Computer Networks 209 (2022) 108925

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Analysis of mobility support approaches for edge-based IoT systems using high data rate Bluetooth Low Energy 5

Risto Katila, Tuan Nguyen Gia *, Tomi Westerlund

Department of Technology, University of Turku, 20500, Turku, Finland

ARTICLE INFO

Keywords:
Internet-of-Things
Bluetooth-Low-Energy (BLE)
Mobility support
Handover
RSSI
Low latency
Real-time
Energy efficiency

ABSTRACT

In remote monitoring edge-based IoT applications, high latency caused by the mobility of a sensor device can cause serious consequences such as inaccurate analysis and low quality of services. Therefore, it is required to have mobility support approaches that help reduce latency while maintaining a connection, high quality of service, and energy efficiency. However, the number of mobility support approaches for high data rate IoT applications using Bluetooth Low Energy (BLE) is limited and they have some disadvantages. For example. they have not been designed for edge-based applications where local computation occurs frequently. Many of them have not been implemented and tested in daily working environments with actual mobility cases. They have not comprehensively analyzed the mobility latency and energy consumption of sensor devices. Hence, this paper presents three possible mobility support approaches including passive and active handover mechanisms for edge-based IoT applications using high data rate BLE5. These approaches based on passive and active handover mechanisms are implemented and tested in an office environment. The results of latency and power consumption of a sensor device via many experiments are measured and analyzed. The results show that the presented mobility support approaches maintain the connection during mobility with a latency of around 900ms for many cases. The results also show that using BLE5's LE 2M physical layer consumes less power than using LE 1M physical layer. Specifically, it can reduce energy consumption when sending or receiving larger data sizes at faster rates.

1. Introduction

Internet of Things (IoT) can be described as a platform where humans and objects from different disciplines can be interconnected and communicate with each other [1]. IoT has been applied in many applications, such as smart homes and smart factories. A conventional IoT-based system often consists of three main layers including sensor layer, gateway layer, cloud and terminal application layer. However, a three-layer IoT system architecture still has some limitations, such as high latency and energy inefficiency. Edge computing can be described as processing at the edge of a network closer to a location where data is collected, can be a suitable candidate for overcoming the limitations. Edge computing brings the cloud paradigm to the edge of a network. It also supports advanced features which are not proffered by cloud computing [2]. Particularly, edge computing provides distributed data storage and computational resources including low-latency data processing and analysis. Depending on the application, an edge-based IoT system can have one or two more edge layers [3] that are located in between a sensor layer and a cloud layer.

In edge-based IoT systems, sensor devices of a sensor layer can be equipped with sensors and a wireless module for collecting and transmitting data wirelessly to a gateway, respectively. The wireless module can support one of the protocols such as 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks), Wi-Fi, Bluetooth Classic, ZigBee, Bluetooth Low Energy (BLE), and LoRa depending on the application [4–6]. For instance, Wi-Fi is often used for high data rate applications but Wi-Fi consumes high power (e.g., approximately 360–500 mW) for data transmission. BLE 4 is often used for many energy-efficient applications that focuses on low data rate (e.g., less than 250 kbps) [7]. Recently, BLE 5 has been introduced to improve BLE 4 with extra features that support high data rate IoT applications. BLE 5 allows different Bluetooth PHY configurations including 1M PHY, 2M PHY, and Coded PHY that have tradeoffs between throughput and range. Particularly, LE 2M feature allows the highest data rate (around 1.3Mbps in practice) among all the supported features. The IoT systems using LE 2M can be suitable for energy-efficient and high data rate health monitoring applications such as the multi-channel

---

* Corresponding author.
E-mail addresses: rijukat@utu.fi (R. Katila), tunggi@utu.fi (T.N. Gia), tovewe@utu.fi (T. Westerlund).

https://doi.org/10.1016/j.comnet.2022.108925
Received 19 July 2021; Received in revised form 22 February 2022; Accepted 15 March 2022

electrocardiogram (ECG) and electromyography (EMG) monitoring that can send 24-bit 1000 samples per second data [8].

Edge gateways, which are interconnected and communicate with each other, are equipped with hardware resources to offer edge services. Each gateway covers a specific area, for example, having a 20–30 m radius. The achieved range can significantly differ depending on the technology, wireless conditions, and environment. For example, when the communication path between a sensor device and a gateway is not light-of-sight (e.g., having objects in the communication path), the covered area is often set smaller to ensure a good connection. In remote monitoring applications such as smart buildings and smart hospitals, the movement of a sensor device is unavoidable. In general, each device is connected to a specific BLE-based gateway. When a device moves out of the working range of the gateway, it needs to reconnect to another gateway to maintain the connection. Switching connection from one gateway to another gateway can cause long latency that is unacceptable in real-time applications such as remote health monitoring. To minimize the mobility latency, a mobility handover necessitates designing an intelligent algorithm. However, it is challenging to design advanced mobility handover approaches for high data rate IoT applications using BLE. For example, it is required that the mobility support approaches must not negatively impact the working time of the BLE-based sensor device while minimizing the mobility latency.

Currently, the number of mobility support approaches for IoT applications using BLE-based gateways is limited [9,10] and these approaches still have many limitations. For example, they were built only for low data rate applications, for example, less than 250 kbps. Most of the mobility support IoT systems using BLE gateways were not implemented with actual hardware such as sensors and gateways. Accordingly, the actual latency and energy consumption of sensor devices could not be properly captured, analyzed, and discussed. Therefore, this paper presents three possible mobility support approaches for the edge-based IoT system using high data rates BLE5. The mobility support approaches are implemented and tested with a complete edge-based IoT system in a working office environment. Many mobility cases are applied for evaluating the system latency and power consumption of a sensor device when mobility occurs. The results of the experiments are compared and comprehensively analyzed. Parameters affecting the handover latency are also tested and reported.

The structure of the paper is as follows: Section 2 discusses Bluetooth and BLE mobility support approaches. Section 3 provides the background of Bluetooth Low Energy and impact factors on mobility support. Section 4 explains the possible mobility support approaches and architectures for edge-based IoT applications. Section 5 represents the setup and implementation of the presented mobility support approaches. Section 6 compares and analyzes the results of mobility support. Section 7 concludes the work.

2. Related work

According to [11], mobility in IoT applications is complex and can be categorized into two main types including physical-based mobility and architecture-based mobility. Physical-based mobility consists of movement type and element, whereas architectural-based mobility contains entity handler and mobility protocol. Movement type includes pre-order mobility, controlled mobility, and random mobility, while movement element comprises node mobility and sink node mobility. In many applications such as smart homes and smart healthcare, random mobility and node mobility mainly occur. Particularly, random mobility means that sensor devices are moving freely without knowing in advance mobility parameters such as destination locations and movement duration. Node mobility means that only sensor devices are moving while gateways are fixed at some specific locations. These imply that the system architecture will remain without any changes when node mobility and random mobility occur. Hence, this paper focuses on node and random mobility.

Many researchers have proposed mobility approaches for IoT systems. However, most of the works focus on Wi-Fi, 6LoWPAN, and IEEE 802.11 [11–13]. In contrast, there has been a limited number of BLE-based mobility support approaches. Although this paper focuses on the BLE-based mobility support approaches, an overview of mobility approaches for both Bluetooth Classic and BLE are still discussed.

In [14], the authors proposed a mobility support approach for Bluetooth-IP based systems. The suggested approach has three procedures including a preliminary procedure, a handover procedure, and an update procedure. In the preliminary and handover procedure, the mobile node periodically inquires information (Bluetooth address and received signal strength indicator—RSSI) from gateways. The mobile node relies on the RSSI values to decide the particular gateway to which the node will send data. Particularly, when an RSSI value is less than a pre-defined threshold, the handover occurs. Then, the node informs the current gateway that it will move to the next gateway. Then, the current gateway will inform the next gateway and wait for the acknowledgment. If this acknowledgment is successfully received, the node will start handover to a new gateway. The result for the simulated approach shows that the connection and handover time is 2.388 s and 15.4 ms, respectively.

Three handover approaches were presented in [15]. In the first approach, while maintaining the connection with a gateway, a Bluetooth device collects gateway addresses periodically. These addresses are recorded in the order of the fastest response. When an RSSI value between a Bluetooth device and its connected gateway is less than a pre-defined level, the gateway sends a message to the device to ask for increasing the transmission power. In the second approach, a gateway collects RSSI values regularly at a specific interval. Then, the procedure is similar to the first approach. When the RSSI value is smaller than a pre-defined value, similar operations are conducted. The third approach uses a backup link to ensure connectivity. If an RSSI value between a device and a gateway is smaller than a threshold value, the device chooses a backup link having a higher RSSI value. After that, the device periodically inquires to find a new gateway for a backup link and connects to any available gateway.

In [9], the authors presented a mobility approach for BLE-capable IoT devices. Two types of methods including full and partial BLE stack cloning are utilized to transfer the pairing and bonding information to the alternative gateways. The handover decision-making is based on RSSI values from scanned advertising packets. When the RSSI value is smaller than the pre-defined threshold value, the handover procedure is started. At this moment, the currently connected gateway will decide to choose a candidate gateway that a sensor device will be connected to. Then connection and its information are transferred from the current gateway to the candidate gateway. The proposed approach has been experimented with and evaluated via a testbed consisting of smartphones and tablets that are used as gateways and sensing devices communicating via BLE. The results show that the proposed approach reduces the communication overhead and latency. Particularly, the time required for connection migration is from 400 to 1600 ms depending on the configuration such as partial stack cloning, trusted full-stack cloning, or untrusted full-stack cloning. However, the proposed approach still has some limitations when sending data with a 20 ms interval. For instance, it has a high number of lost packets (about 40 to 90 packets depending on the configuration mode).

In [10], the authors proposed the BLE-based architectures for handover support for mobile BLE devices. One architecture targets IPv6 over BLE while another architecture focuses on BLE without IP connection. Two approaches including passive and active handover are proposed. In a passive handover approach, a BLE connection is terminated when the supervising timeout has passed. A BLE device starts to establish a new connection via using BLE advertising. In the active handover approach, BLE gateways scan for RSSI values periodically.

The collected RSSI values are sent to a controller that has a mapping table used for monitoring the connection state between a BLE device and a gateway. The handover mechanism starts when a current connection between the BLE device and its connected gateway is poor and the connection with a new gateway is good enough. Connection quality is assessed via RSSI values that are compared with a pre-defined threshold. The proposed approaches and architectures are implemented and evaluated via a setup of the nRF52840 development kit and Raspberry Pi 3. The results show that the proposed approaches help switch from a poor connection to a good connection when the alternative gateway is available.

Although these approaches help maintain the connection between a BLE device and a system during mobility, they only focus on low data rate (e.g., less than 250 kbps) IoT applications. In addition, the approaches have not been comprehensively analyzed in terms of the energy consumption of a BLE sensing device and latency. Many of the approaches have not been implemented and evaluated with actual mobility cases and actual systems including BLE sensing devices and BLE-based gateways. Therefore, this paper targets to overcome the limitations via the actual implementation of the entire Edge-based IoT system supporting BLE 5 mobility for high data rate (e.g., around 1 Mbps) applications. Many practical test cases have been carried out to evaluate the mobility support approaches in terms of latency and energy consumption of BLE sensing devices.

3. Background and impact factor on mobility support

BLE functioning in the Industrial Scientific and Medical frequency band has a spectrum that ranges from 2400.2 MHz to 2483.5 MHz [10]. The frequency band is divided into 40 channels from channel 0 to channel 39, in which each channel has a 2 MHz bandwidth. The arrangement of channels helps avoid interference from other devices that operate in the same spectrum, such as Wi-Fi and Bluetooth classic devices. The allocated primary advertising channels are the last three channels (i.e., channel index 37 at 2402 MHz, channel index 38 at 2426 MHz, and channel index 39 at 2480 MHz). A device can advertise on one, two, or three of these channels, and it is possible to modify the device to advertise only on selected channels. An advertising event is shown in Fig. 1 when all three primary channels are used in advertising [16]. In each advertising packet, an advertisement data is less than or equal to 31 bytes, and the amount of time needed for sending the packet is at most 10 ms. The channel indexes from 0 to 36 are data channels exchanging data packets. In the extended advertising mode introduced in BLE 5, it is possible to use one or more data channels as a secondary advertising channel to send more advertising packets. Extended advertising on secondary channels can use one of available LE PHY, such as LE 1M, LE2M, or Coded S = 8 or S = 2 PHY. The data field *AdvData in BLE 5 common extended advertising payload is allowed to be up to 254 bytes long* [17]. Extended advertising also allows advertising data to be chained by fragmenting the data and sending parts of the data on different secondary channels (1650 bytes max). In chained advertising, the advertising packets have a header field containing the information about the next channel containing the next packet on the chain. In Fig. 2, the primary advertising channels point to the offloaded advertising data on secondary channels in extended advertising [17].

BLE supports collocation and coexistence with adaptive frequency hopping (AFH) that mitigates unilateral and mutual interference by using a pseudo-random convention of hopping channels. Frequency hopping is a technique in which once a link is formed, BLE devices synchronize and change to a channel together for each connection event with a high speed (e.g., many times per second). Channels that are chosen for connection events, are free from interference and stored in a channel map. Generic access profile (GAP) is used as a protocol framework for defining how BLE devices interact with each other. GAP encompasses connection and advertising parameters. GAP specifies four roles that BLE devices can use to join BLE piconet including broadcaster, observer, central (master), and peripheral (slave). Particularly, a broadcaster is only used for sending advertisement packets. Broadcaster cannot participate in any connections or receive packets from other devices. An observer is only used for listening to incoming advertisement packets and the observer cannot initiate connections. Central (master) can discover devices, listen to advertising packets, and initiate multiple connections. Peripheral (slave) can send advertisement packets that can be discovered by central devices and form connections with central devices. Each BLE 5 device can operate with multiple roles concurrently [18].

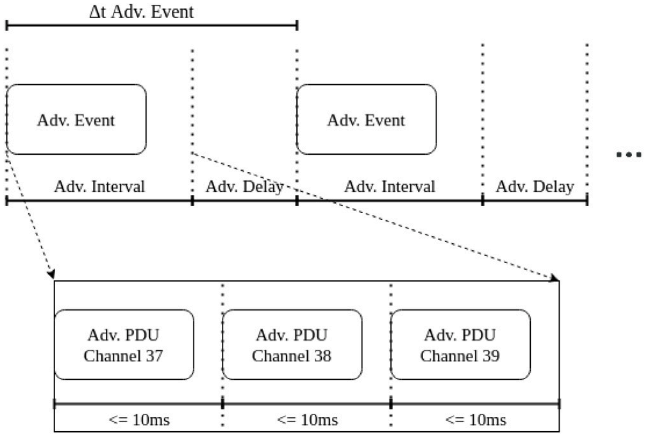


Fig. 1. A diagram showing an advertising event when advertising on all three channels 37, 38 and 39.

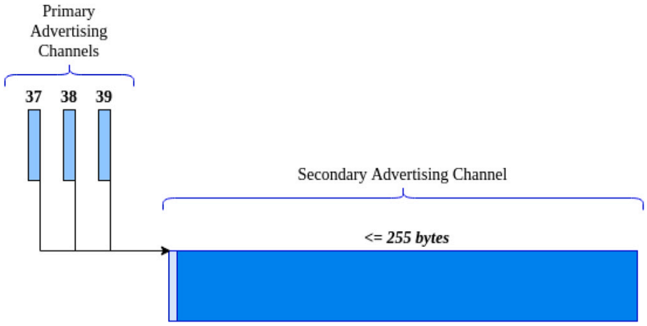


Fig. 2. BLE 5 extended advertising.

The BLE link layer defines different states which enable the BLE device to operate in various roles. A device operating in the advertising state is called an advertiser. An advertiser sends advertisement packets periodically while a device operating in the scanning state named a scanner listens to incoming advertising packets from advertisers. An advertisement packet consists of an advertising protocol data unit (PDU) that can have the following functionalities:

- Connectable versus non-connectable: determining whether a device can be connected or not
- Scannable versus non-scannable: determining whether an advertising device can handle scan requests or not. Scan requests allow advertising more data that can be fit to an advertising packet
- Directed versus undirected advertising: defining establishing a connection with a known specific peer device in case of directed advertising or with any peer device in case of undirected advertising
- Extended advertising: allowing sending more advertising data in the secondary advertising channels
- Periodic advertising: allowing non-connectable advertisements to be transmitted at a predefined and fixed interval

An advertiser has a defined advertising interval that can be set from 20 ms to 10 485.759375 s with 0.625 ms steps according to BLE 5 specification [17]. The advertising interval affects the sending frequency of advertisement packets on each primary advertising channel. The time between advertising events depends on advertising interval and a pseudo-random advertising delay generated by the link layer that can range from 0 to 10 ms. Lower advertising interval causes increased energy consumption but decreases the discovery latency [19,20]. Based on simulated results in [21] and experimented results in [22], discovery time for 20 ms advertising interval is almost unaffected by the interference caused by other BLE devices in case of less than 40 BLE devices advertising at the same time.

In BLE 5 specification, the scan interval and window for the scanner specified to be from 10 ms to 40.96 s, and it is adjustable with 0.625 ms steps. Scanning interval is the time between repeated subsequent scans on advertisement channels, while scan window defines the amount of time used for scanning the advertising channels. Scan window can be equal to or less than scan interval. There are two modes of scanning, including passive scanning and active scanning. In active scanning, the scanner requests more information with a scan request packet. In passive scanning device only listens for advertisement packets and cannot request additional information with a scanning request packet.

A BLE connection can be only initiated by a master device, which can establish connection to many slave devices. The limitation on the number of connected slave devices depends on the hardware specification. In contrast, Bluetooth classic (Basic rate/Enhanced data rate) BR/EDR only allows seven connected slave devices [17]. Maintaining a connection between devices requires regular connection events for exchanging synchronization information and data. Connection interval can be adjusted from 7.5 ms to 4 s with 1.25 ms increments, and it determines the time between connection events. There is a compromise between connection latency and power efficiency. A longer connection interval can help achieve less power consumption but increases the connection latency and vice versa.

The connection interval determines the frequency of sending consecutive connection packets. If there is no data to send, devices still exchange packets having zero-size payload to keep the connection alive. Slave latency determines the number of consecutive connection events that the peripheral device can skip before the connection is lost. The supervision timeout is separately monitored by each connected device. The connection supervision timeout is the maximum time allowed between two connection events before the connection is regarded as lost and the device is disconnected. The supervision timeout according to BLE 5 standard can be from 100 ms to 32 s [17].

The received signal strength indicator (RSSI) is often used in many mobility support approaches, including Bluetooth Classic and BLE approaches. Capturing of RSSI values is nearly always supported, whereas the link quality indicator and other wireless-related parameters are often not available. The RSSI expresses the actual power of signals in receiving antenna in dBm units. In an ideal case, the RSSI value would depend on free-space path loss, and there would be a direct line of sight between receiving and transmitting antennas. However, in real environments, there might be many obstacles in between the communication path and interference from the environment, such as multipath propagation, signal drift, white noise, and signal absorption [23]. Therefore, RSSI values need to be filtered to remove noise that causes the fluctuation. There are several ways of filtering and processing the RSSI values. One of them, known as the Kalman filter, has been widely used because it achieves correctly filtered results with fewer errors. Therefore, all the presented approaches in this paper use Kalman filters [24].

Indoor gateway and outdoor gateways in IoT systems usually have different RSSI values for the same device having the same distance. Particularly, an indoor environment may have more obstacles (e.g., furniture and walls) but a stable environment temperature (e.g., 22 degrees Celsius for room temperature) whereas outdoor environments may have fewer obstacles but their temperatures vary significantly depending on the time of the day [25]. These impact the results of RSSI values. Therefore, bias values need to be considered when dealing with RSSI values.

In order to support mobility, it is required that two adjacent gateways of an IoT system need to have an overlapped area (see Fig. 3). The overlapped area's size depends on several parameters such as the gateways' specifications, system architecture, and surrounding environments. When the overlapped area is larger, the possibility to miss a mobility case may reduce. Therefore, in an environment having a lot of noise and interference, it is recommended to have a larger overlapped area if possible.

3.1. *Mobility in edge-based IoT systems*

Mobility in edge-based IoT systems can be categorized into two main groups, including architectural-based mobility and physical-based mobility. Architectural-based mobility consists of network-based mobility and mobility protocol (e.g., mobility at a network layer or MAC layer). Physical-based mobility means the actual movement of nodes such as sensor devices or sink nodes (i.e., an edge-based gateway). Physical-based mobility can be divided into movement type and movement element. The movement type represents how a node moves. Particularly, a node can move randomly without any pre-orders or move in an ordered and controlled way. When the random movement is supported by a handover algorithm, then the ordered and controlled movement is also covered. The movement element consists of sensor device mobility and sink node (access-point) mobility. In practice, sensor device mobility and random movement often occur, while other types are very seldom or even do not occur in many applications. Therefore, this paper focuses on sensor device mobility and random mobility.

3.2. *Industrial standards for mobility support in edge-based systems*

European Telecommunications Standards Institute (ETSI) has made several industrial standards for mobility support in mobile edge-based systems [26–29]. Most of the standards for mobile edge mobility focus on the mobility of applications and services. The following metrics need to be considered when dealing with mobility latency (e.g., round-trip time, one-way delay, setup time, service update time, and context-update time); energy efficiency; network throughput; system resource footprint; and quality of service. Particularly, the mobile edge-based system needs to support continuity of the service, mobility of application, application state recollection, and mobility of application-specific user-related information. The connectivity and services need to be maintained when mobility occurs. However, the industrial standards do not specify the specific values for the requirements and metrics. Therefore, the requirements and metrics depend on the specific application.

4. Mobility support and architecture

This section presents three potential approaches for facilitating mobility support in edge-based IoT systems. Three handover approaches are presented, including one passive handover and two active handover approaches. In this architecture, a gateway always takes the central role. Therefore from here on in, central and gateway terms are used synonymously. These approaches apply the same scanning interval and window of 10 ms in gateways. Peripherals use a 20 ms advertising interval with connectable, scannable, and undirected advertising to initiate connections. These values are chosen to ensure a minimum discovery latency.

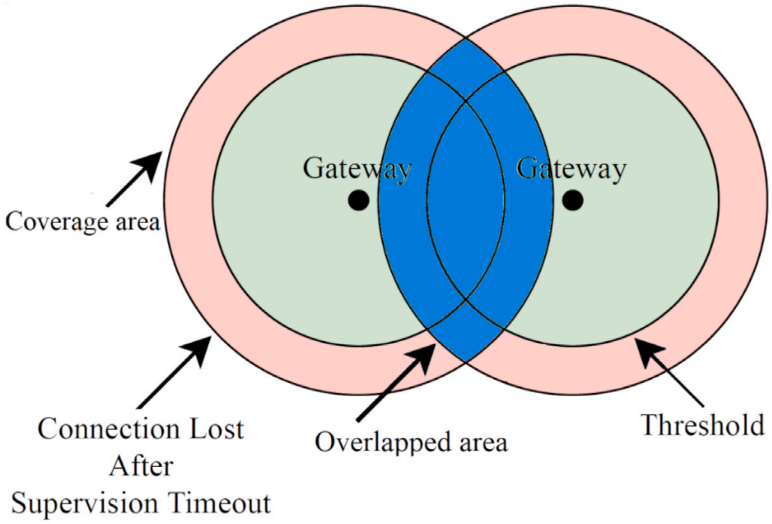


Fig. 3. Overlapping piconets with different areas defined.

4.1. Handover approaches

Three mobility handover approaches named Case 1, Case 2, and Case 3 are presented in this section. In addition, algorithms 1, 2 and 3 which are developed for carrying out these approaches are also included. Two presented approaches including Case 1 and Case 3 are only advertising with undirected connectable scannable advertising packets. Case 2 uses undirected non-connectable advertising when the peripheral device has a connection.

The first approach (Case 1) is a passive handover that is the most elementary approach out of the three approaches. The different states of the passive handover approach are shown in Fig. 4. In passive handover, gateways will attempt to connect to a peripheral device advertising with a correct universally unique identifier that corresponds to the desired Bluetooth service.

In the passive handover approach, when a supervision timeout (that can be either timeout of a gateway or a peripheral device) expires, the handover event happens. The connection between a gateway (named as a source gateway) and a peripheral device is then closed, and the peripheral device begins advertising with a 20 ms advertising interval to establish a connection with another gateway. When a valid packet is received by the device or the gateway, the particular supervision timeout is reset. It is noted that a peripheral and a gateway have their separate supervision timeout and they work based on their timeout. Supervision timeout and slave latency should be adjusted so that they follow the rule shown in the following formula: $Supervision\_timeout > (1 + slave\_latency) * connection\_interval * 2$. After disconnecting with the source gateway, the device can connect to any gateway as long as the gateway first sends connection indication packets to establish a connection. This happens because there is no control mechanism in this approach.

In the second approach (Case 2) shown in Fig. 5, adjacent gateways exchange information such as RSSI values and their connection status to those devices. Gateways make the handover decision independently based on the RSSI values and connection information. RSSI threshold is defined for the system where the handover is initiated and this value has to be carefully chosen according to the environment and gateway placement. During handover, the adjacent gateway that has the highest measured RSSI will become the new central gateway that connects with the device after the device disconnects with the source gateway. In this approach, the peripheral is advertising continuously and changes between two advertising modes: connectable scannable undirected advertising with a 20 ms advertising interval and non-connectable non-scannable undirected advertising with a 100 ms advertising interval.

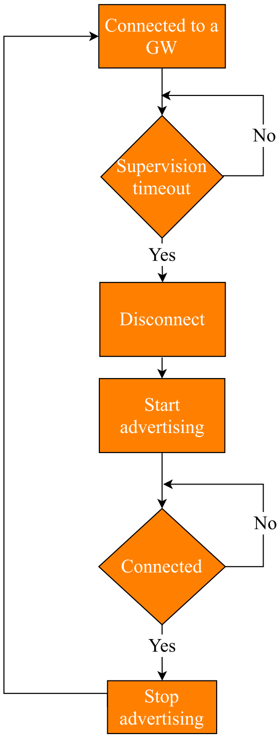


Fig. 4. Case 1 passive handover advertising states.

The peripheral uses connectable scannable undirected advertising (e.g., with 20 ms advertising interval) when it has no connection to a gateway. When the peripheral connects to a gateway, it switches to a non-connectable non-scannable undirected advertising with an interval of 100 ms. The purpose of slower advertising intervals during connection is to reduce energy consumption, avoid overlapping between connection events, and keep advertising so that RSSI values can be updated from advertising packets.

In the third approach (Case 3), advertising is stopped after the peripheral device has been connected and the measured RSSI values have been higher than the RSSI threshold for over 20 s. There is also a

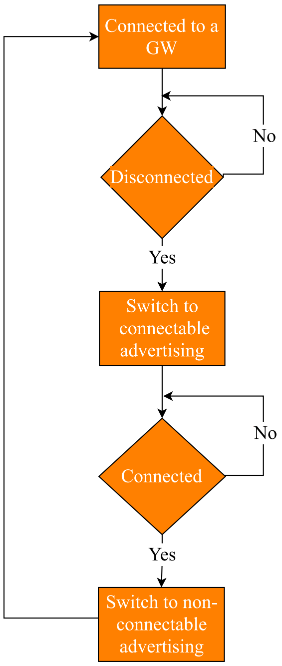


Fig. 5. Case 2 active handover advertising states.

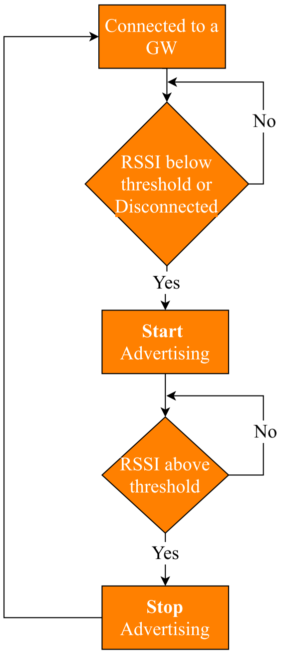


Fig. 6. Case 3 active handover advertising states.

Algorithm 1 for Case 1

```
while true do
    if Peripheral connected to a GW then
        if Supervision timeout then
            Disconnect from the GW
            Start Advertising
        else if Advertising then
            Stop Advertising
        else
            Continue
        end if
    else
        if not Advertising then
            Start Advertising
        else
            Continue
        end if
    end if
end while
```

Algorithm 2 for Case 2

```
while true do
    if Peripheral connected to a GW and Connectable Advertising then
        Switch to Non-Connectable Advertising with a longer advertising interval
    else if Peripheral not connected to a GW and Non-Connectable Advertising then
        Switch to Connectable Advertising
    else
        Continue
    end if
end while
```

Algorithm 3 for Case 3

```
while true do
    if Peripheral connected to a GW then
        if $RSSI < Threshold_{RSSI}$ and not Advertising then
            Start Advertising
        else if $RSSI > Threshold_{RSSI}$ and Advertising then
            Stop Advertising
        else
            Continue
        end if
    else
        if not Advertising then
            Start Advertising
        else
            Continue
        end if
        Continue
    end if
end while
```

short delay for activating the advertising where a few values below the RSSI threshold are also needed. Starting and stopping the advertising are carried out only after the measured RSSI values are stable. This is necessary to prevent quick successions of starting and stopping the advertising. Different states for the advertising for Case 3 are shown in Fig. 6.

The RSSI values measured during the connection are read from the connection data packets. This means that the peripheral device has to inform the gateway immediately via a notification when the values are below the RSSI threshold. When the gateway receives the notification of the RSSI value, it will disconnect from the peripheral device and inform other gateways about the disconnection via MQTT.

The RSSI threshold, in this case, must be set specifically based on the RSSI if gateways have a different type of BLE radio than the peripherals. As mentioned, if the gateways are the same types such as all indoor gateways or all outdoor gateways, bias values for RSSI can be ignored. If the gateway types include indoor and outdoor gateways, bias values for RSSI need to be considered. Advertising interval is 20 ms in this

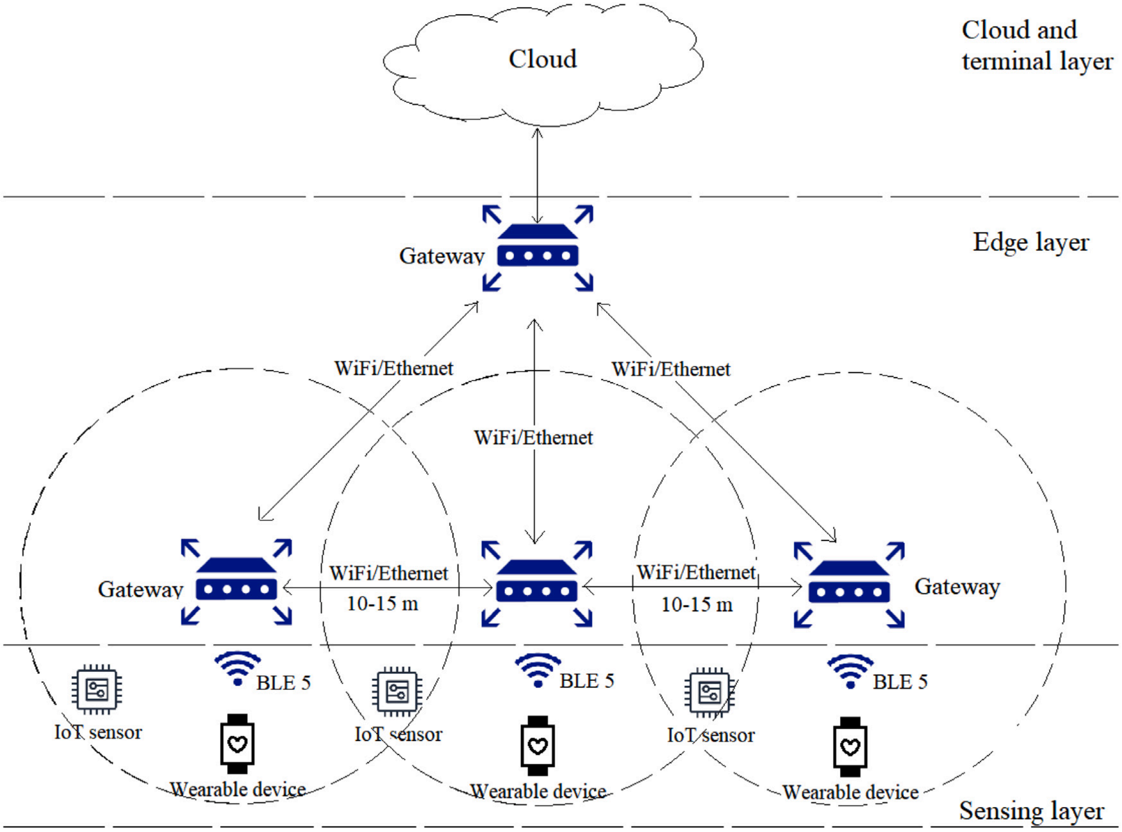


Fig. 7. The edge-based IoT system architecture.

approach. Setting the advertising interval to a higher value, in this case, does not significantly impact energy consumption because the device always tries to connect to a gateway when it is not in a connection.

4.2. System architecture

The edge-based IoT system architecture shown in Fig. 7 includes the sensing layer, edge layer, cloud and terminal layer. The sensing layer consists of IoT sensor devices such as wearable sensors that acquire high data rate signals (e.g., 500–1000 sample/s ECG data), and transmit the collected data to a gateway via BLE 5 with LE 2M feature. The distance between an IoT device and an edge gateway can be from a few meters to 10 m. When the IoT devices move out of the support range of a gateway, a mobility event occurs.

In the edge layer, smart edge-based gateways in the same virtual network (VLAN) are interconnected and communicate between themselves. Gateways are placed so that they have overlapping areas, and the distance between two gateways can be 10–15 m depending on the situation. To achieve good signal coverage in environments with various obstacles, the use of different distances is necessary, e.g., one gateway can be placed in an office room and another one in a corridor. The edge gateways can offer edge services such as distributed data storage, data processing, data compression, and other advanced edge services.

Cloud and terminal layer consist of cloud servers and services such as global big data storage and big data analysis. End-user terminals can be a web browser or a mobile app which can be used to access real-time data and receive real-time push notifications from servers. The detailed information of the architecture can be retrieved from our previous papers [3,30,31].

5. Setup and implementation

Experiments were conducted in the office rooms of the University of Turku. These rooms have different devices using different wireless technologies such as Wi-Fi, Bluetooth Classic, and BLE.

In experiments, the transmit power for the devices is set to 0 dBm because RSSI values below the threshold are reached easier, and the required distance to trigger the handover in the passive handover approach is shorter. The gateway placement for the passive handover is shown in Fig. 8, while room setup for active handovers (Case 2 and 3) is shown in Fig. 9. The distance between gateways is 6 m for active handover approaches and 15 m for passive handover approach. The supervision timeout used in case 1 is 100 ms which is the minimum allowed by the standard, and it will show as an additional delay in the handover latency.

The peripheral device utilized in setup is a development board from Nordic Semiconductor's nRF52833 DK that is programmed with the Zephyr RTOS capable of supporting the BLE 5 stack. Gateways are Raspberry Pi 4 single-board computers running Ubuntu Server 20.04 LTS operating system. The official Linux Bluetooth protocol stack BlueZ version 5.58 was installed in Ubuntu. Each gateway is equipped with an additional BLE Nordic Semiconductor's nRF52840 dongle to support LE 2M PHY feature because the Pi 4 Bluetooth module does not support the optional Bluetooth 5 features. Gateways in the system are in the same VLAN and communicate with each other via a lightweight messaging protocol MQTT (Message Queuing Telemetry Transport) [32].

The supply voltage in nRF52833 DK is regulated to 3.0 V by an on-board regulator when the (interface, MCU) USB port (J2) is used to power the board [33]. Power consumption measurements were conducted with Nordic Semiconductor's Power Profiler Kit II (PPK2). PPK2 has a 100 kHz sampling frequency and a readout on average value has an accuracy of ±10%, offset of ±2%, and resolution of 5 μA for measurement range from 500 μA to 5 mA. For measurement range

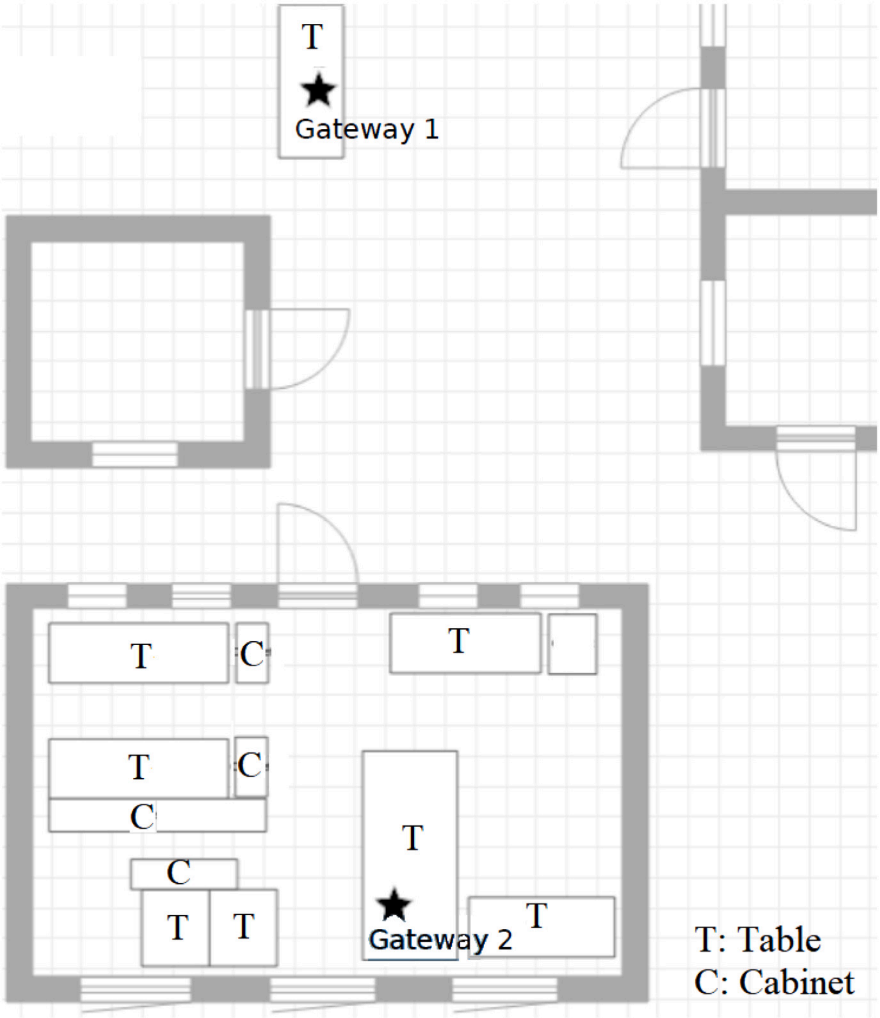


Fig. 8. Passive handover approach gateway placement.

5 mA to 50 mA, the resolution is 50 μA [34]. Reported current and power measurement values are averaged from 1-minute measurements.

Handover latency is determined by measuring the time difference in nRF52833 DK when the device is disconnected from the gateway and reconnects to a new gateway after generic attribute profile services and characteristics are discovered. For all the approaches, 50 samples of handover latency values are measured.

6. Results and discussions

RSSI values of a sensor device (the Nordic nRF52833 development board) in several positions towards a BLE5-based gateway are measured. In each position, RSSI values are measured 20k times. The abnormal values (e.g., error measurements) are filtered and the mean value of these filtered measurements is calculated. All the experiments are carried out in the same working environment. Transmitting power including 0 dBm and 8 dBm are applied for the experiments as they are some of the highest transmitting powers supported by the development board and these values help achieve a good connection. Results of these measurements shown in Fig. 10 indicate that the RSSI value decreases when the distance between the device and the BLE5-based gateway increases. However, the decreasing and increasing rates are not similar and linear. From the experiments, with every 2 m added, the RSSI value reduces about 4–6 dBm.

At a position of 8 m far from a gateway, an RSSI value is around −63 dBm (in the case of a sensor device with 8 dBm transmitting power). This position can be considered as a border threshold for the proposed mobility approaches. This selection can improve the capability of providing an appropriate connection between the gateway and a sensor device in the gateway's coverage area.

In this section, the handover latency and the energy consumption results of different mobility support approaches are presented. Some of the results relate to different parameters such as transmit power level, advertising interval, sending scan response packets, sending data packets with BLE5's LE 1M and 2M. It is noted that these parameters may apply to one or more approaches due to the relevancy for data rate, device discovery, initiating, and establishing connections. Adjusting the connection interval is not deeply studied because it is assumed that the target application would target the lowest latency for the data transfer. In the experiments, power consumption for sending the data packets with a 100 ms connection interval is 1.614 mW while power consumption for sending the data packets with a 7.5 ms connection interval is 2.823 mW. The handover latency for different approaches is presented in Table 1. The results show that the minimum handover latency for the passive handover is much less than the minimum handover latency for the active handover approaches. Particularly, the minimum handover latency for the passive handover approach is equal to one-half and one-fourth of the latency in other active approaches. One of the reasons causing the differences between these cases is

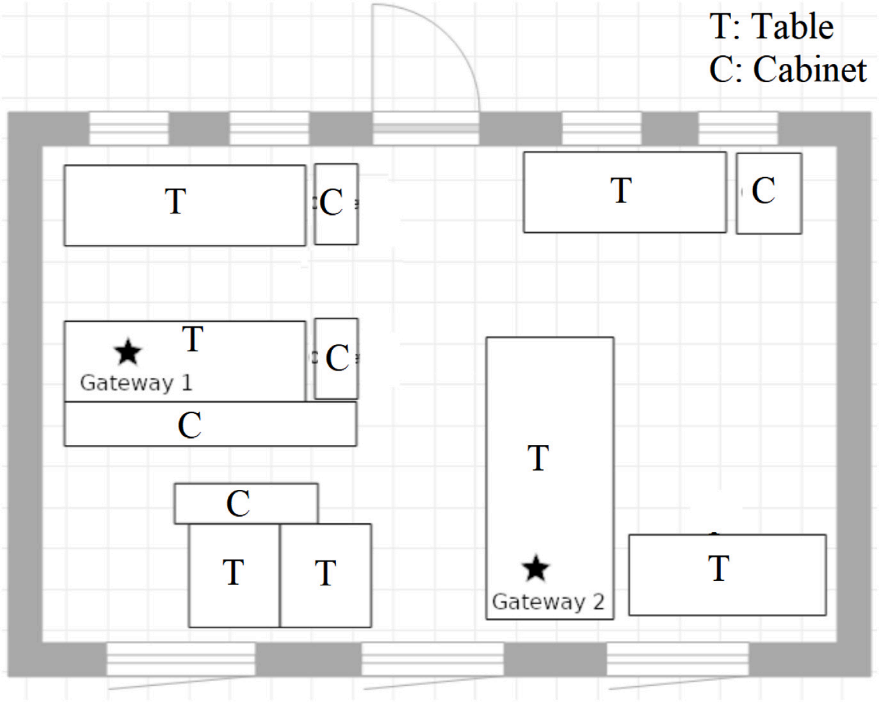


Fig. 9. Gateway placement used in test setup for active handover approaches.

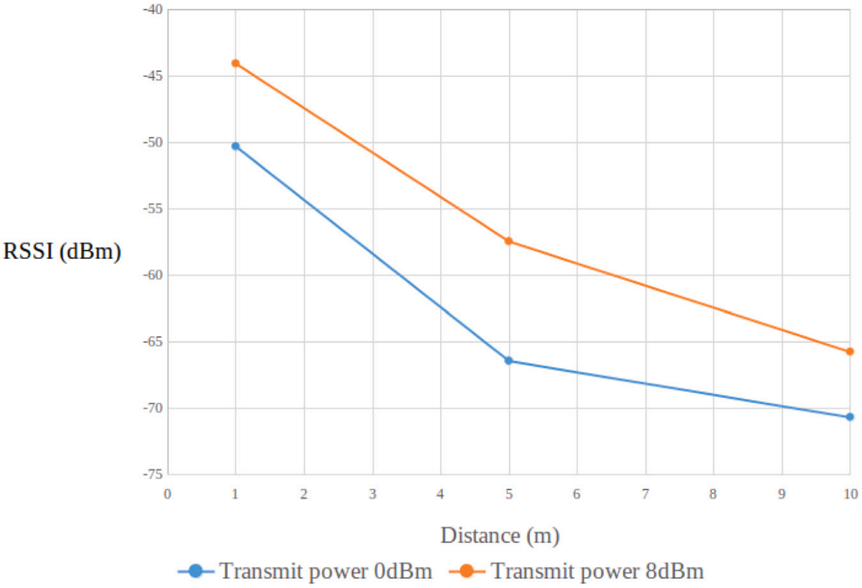


Fig. 10. RSSI values of a sensor device at different distances from a BLE5-based gateway.

that Case 2 and Case 3 have an additional delay from processing the RSSI values and communicating via MQTT. However, active handover helps reduce the error rate of mobility and provides a foundation for developing advanced handover mechanisms. These results show that the handover mobility support approaches for edge-based systems can be suitable for remote monitoring applications that do not need very strict latency requirements. However, the maximum latency of these approaches is around 2 s that exceeds the latency requirement of time-critical applications where the time requirement is often less than 500 ms [35]. Therefore, it is recommended that more advanced edge-based approaches for mobility support should be applied. The approaches should consist of smart algorithms for controlling new central gateway and deal with abnormalities such as bad cases or harsh environments. In the experiments, each approach is tested with 50 mobility trials from a volunteer having a normal working speed (e.g., around 3 m/s). The result shown in Table 1 shows that the success rate of three mobility support approaches is 100%. Although the average latency of the three presented mobility support approaches is not low, the success rate is high.

In the experiments, handover latency results are not stable for the cases. In some instances, latencies are high and significantly different from others. It is suspected that there might be some problems related to the connection between nRF52840 dongle USB and the Linux DBus in gateways that implement host controller interface communicating with the controller part of the BLE stack.

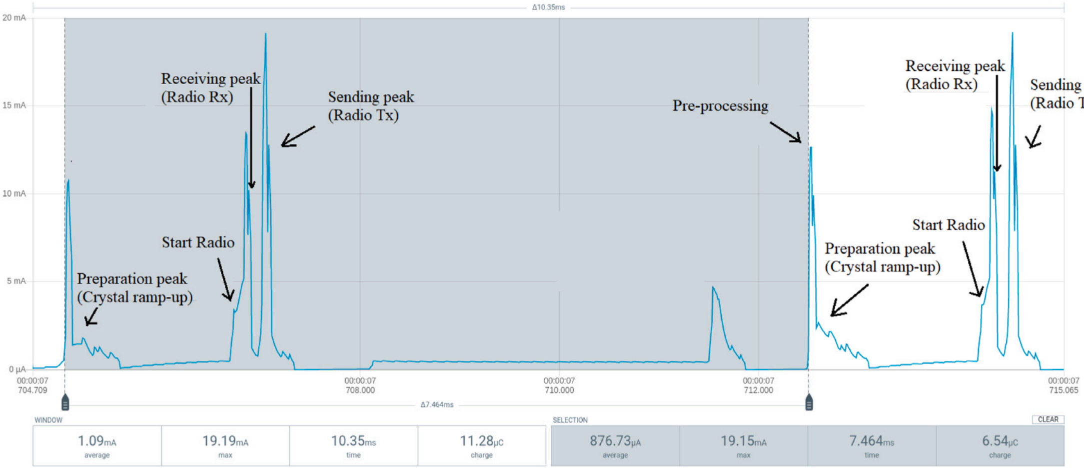


Fig. 11. Power consumption of the nRF52833 board in the active mode without sending any user data monitored by PPK monitor.

Table 1
Comparison of three mobility support cases.

	Case 1	Case 2	Case 3
MIN latency	156 ms	342 ms	710 ms
MAX latency	2275 ms	2032 ms	2664 ms
AVG latency	920 ms	1247 ms	1414 ms
Mobility successful rate (with 50 trials)	100%	100%	100%

The passive handover (Case 1) seems to have the fastest handover latency. However, the passive handover does not incorporate any control mechanism. There is potential data loss even before the supervision timeout happens because it might not trigger even if several packets are missed. Therefore, it is hard to achieve an accurate measurement of the handover latency for this approach.

Case 3 has the longest handover latency of handover approaches because advertising is done only for the duration of the handover event and then stopped after the connection is established. The power consumption is similar for Case 1 and 3 because the advertising is activated only during handover events whereas in Case 2, advertising is not stopped during a connection. In Case 2, frequent MQTT messages are sent between the gateways because of the continually updated RSSI values that cause the unnecessary use of resources in gateways compared to Case 3. In Case 3, the peripheral monitors the if connection RSSI value is under the threshold, then the peripheral advertises to adjacent gateways that measure and update RSSI values for the peripheral.

In our test cases in the office, the three mobility support approaches successfully maintained a connection between a sensor device (nRF52833 DK) and the system's gateways (Raspberry Pi 4) during mobility. However, these approaches have not been evaluated in harsh environments and with a large number of experiments. In the future, these approaches will be applied together with smart algorithms in both normal and harsh environments.

The power consumption of the nRF52833 development board in an active mode without sending user data is measured with the PPK2 and the result is shown in Fig. 11. The active mode without sending user data means that the connection between a peripheral and a gateway is correctly maintained while the device does not send any sensing data to the gateway. However, to maintain a connection, the peripheral (the nRF52833 development board) in active mode needs to send a packet with a 0 Byte attribute protocol (ATT) payload every 7.5 ms. The interval of 7.5 ms can be counted from the 1st start radio peak to the 2nd start radio peak or from the 1st sending peak to the 2nd sending peak in Fig. 11. Sending a packet requires at least three consecutive contiguous peaks, including (e.g., the preparation peak for crystal ramp-up, start radio peak, and the sending peak (radio Tx)). When the receiving (radio Rx) is enabled, there would include four peaks including preparation peak for crystal ramp-up, start radio, receiving peak, and sending peak to send a packet. These peaks do not locate right after each other. Between receiving peak and sending peak, there is a radio switch latency which is often around 140 μs. The preparation peak for crystal ramp-up and starting radio is often around 5 mA and 12.5 mA, respectively. In this case, the sending peak is around 13 mA. In Fig. 11, the highest peak, which is not the actual sending, might be caused by the radio switching from receiving to sending state and errors in the measurements.

The power consumption of the nRF52833 development board when sending user data in every 7.5 ms with a LE 2M feature is shown in Fig. 12. Similar to the case of active mode without sending user data, it has also four peaks including preparation peak for crystal ramp-up, start radio, receiving peak, and sending peak. The preparation peak, in this case, is almost the same as the preparation peak in the case of active mode without sending user data discussed above. However, the sending peak, in this case, is not just a single peak but represents many contiguous peaks that turn into a rectangle shape. The duration of the peak values of the rectangle is around 1 ms that is the latency needed for sending 244 Bytes ATT payload in a packet. The miscellaneous peaks are around 4–5 mA and they stand for some of the activities such as post-processing or data buffering.

Different payload sizes including 52 Bytes, 100 Bytes, 196 Bytes, and 244 Bytes are applied in the experiments with LE 2M feature, and the results are shown in Fig. 13. These cases are similar in terms of the number of peaks and amplitude of the peaks.

Several payload sizes including 52 Bytes, 100 Bytes, and 244 Bytes are applied for the experiments with LE 1M feature. The results are shown in Figs. 14 and 15. The cases of sending 52 Bytes, 100 Bytes, and 244 Bytes ATT payload are similar to the cases of sending similar payloads with the LE 2M feature in terms of the required number peaks and amplitude of the peaks. The only difference is that the sending latency of the 244 Bytes ATT payload case lasts 2.1 ms per packet.

BLE stack needs to run multiple tasks that use the radio such as packet sending/receiving, listening to incoming packets on a channel. Depending on the specific BLE stack, the priority can be different. In the experiments, only a single task can be executed at a moment. Therefore, each task needs to be scheduled. It is noted that other BLE stacks might have different priorities and task execution. Based on the experiments, rescheduling and blocking will happen more frequently when either a faster connection or advertising interval is used and especially in the case of having multiple simultaneous connections.

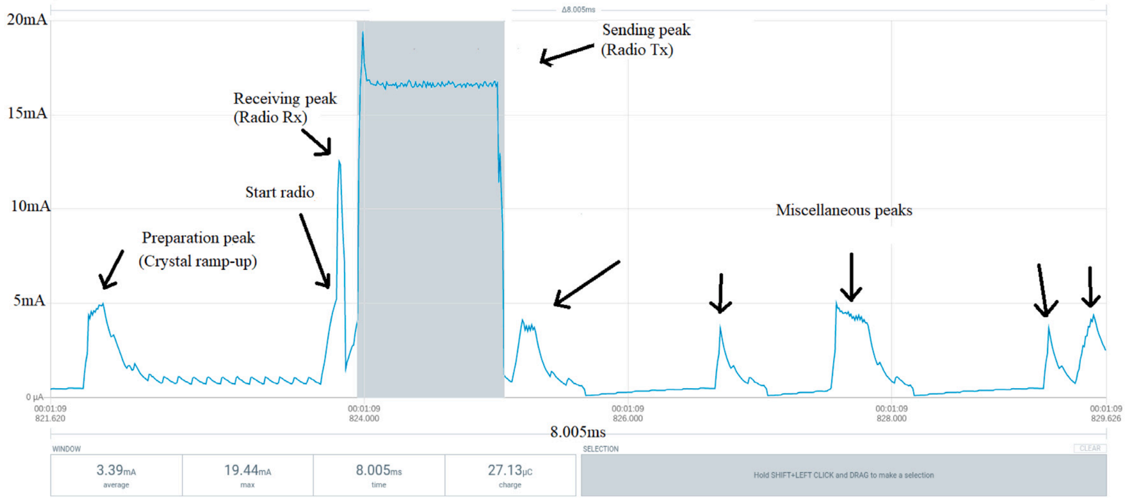


Fig. 12. Power consumption of the nRF52833 board in the active mode when sending 244 Bytes ATT payload user data with LE2M feature monitored by PPK monitor.

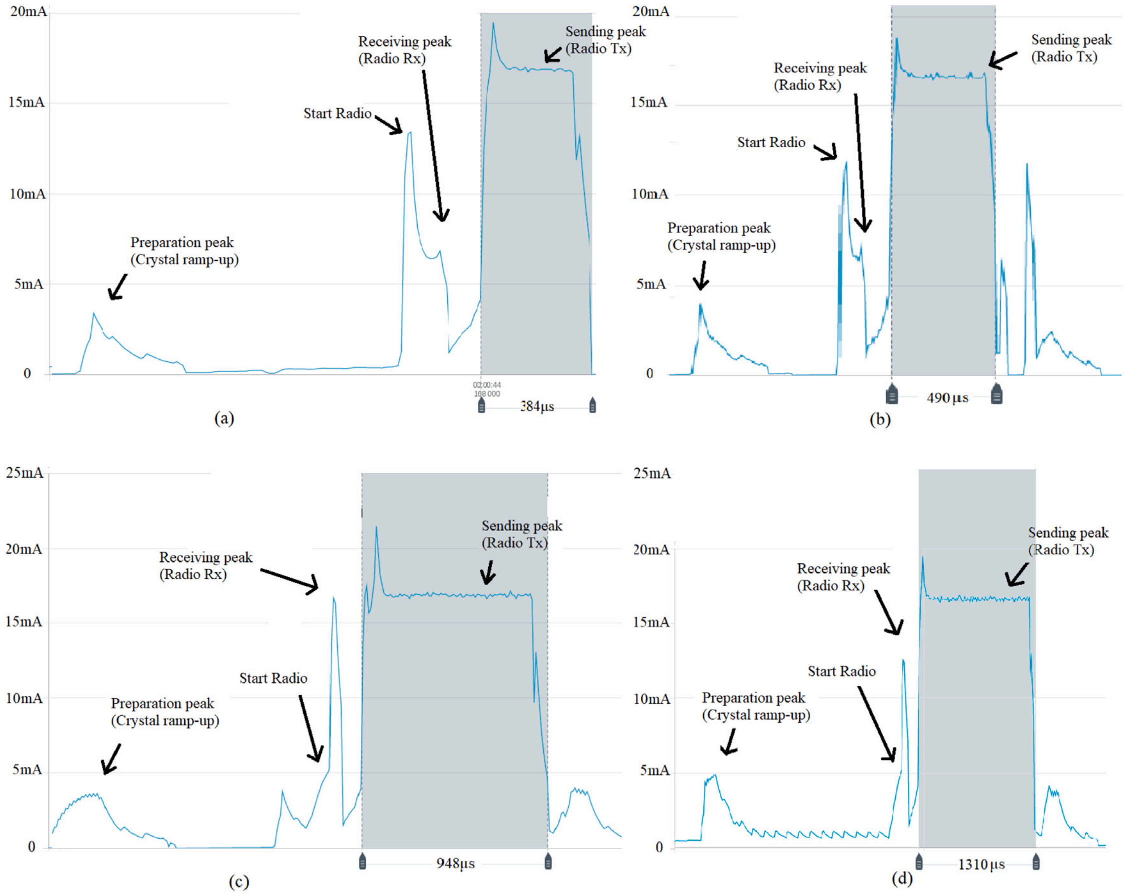


Fig. 13. Power consumption of the nRF52833 board when sending different data sizes with LE 2M feature monitored by PPK monitor. (a) 52 Bytes ATT payload per packet (b) 100 Bytes ATT payload per packet (c) 196 Bytes ATT payload per packet (d) 244 Bytes ATT payload per packet.

Advertising and connection events once in a while overlap in Case 2. Advertising and connection event colliding can be seen in Fig. 16, where a connection event is supposed to take place after the second marker but is disregarded, and an advertising event takes its place. After the advertising event, data packages are sent in which the first one is from a previously missed connection event. Colliding lower priority tasks are discarded if they cannot be scheduled later. Some tasks get elevated priority if they have been disregarded multiple times.

Measurement results for power consumption for adverting with different parameters are shown in Table 2. Particularly, connectable and scannable undirected advertising is used in all of the explored approaches (i.e., Case 1, 2, and 3) with the following parameters: 0 dBm transmit power, 20 ms advertising interval and a user payload data length of 25 bytes.

It can be seen in Table 2 that advertising interval impacts power consumption. However, depending on application requirements for the device discovery, a longer advertising interval might not be possible.

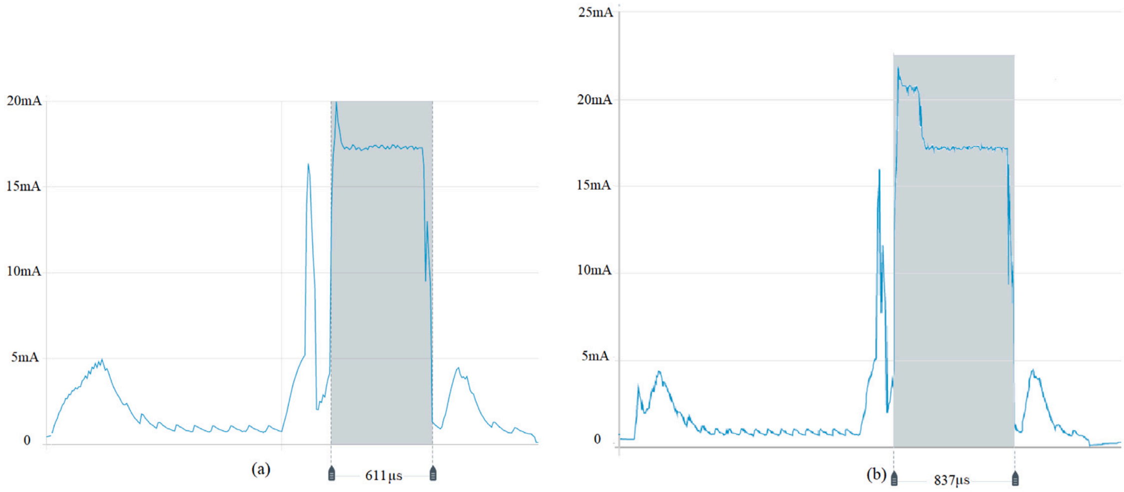


Fig. 14. Power consumption of the nRF52833 board in the active mode when sending 52 Bytes and 100 Bytes ATT payload user data with LE1M feature monitored by PPK monitor. (a) 52 Bytes ATT payload per packet (b) 100 Bytes ATT payload per packet.

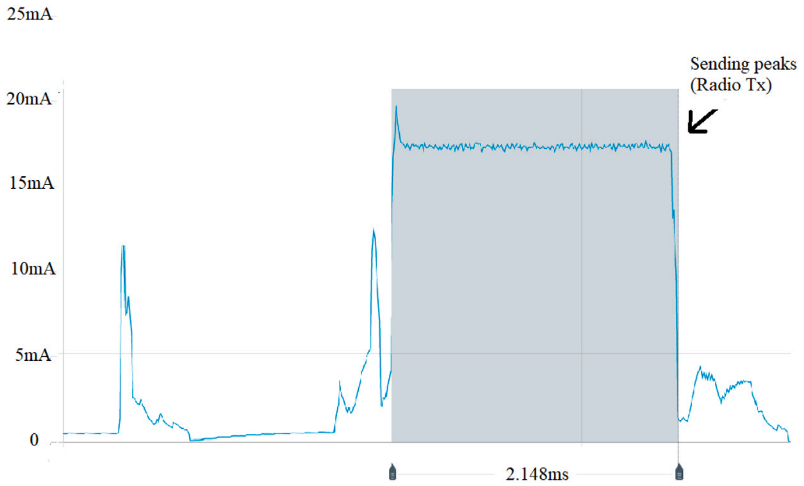


Fig. 15. Power consumption of the nRF52833 board in the active mode when sending 244 Bytes ATT payload user data with LE1M feature monitored by PPK monitor.

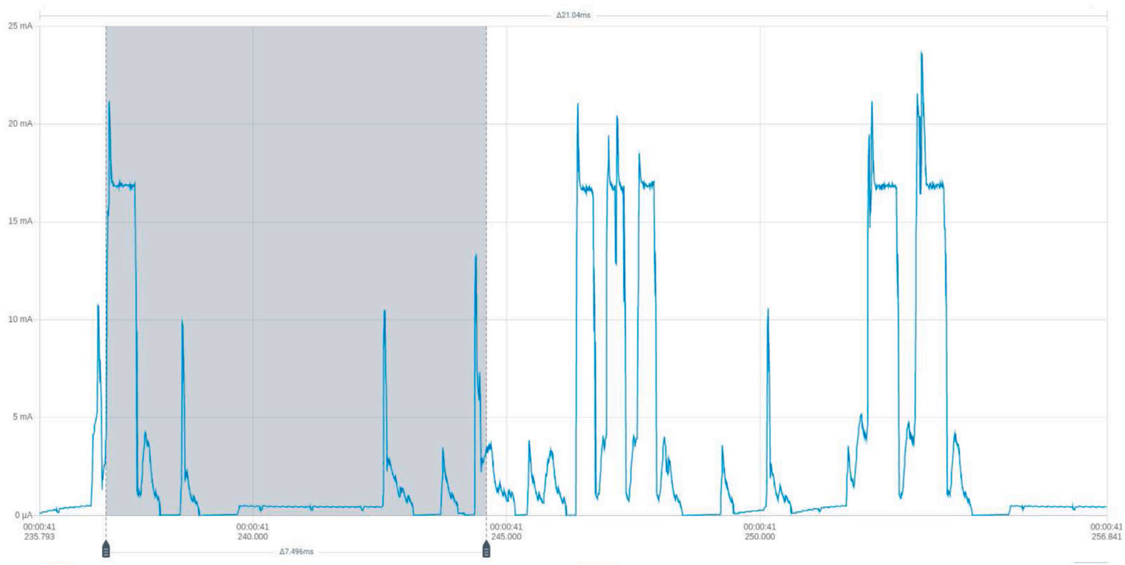


Fig. 16. Advertising overlapping with a connection event.

Table 2
Power measurements.

State	Transmit power (dBm)	Advertising Interval (ms)	Current (µA)	Power (mW)
Connectable scannable undirected advertising	0	150	695	2.08
Connectable scannable undirected advertising	0	60	783	2.35
Connectable scannable undirected advertising	0	20	1040	3.12
Connectable scannable undirected advertising	4	20	1220	3.66
Connectable scannable undirected advertising	8	20	1460	4.38
Connectable scannable undirected advertising using BLE 5 extended advertising	0	150	496	1.49
Connectable scannable undirected advertising using BLE 5 extended advertising	0	60	547	1.64
Connectable scannable undirected advertising using BLE 5 extended advertising	0	20	722	2.17
Connectable scannable undirected advertising using BLE 5 extended advertising	8	20	1010	3.03
Connectable scannable undirected advertising /Scan request and response	8	20	1620	4.86
Non-connectable non-scannable Undirected advertising/Connection	0	100	817	2.45
Connection **and not** advertising	0	N/A	796	2.39
Connectable scannable undirected advertising/Connection	0	20	1350	4.05

Adjusting the transmit power impacts the total power consumption in all data transmissions, where higher transmit power increases power consumption. However, adjusting the transmit power is restricted by the range requirements. Using the BLE 5 extended advertising feature with LE 2M PHY consumes less energy compared to advertising only in primary channels with the same advertising parameters. Energy is saved due to the shorter time required to send the advertising data in secondary advertising channels with LE 2M PHY. However, sending small advertising packets using extended advertising has no measurable difference. For example, sending advertising packets with a 9-byte length user payload and 60 ms advertising interval, the average current is 539 µA for using only primary advertising channels and 540 µA for extended advertising. Similarly with the same advertising package size, when using a 30 ms advertising interval to send the advertising packets, the average current is 630 µA for only advertising on primary advertising channels and 632 µA for extended advertising. In [36], authors experimented with simulated models of BLE 4 advertising and BLE 5 extended advertising. The analyzed results showed that BLE 5 advertising has lower energy consumption and latency in device discovery in most cases. Only in cases where the probability of signal collisions is infrequent due to a low amount of devices or slower advertising interval does the BLE 4 advertising perform better.

Due to the hardware, transmit power is limited values between −20 dBm and 8 dBm and values such as 0, 4, and 8 dBm are chosen because they can provide a practical range of operation. The power consumption of the device when applying different transmit power values slightly varies. For example, when applying the same advertising interval such as 20 ms and sending equal sized payload, the total power consumption of a device may increase around 0.6 mW for each 4 dBm increase of transmit power.

Sending a scan response can happen after each advertising packet is transmitted on a primary advertising channel. When a scannable advertising device is continually scanned with active scanning, it has to respond to a scan request that causes additional radio transmissions, thus increasing the overall power consumption. The different phases of advertising are outlined and additional power consumption caused by the scan request and scan response can be seen in part (3) in Fig. 17. The peak current for advertising, scanning request, and response is around 17 mA. In Table 2 power consumption is 4.86 mW when a peripheral device is scanned with 10 ms scan interval and window. It has 0.5 mW more power consumption compared to a device that is not responding to scan requests.

The payload part of uncoded LE packets is shown in Fig. 18, the user data in the payload is limited to only 244 bytes (ATT payload) even if the whole payload is 251 Bytes.

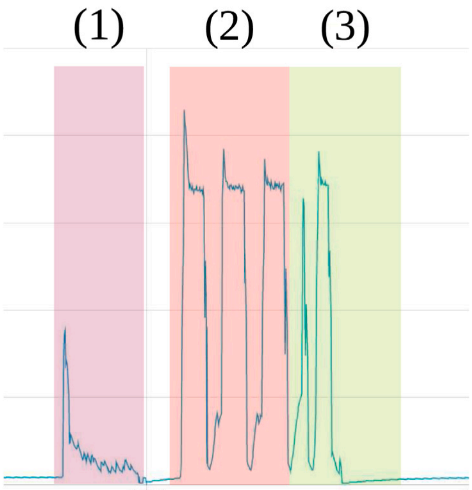
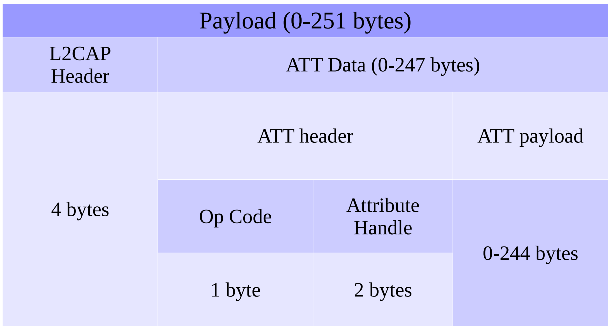


Fig. 17. (1) Different phases of advertising affecting the power consumption: Pre-processing and crystal ramp-up, (2) advertising event, (3) scan request and response.



<table>
<tr><th colspan="4">Payload (0-251 bytes)</th></tr>
<tr><td>L2CAP Header</td><td colspan="3">ATT Data (0-247 bytes)</td></tr>
<tr><td rowspan="3">4 bytes</td><td colspan="2">ATT header</td><td>ATT payload</td></tr>
<tr><td>Op Code</td><td>Attribute Handle</td><td rowspan="2">0-244 bytes</td></tr>
<tr><td>1 byte</td><td>2 bytes</td></tr>
</table>

Fig. 18. Payload part of an LE data packet.

The average energy consumption of the device when applying LE 1M and LE 2M features for sending different size ATT payloads with 8 dBm transmit power is presented in Table 3. The results show that sending the larger payload in each packet with the same LE feature (such as LE 1M or LE 2M) causes higher energy consumption. The

Table 3
Energy used in transmission (Tx) for different ATT payload sizes.

LE 2M					
Bytes	20	52	100	196	244
Energy (mJ)	8.02	17.54	24.55	47.60	51.60
LE 1M					
Bytes	20	52	100	196	244
Energy (mJ)	16.53	33.06	51.6	93.19	110.22

increasing rate of energy consumption is the same as the increasing rate of the payload size. Particularly, when raising the payload size from 52 Bytes to 244 Bytes, the energy consumption merely increases from 17.54 mJ to 51.6 mJ. This shows that using LE 2M PHY feature will save significantly more energy than using LE 1M PHY feature when larger payload sizes are transmitted.

The results also show that using the LE 2M PHY feature for sending large data with a high data rate is more energy-efficient than the using LE 1M feature. It is noted that the ATT payload size of 52 Bytes, 100 Bytes, and 196 Bytes would fit ECG monitoring applications using 250 Hz, 500 Hz, and 1 kHz sampling frequency (in which each sample has 24 bits and 8 channels are applied), respectively. This implies that the LE 2M can be suitable for high data rate applications while maintaining a high level of energy efficiency.

7. Conclusions

This paper presented possible mobility support approaches for edge-based IoT systems. The approaches were implemented in an entire system placed in an office environment. Different parameters affecting the handover mechanisms were applied and tested. A device's energy consumption and the system latency were measured and evaluated via concrete practically implemented mobility. The results show that using the BLE LE 2M PHY feature, especially when transferring larger ATT payloads reduces energy consumption significantly because it takes a shorter time to send the same amount of data compared to the LE 1M PHY feature. Switching the advertising interval length and connectivity as implemented in Case 2 or turning advertising off during connected state as implemented in Case 3 helps to reduce total energy consumption. Using extended advertising with LE 2M PHY if the feature is available reduces energy consumption provided that the advertising packets are larger in size. From experiments, it can be concluded that applying mobility support approaches can aid in properly maintaining the connection between an IoT device and edge-based IoT systems during mobility with low latency. Simple handover approaches, such as Case 1, 2, and 3 presented in this paper are not entirely suitable for real-time applications, especially if uninterrupted connections are required. It is recommended to have more advanced mobility support approaches for edge-based IoT systems. The handover approaches should both reduce system latency during mobility and maintain a high quality of edge services. Additionally, handle the worst mobility cases and challenging environments to suit time-critical high data rate applications.

CRediT authorship contribution statement

Risto Katila: Methodology, Writing – original draft, Editing, Visualization, Software, Formal analysis, Validation. **Tuan Nguyen Gia:** Conceptualization, Methodology, Writing – original draft, Visualization, Investigation, Reviewing and editing, Formal analysis, Supervision. **Tomi Westerlund:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] I. Tcarenko, et al., Energy-efficient iot-enabled fall detection system with messenger-based notification, in: International Conference on Wireless Mobile Communication and Healthcare, Springer, 2016.

[2] T.N. Gia, et al., Edge-AI in smart farming IoT: CNNs at the edge and fog computing with LoRa, in: 2019 IEEE AFRICON, 2019.

[3] A.M. Rahmani, et al., Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach, Future Gener. Comput. Syst. (2017).

[4] J. Ding, et al., IoT connectivity technologies and applications: A survey, IEEE Access 8 (2020) 67646–67673.

[5] J. Queralta, et al., Edge-AI in LoRa-based health monitoring: Fall detection system with fog computing and LSTM recurrent neural networks, in: 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), 2019, pp. 601–604.

[6] T.N. Gia, et al., Fault tolerant and scalable IoT-based architecture for health monitoring, in: 2015 IEEE Sensors Applications Symposium (SAS), IEEE, 2015, pp. 1–6.

[7] J. Tosi, et al., Performance evaluation of bluetooth low energy: A systematic review, Sensors 17 (12) (2017).

[8] M. Jiang, et al., IoT-based remote facial expression monitoring system with sEMG signal, in: IEEE SAS, IEEE, 2016, pp. 1–6.

[9] S.R. Hussain, et al., Secure seamless bluetooth low energy connection migration for unmodified IoT devices, IEEE Trans. Mob. Comput. 17 (4) (2018) 927–944.

[10] M. Baert, et al., A BLE-based multi-gateway network infrastructure with handover support for mobile BLE peripherals, in: 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2019, pp. 91–99.

[11] T.N. Gia, et al., Fog computing approach for mobility support in internet-of-things systems, IEEE Access 6 (2018).

[12] H. Fotouhi, et al., Reliable and fast hand-offs in low-power wireless networks, IEEE Trans. Mob. Comput. 13 (11) (2014) 2620–2633.

[13] A.J. Jara, et al., An initial approach to support mobility in hospital wireless sensor networks based on 6LoWPAN (HWSN6), 2010.

[14] M. Chen, et al., Efficient handoff algorithm for Bluetooth networks, in: 2005 IEEE International Conference on Systems, Man and Cybernetics, Vol. 4, 2005, pp. 3884–3889.

[15] M.L. George, L.J. Kallidukil, J.-M. Chung, Bluetooth handover control for roaming system applications, in: The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002. Vol. 1, 2002, pp. I–404.

[16] Bluetooth low energy scanning and advertising, 2021, Accessed: Dec. 2021, https://dev.ti.com/tirex/explore/node?node=AKvX4BPHvI6W3ea9a0OTxA__pTTHBmu__LATEST.

[17] Bluetooth core specification 5.2, 2019, Accessed: June. 2021, https://www.bluetooth.com/specifications/bluetooth-core-specification/.

[18] K. Townsend, C. Cufi, Akiba, R. Davidson, Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking, first ed., O'Reilly Media, Inc., 2014.

[19] K. Cho, et al., Analysis of latency performance of bluetooth low energy (BLE) networks, Sensors 15 (1) (2015) 59–78.

[20] J. Liu, C. Chen, Y. Ma, Modeling neighbor discovery in bluetooth low energy networks, IEEE Commun. Lett. 16 (9) (2012) 1439–1441.

[21] G. Shan, B.-H. Roh, Advertisement interval to minimize discovery time of whole BLE advertisers, IEEE Access 6 (2018).

[22] J.J. Treurniet, et al., Energy consumption and latency in BLE devices under mutual interference: An experimental study, in: International Conference on Future Internet of Things and Cloud, 2015, pp. 333–340.

[23] T.K. Sarkar, et al., A survey of various propagation models for mobile communication, IEEE Antennas Propag. Mag. 45 (3) (2003) 51–82.

[24] H. Nguyen, et al., Optimized indoor positioning for static mode smart devices using BLE, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017, pp. 1–6.

[25] A. Guidara, et al., Impacts of temperature and humidity variations on RSSI in indoor wireless sensor networks, Procedia Comput. Sci. 126 (2018) 1072–1081.

[26] Mobile edge computing (MEC); Technical requirements, 2021, Updated: Mar. 2016, Accessed: Dec. 2021, https://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf.

[27] Multi-access edge computing (MEC); Framework and reference architecture, 2021, Updated: Dec. 2020, Accessed: Dec. 2021, https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.02.01_60/gs_MEC003v020201p.pdf.

[28] Mobile edge computing;market acceleration; MEC metrics best practice and guidelines, 2021, Updated: Jan. 2017, Accessed: Dec. 2021, https://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/006/01.01.01_60/gs_MEC-IEG006v010101p.pdf.

[29] Mobile edge computing (MEC);end to end mobility aspects, 2021, Updated: Oct. 2017, Accessed: Dec. 2021, https://www.etsi.org/deliver/etsi_gr/MEC/001_099/018/01.01.01_60/gr_MEC018v010101p.pdf.

[30] T.N. Gia, et al., Energy efficient fog-assisted IoT system for monitoring diabetic patients with cardiovascular disease, Future Gener. Comput. Syst. 93 (2019) 198–211.

[31] T.N. Gia, et al., Low-cost fog-assisted health-care IoT system with energy-efficient sensor nodes, in: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 2017.

[32] A. Banks, et al., MQTT Version 5.0, OASIS Standard, 2019, https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html. Latest version: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html.

[33] Nordic Semiconductor, NRF52833 user guide, 2021, Updated Dec 2020, Accessed: Jun. 2021, https://infocenter.nordicsemi.com/pdf/nRF52833_DK_User_Guide_20201203.pdf.

[34] Nordic Semiconductor, Power profiler kit II user guide, 2021, Updated Jun 2021, Accessed: Jun. 2021, https://infocenter.nordicsemi.com/pdf/PPK2_User_Guide_v1.0.1.pdf.

[35] T.N. Gia, et al., Customizing 6LoWPAN networks towards Internet-of-Things based ubiquitous healthcare systems, in: NORCHIP, 2014, IEEE, 2014, pp. 1–6.

[36] G. Shan, B. Roh, Performance model for advanced neighbor discovery process in bluetooth low energy 5.0-enabled internet of things networks, IEEE Trans. Ind. Electron. 67 (12) (2020) 10965–10974.

Risto Katila received a bachelor of engineering degree in electronics specializing communication systems from Turku University of Applied Sciences, in 2015 and is working towards receiving a master's degree in information and communication technology in 2021 from University of Turku. His research interest include wireless networks and Internet of Things.

Tuan Nguyen Gia received the Ph.D. degree in technology. He is currently a Senior Researcher and the Vice Leader of the Turku Intelligent Embedded and Robotic Systems Research Group (tiers.utu.fi), University of Turku, Finland. He has published more than 60 international peer-reviewed articles. He is working on edge and fog computing. His research interest is e-health, drones, autonomous and embedded systems, smart cities, and blockchain. He is also working on reconfigurable computing with FPGA and CGRA, energy efficiency of wearable devices and remote monitoring systems. He is also a Topic Editor of Vehicles, Future Internet and Sensors journals.

Tomi Westerlund is an Associate Professor of Autonomous Systems and Robotics at the University of Turku and a Research Professor at Wuxi Institute of Fudan University, Wuxi, China. Dr. Westerlund leads the Turku Intelligent Embedded and Robotic Systems research group (tiers.utu.fi), University of Turku, Finland. His current research interest is in the areas of Industrial IoT, smart cities and autonomous vehicles (aerial, ground and sur-face) as well as (co-)robots. In all these application areas, the core research interests are in multi-robot systems, collaborative sensing, interoperability, fog and edge computing, and edge AI.