SoftBLE: An SDN Framework for BLE-based IoT Networks

Mehdi Jafarizadeh, Xingzhi Liu
Dept. of Computing and Software, McMaster University
Hamilton, Ontario, Canada
(jafarizm,liux57)@mcmaster.ca

Rong Zheng
Dept. of Computing and Software, McMaster University
Harbin Institute of Technology (Shenzhen), China
rzheng@mcmaster.ca

ABSTRACT

Today's Industrial IoT (IIoT) applications often employ large-scale and dense sensor deployments for environmental monitoring. A hierarchical Bluetooth Low Energy (BLE) based architecture can facilitate power efficiency and reliability for data collection in dense networks. But if the network is static with fixed parameter settings, it can not be adaptable to dynamic application requirements. Although BLE is a parametric protocol, it does not provide any built-in feature for parameter tuning. To achieve network adaptability, we introduce and design SoftBLE, a Software Defined Networking (SDN) framework that provides controllability for BLE based 2-tier networks. 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. Evaluation results from both an experimental testbed and a large-scale simulation study show that almost all the SoftBLE sensors can save around 70% of transmission power while keeping Packet Reception Rate (PRR) above 99.9%.

CCS CONCEPTS

• **Networks → Network protocol design**; *Ad hoc networks.*

KEYWORDS

Bluetooth Low Energy, Software Defined Networking, Internet of Things

ACM Reference Format:
Mehdi Jafarizadeh, Xingzhi Liu and Rong Zheng. 2021. SoftBLE: An SDN Framework for BLE-based IoT Networks. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '21), May 18–21, 2021, Charlottesvle, VA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3450268.3453527

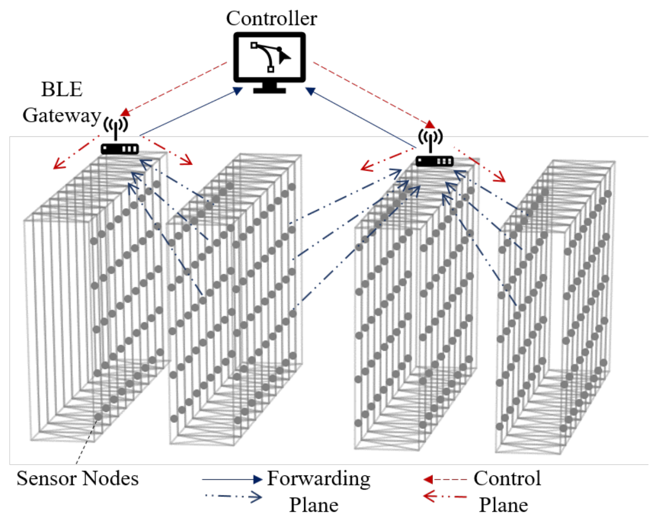


Figure 1: A 2-tier IoT Network for DC monitoring. Thermal sensors are instrumented in the front and back of servers in a DC. Measured data are sent to the controller first via BLE to the gateways and then through Ethernet to the controller. Control packets are disseminated in reverse from the controller via the gateways to the sensors.

1 INTRODUCTION

Environmental and process monitoring are emerging as key applications in many IIoT networks, from water quality assessment in agriculture [26], bridge displacement monitoring [13], smart power [8], and malfunction detection in industrial plants [32], to efficient thermal management in smart buildings [9] and data centers (DCs) [15]. Most of these monitoring applications require dense and/or on-demand installation of thousands of wireless sensors in target environments. For example, in a DC ambient air monitoring system (Figure 1), sensors that are installed in the front and back of servers, can have a density as high as 100 sensors per meter cube.

Due to limited wireless bandwidth, 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 $\Theta\left(\frac{1}{n}\right)$ [34]. Thus, recent successful solutions to DC monitoring ([15] and [7]) employ 2-tier networks where multiple backbone nodes (gateways) connected through wires in the second tier gather data from sensors in the first tier. Battery-powered sensors communicate with the gateways through low-power wireless radios such as ZigBee, BLE or LoRA for better power conservation and low deployment costs.

Among competing short-range RF technologies, BLE has gained popularity due to its low power consumption and wide availability [15]. To avoid the overhead of connection establishment and tear down for short messages, many BLE based monitoring systems adopt connection-less advertising modes for sensor data collection. However, legacy advertising of BLE is proven to be unscalable in neighbor discovery if used haphazardly [25]. Even with recent improvements such as BLE mesh or periodic and extended advertising,

IoTDI '21, May 18–21, 2021, Charlottesvle, VA, USA

ACM ISBN 978-1-4503-8354-7/21/05…$15.00
https://doi.org/10.1145/3450268.3453527

the scalability problem persists [27]. The root cause is that there is no one-size-fit-all fixed BLE parameter settings for all applications or deployment environments. It is important to be able to *configure and adapt* BLE parameters both during deployment and at operational time.

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 [19, 21, 25], or tried to improve the overall network performance by tuning one of them [6, 20, 28]. 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 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.

The operations of SoftBLE over a 2-tier BLE-base IoT network are highlighted in Figure 1. The first tier provides communication between sensors and BLE gateways, and the second tier connects the gateways to a central controller via reliable wired links. SoftBLE enables separation between a control plane, which is responsible for provisioning the sensors, and a forwarding plane, which routes the measurements from provisioned sensors. To facilitate gateway orchestration, we formulate an integer programming problem and propose an efficient heuristics solution. For sensor orchestration, we first develop analytical models to characterize the relationships between key performance metrics such as PRR and power consumption, and control parameters in 2-tier networks. The models are utilized in finding the optimal scanning and advertising parameters that minimize sensor power consumption subject to reliability constrains.

The main contributions of this work are two-fold:

(1) An SDN control plane is designed as an overlay on a two-tier forwarding plane for BLE IoT networks.
(2) Two orchestration algorithms are proposed to optimize the scanning parameters on the gateways and the advertising parameters on the sensors based on run-time measurements.

SoftBLE has been implemented and evaluated using devices equipped with TI cc2640r2 MCU chips. A 48-sensor, 2-gateway testbed has been deployed in a 11m-by-8m laboratory. Experiments show that SoftBLE outperforms baseline approaches in both PRR and power consumption. To further evaluate its scalability, we have implemented SoftBLE in the OMNET++ network simulator. The new SDN based orchestration schemes can save up to 70% energy while maintaining at least 99.9% PRR in a network of 2500 sensors.

In the rest of the paper, after necessary background on BLE in Section 2, the design of the SDN framework and the details of the control plane are presented in Section 3 and 4 respectively. It is followed by the details of the proposed orchestration algorithms for the gateways and the sensors in the sections 5 and 6 respectively. Finally, evaluation results are presented in Section 7, and related works are reviewed in Section 8. We conclude the paper in Section 9.

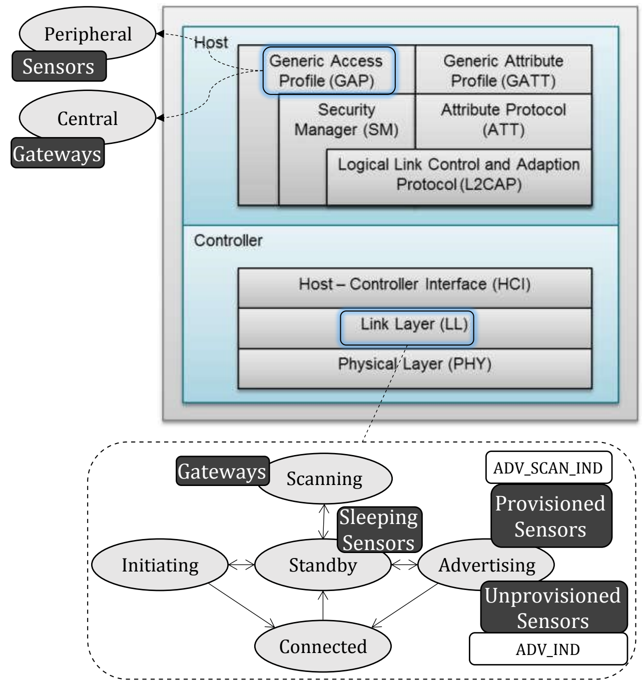


Figure 2: BLE protocol stack.

2 BACKGROUND

In this section, we review the basics of the BLE protocol that are crucial to the design of SoftBLE.

2.1 BLE Protocol Stack

As shown in Figure 2, 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-LL): Standby, Advertising, Scanning, Initiating, and Connected. Three states happen in our base solution. Sensors periodically switch between *Advertising* and *Standby*, while gateways are always in the *Scanning* state. Connection states are omitted 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.

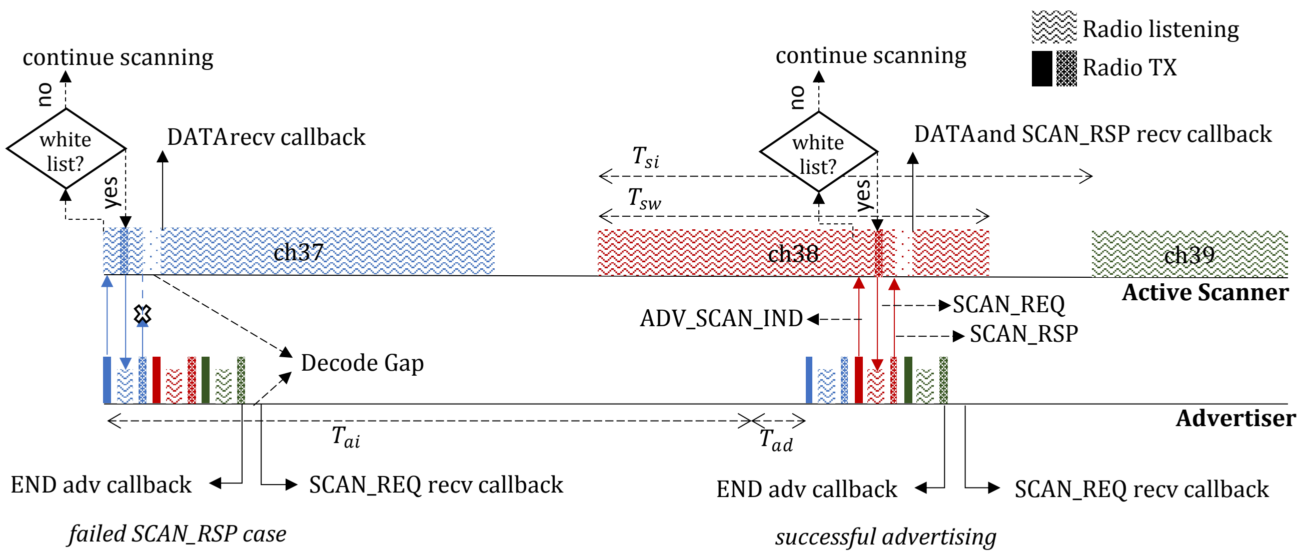


Figure 3: The time line of advertising *ADV_SCAN_IND* packets to an active scanner.

2.2 PDU Types of BLE Advertising Packets

Every BLE packet PDU contains two-byte header representing its type code, followed by 6 bytes of advertising address 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 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 3 shows the timeline of packet transmissions between an *ADV_SCAN_IND* advertiser, and an active scanner.

3 SDN FRAMEWORK COMPONENTS

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 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

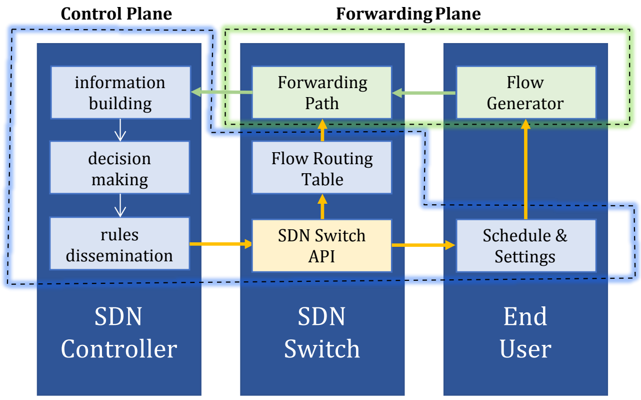


Figure 4: The components of SDN framework in a SoftBLE.

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 4, 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 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) [15]. 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).

4 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)

Sensors orchestration determines and configures the TX power and advertising channel map of the sensors, and consequently the whitelists of gateways (Section 6).

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.

4.1 Provisioning Timeline

The detailed message exchanges in different stages of SoftBLE are shown in Figure 5. 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.

4.2 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

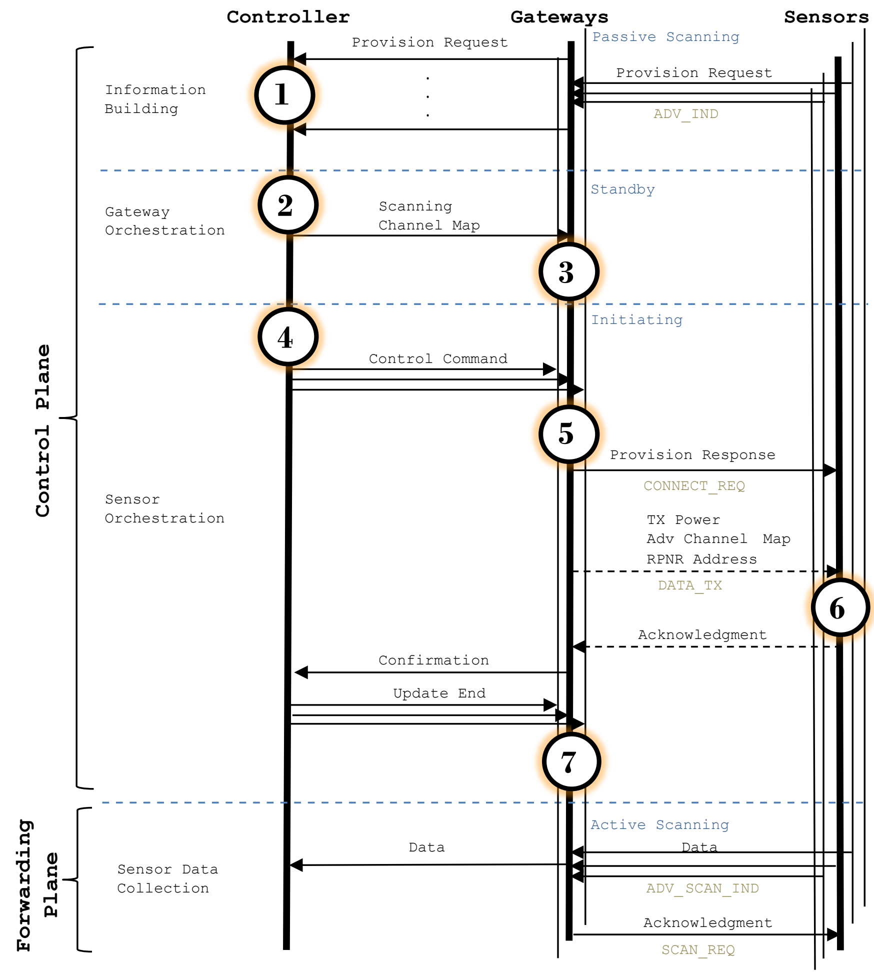


Figure 5: The timeline of packet transmission in the control and forwarding plane of SoftBLE.

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:

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

Unlike the analytical model of LEMoNet, where the RSS values are estimated based on distance, in (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.

4.3 Control Knobs

Five parameters can be tuned on the BLE devices, three on sensors and two on gateways. These parameters, as listed in Table 1, are control knobs of the framework. The sensor control knobs include:

Table 1: Control Knobs of SoftBLE devices

Name	Description	Default
C_s^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^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

Algorithm 1: Channel assignment to gateways

```
input  : observation matrix (O_{M×N})
output : assigned gateways channels (C^𝒢)
// Constructing A^𝒢_{M×M}
for i ← 1 to M do
    for j ← 1 to M do
        A^𝒢_{i,j} ← ‖O⃗_i ∧ O⃗_j‖;
// Coloring of the graph with adjacency A^𝒢_{M×M}
Initialize SCS and C^𝒢 to {0}_M
                        ▷ SCS: Sum of the Covered Sensors;
for i ← 1 to M do
    candidGW ← arg max (SCS);
    Initialize CC to {0, 0, 0}          ▷ Channel Covered;
    for j ← 1 to M do
        if C^𝒢_j > 0 then
            CC_(C^𝒢_j − 36) +← A^𝒢_{candidGW,j}
    C^G_{candidGW} ← arg min(CC) + 36;
    for k ← 1 to M do
        if C^G_k = 0 then
            SCS_{candidGW} +← A^G_{candidGW,k}
        else
            A^G_{candidGW,k} = 0
```

5 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 [2] in graph theory and *Minimum Interference Frequency Assignment Problem (MI-FAP)* [1] in wireless communication. The variations of MI-FAP that deal with channel assignment in cellular networks, such as the one in [12], are the closest to our problem. The problem was initially formulated as Integer Linear Programming in [1] has been proven to be NP-Hard in [35]. Heuristic solutions, including greedy search [1], tree search [22], or branch and cut algorithm [10] 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 (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,

$$\begin{aligned} &\min \sum_{(u,v) \in E} w_{uv} \mathbb{I}(x_u = x_v) \\ &s.t. \quad x_u \in \{37, 38, 39\}, \forall u \in V, \end{aligned} \tag{2}$$

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.

6 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.

6.1 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,

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

The definitions of the parameters in above formulation are listed in Tables 1 and 2.

Two control knobs are involved in the process of minimization: advertising channel map ($C^{\mathcal{S}}_s$), 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.

Table 2: The parameters SoftBLE analytical model

Name	Description	Default
$\mathcal{G}$	Set of gateways	
$\mathcal{S}$	Set of sensors	
N	Number of sensors	$\lvert\mathcal{S}\rvert$
M	Number of gateways	$\lvert\mathcal{G}\rvert$
δ	Duty cycle	3s
T	Application defined PRR threshold	99.9%
R	Maximum re-transmissions	3
μ	BLE bit rate	1Mbps
$\lvert PDU\rvert$	Length of an advertisement's PDU in bits	16*8
$\lvert HEADER\rvert$	Length of advertising packet header	16*8
$\lvert DATA\rvert$	Bit length of *SCANNABLE_ADV_IND* packet	|HEADER| + |PDU|
$\lvert SR\rvert$	Bit length of *SCAN_REQ* packet	|HEADER|
$\lvert SS\rvert$	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}	Default TX power consumption of sensors	(V*8.9)mW
P_{rx}	Listening power consumption of sensors	(V*5.9)mW
P_{ifs}	Power consumption of intra-frame spacing	(V*3.8)mW

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 C and TX in f^s_{PRR} by the observation matrix (O), defined earlier in (1). Subsequently, the problem formulation in (3) is equivalent to the minimization of pwr_s on each sensor $s \in \mathcal{S}$ individually, as:

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

6.2 Estimating PRR and Power Consumption

A similar analytical approach that is used to evaluate LEMoNet in [14], can be utilized to estimate the PRR and the mean power consumption of sensors in SoftBLE as well. As listed in Table 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.
(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 ($\boldsymbol{E[PRR]_s}$). Since the number of gateways that are assigned to a sensor is at most 3 (one for each channel) in SoftBLE, the aggregated PRR of sensor s is calculated as:

$$\begin{aligned}
\hat{prr}_s =& c_s^{37} prr_s^{\{g_s^{37}\}} + c_s^{38} prr_s^{\{g_s^{38}\}} + c_s^{39} prr_s^{\{g_s^{39}\}} \\
& - c_s^{37} c_s^{38} prr_s^{\{g_s^{37}, g_s^{38}\}} - c_s^{37} c_s^{39} prr_s^{\{g_s^{37}, g_s^{39}\}} - c_s^{38} c_s^{39} prr_s^{\{g_s^{38}, g_s^{39}\}} \\
& + c_s^{37} c_s^{38} c_s^{39} prr_s^{\{g_s^{37}, g_s^{38}, g_s^{39}\}},
\end{aligned} \tag{5}$$

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 $i \in C_s^{\mathcal{S}}$, otherwise zero. Accordingly, the expected PRR for the packets of sensor s is given by:

$$E[PRR]_s = 1 - (1 - \hat{prr}_s)^R. \tag{6}$$

Per Gateway PRR ($\boldsymbol{prr_s^{\chi}}$). Since all the sensors are in the scannable advertising mode in SoftBLE, prr_s^{χ} in (5) relies on both data and SCAN_REQ receptions:

$$prr_s^{\chi} = \left(1 - col^{\chi}\right) \times \prod_{g \in \chi} \left(1 - \left(ber_s^g\right)^{(\lvert DATA\rvert)}\right)\left(1 - \left(ber_s^g\right)^{\lvert SR\rvert}\right), \tag{7}$$

where col^{χ} denotes the collision probability of the packets sent from sensor s at all the gateways $g \in \chi$, and ber_s^g denotes the Bit Error Rate (BER) of advertising packets from sensor s at gateway g. Apparently, if χ is empty, prr_i^{χ} will be zero.

col^{χ} is calculated using ALOHA based collision estimation as,

$$col^{\chi} = e^{\left(-2I^{\chi} \cdot \lambda \cdot \overline{ret}\right)}, \tag{8}$$

where I^{χ}, namely interference counter, is the number of potentially colliding sensors, and $\overline{ret}$ is the expected number of re-transmissions, both will be calculated later on. For ber_s^g, BER of BLE packets can be estimated with the GFSK BER model [18] as,

$$ber = \frac{N_h - 1}{N_h} P_{e(N_h-1)} + \frac{1}{N_h} P_{e1}, \tag{9}$$

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 [18].

Interference Counter ($\boldsymbol{I^{\chi}}$). 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 $\chi = \{g_1, g_2, g_3\}$, it can be counted using the principle of exclusion and inclusion as,

$$I^{\chi} = \sum_{i=1}^{3} \left\| \vec{O_{g_i}} \right\| - \sum_{i<j=1}^{3} \left\| \vec{O_{g_i}} \wedge \vec{O_{g_j}} \right\| + \left\| \vec{O_{g_1}} \wedge \vec{O_{g_2}} \wedge \vec{O_{g_3}} \right\|, \qquad (10)$$

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

Traffic Load ($\boldsymbol{\lambda}$). 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*, reception time of *SCAN_RSP*, and the decode gap, as:

$$\lambda = \frac{1}{\delta} \left(\frac{|DATA| + |SR| + |SS|}{\mu} + gap \right), \qquad (11)$$

Expected Number of Retransmissions ($\overline{\boldsymbol{ret}}$). 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:

$$\overline{ret} = \left[\sum_{i=1}^{R-1} \left(\sqrt[R]{T} \right)^{i-1} \left(1 - \sqrt[R]{T} \right) \right] + \left(\sqrt[R]{T} \right)^{R-1}, \qquad (12)$$

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

$$\begin{aligned} E[PWR]_s &= \sum_{r=1}^{R} \frac{1}{\delta} \cdot E_s^{adv} \cdot E[PRR]_s (1 - E[PRR]_s)^{(r-1)} \\ E_s^{adv} &= P_s^{tx} \cdot \left\| C_s^S \right\| \cdot \left(\frac{|DATA|}{\mu} + P_{ifs} \right) \\ &\quad + P_{rx} \cdot \left(\frac{|SR|}{\mu} \right) + P_{ifs} + P_s^{tx} \cdot \left(\frac{|SS|}{\mu} \right) \\ P_s^{tx} &= P_{tx} \cdot 10^{(TX_s/10)}, \end{aligned} \qquad (13)$$

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

6.3 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. 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 ($\vec{rss}_s$)

output : assigned advertising channel map (C_s^S) and TX power (TX_s^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 $C, TX, \vec{rss}_s, O$ using (5);
    Estimate $E[PWR_s]$ based on $E[PRR_s]$ using (13);
    if $E[PRR_s] > T$ ***and*** $E[PWR_s] <$ *bestPWR* **then**
      bestC $\leftarrow C$;
      bestP $\leftarrow PTX[p]$;
      bestPWR $\leftarrow E[PWR_s]$;
$C_s^S \leftarrow$ bestC;
$TX_s \leftarrow$ bestP;

7 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.

7.1 Baseline Approaches

We compare the performance of SoftBLE with a similar 2-tier BLE-based network called LeMoNet [15]. Compared to SoftBLE, LeMoNet uses fixed parameter settings and cannot be reconfigured centrally except for over-the-air reprogramming. Sensors can operate in two modes in LeMoNet:

- **Normal Connectionless (NCL) mode** where sensors advertise their data through *ADV_NONCONN_IND* packets.
- **Scannable Connectionless (SCL) mode** where sensors advertise *ADV_SCAN_IND* packets, and if no *SCAN_REQ* is received from any gateway (as acknowledgments), packets will be re-transmitted up to R re-tries. Sensors in the SCL mode have the similar behavior as sensors in SoftBLE in the forwarding plane.

7.2 Experimental Validation

48 BLE sensor nodes and two gateway devices have been deployed in a 11m-by-8m laboratory (Figure 6). Both sensors and gateways are equipped with TI cc2640r2 MCU chips for BLE communication. The gateway devices are implemented on 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 2.

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 7a shows the results of sensor orchestration and the average RSS

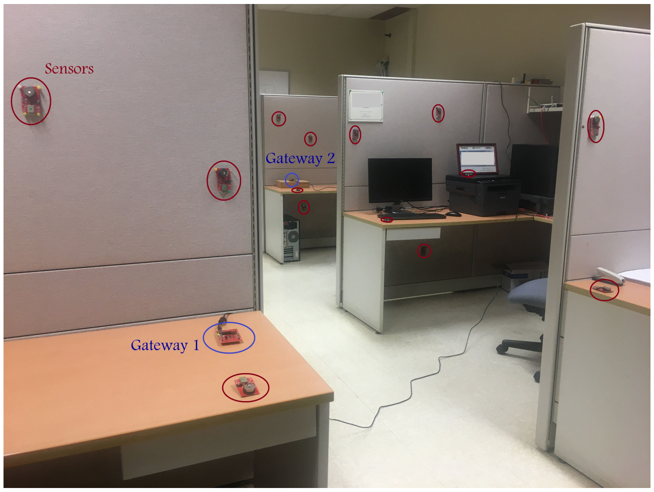


Figure 6: A scene of the experiment setup in our lab.

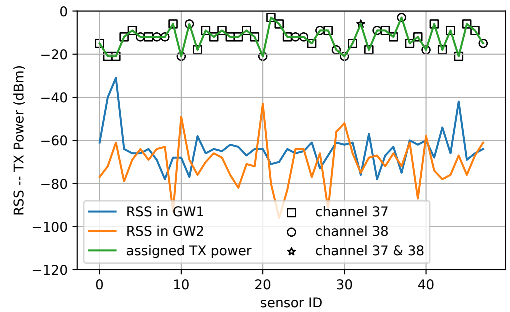
values at the two gateways from each sensor. Gateway 1 and 2 are assigned Channel 37 and Channel 38, respectively. The majority of sensors are assigned to a single channel associated the gateway with a higher RSS except for sensor 32, which has low RSS to both gateways. To ensure a high PRR, it advertises 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 7b shows PRRs under different schemes. Both SoftBLE and LEMoNet-SCL can achieve high PRRs due to the use of SCAN_REQ messages as acknowledgment and possible re-transmissions. But 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 6.

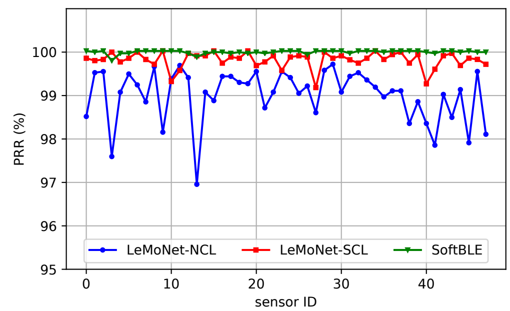
We observe from Figure 7c, LEMoNet-NCL sensors roughly spend a constant amount time in advertising in each duty cycle. This is due to the predefined length of legacy advertising events. Among SCL mode sensors, the advertisement duration varies because of the additional time to receive SCAN_REQ. Furthermore, 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 7d.

7.3 Simulation Study

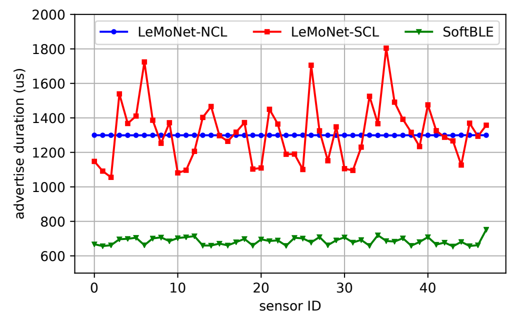
To study the scalability of the proposed SDN framework and the effects of different parameters, we have implemented BLE and SoftBLE in OMNET++ [30], an event-driven network simulator.



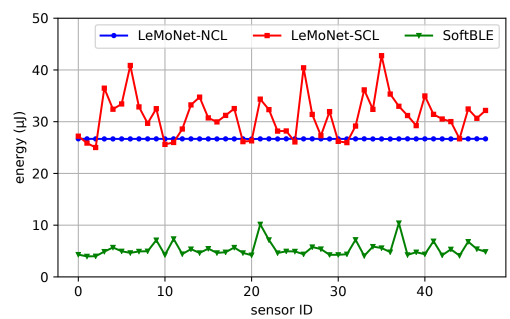
(a) RSS at gateways, sensor channel maps and TX power levels



(b) Packet Reception Rate



(c) Advertise duration per duty cycle



(d) Mean energy consumption for TX per duty cycle

Figure 7: 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.

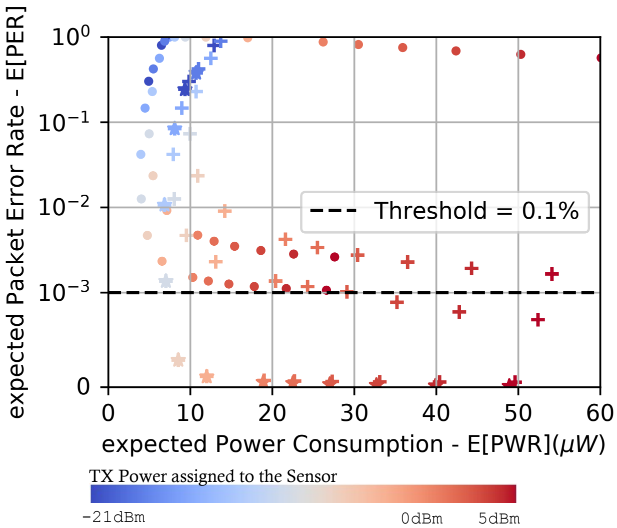


Figure 8: The effect of TX power and channel assignment of a randomly selected sensor node. All other sensors' parameters remain fixed. Results include channel assignments of a single channel (•), two channels (+), and three channels (⋆) at different TX power levels.

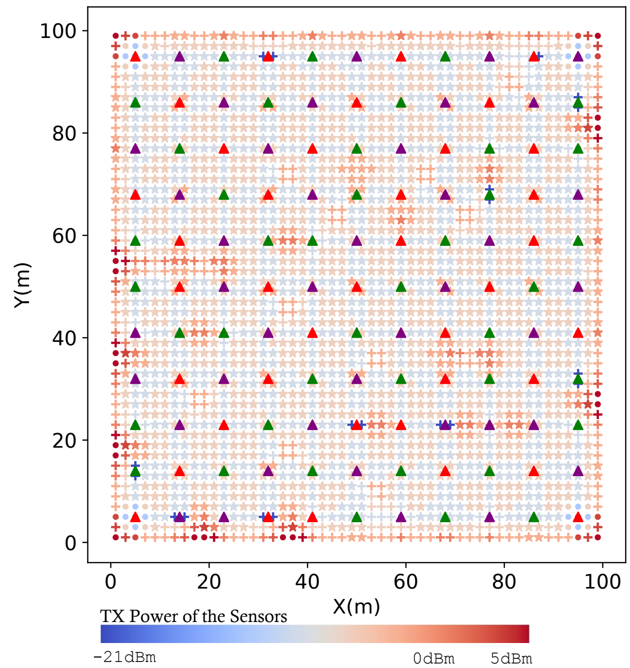


Figure 9: The output of running orchestration algorithms on the scaled scenario including 2500 sensors advertising on 5s duty cycles, and 121 gateways. The sensors are symbolized based on whether they assigned to one channel (•), two channels (+), or three channels (⋆). Gateways are shown with △, colored based on their scanning channel (red: 37, purple:38, green:39).

7.3.1 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.

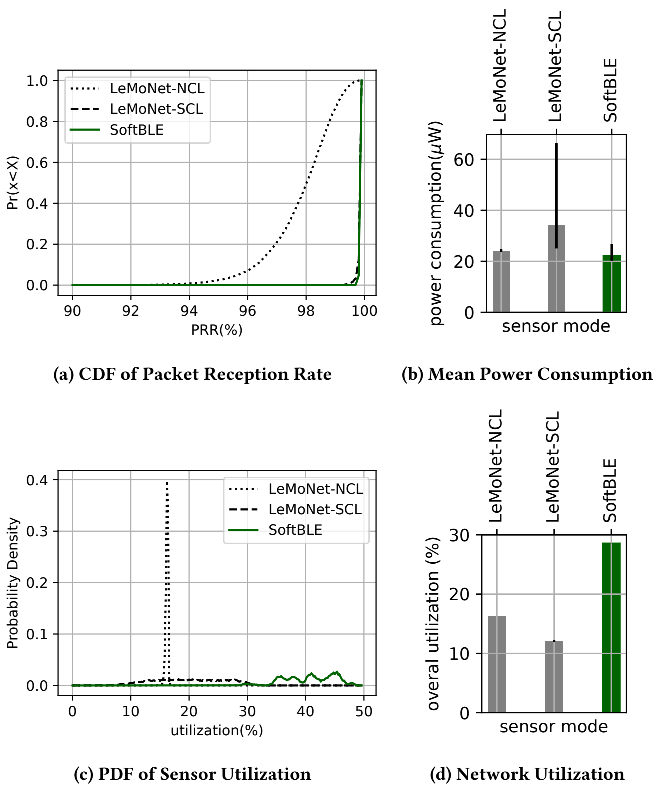
The remaining fixed parameters can be found in Table 2. The results of each scenario are the averages of 5 runs, each lasting for 10,000s and with a different random seed. In addition to PRR and the power consumption of sensors, we also evaluate sensor utilization defined as,

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

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

7.3.2 Performance at Scale. Figure 8 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 9 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 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 10 compares the performance of SoftBLE and LEMoNet in the large-scale network. It is seen from Figure 10a 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 use of acknowledgment and re-transmissions. However, Figure 10b 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 10c, all sensors in



(a) CDF of Packet Reception Rate **(b) Mean Power Consumption**

(c) PDF of Sensor Utilization **(d) Network Utilization**

Figure 10: Performance of SoftBLE in a scaled network with 2500 nodes.

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 10d.

7.3.3 Impacts of Network Size and Duty Cycle. Figures 11a – 11c 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 12a – 12c show the effects of sensor duty cycle. Reducing duty cycles (or 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 12c, 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.

8 RELATED WORK

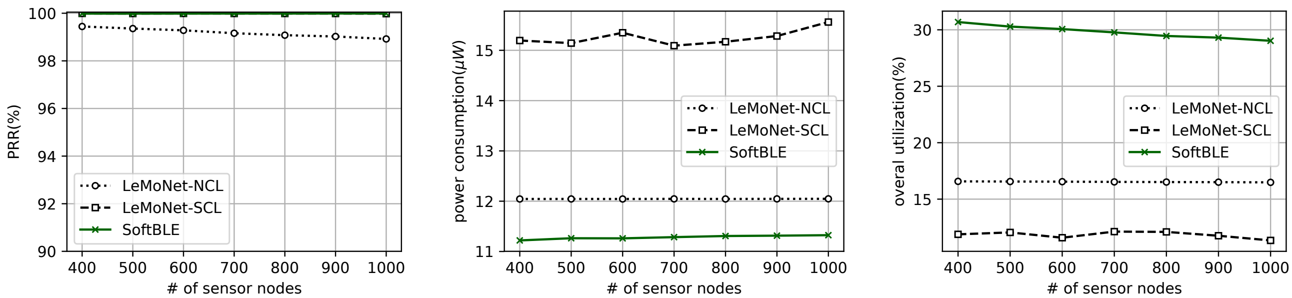
This section reviews related works in two categories. The first category includes the recent SDN developments in Internet of Things (IoT) edge networks, and the second category consists of the studies on BLE parameter tuning.

8.1 SDN in IoT Edge

In the last decade, SDN controllers have become popular in IoT edge to enhance the efficiency of data transmission protocols[5]. Zheng *et, al.* [33] 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.* [16] utilize path difference degree (PDD) in an SDN-based IIoT network to find the optimum flow path with respect to time delay and goodput. In [31], 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.

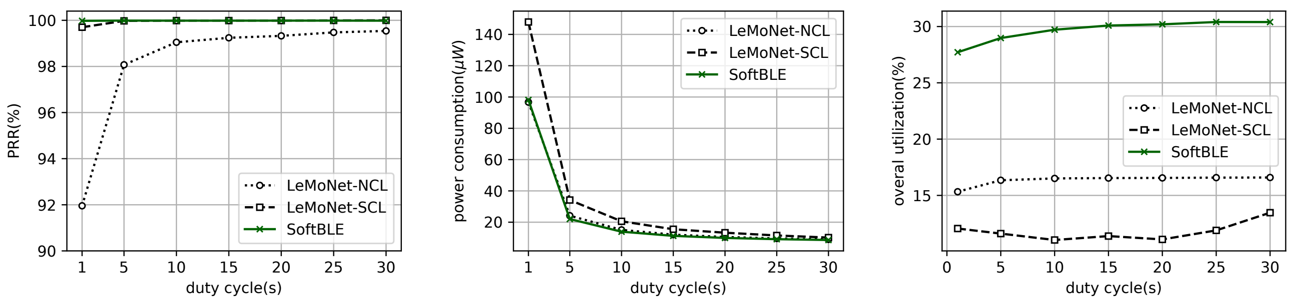
Although Zigbee and LoRaWAN do not explicitly incorporate SDN, they utilize coordinators and network servers as central controllers. A Zigbee Coordinator is responsible for bootstrapping its network by selecting a Personal Area Network (PAN) identifier, and an operating channel for 802.15.4 based devices. Taking advantage of such capabilities, several work introduces SDN improvements for 802.15.4 based networks, including Atomic-SDN [3], SDN-based topology management [4], or WISE-SDN [11]. 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 [24], 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 *et al.* [29] 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



(a) Packet Reception Rate (b) Mean Power Consumption (c) Utilization

Figure 11: 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 12: Effects of duty cycle on the performance of SoftBLE. 600 sensors are deployed randomly in a $50m \times 50m$ area.

impose significant delay and extra power consumption to the network. A SDN framework is also proposed in [23] 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, SoftBLE utilizes legacy advertising to avoid connection overhead and can be implemented on most BLE commercial-off-the-shelf devices.

8.2 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 [19] and the neighbor discovery latency [21] 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.* [28] 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 [6], advertising interval [20], or both [17]. Comparing to these works that deal only with interval settings, 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.

9 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 reduce the average power consumption of sensor nodes while meeting the application-defined performance requirements. Our proposed framework enables long lifetime IIoT network deployments for large-scale and dense monitoring applications.

ACKNOWLEDGMENT

This research was in part supported by the collaborative research and development grant *NSERC CRDPJ 500720* and the Discovery grant from the Natural Science and Engineering Research Council of Canada.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] Mehdi Jafarizadeh and Rong Zheng. An analytical study of low energy monitoring networks for large-scale data centers. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

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

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

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

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