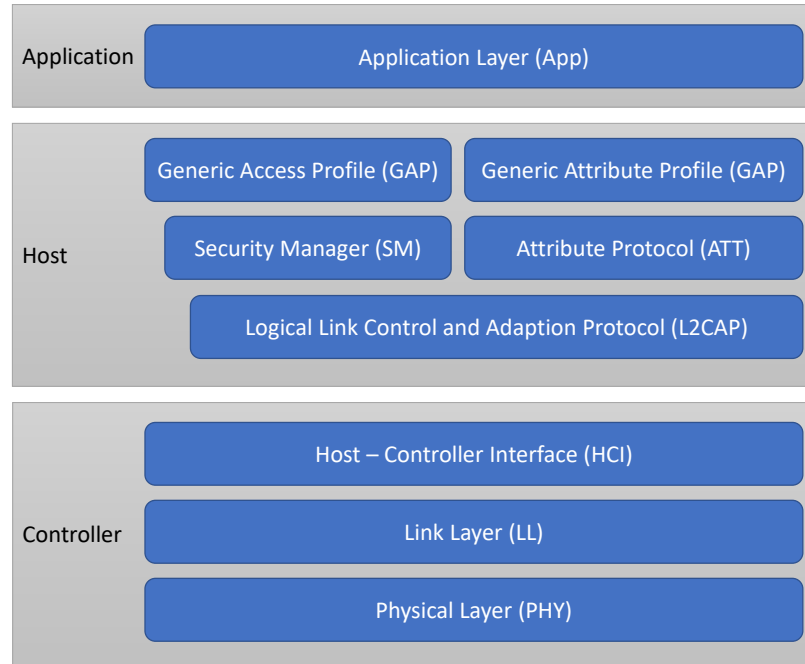


Figure 2.1: BLE Protocol Stack

broadcast payloads up to 31 bytes using the advertising channels for 1 to n communication.

The BLE architecture consists of application, host (upper layer), and controller (lower layer) [13]. User applications interface with the BLE stack (Fig. 2.1). The host and controller are implemented separately and usually run on different hardware. Within the controller, the host control interface (HCI) layer provides communication between the host and the controller through a standardized interface such as USB, UART, or SPI. The link layer controls the RF state of the device. The link layer can be in one of the five states: standby, advertising, scanning, initiating, and connecting (Fig. 2.3).

The generic access profile (GAP) of the BLE stack is responsible for connection

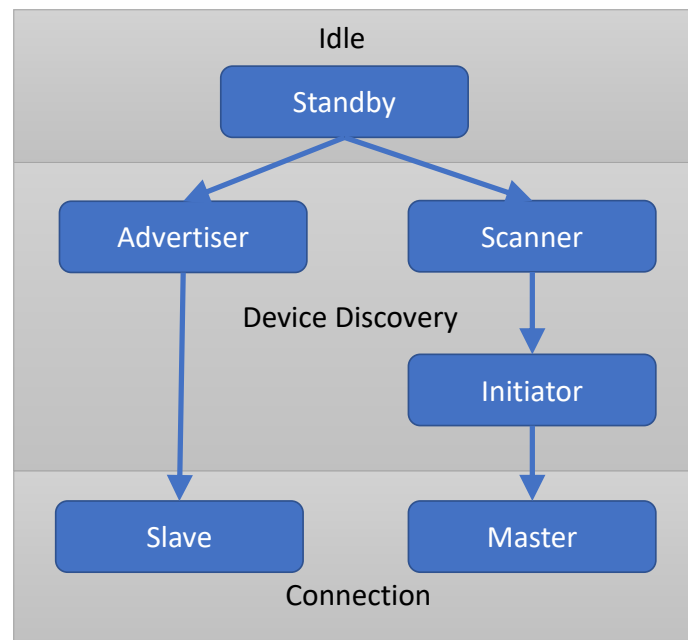


Figure 2.2: GAP State Diagram

and advertising functionalities (see Fig. 2.2). It handles device discovery, link establishment, link termination, initiation of security features, and device configuration. A central device must recognize the peripheral device to establish a connection. Therefore, a peripheral send advertising packets. The central device scans the advertising packets to discovery the advertisers. The initiator must specify a peer device to connect, and thus sends a connection request with the connection parameters after receiving another advertising packet. Once the connection is formed, the initiator (central device) is the master and the advertiser (peripheral device) is the slave.

The generic attribute profile (GATT) defines how data is stored and exchanged between connected devices. A peripheral device usually operates as a GATT server, while a central device operates as a GATT client. A GATT server provides multiple services, which are collections of characteristics. A GATT client can read from or write to characteristics and get notified of value changes.

Table 2.1: Advertising Packet Types

Advertising PDU Type	Max Data length	Allow scan	Allow conn.
ADV_IND	31 bytes	YES	YES
ADV_DIRECT_IND	N/A	NO	YES
ADV_NONCONN_IND	31 bytes	NO	NO
ADV_SCAN_IND	31 bytes	YES	NO

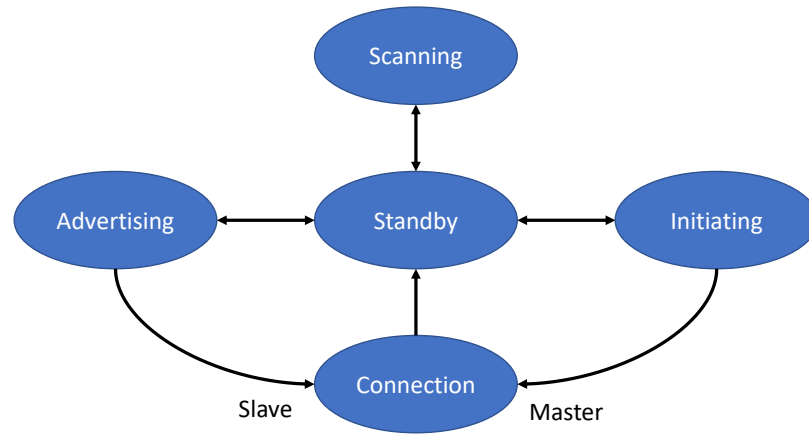


Figure 2.3: BLE Link Layer State

The four types of advertising packets are ADV_IND, ADV_DIRECT_IND, ADV_NONCONN_IND and ADV_SCAN_IND (Table 2.1). A peripheral device advertises connectable undirected advertising packets (ADV_IND) to request a connection to central devices. Non-connectable devices advertise non-connectable undirected packets (ADV_NONCONN_IND) to scanning devices. The scannable undirected advertising packet (ADV_SCAN_IND) allows additional information exchange via scan responses.

A scanner listens for advertising packet on the advertising channels and can listen on a primary advertising channel for at most 40.96s. It changes to other advertising channels. Scanners are either active or passive. A passive scanner receives data from an advertising device, while an active scanner may send scan requests to an advertising

device for additional information. The BLE link layer of an active scanner handles scan requests. After entering the scanning state, an active scanner sends a scan request packet after receiving an ADV_IND or ADV_SCAN_IND protocol data unit (PDU). The scanner listens for the scan response packet and continues to respond to the same advertiser until the scan request packet is received. The payload of scan request packets only contains the advertiser's and the scanner's MAC addresses. BLE uses a backoff procedure to minimize collision of scan request PDUs from multiple scanners. Link layer resets backoff machine when a device enters the scanning state (Fig. 2.3).

The BLE link layer uses a whitelist to filter advertisers, scanners or initiators. The whitelist of an advertiser contains a set of device MAC addresses for link layer filtering so that only scan or connection requests from devices on the list are processed. A scanner's whitelist contains a set of device MAC addresses to filter advertising packets.

The BLE standard supports star topologies and more recently mesh topologies [14]. Some BLE 4.x or 5.0 devices support multiple roles simultaneously. In LEMoNet, only the star topology is used.

2.3 Message Queue Telemetry Transport (MQTT)

MQTT [15] was invented by Stanford-Clark of IBM[®], and Nipper of Arcom (now Eurotech) in 1999 and became an ISO standard (ISO/IEC PRF 20922) in 2016. Previous studies show MQTT is a stable connectivity protocol for machine-to-machine (M2M) and Internet of Things (IoT) [16, 17, 18, 19]. MQTT aimed to connect embedded devices and network applications using a lightweight publish-subscribe messaging protocol. MQTT runs over TCP/IP by default.

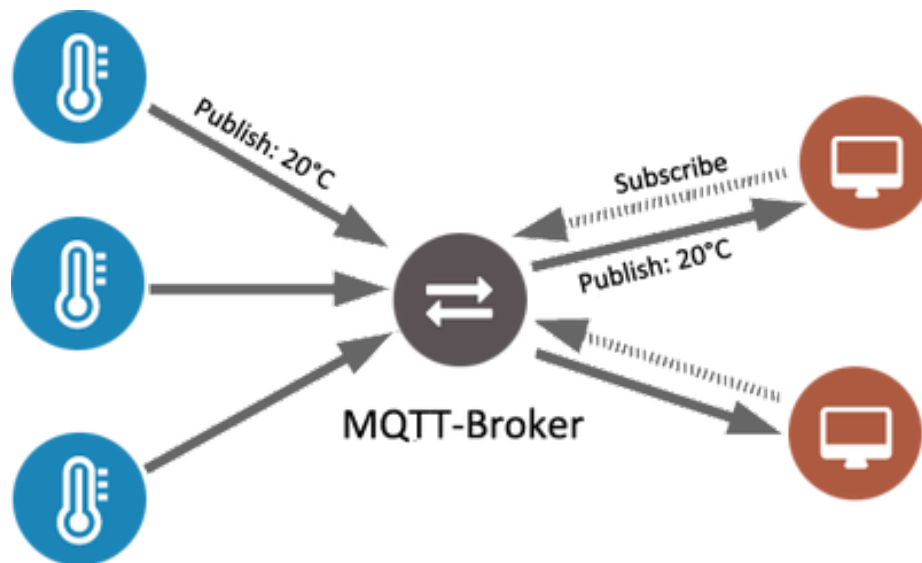


Figure 2.4: MQTT Architecture

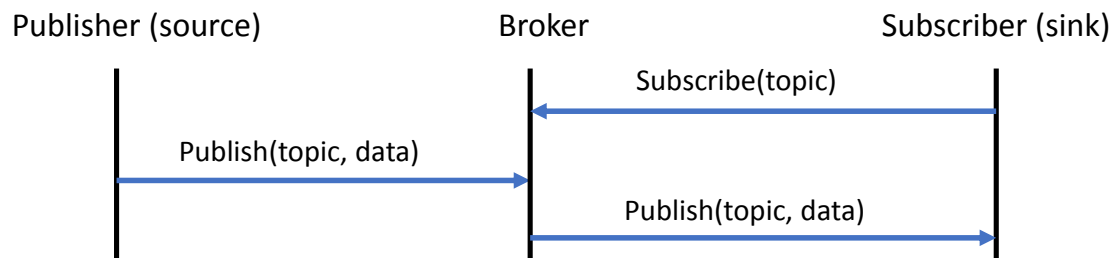


Figure 2.5: Publish/Subscribe Communication

2.3.1 Publish/Subscribe Pattern

Each client in the publish-subscribe pattern can act as a publisher and a subscriber at the same time. A publisher who sends the data does not know any information about the existence of subscribers. The publish-subscribe pattern enables easy broadcasting of messages from one publisher to many subscribers. The publisher and subscriber do not have to run at the same time and they can publish and subscribe asynchronously. All messages are exchanged in a broker (Fig. 2.4). The broker filters and sends the published message to the subscribers based on the message filters.

2.3.2 Message Filtering

Publisher publishes data under a specific "Topic" (Fig. 2.5). The topic is a string and is used by the broker to filter messages for each connected client. To receive the data associated with a topic, the subscriber must connect to the same broker and subscribe to the same topic as the publisher. A topic consists of one or more topic levels. Each topic level is separated by a topic perpetrator. For example, in

/floor1/sensor1/temperature

"floor1" is the first topic level, "sensor1" is the second topic level and "temperature" is the third topic level. Any subscriber to this topic will get the published messages related to temperature readings of sensor 1 on floor level 1. The subscriber can also subscribe to multiple topics at once using wildcards: the plus sign (+) matches only one topic level and the number sign (#) matches any number of levels within a topic.

For example,

/floor1/ + /temperature

The topic above uses a single level wildcard and it can be used to subscribe to the temperature reading of all sensors on floor level 1.

Below is a second example,

/floor1/#

The topic above uses a multilevel wildcard, and it can be used to subscribe to all message on floor level 1.

The broker accepts each valid topic without any prior initialization. A client publishes or subscribes messages by attaching a topic. Moreover, the broker uses

topics to decide if a subscribing client will receive the message or not.

2.4 Error Detection and Correction

Many communication channels are subject to noise and interference, and hence errors may be introduced during transmission from the source to a receiver. There are two sources facets of packet reception failure: packet corruption (packets received with error onerous bits) and packet losses. Packet corruption occurs when interference and noise result in incorrectly decoded symbols. Packet loss occurs due to failure to extract physical-layer preamble.

Error detection and correction techniques enable reliable delivery of digital data over unreliable communication channels. Error detection techniques allow detecting bit errors. Error correction enables reconstruction of the original data.

In IEEE 802.15.4, sensors retransmit the original data packets after packet failure occurs [20]. Upon receiving a data packet, the intended receiver sends an acknowledgment frame. Absence of acknowledgment triggers retransmission of the data packet. As a result, the energy waste from listening to ACK messages and retransmission can be significant.

2.4.1 CRC Checksum

The cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks to detect accidental changes in raw data. The BLE standard employs CRC code for error detection. Each packet is encoded such that twenty-four redundant bits are generated and appended to the packet. The BLE standard adopts the

CRC polynomial: $x^{24} + x^{10} + x^9 + x^6 + x^4 + x^3 + x + 1$ (Fig. 2.6). The initial value for advertising packet is 0x555555. At the receiver side, the packet is forwarded to the upper layer if the packet passes the CRC check. Otherwise, the packet is discarded. A previous study[21] shows the n-bits CRC can correct single-bit errors in data less than $2^{(n-1)}$ bits. The BLE receiver has the ability to correct 1 error bit.

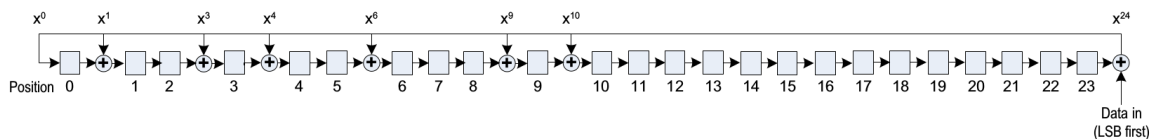


Figure 2.6: CRC-24 Generator ($x^{24} + x^{10} + x^9 + x^6 + x^4 + x^3 + x + 1$)

2.4.2 Forward Error-Correction

The Forward Error Correction (FEC) method adds redundancy to information bits, to correct errors introduced during transmission. Two efficient FEC codes are Hamming code and Reed-Solomon Code. Hamming code [22] adds parity bits to fixed-bit data. For example, hamming(7,4) code maps four data bits into seven bits by adding three parity bits. This method can detect and correct single-bit errors in seven bits. By adding $2t$ check symbols to the data, Reed-Solomon code [23] can detect any combination of up to $2t$ erroneous symbols, and correct up to t symbols.

Chapter 3

DESIGN of LEMoNet

In this chapter, we present an overview of LEMoNet design and components, as well as details of LEMoNet protocol.

3.1 Design Overview

We design and develop LEMoNet, a low energy wireless monitoring Network for DCs. LEMoNet is energy efficient, easy to install and manage within existing IT infrastructure while delivering better visibility into a DC's thermal profile. It also supports real-time alerts in the event of outages such as overheating.

LEMoNet consists of sensor nodes, gateways and a central controller organized in a two-tier network architecture (Fig. 3.1). The first tier is between sensor nodes and gateways; the second tier is between gateways and a central controller. Sensor nodes are battery powered and operate at low duty cycles. They measure and report environmental conditions (such as temperature, humidity, pressure and air flow). The wall-powered Gateways are responsible for collecting data from sensor nodes and

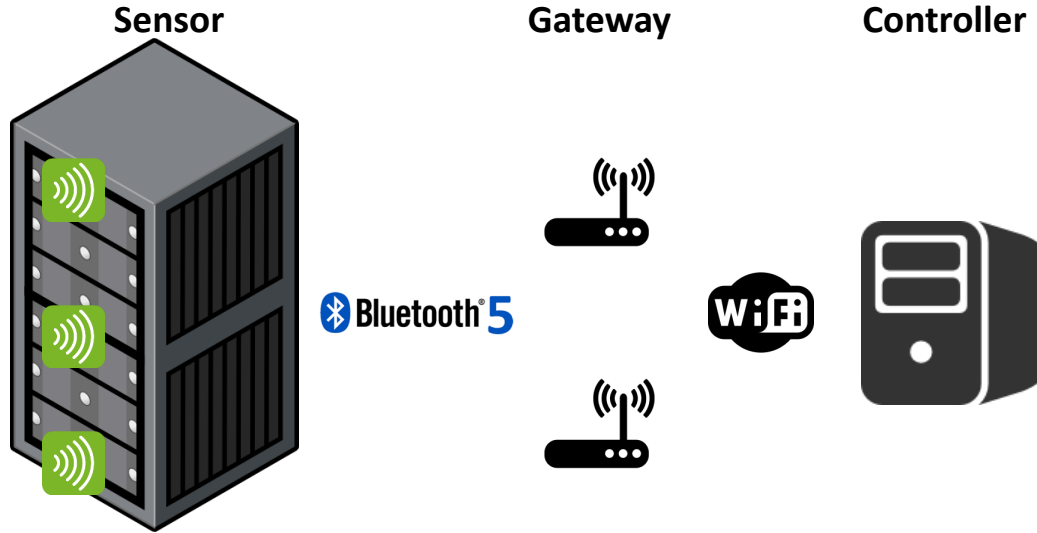


Figure 3.1: LEMoNet system architecture

wirelessly forwarding the data to a central controller. There is no explicit association between sensor nodes and gateways. This architecture makes the network scalable with little management overhead since adding a new sensor node only involves deciding its location and installing it.

3.1.1 Sensor to/from Gateway

Sensor nodes communicate with gateways using BLE. BLE defines two modes of communication: connectionless and connection oriented [3.2]. The broadcast connectionless mode consumes less power due to low protocol overhead. On the other hand, connection oriented communication requires pairing of devices and supports reliable data transfer.

In DCs, unlike computer workloads, thermal conditions tend to change slowly. The majority of sensor data are non-mission critical and do not require high reliability. Broadcast connectionless communication is thus a good fit for collecting sensor

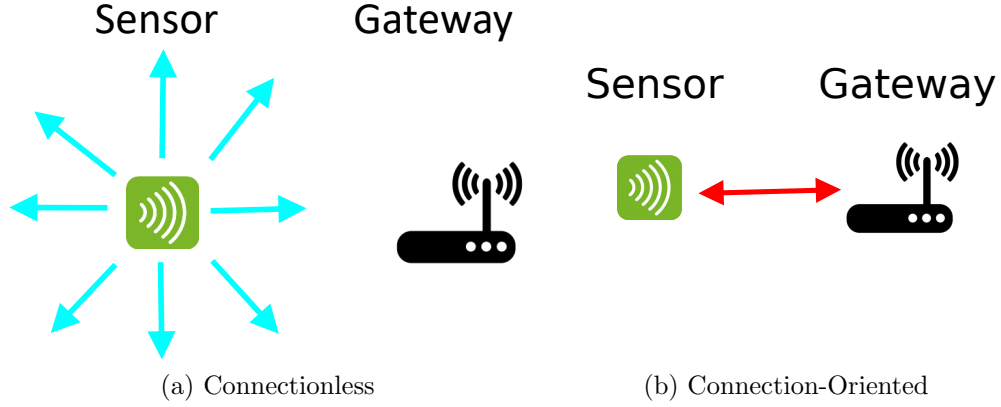


Figure 3.2: Sensor to/from Gateway communication

data. Instead, multi-packet reception at gateways offers opportunities to improve the reliability of data collection at no extra cost.

There are however two scenarios when reliability is highly desirable. First, when sensors detect outage events such as overheating, alerts should be generated as soon as possible. Second, updating configuration parameters and software images on sensor nodes require reliable data transfer. To support alerts, we leverage a particular message type in BLE to provide an acknowledgment to sensor data without device pairing. For the latter, connection-oriented communication in BLE is utilized.

To this end, LEMoNet supports three service types between sensor nodes and gateways, namely, i) unreliable uplink (*sensor node to gateway*) communication for non-urgent sensor data, ii) reliable uplink traffic for urgent sensor data, and iii) reliable downlink (*gateway to sensor node*) communication for commands and software updates.

3.1.2 Gateway to Controller

Gateways communicate with a central controller using Wi-Fi channels 1, 6 and 11. This minimizes overlapping between WiFi channels and BLE advertising channels. For situations where Wi-Fi coverage is insufficient, a multi-hop mesh network among the gateways can be employed. The MQTT protocol [24], a lightweight machine-to-machine publish/subscribe messaging protocol, is employed at the application layer. In MQTT, gateways publish non-urgent data periodically and urgent data immediately. The central controller subscribes to sensor reading topics. Merging multiple non-urgent data and sending them once reduce the number of packets between gateways and the controller, and limit interference with BLE traffic.

3.1.3 Sensor Operations

To maximize battery lifetime, sensor nodes stay in low-power mode as much as possible. In LEMoNet, a sensor node can be in one of six states (Fig. 3.3): measuring, reporting non-urgent data in the normal connectionless (NCL) mode, reporting urgent data in the scannable connectionless (SCL) mode, and sleeping. After booting-up, on-board sensor readings are collected. When a reading is considered out of bound (e.g., the temperature exceeding a pre-defined threshold in a cold aisle), the sensor node goes into the scannable connectionless (SCL) mode and attempts to reliably transfer the measurement data to gateways. Otherwise, the sensor node broadcasts the data once. The reliable connection (RC) mode is used for reliable downlink communication of commands and software updates. Since all communications are initiated by the sensor nodes (as opposed to the gateways), the RC mode is locally scheduled at a long interval (e.g., daily) on each sensor node.

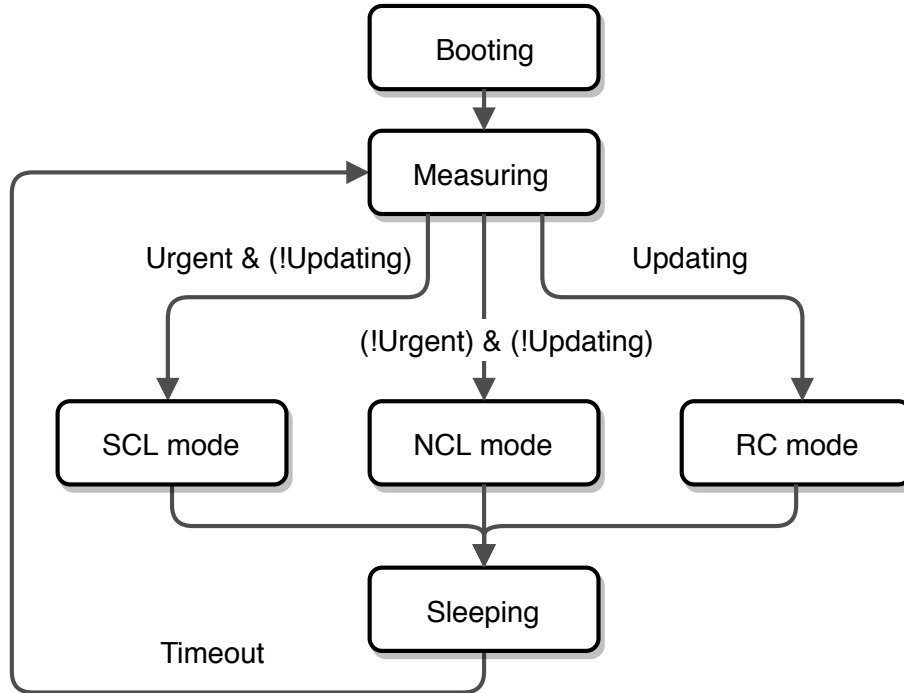


Figure 3.3: Sensor State Diagram

In all cases, once all transmissions are complete, the sensor node goes into sleep state. A timer is set and, upon expiration, the sensor node goes into measuring state.

3.2 LEMoNet Protocol

LEMoNET builds upon the BLE stack and leverages existing message types.

3.2.1 LEMoNet Messages

LEMoNet uses standard BLE messages for scan request/response, connection request/response, and more. To support the three service types between sensor nodes and gateways, we customize the payload of advertising packets to include additional

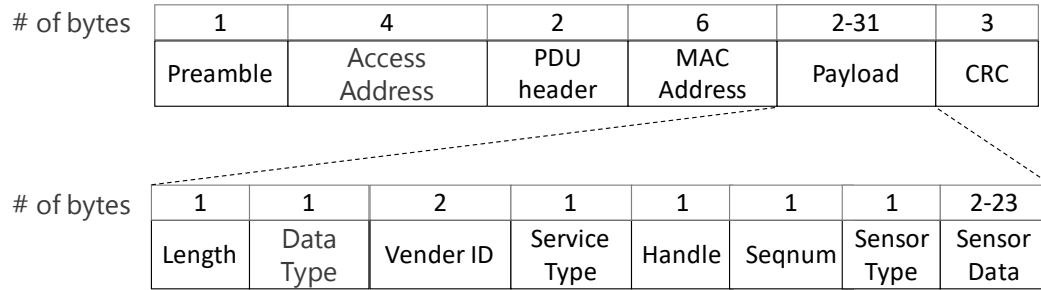


Figure 3.4: LEMoNet Advertising Packet Format

information. The format of advertising packets is shown in Fig. 3.4. More specifically, in the payload field,

- **Vendor ID** distinguishes LEMoNet nodes from other BLE devices.
- **Service Type** specifies the service type, such as 0 for non-urgent sensor data, 1 for urgent sensor data, 2 for commands and software updates.
- The parameter characteristic value **Handle** allows a gateway to directly update sensor parameters after paring with the sensor node.
- **Sequence number** tracks unique sensor data transmitted from each device.
- **Sensor Type** is composed of one byte. Each set bit indicates that the packet contains sensor data of the related type. More specifically,
 - **Bit Position 7** indicates temperature data with 0.01 $^{\circ}C$ resolution.
 - **Bit Position 6** indicates humidity data is with 0.01 % resolution.
 - **Bit Position 5** indicates pressure data is with 0.0 2hPa resolution.
 - **Bit Position 4** indicates battery voltage data is 1 mV.

- **Sensor Data** contains sensor readings. Each reading is of two-byte to nine-byte long. Each reading of the assigned sensor types is of two-byte long.

3.2.2 Operation Modes

NCL Mode

In NCL mode, a sensor sends ADV_NONCONN_IND advertising packets (Service Type 0) containing non-urgent sensing data one time and immediately goes to sleep (Fig. 3.5a). When a gateway receives the advertising packet, it puts the packet into a non-urgent message queue. Gateways publish non-urgent messages every second. Due to the broadcast nature of advertising packets, multiple gateways can receive each topic. By combining multi-packet receptions at the gateways, the controller tolerates packet losses at individual gateways and can recover corrupted packets. Further details on this subject are discussed in Section 3.2.3.

SCL Mode

A sensor node enters the SCL mode when at least one sensor reading falls outside the pre-defined range, typically indicating abnormal thermal conditions¹. In this mode, a sensor indicates the urgency of the conveyed data from an ADV_SCAN_IND advertising packet with Service Type field 1. Upon receiving the advertising packet, a gateway sends a scan response message acting as an acknowledgment (Fig. 3.5b). The gateway then immediately publishes the urgent data to the central controller via MQTT. A sensor node retransmits the data up to 10 times if the scan response message is not received. Upon receiving an acknowledgment or exceeding maximum

¹In this study, we do not consider abnormal readings from sensor failure.

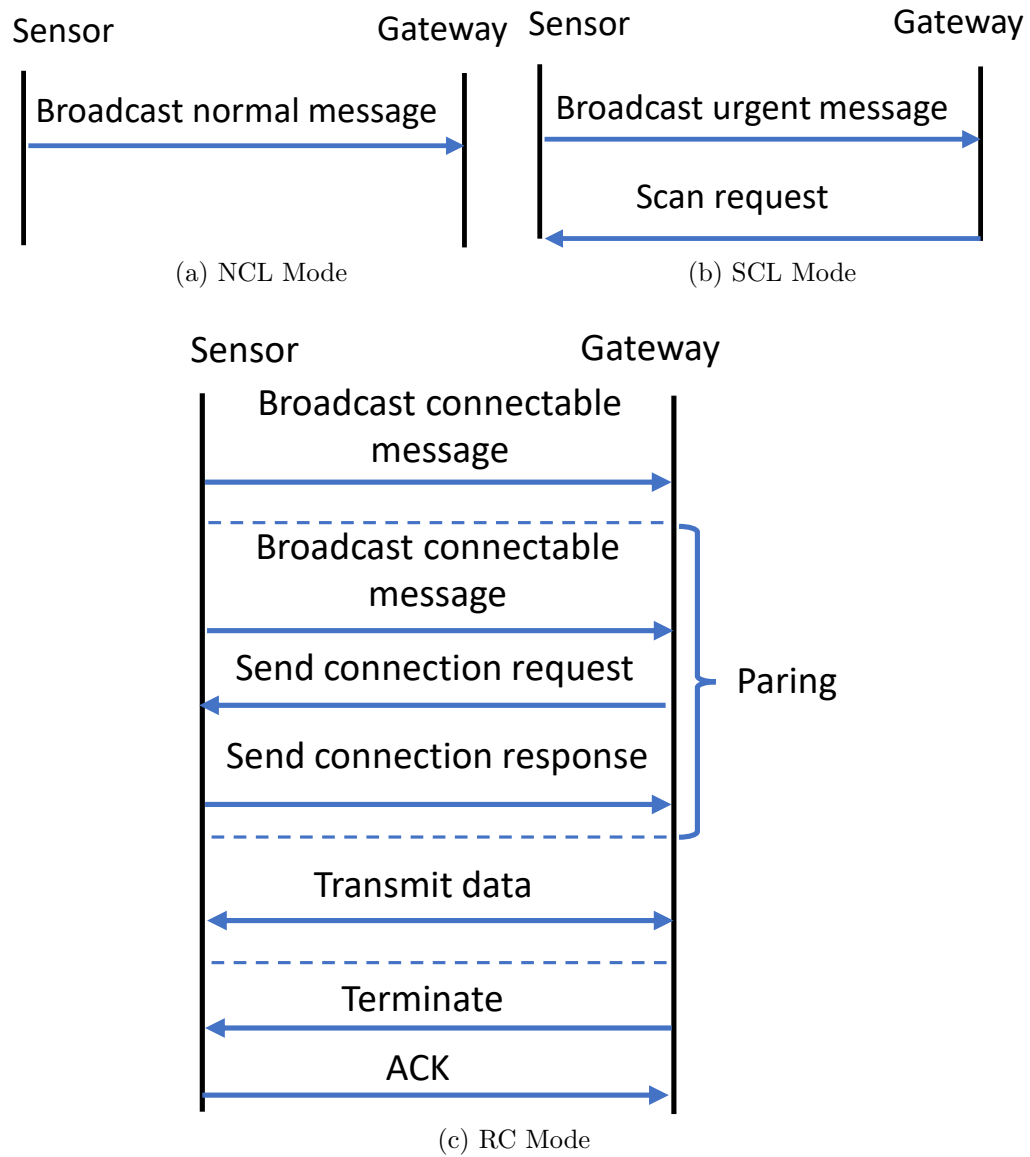


Figure 3.5: Data Transmission Timeline

retransmission attempts, the sensor node goes to sleep.

RC Mode

The RC mode is used for downlink communication from gateways to sensors to send commands, configuration parameters and software updates (Fig. 3.5c). In LEMoNet, sensor nodes initiate all communications, and thus operations are completed asynchronously. Sensor nodes transmit an ADV_IND advertising packet with Service Type 2. When an update is needed, the gateway contains the sensor parameters responds with a connection request to pair with the sensor. After successfully pairing, the gateway sends relevant information to the sensor via data channels. Upon transfer completion, the gateway terminates the pairing. The sensor acknowledges the termination and goes to sleep. In the absence of connection requests from any gateway, the sensor retransmits the ADV_SCAN_IND advertising packet up to ten times.

3.2.3 Multi-packet Reception

In the NCL mode, an advertising packet containing non-urgent sensor data can become lost. Failures to receiving a packet arise from two sources: packet corruption (received with error) and packet loss (packet not received). Packet corruption occurs due to low Signal Noise and Interference Ratio (SNR). Packet loss occurs when the physical-layer preamble cannot be extracted. Therefore the packet cannot be decoded at all. Traditionally, to improve the reliability of data transfer, two mechanisms can be utilized, namely, packet retransmission and forward error correction (FEC). Both mechanisms incur additional power consumption on the sensors as the result of either extra transmissions or extra bits in the packets. We improve the reliability of

non-urgent data without compromising the battery lifetime of sensors.

This improvement is possible because advertising packets are broadcast to any gateway in the vicinity of the transmitting sensor node. In LEMoNet, an algorithm takes advantage of multi-packet receptions at gateways. The gateways are configured to receive advertising packets without CRC checksum on the advertising channels. Upon receiving some sensor data with a desired sequence number that passes CRC from a gateway, the controller saves the data in the storage. If a packet is corrupted and no other correct duplicated copy is available, the controller attempts to salvage the data in two steps. First, it tries to recover the PDU header, MAC address, length, sequence number and sensor type fields using historical data. If the resulting packet still fails CRC, we brute-forcelly search for up to 2-bit errors in the sensor data and checksum fields and verify if the packet passes the CRC checksum. Further details regarding the multi-packet reception algorithm are given in Fig. 3.6.

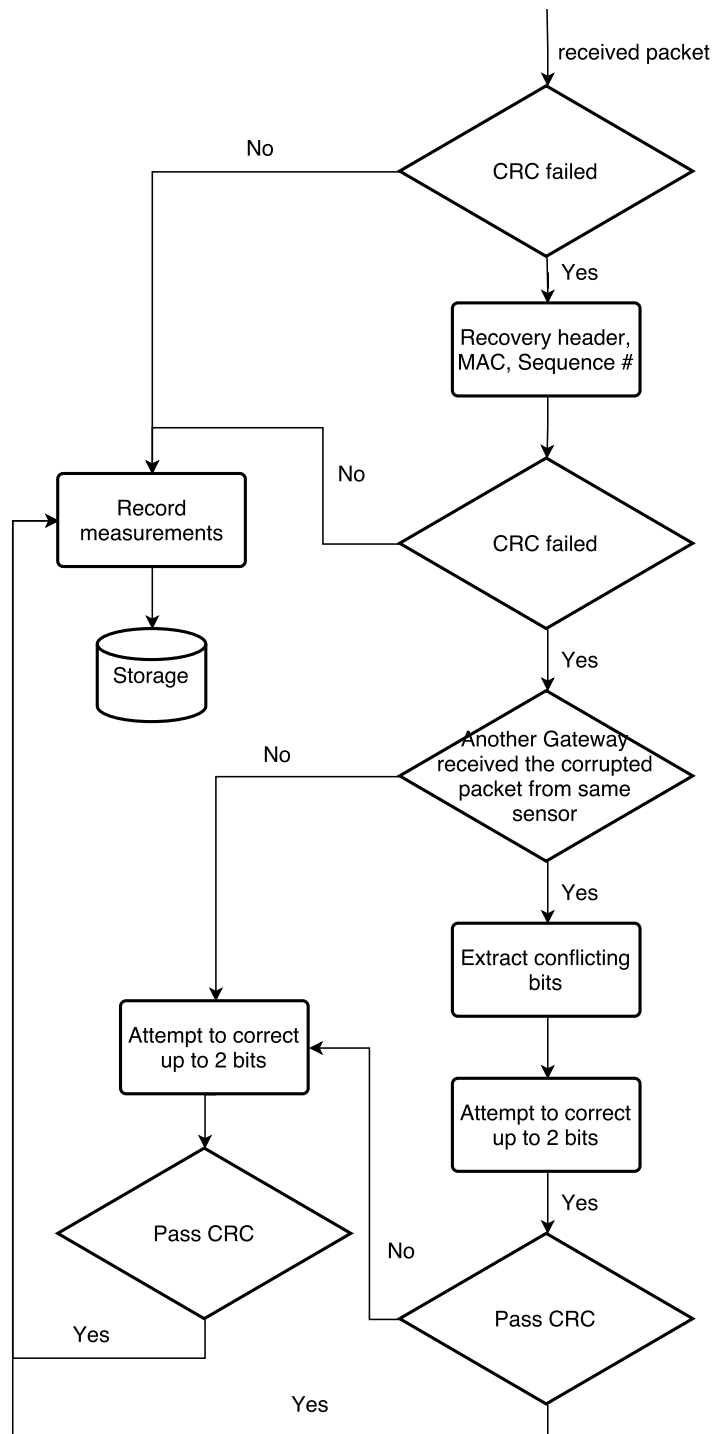


Figure 3.6: Recovery from Packet Losses and Corruption

Chapter 4

Implementation

In this chapter, we present the LEMoNet hardware and software implementations.

4.1 Hardware

In choosing hardware components, we consider power consumption, response time and resolution requirements following industry standards and best practices. For example, in DC monitoring, the accuracy of temperature measurement must be within ± 0.33 °C, and pressure sensors are required to distinguish differential pressures between -2.491 hPa to 4.982 hPa [25]. Consequently, integrated digital sensor chips are suitable to build LEMoNet sensor nodes for their ultra-low power consumption and precise measurement results. A LEMoNet gateway BLE shield and a sensor node are shown in Fig. 4.1.

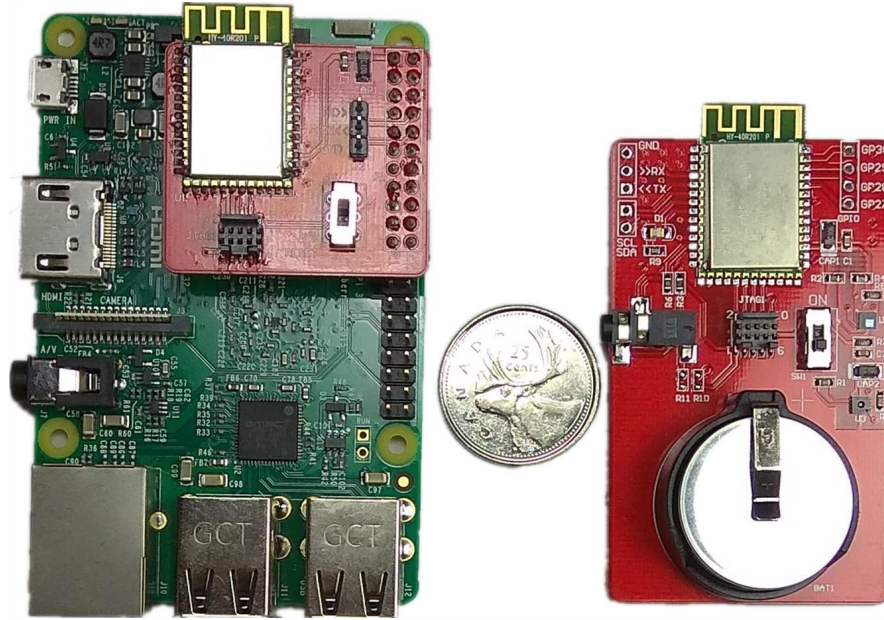


Figure 4.1: LEMoNet BLE shield and Sensor

4.1.1 Sensor Node

The core of a LEMoNet sensor node is an HY-40R201PC module [26] featuring a TI CC2640r2f wireless micro-controller. The module contains a 32-bit ARM Cortex-M3 processor running at 48 MHz. This is compatible with BLE 5 specifications. In addition, the HY-40R201PC module supports a wide supply voltage range from 1.8 to 3.8V, and the standby current usage is less than $2 \mu\text{A}$.

The sensor node contains an SHT-31-DIS temperature and humidity sensor [27]. It achieves a typical humidity accuracy of $\pm 2\% \text{RH}$, and temperature accuracy of $\pm 0.2^\circ \text{C}$ in the range from 0°C to 90°C . The sensor consumes $0.2 \mu\text{A}$ in idle state and $800 \mu\text{A}$ during measurements.

The LPS25HB pressure sensor[28] is a piezoresistive, digital output barometer.

The operating pressure range covers from 260 hPa to 1260 hPa, with ± 0.2 hPa accuracy. The sensor only consumes 700 μA during measuring and only 0.5 μA in idle mode.

4.1.2 Gateway's BLE Shield

For fast prototyping, we develop a BLE shield composed of an HY-40R201PC module that can be attached to Raspberry Pi 3 (RPI3), a single board computer that supports Linux systems. Existing BLE module on RPI3's can only receive 90% of advertising packets with the BlueZ library [29]. This may result from the WiFi and BLE transceivers on RPI3 share the same antenna. Such a low packet reception rate necessitates the design of our own BLE shield. The BLE shield communicates with RPI3 via a UART interface. The whitelist of TI CC2640r2f device stores up to 16 MAC addresses. We fill the sensor node whitelists with the MAC addresses of the gateway BLE shields.

4.2 Communication between Gateway and Controller

Controllers exchange information with gateways using the MQTT protocol. Controllers subscribe to sensor data and publish parameters for configuration updates. The gateways do not process the incoming sensor messages but, rather, publish all sensor messages. The controller subscribes and publishes the following topics:

- `/N/#` : subscribe to normal messages from all gateways

- `/E/#` : subscribe to emergency messages from all gateways
- `/R/#` : subscribe to raw data from all gateways
- `/G/gateway_id` : publish configuration parameters to gateway

The gateways subscribe and publish the following topics:

- `/G/gateway_id` : subscribe to sensor parameters to gateway
- `/N/gateway_id` : publish normal messages
- `/E/gateway_id` : publish emergency messages
- `/R/gateway_id` : publish raw data messages

The format for normal, emergency and raw data message is as follows:

sensor_id, RSS, payload

The controller identifies message types using the first topic level; and gateway use the first and the second topic levels.

Chapter 5

Evaluation

In this chapter, we present performance evaluation results. We profile the power consumption of LEMonNet sensor nodes in different modes and evaluate the performance of the LEMonNet protocol using a testbed in SHARCNET DC.

5.1 Power Consumption Profiling

A Power Monitor FTA22D [30] and a Fluke 87V industrial multimeter are used to profile the power consumption of LEMonNet sensor nodes. FTA22D has a resolution of $2.86\mu\text{A}$, while Fluke 87V is a high accuracy digital multimeter with $0.01\mu\text{A}$ resolution. However, Fluke V87 cannot log measurement data and the digital display is updated only four times per second. Therefore, we use Fluke V87 to measure the standby current and FTA22D for all other measurements. Examples of logged data from FTA22D are given in Fig. 5.1.

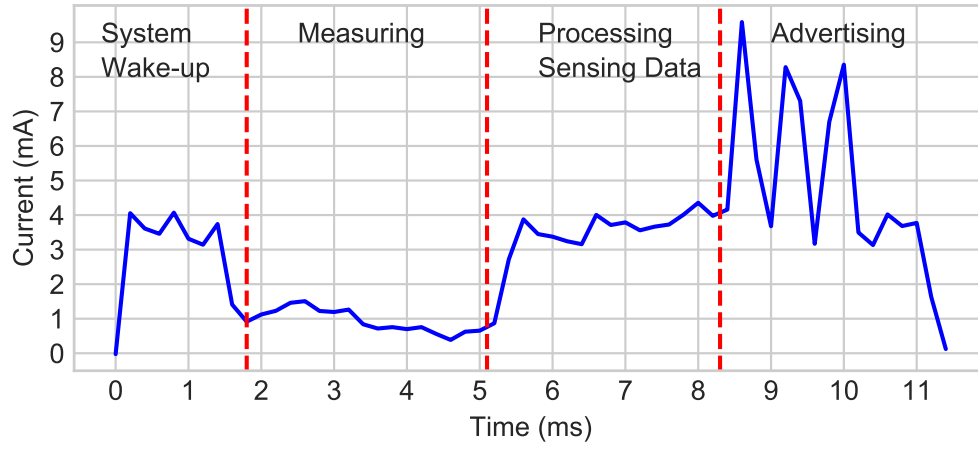
As described in Section 3.2 there are three modes of sensor operation, and each of them has different average power consumption. After waking up, the processor

Table 5.1: Current Usage Profile under Different Working Mode

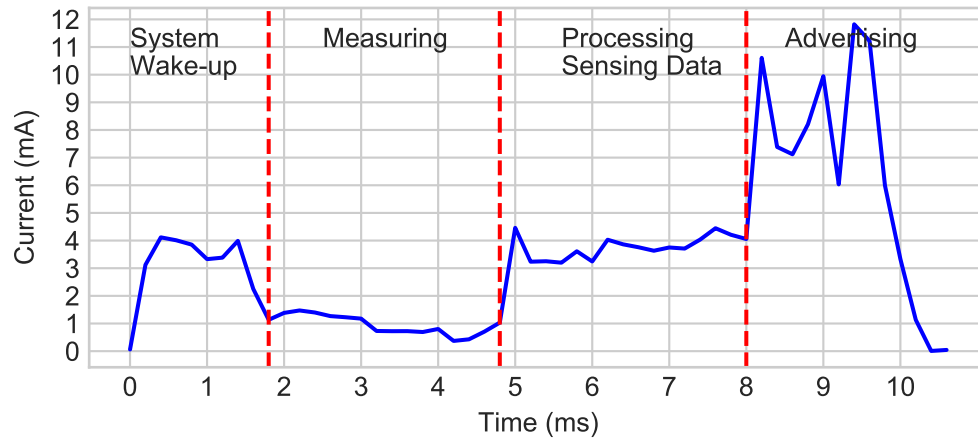
Mode	MCU	Sensors	RX	TX	Current
Standby	OFF	OFF	OFF	OFF	2 μ A
Wake-up	ON	ON	OFF	OFF	3.8 mA
Measuring	OFF	ON	OFF	OFF	1.2 mA
Processing	ON	ON	OFF	OFF	3.8 mA
Advertising	ON	OFF	OFF	ON	8.9 mA
Unicast TX	ON	OFF	OFF	ON	8.9 mA
Receiving	ON	OFF	ON	OFF	5.9 mA

sends commands to SHT-31-DIS and LPS25HB sensors to initiate the start of data sampling. After this, the processor enters standby for 3ms. The processor reads data from sensors and determines the next state. In the NCL mode, data is broadcast to gateways and the sensor node enters a low-power mode (see Fig. 5.1a). There are slight differences between SCL mode and NCL mode during the transmission of the measurement data. Specifically, depending on how long it takes to receive a SCAN_REQUEST message that confirms the reception of the data message, the transmission time varies (see Fig. 5.1b). Fig. 5.1c) illustrates the current usage in the RC mode. The sensor node pairs with a gateway to transmitting data. The pairing and transmitting time depends on how long it takes to receive a connection request packet from gateways and the size of the data, the transmission time varies.

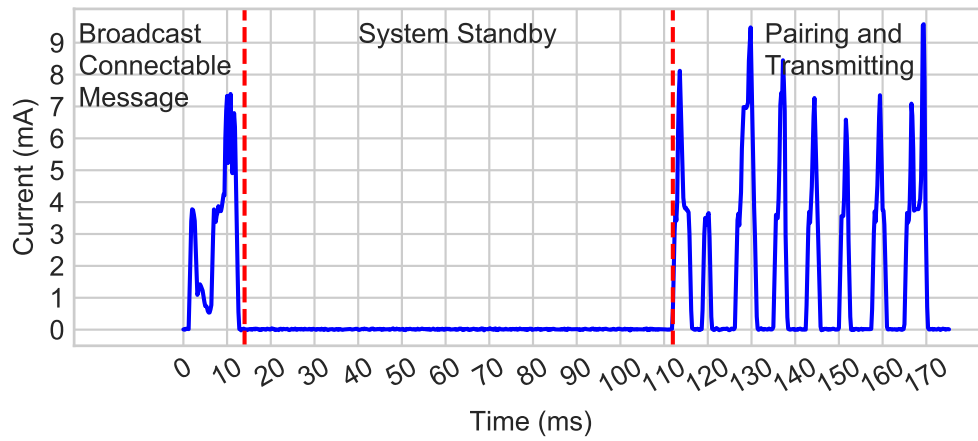
Table 5.1 shows the average current when a sensor node is in different working modes. Assuming a sensor is powered by a lithium battery with capacity 1000mAh, we can estimate the battery lifetime when operating in the NCL mode. This is calculated by considering the average current in different states. Table 5.2 summarizes the battery lifetime for different periods of collecting sensor measurements. For instance, 10s indicates the sensor node wakes up every 10 seconds to collect and transmit data via advertising. When a sensor node operates in the SCL or RC modes, it has to stay



(a) NCL Mode



(b) SCL Mode



(c) RC Mode

Figure 5.1: Current measurements of a LEMoNet sensor node in different modes.
The supply voltage is 3V

Table 5.2: Current Usage in the NCL mode

Sampling Period	Current usage (μA)	Life time (Year)
60 s	6.77	16.8
30 s	7.65	14.9
15 s	9.32	12.2
10 s	10.78	10.6
5 s	15.66	7.3
1 s	54.55	2.1

for longer periods of time and thus battery lifetime will be reduced. However, both are rare occurrences in DCs.

5.2 Testbed Evaluation

To evaluate the performance of LEMoNet, we deployed 58 sensor nodes and 3 gateway nodes in SHARCNET DC. Fig. 5.2 shows the floor plan of the DC. In DCs with air cooling, server racks are lined up in alternating rows cold air intakes facing one way and hot air exhausts facing the other. The rows composed of rack fronts are called cold aisles. Typically, cold aisles face air conditioner output ducts. The rows the heated exhausts pour into are called hot aisles. A total of 36 sensor nodes were deployed on the top, middle and bottom of 6 racks (in the first three rows) on both the front and back sides. The remaining 22 sensors were instrumented on a single rack in the bottom row with 40 servers. Gateways were placed on the top of three racks. We set the transmission power of both sensor nodes and gateways to 5 dBm.

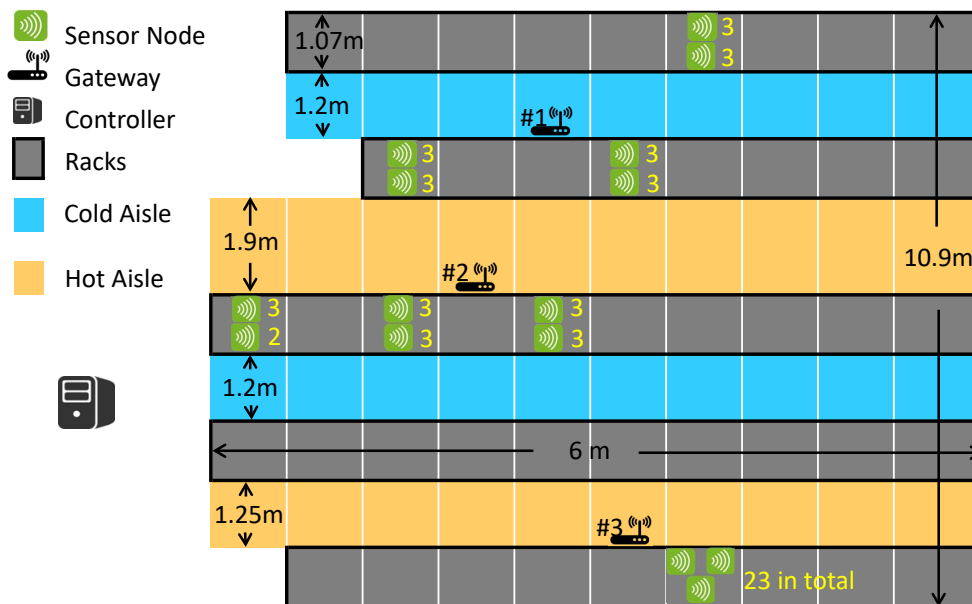


Figure 5.2: Placement of 58 sensors and 3 gateways in SHARCNET DC

5.2.1 Effects of Multi-Packet Reception

In this set of experiments, we evaluate the effect of multi-packet reception. The sampling periods are set to 30 seconds. All gateways are configured to receive advertising packets without CRC checksum on the advertising channels. The packet reception rates (PRR) for all sensor nodes combined and two sensor nodes are shown in Fig. 5.3. The two sensor nodes chosen have relatively low received signal strength from all three gateways. The average RSS of Sensor 1 at gateway 1, 2, 3 are respectively, -84.6dBm, -77.7dBm and -71.6dBm; and the average RSS of Sensor 2 at gateway 1, 2, 3 are respectively, -60.1dBm, -90.1dBm, and -74.8dBm. In Fig. 5.3, in each group, we report the percentage of received packets that pass CRC and that can be recovered using error correction at each gateway, and at the controller. A couple of observations can be made. First, there exist much diversity in packet reception across different gateways. Though the PRR at individual gateways before error correction



Figure 5.3: Packet reception rate. GW1_recv, GW2_recv, GW3_recv and C_recv are the percentages of received packets passing CRC at the three gateways and at the controller. GW1_rec, GW2_rec, GW3_rec and C_rec are the percentage of packets recovered.

can be as low as 93.5%, the PRR at the controller is higher than the best gateway. Second, though error correction is useful to recover certain sensor nodes' packets at a particular gateway (e.g., Sensor 2 at Gateway 2), its impact on PRR is marginal at the controller. This is because the controller by combining correctly received packets from multiple gateways, can attain a very high PRR (at 99.4%).

5.2.2 Effects of Sampling Period

Next, we investigate the effect of sensor sampling periods. Smaller sampling periods result in higher traffic loads in the network. Fig. 5.4 reports the PRRs at the controller after multi-packet reception. The PRRs are calculated for every 512 packets transmitted by each sensor nodes. In accordance with our prediction, as the sampling

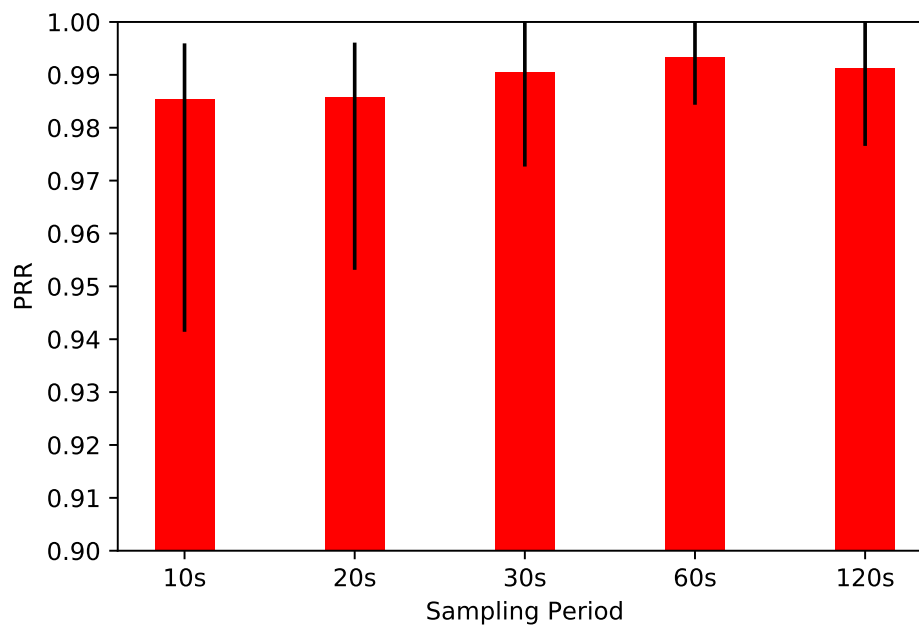


Figure 5.4: Effects of sampling interval on PRR in the NCL mode

period increases, the traffic load decreases and overall the PRR increases. The average PRR in all cases is above 98.4%. Note this is accomplished with no extra power consumption on the sensor nodes.

5.2.3 Number of Transmissions in the SCL mode

In this set of experiments, we fix the sampling period of sensor node to be 30 seconds and evaluate the co-existence of NCL and SCL modes. Specifically, three scenarios are evaluated,

- **Normal operation:** The temperature threshold for sensors in cold aisles is set to $24^{\circ}C$ ¹. When the measured temperature exceeds the threshold, a sensor node operates in the SCL mode to ensure reliable transfer of the measurement.

¹High cold aisle temperatures are often indications of potential hotspots and inefficient cooling.

- **Sensors in cold aisle:** All sensor nodes in cold aisles are in the SCL mode while sensor nodes in hot aisles are in the NCL mode unless an outage occurs. This and the next scenario are used to stress-test the network.
- **All sensors:** Sensor nodes in both hot and cold aisles are in the SCL mode.

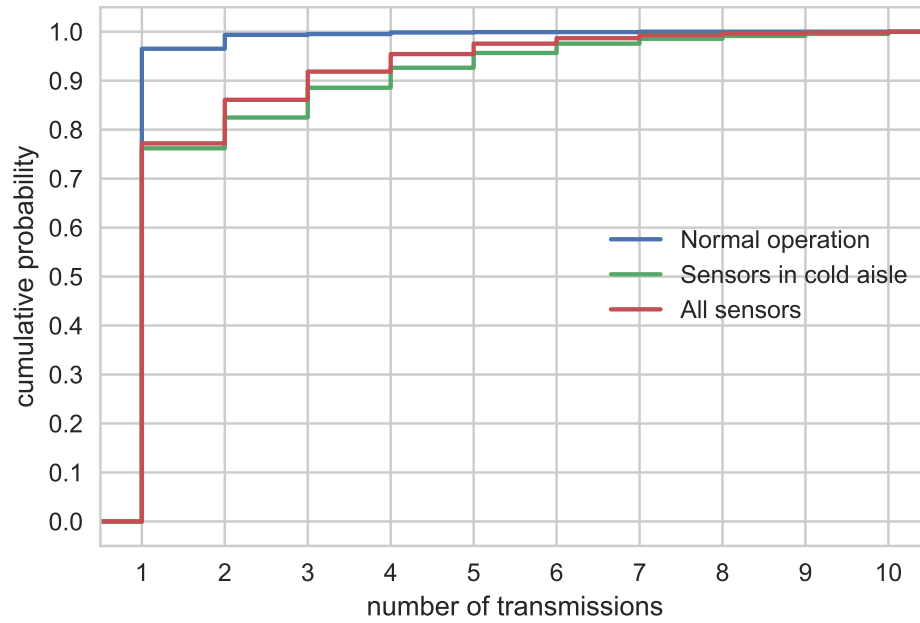


Figure 5.5: The number of transmissions before receiving a scan request

Fig. 5.5 shows the number of transmissions a sensor node makes in the SCL mode before receiving SCAN-REQ (as acknowledgment). In the first scenario, at most one retransmission is needed to deliver measurement data reliably while in the second and third scenarios, around 90% of data can be delivered reliably with four transmissions (or three retransmissions). In the first and second scenarios, the average PRRs for sensor nodes in the NCL mode are 98.6% and 99.0%. No discernible changes can be observed in PRR for sensor nodes in the NCL mode.

Chapter 6

Insights from Deployment

In this chapter, we present our insights on DC operations gained from deploying LEMoNet. This is done in a modular DC with in-row cooling called MCX and a high-performance DC for SHARCNET¹ that features raised floor and overhead cooling.



Figure 6.1: MCX DC layout. The cooling units are at both sides of the DC.

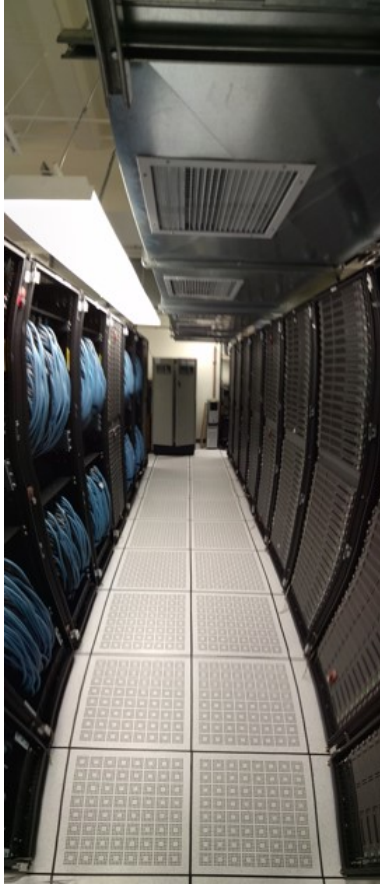
¹The Shared Hierarchical Academic Research Computing Network (SHARCNET) is a consortium of 18 universities, colleges and research institutes providing a range of high-performance computers and software, linked by an advanced fibre optics network.

The MCX DC is an experimental DC designed to evaluate different cooling and IT strategies. The researchers fully control it. It contains 2 cooling units and 5 racks as shown in Fig. 6.1. The power distribution unit (PDU) can only supply enough power for 61 servers therefore we deployed 61 servers in five racks MCX DC. The servers are at top, middle, and bottom of the racks and empty slots are covered by blanking panels to reduce local air recirculation. The servers provide on-board measurements such as inlet temperature, CPU core temperature, and memory temperature for intelligent management as summarized in Table 6.1. The resolution of inlet temperature reports is 1 °C.

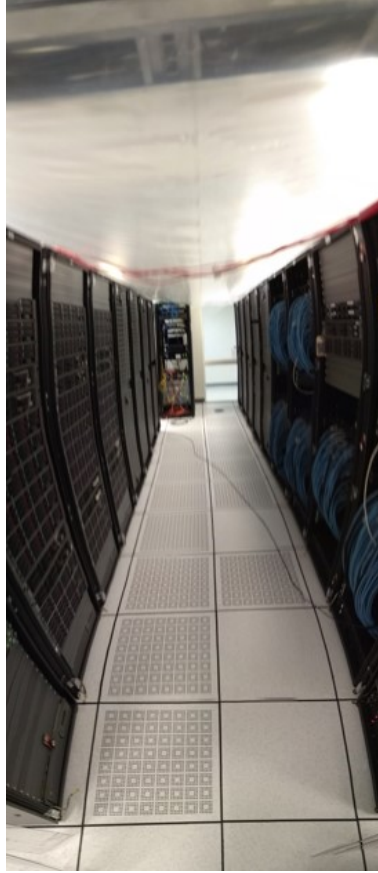
Table 6.1: The Number of Servers Providing Thermal Measurements in MCX DC

Vendor	Number of Servers	Inlet Temperature	Power	CPU Core Temperature	Memory Temperature
DELL	23	22	2	19	16
HP	38	21	20	21	21

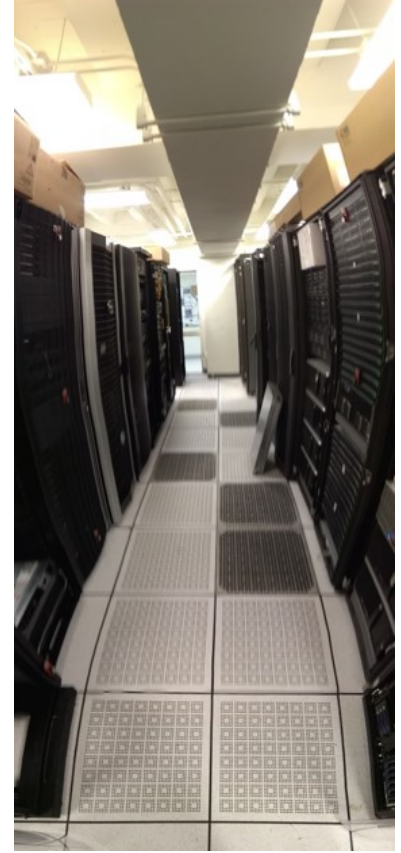
The SHARCNET DC is a high-performance co-location DC. It has 3 different cold aisle layouts as shown in Fig. 6.2. It is a production DC. Researchers have no control over server utilization, cooling settings and have no access to temperature information from the servers. Default perforated tiles have a 22% opening. The servers on cold aisle 1 (Fig. 6.2a) are computational servers that typically operate at close to 100% utilization. The raised floor cooling system cannot supply enough cold air to all the servers. An overhead cooling system is added to cold aisle 1 to provide additional cooling capacity. Most servers on cold aisle 2 (Fig. 6.2b) are data storage servers with relatively low heat load. Cold aisle 2 is covered from the top to avoid mixing cold and hot air. Most servers on cold aisle 3 (Fig. 6.2c) have low utilization; therefore racks with high utilization contain perforated tiles with 56% opening in front to allow more



(a) Cold Aisle 1. The aisle is cooled by both raised floor and an overhead cooling unit.



(b) Cold Aisle 2. The aisle is cooled by raised floor cooling. Additionally, containment is provided on top using plastic wraps.



(c) Cold Aisle 3. The aisle is cooled using raised floor cooling. Some perforated tiles (darker ones) have 56% openings to allow more cold air.

Figure 6.2: Cold aisle layouts of an SHARCNET DC

cool air. Each cold aisle has its own cooling unit. There is very little airflow across cold aisles, primarily because the fan speeds of the cooling units are constant. The cooling units use an On/Off controller. When the measured temperature is below a threshold (e.g., T_H), the chiller is turned off. When the temperature is above another threshold (e.g., T_L), the chiller is turned on.

6.1 Impacts of Cold Aisle Layouts in the SHAR-CNET DC

We deployed 54 sensor nodes on the top, middle and bottom of 18 racks in cold aisles to investigate the impact of different cold aisle layouts. The 1 hour inlet temperature readings are shown in Fig. 6.3. Several observations can be made from the results. Firstly, as indicated in Fig. 6.3 (a) – (c), the additional overhead cooling unit has in fact over cooled the servers to below 18 °C. The acceptable operating temperature range by ASHRAE is between 18 °C and 27 °C[31]. This leads us to believe that the overhead cooling may not be necessary or can be configured with a higher set point temperature resulting in energy savings. Another interesting observation is that the two cooling units appear to operate out of phase from the “peak shedding” effects. Secondly, the variation of the inlet temperature in the cold aisle with top containment is less than the other two aisles. This can be attributed to the low heat generated from the servers and possibly the effect of containment. Thirdly, comparing the temperature readings of the top, middle and bottom locations, there are no obvious locations that have the lowest temperature consistently. Though heterogeneous workload on the servers may play a role here, we believe the phenomenon is caused by non-trivial interactions among cold air from the cooling unit(s) and the server fans.

6.2 Measurement Study in the MCX DC

For the remaining of this chapter, we review the results of experiments conducted in the MCX DC. First, we evaluate the correlation between the inlet temperature of readings from on-board sensors and LEMoNET sensors deployed in MCX DC.

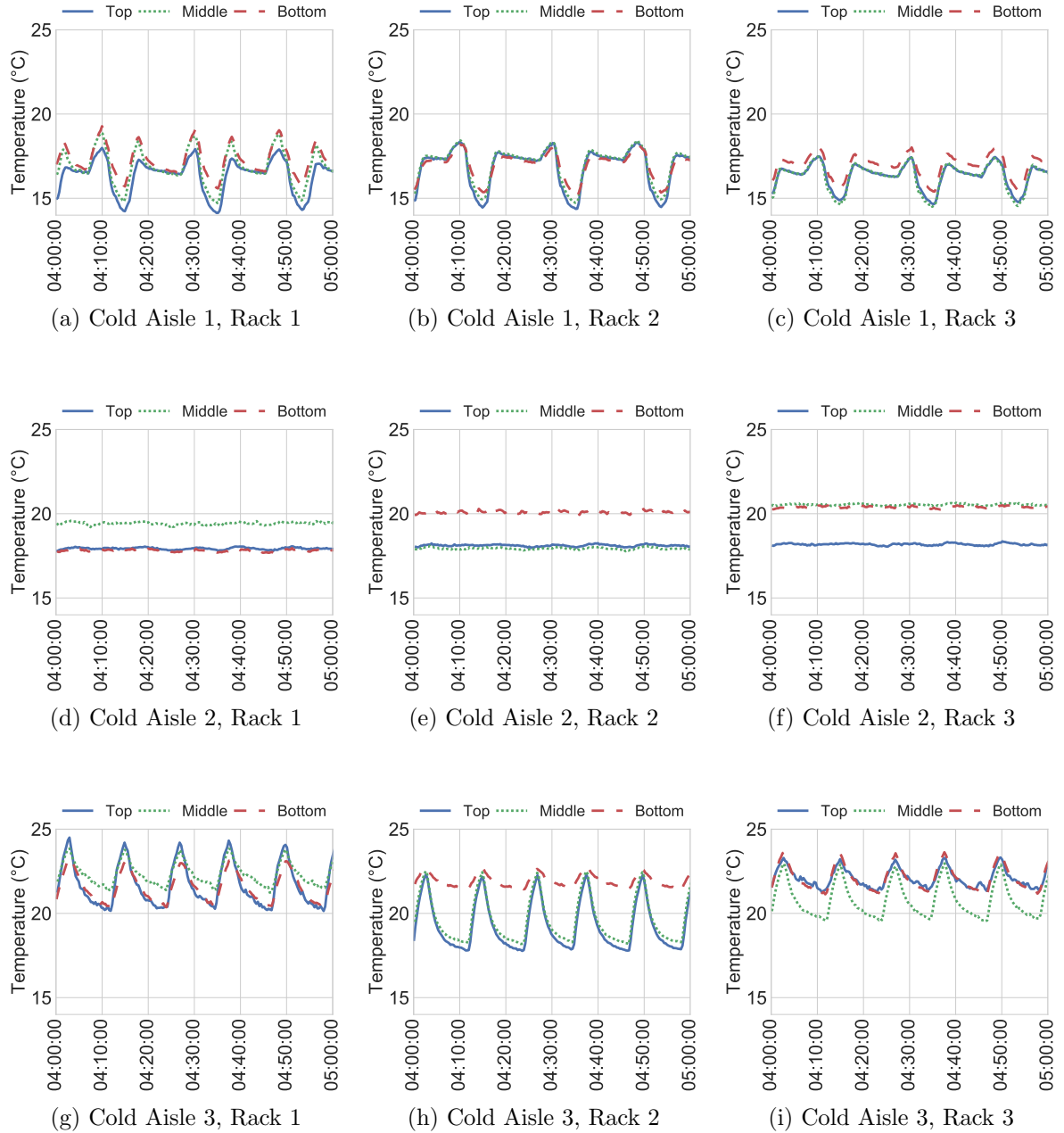


Figure 6.3: Inlet temperature readings of 9 racks at SHACNET DC

We instrumented 6 sensors at the top, middle and bottom of each rack in both cold and hot aisles. Table 6.2 shows the number of servers reporting inlet temperature

reading and we find that there is at least one server providing inlet temperature in each zone. Measurements are collected over a two-hour period. Table 6.3 summarizes the utilization of all servers, the set points of the two cooling units and the state (“on” or “off”) of the cooling unit next to rack 5 at different time intervals during the two-hour evaluation.

Table 6.2: The Number of Servers Provides Inlet Temperature at Different Zone of MCX DC

	Rack 1	Rack 2	Rack 3	Rack 4	Rack 5
Top	1	3	2	3	4
Middle	4	1	2	2	4
Bottom	4	4	5	3	1

Table 6.3: Settings at Different Time Interval During the Correlation Experiment at MCX DC

	1	2	3	4	5	6	7
Cooling Unit Set Point	20 °C	20 °C	15 °C	15 °C	20 °C	20 °C	20 °C
Utilization of All Servers	70%	0%	0%	70%	0%	70%	70%
Cooling Unit 2	ON	ON	ON	OFF	ON	ON	OFF

Correlations among temperature readings

From Fig. 6.4, we observe that the inlet temperature reported by the deployed sensors and the servers follow the same trend in most cases except for some offsets between readings. Define the correlation of two time series data as, Fig. 6.5 shows the CCDF of the correlation coefficients between the inlet temperature reported by the server and deployed sensors in the same zone. The lowest correlation is 0.68 and is at the bottom of rack 5. The offset between the readings from the deployed sensor and HP server 49 at the bottom of rack 5 is around 5 °C. HP server 49 takes longer

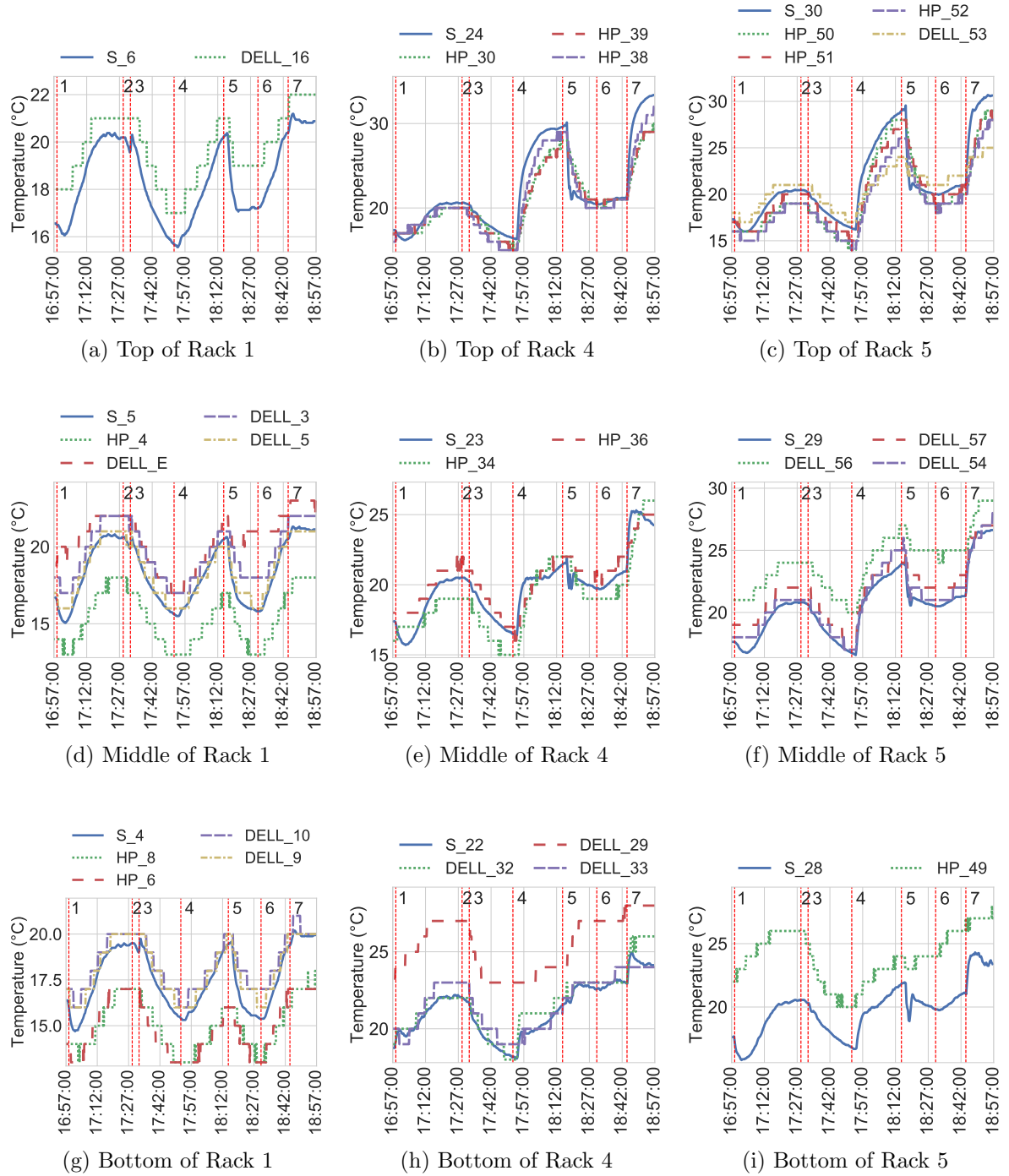


Figure 6.4: Inlet temperature readings reported by deployed sensors and servers. The vertical red lines indicate the starts and ends of evaluation periods with different configurations (Table 6.3).

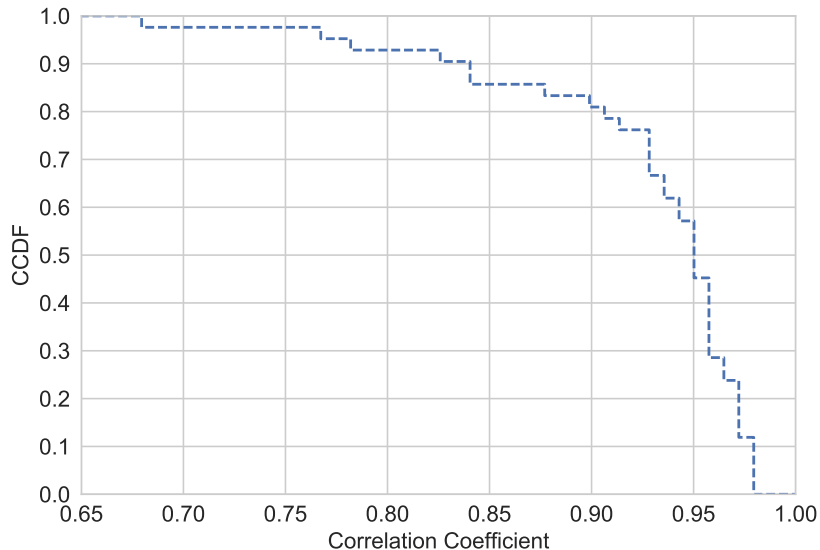


Figure 6.5: The CCDF of the correlation coefficients between the inlet temperature of the server and the deployed sensor in the same zone at MCX DC.

time to heat up when the cooling unit was off (see Fig. 6.4i). Overall, we observe good agreement between the inlet temperature readings reported by the deployed sensors and the servers. This confirms the fidelity and utility of LEMoNet for co-location data centers, where on-board sensor readings are not available. However, some discrepancy does exist. There are a number of contributing factors. First, the sensors are instrumented at the center of each zone and slightly away from the rack. The thermal conditions are different from those inside the server. Second, some of the servers are relatively old (up to 10 years). It is possible the fidelity of their on-board sensors degrade over time or they need to be recalibrated. Third, the resolution of on-board sensors is quite low $1\text{ }^{\circ}\text{C}$.

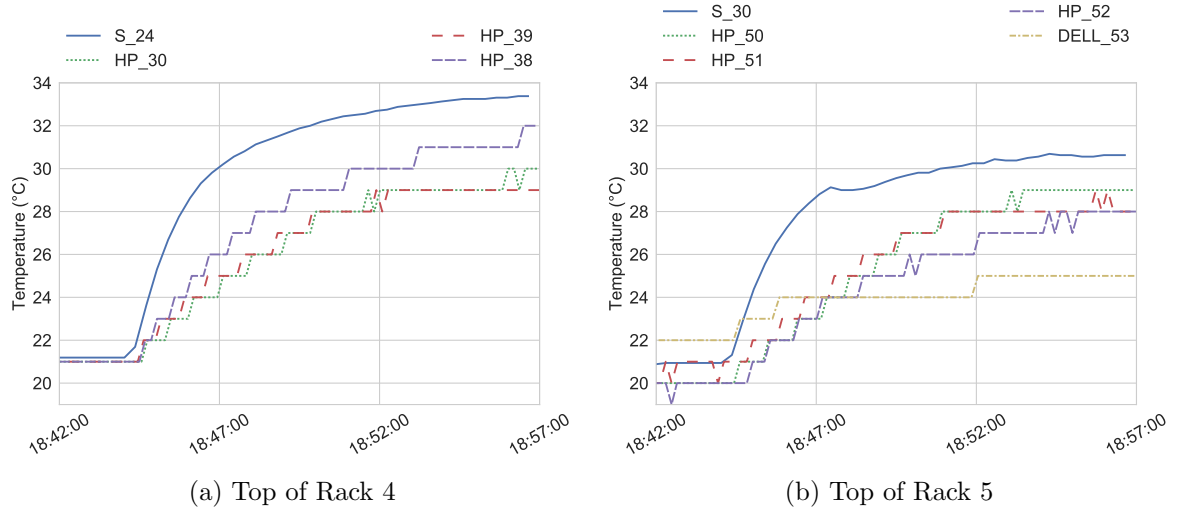


Figure 6.6: Inlet temperature timelines at top of rack 4,5 during time interval 7 of the correlation experiment. The set point was 20 °C. Cooling unit next to rack 5 was off. The utilization of all servers are 70%.

6.2.1 Effects of a single cooling unit

During the evaluation in the SHARCNET DC, the cooling units were off several times due to power outage and broken pipes. To protect IT equipments, sensors should report temperature hot spots in time. After one cooling unit is turned off in MCX DC, the temperature reported by both types of sensors rises quickly. However, it is interesting to see that the readings from deployed sensors have a large gradient. For example, readings from the deployed sensor reached 28 °C in 3 minutes while the on-board sensors reached 28 °C in 5 minutes at the top of rack 4 (see Fig. 6.6a).

6.2.2 Effects of server utilization

In this section, we evaluate server utilization impacts on inlet temperature in the MCX DC. We set the set points for the two cooling units at 20 °C and vary the utilization of all servers. The fan speed and chiller of cooling units are set to automatic. Table 6.4 and 6.5 shows the inlet temperature reported by the deployed sensors when the servers are idle and at 70% utilization, respectively. Somewhat counter-intuitively, the inlet temperature is higher when servers are idle. This is because the cooling units are operating more frequently to remove heat generated by servers.

Table 6.4: Temperature at Different Zones of the MCX DC When Servers are Idle. Set point for both cooling units are 20 °C.

	Rack 1	Rack 2	Rack 3	Rack 4	Rack 5
Top	20.19 °C	22.19 °C	22.13 °C	21.06 °C	20.75 °C
Middle	20.56 °C	20.69 °C	21.06 °C	20.50 °C	21.31 °C
Bottom	19.38 °C	19.88 °C	21.25 °C	22.25 °C	20.24 °C

Table 6.5: Temperature at Different Zones of the MCX DC When Servers are at 70% Utilization. Set point for both cooling units are 20 °C.

	Rack 1	Rack 2	Rack 3	Rack 4	Rack 5
Top	19.63 °C	20.19 °C	20.75 °C	20.31 °C	20.25 °C
Middle	20.19 °C	19.88 °C	20.38 °C	20.19 °C	20.63 °C
Bottom	19.06 °C	19.50 °C	20.56 °C	21.75 °C	20.13 °C

6.2.3 Effects of blanking panel

Blanking panels block openings on server racks to reduce local air recirculation. In this set of experiments, we evaluate the temperature changes with/without blanking panel in MCX DC. The set point for both cooling units are both at 20 °C. Only the blanking panels between the middle and top servers on are removed rack 5. Table 6.6

shows the temperature in steady state before and removed the blanking panels are removed. The results show that temperature increases more than 1 $^{\circ}C$ at the top and more than 0.5 $^{\circ}C$ at the middle. Therefore, we conclude the blanking panel eliminates local recirculation.

Table 6.6: Rack 5's Inlet Temperature With/Without Blanking Panel at Middle up of Rack 5. The Set Point is 20 $^{\circ}C$ and servers are at 70% utilization.

	With Blanking Panel	Without Blanking Panel
Top	20.25 $^{\circ}C$	21.81 $^{\circ}C$
Middle	20.63 $^{\circ}C$	21.16 $^{\circ}C$
Bottom	20.13 $^{\circ}C$	20.28 $^{\circ}C$

Chapter 7

Conclusion

In this thesis, we proposed LEMoNet, a low energy wireless sensor network design for data center monitoring. It employs a two-tier network architecture and a multi-mode data exchange protocol to balance the trade-offs between low power consumption and high data reliability. The experiments conducted at the SHARCNET data center demonstrated the high data reliability of the proposed design. Insights from our study demonstrated the demands for deploying wireless sensor network for monitoring co-location DC.

As described in Section 2.2, BLE link layer uses whitelist to filter advertisers and scanners. In the future research, we will investigate the use of whitelist to filter advertisers and scanners given the restriction of the size of whitelist. The TI BLE chip's whitelist can store at most 16 MAC addresses and the Nordic BLE chip's whitelist can store 8 MAC addresses. One possible solution is to change the firmware to increase the white list size. Another option is to devise a cache replacement mechanism to effectively utilize the limited whitelist.

We will also perform longitudinal study in different DC layouts to determine

the sensor placement based on measurements collected by LEMoNet. The sensor placement problem can be formulated as a constrained optimization problem. The optimization goal is to maximize the likelihood of anomaly detection. For example, the temperature anomalies during the cooling system failure.

Bibliography

- [1] P. Delforge, “Americas data centers are wasting huge amounts of energy,” *National Resources Defense Council, vol. Issue Brief*, pp. 14–08, 2014.
- [2] R. R. Schmidt, E. Cruz, and M. Iyengar, “Challenges of data center thermal management,” *IBM Journal of Research and Development*, vol. 49, no. 4.5, pp. 709–723, 2005.
- [3] X. Wang, X. Wang, G. Xing, and C.-X. Lin, “Leveraging thermal dynamics in sensor placement for overheating server component detection,” in *Green Computing Conference (IGCC), 2012 International*. IEEE, 2012, pp. 1–10.
- [4] G. Bell, “Wireless sensors improve data center energy efficiency,” *US Dept. Energy, Lawrence Berkeley Nat. Lab., Berkeley, CA, USA*, 2010.
- [5] R. Mahdavi and W. Tschudi, “Wireless sensor network for improving the energy efficiency of data centers,” Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), Tech. Rep., 2012.
- [6] C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao, “RACNet: a high-fidelity data center sensing network,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2009, pp. 15–28.

-
- [7] M. G. Rodriguez, L. E. O. Uriarte, Y. Jia, K. Yoshii, R. Ross, and P. H. Beckman, "Wireless sensor network for data-center environmental monitoring," in *Sensing Technology (ICST), 2011 Fifth International Conference on*. IEEE, 2011, pp. 533–537.
- [8] K. Hong, S. Yang, Z. Ma, and L. 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*. IEEE, 2013, pp. 263–271.
- [9] J. Chen, R. Tan, Y. Wang, G. Xing, X. Wang, X. Wang, B. Punch, and D. Colbry, "A high-fidelity temperature distribution forecasting system for data centers," in *Real-Time Systems Symposium (RTSS), 2012 IEEE 33rd*. IEEE, 2012, pp. 215–224.
- [10] A. Saifullah, S. Sankar, J. Liu, C. Lu, R. Chandra, and B. Priyantha, "Capnet: A real-time wireless management network for data center power capping," in *Real-Time Systems Symposium (RTSS), 2014 IEEE*. IEEE, 2014, pp. 334–345.
- [11] J. Hughes, J. Yan, and K. Soga, "Development of wireless sensor network using bluetooth low energy (BLE) for construction noise monitoring," *International Journal of Smart Sensing and Intelligent Systems*, vol. 8, no. 2, pp. 1379–1405, 2015.
- [12] L. Alliance, "A technical overview of lora and lorawan," 2016.
- [13] B. S. I. Group, "Bluetooth core specification 5.0," 2016.
- [14] M. W. Group, "Bluetooth® mesh profile specification 1.0," 2017.
- [15] A. Banks and R. Gupta, "Mqtt version 3.1. 1," *OASIS standard*, vol. 29, 2014.

- [16] A. Stanford-Clark and H. L. Truong, “Mqtt for sensor networks (mqtt-sn) protocol specification,” *International business machines (IBM) Corporation version*, vol. 1, 2013.
- [17] T. Rault, A. Bouabdallah, and Y. Challal, “Energy efficiency in wireless sensor networks: A top-down survey,” *Computer Networks*, vol. 67, pp. 104–122, 2014.
- [18] S. Bandyopadhyay and A. Bhattacharyya, “Lightweight internet protocols for web enablement of sensors using constrained gateway devices,” in *Computing, Networking and Communications (ICNC), 2013 International Conference on*. IEEE, 2013, pp. 334–340.
- [19] M. Singh, M. Rajan, V. Shivraj, and P. Balamuralidhar, “Secure mqtt for internet of things (iot),” in *Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on*. IEEE, 2015, pp. 746–751.
- [20] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda *et al.*, “A line in the sand: a wireless sensor network for target detection, classification, and tracking,” *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.
- [21] P. Koopman, “32-bit cyclic redundancy codes for internet applications,” in *Dependable Systems and Networks, 2002. DSN 2002. Proceedings. International Conference on*. IEEE, 2002, pp. 459–468.
- [22] R. W. Hamming, “Error detecting and error correcting codes,” *Bell Labs Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

- [23] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300–304, 1960.
- [24] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "Mqtt-sa publish/subscribe protocol for wireless sensor networks," in *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*. IEEE, 2008, pp. 791–798.
- [25] P. Rod Mahdavi, "Data center energy efficiency measurement assessment kit guide and specification," 2014.
- [26] L. SHENGRUN Rechnology Co. (2017) Hy-40r201 bluetooth ble 5.0 module specifications. [Online]. Available: <http://www.tuner168.com/uploads/soft/171114/3-1G114161337.pdf>
- [27] Sensirion. (2017) Sht3x-dis datasheet. [Online]. Available: https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/2_Humidity_Sensors/Sensirion_Humidity_Sensors_SHT3x_Datasheet_digital.pdf
- [28] STMicroelectronics. (2016) Lps25hb datasheet. [Online]. Available: <http://www.st.com/content/ccc/resource/technical/document/datasheet/9a/4c/aa/72/1f/45/4e/24/DM00141379.pdf/files/DM00141379.pdf/jcr:content/translations/en.DM00141379.pdf>
- [29] M. Krasnyansky, "Bluez: Official linux bluetooth protocol stack," 2003.
- [30] M. S. Inc. (2014) Mobile device power monitor manual. [Online]. Available: <http://msoon.github.io/powermonitor/PowerTool/doc/LVPM%20Manual.pdf>

- [31] ASHRAE. (2016) Data center power equipment thermal guidelines and best practices. [Online]. Available: https://tc0909.ashraetcs.org/documents/ASHRAE_TC0909_Power_White_Paper_22_June_2016_REVISED.pdf