Edge-Based Hybrid System Implementation for Long-Range Safety and Healthcare IoT Applications

Fan Wu, *Member, IEEE*, Chunkai Qiu, *Graduate Student Member, IEEE*, Taiyang Wu, *Member, IEEE*, and Mehmet Rasit Yuce, *Senior Member, IEEE*

***Abstract*—In many conventional Internet-of-Things (IoT) applications, data are transferred directly from the sensor network to the cloud via a gateway for further data processing. However, this typical usage of a gateway is not suitable for every application. For short-range IoT wireless protocols [e.g., Bluetooth low energy (BLE)], multiple gateways are required to achieve broader coverage, which is inconvenient. In this article, an edge-based hybrid network system architecture is presented. The proposed system consists of hybrid routers and an IoT gateway. The router supports two wireless protocols, BLE and long range (LoRa), and is equipped with a solar energy harvester to extend the router's lifetime. It can extend the coverage of short-range BLE network by utilizing LoRa wireless technology, and support fundamental edge computing tasks such as preliminary data processing. The IoT gateway can support multiple IoT protocols, including LoRa, BLE, and XBee. It can perform more advanced edge computing tasks, such as data filtering, storage, processing, user interface, and cloud connection. Three case studies incorporating a wearable safety monitoring sensor network, a healthcare monitoring application, and a smart hospital application are studied with the proposed edge network system to demonstrate its promising capabilities to support IoT applications. Experimental evaluations indicate that by processing data at the edge, the minimal delay is only 11.5 ms. Furthermore, with the hybrid LoRa network implementation, the BLE network can be extended to 2.4 km.**

***Index Terms*—Bluetooth low energy (BLE), edge computing, gateway, Internet of Things (IoT), long range (LoRa), wireless sensor network.**

I. Introduction

INTERNET of Things (IoT) has become a promising technological paradigm in current and future generations of networking, sensing, and data collection in many environments. It is predicted that by 2030, 100 billion devices will be connected to the Internet [1]. These devices will be widely deployed in different IoT application domains, such as smart cities, healthcare monitoring, smart agriculture, and campus monitoring [2]–[7]. With the increasing demand for IoT devices, there is a need for a generalized IoT architecture that can support better multiple IoT applications and standards.

Manuscript received September 24, 2020; revised December 15, 2020; accepted December 23, 2020. Date of publication January 11, 2021; date of current version June 7, 2021. This work was supported by the Australian Research Council Future Fellowships under Grant FT130100430. *(Corresponding author: Mehmet Rasit Yuce.)*

The authors are with the Department of Electrical and Computer Systems Engineering, Monash University, Melbourne, VIC 3800, Australia (e-mail: fan.wu@monash.edu; chunkai.qiu@monash.edu; taiyang.wu@outlook.com; mehmet.yuce@monash.edu).

Digital Object Identifier 10.1109/JIOT.2021.3050445

An IoT gateway is a crucial element between various smart devices and cloud applications, which is the core of the network layer connecting smart IoT devices of the bottom layer (perception layer) to various applications in the upper application layer [8]. It is responsible for some essential tasks, such as translating protocols between sensors and the Internet and providing local data storage [9]–[11]. In many cloud-based IoT applications, the data acquired from sensors are directly transferred to a centralized server via gateway devices for data processing, analyzing, and storing, after which data consumers can access the required information [12]. However, with the increasing of IoT devices and data, the network bandwidth (BW) is becoming a bottleneck of cloud computing. The primary limitation of cloud computing is delay [13]. Besides, since the cloud server is typically located in a centralized position that is far away from the IoT devices, cloud computing mainly relies on the Internet protocol (IP) network. As a consequence, once the IP connection becomes intermittent, users may lose access to the data and services. This typical usage of a gateway and cloud-based computing is not suitable for applications that require real-time and low-latency feedback, such as healthcare, emergency monitoring, and safety applications.

Edge computing and fog computing are efficient enabling technologies that allow computation and storage to be performed or located at the edge of the network in proximity to IoT devices and sensors rather than processed in the cloud server [12]–[15]. The concept of edge/fog computing is proposed by researchers in multiple works in the present IoT design paradigm [16]–[21]. They can reduce the delay and latency, improve scalability, address the safety and privacy issue, and mitigate network traffic burden and BW. Both fog and edge computing are heterogeneous network infrastructure and can be connected to edge devices via various wireless technologies [e.g., ZigBee, Bluetooth low energy (BLE), long range (LoRa), etc.]. Edge-/fog-assisted system design can interoperate with all these edge devices and support different IoT services even though the Internet connection is intermittent.

The terms of fog computing and edge computing can be interchangeable, but fog computing focuses more on the infrastructure side, where edge computing focuses more on the things side [12]. In addition, the function of fog computing can sometimes be interchangeable with edge computing. According to [22], fog computing and edge computing are synonymous with the tasks executed at somewhere between the cloud and end devices rather than at the cloud center.

Fog computing emphasizes more on a virtualized platform, which can be a distributed infrastructure, such as a dedicated fog server. In contrast, edge computing is closer to the IoT devices. The computing can be performed in a dedicated gateway device, a simple network router, or even IoT end devices, such as a smartphone or smartwatch.

Different wireless protocols are typically required to connect IoT sensors to the edge computing devices. Short-range wireless protocols such as BLE generally have a limited range of up to 100 m, and they cannot connect to the Internet directly [23], [24]. Hence, multiple gateways are required to enable the Internet connection to cover a larger area and connect these devices to the Internet. However, gateway devices are generally more expensive than IoT end devices and complicated in terms of network protocol, hardware, and software implementation, which are challenging to implement and will increase the total deployment cost. Hybrid networks can be utilized to tackle some of these issues by combining long-range wireless technologies with short-range technologies to improve the coverage. Emerging low-power wide-area network (LPWAN) technologies, such as LoRa, Sigfox, and narrow band Internet of Things (NB-IoT), can be applied to increase the network coverage [25]–[27]. For instance, the work in [25] proposes an information monitoring based on LoRa and NB-IoT. The work in [26] proposes an approach by using various sub-GHz technologies (e.g., XBee-PRO 900HP and XBee 868LP) to extend the range of short-range IoT protocols.

In this article, an IoT edge architecture incorporating an edge gateway and a hybrid edge router that can support multiple wireless technologies and applications is presented. The proposed gateway can support different wireless protocols, provide local storage, enhance data security, and provide local data processing and filtering functions. The edge router can connect the short-range IoT devices with a remote gateway via a LoRa-based LPWAN network. Such an edge system can facilitate the deployment process for various IoT applications, extend the range of short-distance IoT protocols, improve the IoT heterogeneity, as well as provide better Quality of Service (QoS).

The major contributions of this article are as follows.

1) We propose an edge-computing-based system architecture with a hybrid router and an edge gateway for connected safety and healthcare IoT applications.
2) We design the software architecture of the gateway to support advanced edge computing tasks. The gateway supports multiple wireless protocols and provides local database, data processing, and cloud connection.
3) We design the hardware and software components of the hybrid router to extend the coverage of short-range BLE protocol and perform some basic edge computing tasks.
4) We integrate and evaluate the proposed edge architecture with three practical IoT applications, demonstrating its promising capabilities.

The remainder of this article is organized as follows. Section II discusses some related works. Section III presents the system architecture and focuses on both the design and implementation of the gateway and router. Section IV discusses the first use case, namely, a wearable safety monitoring application. Section V discusses the second use case, namely, a healthcare monitoring sensor network. Section VI presents the third use case, namely, a smart hospital application. Section VII concludes this work.

II. Related Works

There have been multiple IoT applications based on edge computing concepts providing and supporting better QoS for the applications. One typical edge-based architecture to support the healthcare industry is presented in [9]. The researchers propose an edge architecture named BodyEdge, which consists of a tiny mobile client BodyClient (BE-MBC) and a gateway (BE-GTW) device. The BE-MBC is a software application that can be installed on a smartphone and communicate with body sensors. Such an edge architecture greatly supports the healthcare applications in local environments, and in addition, it reduces the data traffic toward the cloud.

The work [11] presents an edge-based gateway design for smart IoT healthcare applications. The features of the smart e-Health gateway (UT-GATE) are: 1) local data processing capabilities; 2) adaptivity; 3) local storage; 4) local actuation; 5) security; 6) interoperability and reconfigurability; 7) IoT device discovery and mobility support; 8) energy efficiency support for sensor nodes; and 9) reduced latency. Local data analysis module can greatly improve the system reliability in the case of unavailability of Internet connection. Such an edge-based smart gateway can help to improve challenges, such as mobility, energy efficiency, scalability, interoperability, and security issue.

Another fog computing-based IoT architecture is proposed in [26]. The authors propose a hybrid architecture consisting of micro IoT collectors *u*Hub and more powerful macro gateways to combine the short-range radio technologies with long-range radio technologies. The *u*Hub provides services in microenvironments using IEEE 802.15.4/IEEE 802.11 technologies. It has the ability to collect data from connected networks, store data in the local database, transmit the data to the cloud via Constrained Application Protocol (CoAP) protocol, as well as forward data to macro IoT gateways via a sub-GHz network for further processing. The macro IoT gateway where the data processing is performed has higher performance compared to *u*Hub and is providing services in wider areas. This approach is of great importance for smart cities and smart campus monitoring scenarios with improved connectivity and processing power of the short-ranged IoT devices.

Fraga-Lamas *et al.* proposed a fog computing-based architecture using LoRaWAN for smart campus applications [28]. The architecture comprises three layers: 1) bottom; 2) middle; and 3) top layer. This is one of the few academic solutions that is based on LoRaWAN infrastructure and fog computing methodology. The bottom layer is the node layer, which contains multiple LoRaWAN nodes deployed across the campus. The middle later contains multiple fog gateways that provide fast location-aware responses to the nodes' requests. By adopting fog computing concept, the fog nodes throughout the campus support physically distributed, low latency, and location-aware applications that reduce the network traffic and

TABLE I
COMPARISON OF EXISTING EDGE/FOG COMPUTING SYSTEMS

Study	Targeted applications	Edge/fog computing devices	Sensors	Wireless	Coverage	Main contribution
[5]	Healthcare	HTC One X smartphone	Physiological	Not provided	Not provided	A robust fault-tolerant decision making scheme and advanced authentication scheme for secure and efficient healthcare system
[9]	Healthcare	Raspberry Pi 3, Nano PC, smartphone	Physiological	BLE, WiFi	Not provided	A software framework for healthcare applications at the edge reducing the data traffic towards Internet
[11]	Healthcare	Pandaboard with TI SmartRF06	Physiological	WiFi, Bluetooth, 6LoWPAN	Not provided	A smart e-health gateway which offers high level services
[18]	Human activity recognition	Raspberry Pi 3	Not provided	Not provided	Not provided	Utilize Docker Container to realize intelligent, effective, and lightweight edge computing tasks
[26]	Traffic control, Environmental monitoring	Raspberry Pi 3 B	Picture	XBee, WiFi, IEEE 802.15.4	Long	An architecture to combine the short-range IoT protocols with long-range protocols, enabling long-range wireless communication
[28]	Smart campus	Raspberry Pi 3	Not provided	LoRaWAN	Long	A fog computing architecture for smart campus to support low-latency and location-aware applications with simulations
[29]	Patient monitoring in smart homes	Not provided	Physiological, environmental	Not provided	Not provided	A remote patient health monitoring using the concept of fog computing at smart gateways
Proposed work	Healthcare, Hand hygiene, Safety monitoring	Gateway: Raspberry Pi 3 B+, Router: nRF52840	Physiological, Environmental, Proximity	BLE, LoRa, WiFi, XBee	Short, Long	A short- and long-range multi-protocol edge architecture that supports various IoT applications.

the computational load of the cloud server. The top layer is the cloud, which supports user applications, remote access, and data storage services.

Gioia *et al.* [30] proposed an advanced IoT gateway solution AMBER to support heterogeneous IoT technologies. The AMBER board, which is an open-hardware and open-source software gateway solution to be used in IoT network to interact with different IoT standards, supports modularity, flexibility, and scalability in IoT scenarios, providing distributed processing power at the edge of the network. The design concept of the AMBER is based on extender modules and System on Modules (SoMs). Modularity is addressed by the extender modules, which are basically connection sockets. Common communication interfaces, such as interintegrated circuit (I^2C), serial peripheral interface (SPI), universal asynchronous receiver/transmitter (UART), are exported on the sockets. SoM is an embedded module hosting computational and storage capabilities, which is normally a microprocessor with random access memory (RAM) and communication interfaces.

Several other fog-/edge-based approaches have also been proposed by researchers in work [18], [29], [31], [32]. Most of these works are focusing on one sector, such as healthcare, and there is a need for developing and implementing a practical edge-based gateway architecture that supports multiple wireless technologies and various IoT applications. Table I presents a detailed summary and comparisons of recent edge/fog computing systems. Compared to existing studies, our proposed system presents a promising solution to support various IoT applications at the edge, especially for connected safety and healthcare sectors. It can greatly support heterogeneous IoT applications, realizing better IoT interoperability and QoS.

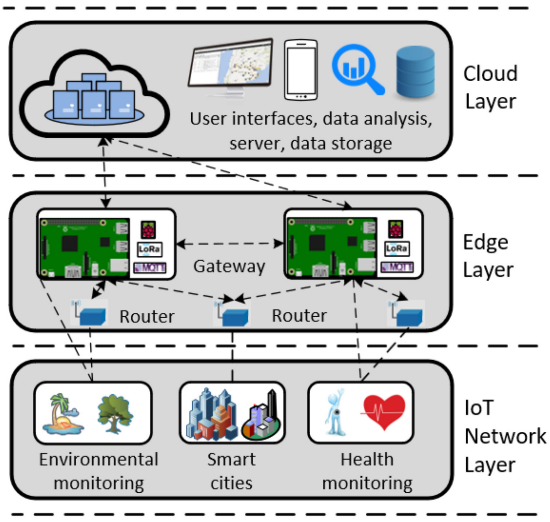


Fig. 1. Typical three layers IoT architecture.

III. SYSTEM ARCHITECTURE AND IMPLEMENTATION

A typical IoT system architecture is depicted in Fig. 1. There are three layers, including an IoT network layer, an edge layer, and a cloud layer. The bottom layer is the IoT network layer consisting of multiple IoT sensor devices, which is mainly for sensing different environments. The cloud layer is located in a remote data center, which receives the data from the edge layer and is responsible for hosting a Web-based application, storing sensor data, processing, and analyzing sensors' data. The middle layer is the edge layer that is mainly in charge of connecting the IoT devices to the cloud and performing some edge computing tasks at the proximity of IoT devices.

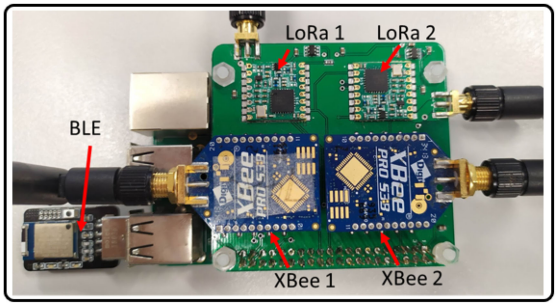


Fig. 2. IoT gateway implementation.

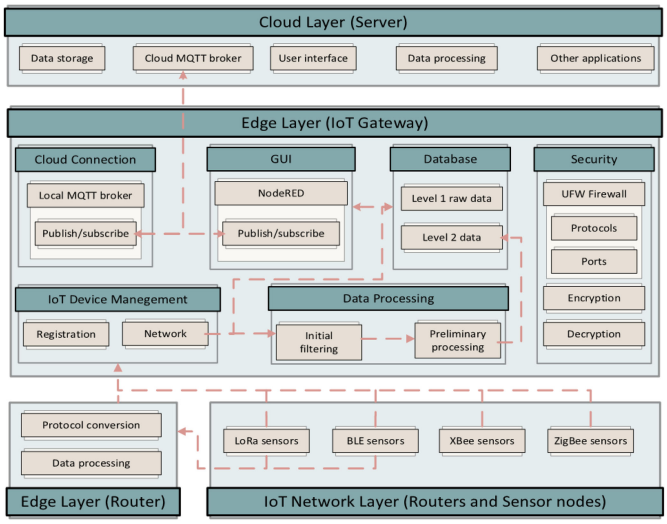


Fig. 3. IoT gateway software architecture.

In this work, we present an edge-assisted hybrid network system that connects short- and long-range IoT applications. The proposed system mainly focuses on the design and implementation of the edge layer, which incorporates an edge gateway and a router. The role of the gateway and router is summarized as follows.

1) The edge gateway provides low-cost and local edge services for different IoT devices in the bottom layer.
   a) Devices management.
   b) Cloud connection.
   c) Data storage.
   d) Data processing.
   e) Security.
   f) Visualization.
2) The hybrid router provides services for nearby short-range IoT devices (i.e., BLE sensors in this work).
   a) Extend the range of wireless sensor nodes if needed.
   b) Provide prompt responses and fundamental edge tasks.
   c) Measure surrounding environments.

A. Edge Gateway Implementation

1) Hardware Implementation: The hardware of the gateway is based on a Raspberry Pi Model 3B+, which is a single board computer with embedded Wireless Fidelity (WiFi), Ethernet, and BLE modules. The operating system (OS) is installed on a micro-SD card. A printed circuit board (PCB) is designed and integrated with Pi to support different wireless protocols, such as LoRa, ZigBee, and XBee. Fig. 2 presents the prototype of the gateway with different wireless modules connected. The gateway consumes much lower power compared to a normal server or computer. The required supply voltage is only 5 V with 2.5-A current. The gateway can be powered by a main power supply through a wall adapter. Besides, it can also be powered by a portable power bank to increase mobility. With a 20 000-mAh power bank, the gateway can maintain operation for a maximum of 8 h.

There is a low-power switch (TPS22918) for each radio frequency (RF) module. Each of the RF can be turned on and off depending on the different requirements of applications. RF modules can be configured by the Raspberry Pi via the UART or SPI interface. The LoRa module embedded in this work is RFM95 from HoperRF electronics, which is a low-cost and low-power LoRa transceiver with high sensitivity. Two LoRa modules are included so that the gateway can support two LoRa networks with two different modem configurations. The XBee module is XBee-PRO 900HP long-range module, which supports long-range mesh networking. The ZigBee module from Digi company has the same footprint as XBee. Therefore, these two modules are interchangeable, and they can be replaced by other RF modules with the same footprint. A BLE dongle based on nRF52840 is connected to the Raspberry Pi's universal serial bus (USB) port, which transfers data to the Pi via USB.

2) Gateway Software Architecture: The Raspberry Pi possesses a Linux distribution (Raspbian) as its OS, where programs can be written in Python, C, or Java. The local storage can be utilized to store data when there is no Internet connection to successfully transmit the data from the edge devices to the cloud. The software architecture is presented in the edge layer of Fig. 3. There are six software modules: 1) IoT device management; 2) data processing; 3) local graphical user interface (GUI); 4) database; 5) security; and 6) cloud connection.

a) IoT devices management module: This module is in charge of sensor nodes' registration and network management. Each RF module has a unique media access control (MAC) address. Hence, the device registration function can differentiate whether the data are from a registered or an unregistered device. If the wireless data are received from a new device, the registration process will be initialized, and the device MAC address and join timestamp will be inserted into the database for future reference. The network management function is utilized to manage different networks, for instance, to perform modem configuration for embedded LoRa network. The data will be inserted into different databases according to the wireless protocol.

b) Data processing module: It is important to process the data at the edge of the network because such edge computing task can provide low latency, prompt feedback, and reduced

network BW. The data processing module contains the following functions: initial filtering and preliminary processing. The initial filtering function removes any unwanted wireless data, such as duplicate data from the same network router or IoT devices. The preliminary data processing processes the filtered data, converts the machine data to meaningful data, and raises an alert if there is an emergency that needs to be resolved or requires attention.

c) Local GUI module: The local GUI provides data visualization and other management functionalities for the network operator and users. Therefore, the network can still be established and managed without Internet connection. The GUI is developed using an open-source and cross-platform Node.js and Node-RED package.

d) Database management module: MongoDB is one of the most popular NoSQL databases. It is installed in the local gateway to support local data storage. The data are stored in JSON-like (JavaScript Object Notation) documents and are very flexible and scalable. MongoDB supports replicated servers and indexing, and it offers drivers for different programming languages [33]. A MongoDB Python driver "PyMongo" is used to access and manage the MongoDB database. Besides MongoDB, MySQL is also installed to support various applications' requirements.

There are two levels of storage data: 1) level 1 and 2) level 2. The level 1 data contain all the raw data received from the wireless sensor nodes, which is stored in the original format for future data recovery purposes. The level 2 data storage contains all the processed data.

e) Security module: Data security issue is a major concern in current IoT design applications. To ensure the data are securely transmitted and stored in the gateway, a security function is implemented in this work, which includes data decryption and encryption. Initially, when all the data are transferred from the sensor nodes to the gateway, the data are encrypted. After the gateway receives the data, the data decryption function will decrypt the RF payload if required. This is achieved by utilizing the advanced encryption standard (AES) 128-b encryption engine. Furthermore, the data will be encrypted when transferred to the cloud.

A firewall named uncomplicated firewall (UFW) is installed to restrict the access to various protocols and ports. Only certain protocols and ports are enabled to allow transfer data between the gateway and the cloud. For example, message queue telemetry transport (MQTT) utilizes transmission control protocol (TCP) to transfer data to the cloud. Therefore, TCP protocol over port 1883 is enabled. Secure shell (SSH) over port 22 is also enabled so that the network operator can access the gateway via SSH to configure the gateway. Current work focuses on protecting the wireless data and gateway access control based on OS level techniques. We plan to utilize advanced authentication method in our future work to improve edge security.

f) Cloud connection module: There are multiple IoT connection techniques that can be utilized to transfer the data from the gateway to the cloud, such as MQTT and CoAP. They are both specially designed for machine-to-machine (M2M) communication. MQTT requires only limited network BW. It is simple and lightweight, which is ideal for resource-constrained IoT applications. It is a publish-subscribe-based two-way communication protocol and can easily support Web-based user user interface (UI) design. Moreover, MQTT has three QoS settings to support reliable data transmission. Therefore, MQTT is installed in the gateway as the default messaging protocols to connect the gateway, cloud, and Web-based UI. In addition, CoAP and representational state transfer (REST) can also be utilized to transfer data between the cloud and gateway depending on the requirements of different applications.

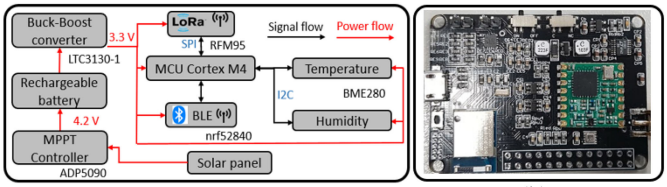


Fig. 4. Router: (a) block diagram and (b) router device.

B. Hybrid Router Implementation

The hybrid router is in proximity to the IoT devices, which is located between the gateway and the IoT sensor nodes as presented in Figs. 1 and 3. For instance, in a hospital environment, the router can be a network router located inside a patient room. Fig. 4(a) and (b) presents the hardware block diagram and the prototype of the router, respectively. The router is composed of a power management unit (PMU), a high-performance microcontroller (MCU), BLE, LoRa, and an onboard temperature and humidity sensor.

The PMU has three components, including an energy harvesting unit (ADP5090), a buck-boost regulator (LTC3130-1), and an energy storage unit (a 2600-mAh rechargeable lithium battery). The energy harvesting circuit based on the ADP5090 has been investigated in our previous work [34]. ADP5090 is a high efficiency and ultralow-power energy harvesting chip with maximum power point tracking (MPPT) and charge management function. It can convert DC power with more than 80% conversion efficiency from a solar cell and store into a rechargeable battery. LTC3130-1 is a low-power buck-boost regulator, which is utilized to provide constant output voltage (3.3 V) for the device. A commercial solar cell (55.0–67.5 mm) is chosen, which provides a maximum of 310 mW power. The energy harvesting circuit can harvest enough energy during daytime and fully charged the battery to realize the continuous operation of the router in outdoor scenarios. The average power consumption for the router is 12 mA in continuous operating mode. Hence, in indoor conditions, the router can continuously operate for about nine days. After that, a battery charging or replacement will be required.

The MCU is based on Nordic nRF52840, which is an advanced and high-performance SOC supporting multiple wireless protocols. BLE 5 is selected as the embedded wireless protocol, which features long-range, high data rate (2 Mb/s), high transmission power (+8 dBm), −95 dBm sensitivity, and low current transmission during TX and RX mode. The LoRa

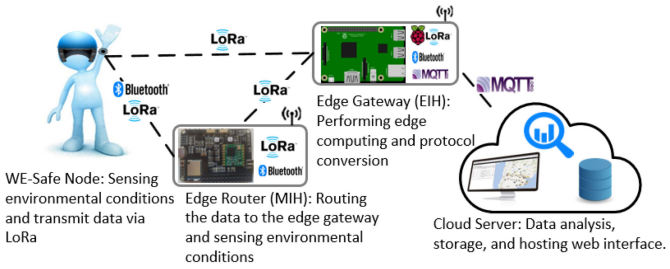


Fig. 5. System architecture of the wearable safety monitoring sensor network.

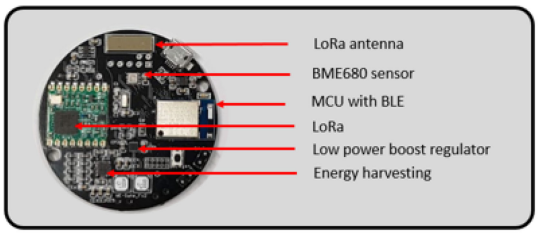


Fig. 6. Wearable sensor node for safety monitoring.

TABLE II
HARDWARE IMPLEMENTATION OF USE CASE 1

Device	MCU	Wireless	Location/ Coverage
Wearable Sensor	nRF52840	LoRa, BLE	Indoor/ 100 m (LoRa) Outdoor/ 400 m (LoRa)
Router	nRF52840	LoRa	Indoor/ 200 m Outdoor/ 2 km
Gateway	Raspberry Pi Model 3B+	LoRa, WiFi	Indoor

[1] The router can be installed indoors or outdoors.
[2] The gateway is installed on a second floor laboratory.

module (RFM95) supports long-distance data transmission. There is an onboard environmental sensor BME280 for sensing the temperature and relative humidity near the router location. Therefore, the router itself can be used as an environmental sensor node to detect the environments.

C. Overview of Case Studies With the Proposed Architecture

The proposed edge architecture can be integrated with different IoT applications. Three use cases are studied to demonstrate how the proposed system can benefit different IoT applications. Different aspects, experiments, and results are evaluated and discussed, demonstrating the edge system's promising capabilities to benefit IoT applications and heterogeneity.

IV. CASE STUDY 1—WEARABLE NETWORK FOR SAFETY MONITORING APPLICATION

A. System Implementation

The first case study is a wearable safety monitoring application, which must be capable of measuring safety-related environments, transmitting data reliably to the gateway and the cloud server, and performing some edge tasks to provide low-latency decision making. The system architecture is presented in Fig. 5. The hardware implementation is presented in Table II. The network is tested in a campus environment to monitor some safety-related environments, such as temperature, humidity, and ultraviolet (UV) index.

The safety monitoring application has two main requirements: 1) short response time and 2) long-range data transmission, which can be realized by the proposed edge system. First, with the integration of the router and gateway, the network system can detect and trigger an alarm for the safety workers as soon as any emergency detected. This is demonstrated by measuring the time delay that the system requires to get a response from the router or gateway. Second, the network can cover a large area with the help of the router. The hybrid network architecture with BLE and LoRa can significantly improve the network coverage and reduce the need for multiple gateway implementations.

1) Hardware Implementation: The wearable sensor node utilized in this work is an improved version of our previous work [34]. A higher performance, BLE embedded, and lower power consumption MCU (nRF52840), and a new power management circuit are implemented in the wearable node as demonstrated in Fig. 6. Apart from the BLE network, a long-range and low-power LoRa module (RFM95) is also included in the wearable sensor node. Therefore, the sensed data not only can be transmitted via BLE network but also by the LoRa network to reduce the dependency on multiple gateways. The power management circuit includes an energy harvesting unit (ADP5090) and an ultralow-power boost regulator (MAX17222), which can harvest the energy from solar and provide enough power for the rest of the circuit with low power consumption.

2) Experimental Implementation: During the test, two wearable sensor nodes are worn by two subjects. The gateway is installed near the window inside a laboratory that is on the second floor. A router node is installed off the campus to help route the data packet from wearable sensor nodes to the gateway. It is installed at a location where it can directly connect to the gateway without additional hops.

For network configuration, sensor nodes can communicate with a router via either BLE or LoRa network or directly with a gateway via the LoRa network. Therefore, if the sensor node is within the BLE network range of a router, the data are transmitted via the BLE. After data are received by the router, the router will forward this information to the gateway via LoRa. If it is outside the BLE range, the data will be transmitted via LoRa, and these data can also be received by the gateway. Therefore, the data processing module of the gateway is required to filter out all the duplicated data transmitted from the same nodes.

B. Experimental Evaluation and Discussion

1) Network Testing: LoRa should be configured to transmit data at a relatively long-range while maintaining data rate and average transmission time. Higher spreading factor (SF) results in longer range but lower data rate and longer transmission time. An increase in BW will reduce the receiver sensitivity and lower the transmission time. Lower coding rate (CR) is

Fig. 7. LoRa network coverage. (a) Wearable sensor transmits data directly to the gateway. (b) Wearable sensor transmits data to the router, then to the gateway.

TABLE III
COMPARISON OF RTT DELAY FROM DIFFERENT EDGE DEVICES AND CLOUD PLATFORM

Network	Route	Time (ms)
BLE	Sensor - Router or Sensor - Gateway	11.5 ms
LoRa	Sensor - Router or Sensor - Gateway	87 ms (SF = 7) 316 ms (SF = 9) 1035 ms (SF = 11)
LoRa	Sensor - Router - Gateway	180 ms (SF = 7) 638 ms (SF = 9) 2078 ms (SF = 11)
MQTT	Gateway - Cloud	83 ms (QoS 1) 155 ms (QoS 2)

more tolerant to interference but will result in a longer transmission time. In this work, SF 9, CR 4/7, BW 125 kHz are selected as the LoRa modem configuration to achieve a better tradeoff between transmission time, transmission range, and tolerance to interference.

Fig. 7 presents the network coverage with the aforementioned LoRa configuration. Fig. 7(a) presents the network coverage without the hybrid router. As can be seen from the figure, LoRa network can only cover 400 m with stable transmission using 2-dBi chip antenna. This result is not promising because such implementation will require multiple gateways to receive the sensors' data. To improve the network coverage, a hybrid router is installed off the campus, which is approximately 2 km away from the gateway, as shown in Fig. 7(b). The router can maintain a stable wireless link with the gateway by using a higher gain antenna (3.5 dBi). With the help of the router, the network coverage is extended to 2.4 km from the original 400 m.

2) Round Trip Time Delay Test: The round trip time (RTT) between sensor nodes, the edge devices, and the cloud server is measured and discussed. The results are presented in Table III. The RTT delay is the length of time it takes from the sensor node transmits a fixed-length packet (10 B) until it receives an acknowledgment from the router, gateway, or cloud. When the sensor node can directly communicate with the router with BLE network, the delay is only 11.5 ms.

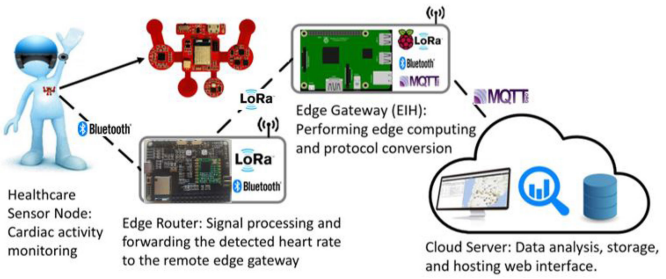


Fig. 8. System architecture of the wearable remote healthcare monitoring system.

However, for LoRa-based wireless links between the router and gateway, it mainly depends on the Time on Air (ToA) of a LoRa wireless packet. The ToA varies according to different modem configurations, such as SF, BW, and CR. Since BW (125 kHz) and CR (4/7) are fixed in our experimental setup, three different SFs are considered to calculate the RTT from the sensor node to the router or gateway. As can be seen from Table III, with SF configured as 9, it takes the sensor node 316 ms to receive an acknowledgment from a router or gateway if the data are directly transmitted to the router or gateway. The time will double if the acknowledgment is sending from the gateway through the router and then reach sensors. Since the ToA of LoRa transm

ission will change according to different configurations of SF, and CR, for different applications that require shorter response time, these parameters can be modified accordingly. For example, by increasing the BW to 250 kHz and reducing the SF to 7, the RTT delay can be reduced to only 44 ms when transmitting the same packet. However, this will reduce the transmission range.

MQTT protocol is used to upload the data from the gateway to the cloud. In this work, the cloud is an Ubuntu server at DigitalOcean. With QoS1 setting, which transmits at least one packet to the cloud and requires an acknowledgment from the cloud, the RTT is 83 ms. It can be clearly concluded that by processing data at the edge level, the RTT can be reduced by at least 83 ms. Therefore, some decision-making tasks can be lowered to the edge devices such as a router to reduce the delay from the cloud. If there is an emergency condition, the system delay can be reduced compared to cloud-based applications, and the whole system can remain normal operation without an IP network.

V. CASE STUDY 2—WEARABLE REMOTE HEALTH MONITORING SYSTEM

A. System Implementation

The second use case studied is a wearable healthcare system for remote cardiac activity monitoring, which is illustrated in Fig. 8. Heart rate, as one of the vital signs, is the most direct sign of our heart health. Long-term and continuous heart rate monitoring is essential to early diagnosis of heart disease. With the proposed router and gateway architecture, both short-distance and long-distance remote healthcare monitoring can be achieved. Shorter response time can also be realized.

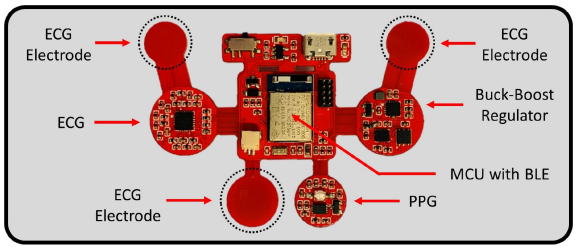


Fig. 9. Healthcare sensor node.

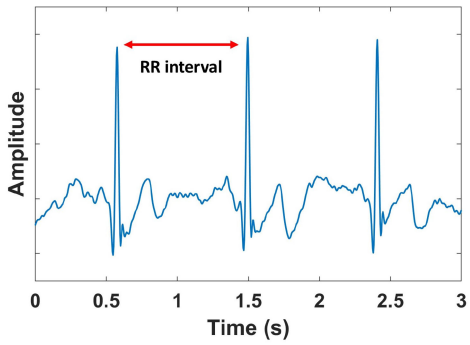


Fig. 10. Filtered ECG signal.

The system is tested in an office to monitor the heart rate of an office worker. The Electrocardiogram (ECG) signal is acquired by the wearable sensor node and then sent to the router through the BLE network. The signal processing and heart rate computation algorithms are performed in the router; then the heart rate data are forwarded to the remote gateway and cloud server. Since the data are processed in the router, the workload of the gateway and cloud server has been reduced. Furthermore, the local router and gateway can provide a faster response if any emergency is detected. The edge system can still provide services offline even if the IP connection to the cloud is lost.

1) Hardware Implementation: Fig. 9 shows the design of the wearable healthcare sensor node. The wearable sensor employs the AD8232 ECG front end to measure the ECG signal of the heart. Three printed dry electrodes are attached to the chest with conductive gel to detect the electrical activity of heartbeats. The analog ECG signal is sampled at the frequency of 250 Hz in this experimental testing and is transmitted from the wearable sensor node to the router in real time via BLE. The data received by the router are filtered by a high-pass digital filter at 0.5 Hz, which follows a low-pass digital filter at 35 Hz to remove the noise such as motion artifacts. The ECG R-peak is identified afterward, which represents a single heartbeat. The time interval between two consecutive ECG R-peaks is used to compute the heart rate, as shown in Fig. 10.

B. Experimental Implementation and Discussion

During the test, the sensor node is attached to the chest of a subject. The device is placed near the heart. Thus, strong cardiac signals can be obtained.

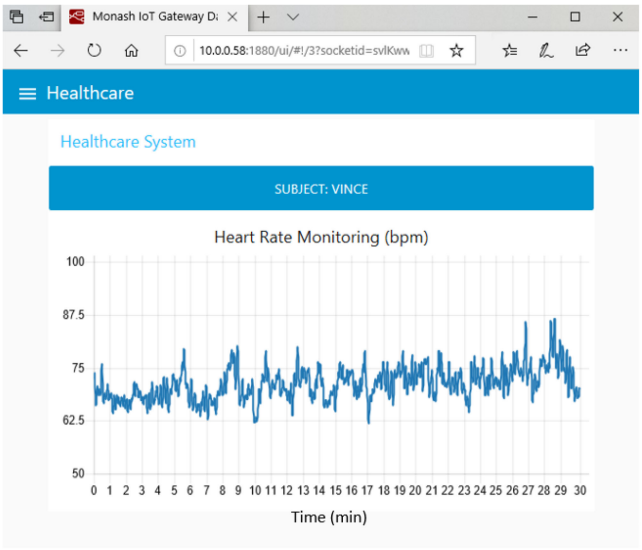


Fig. 11. Website hosted on the local gateway demonstrating the heart rate data.

1) Long-Range Connected Health: The router is located in the center of the office ceiling to maximize the BLE coverage. The experimental measurement runs for 30 min. During the experiment, an office staff is sitting on a chair and performing word typing. The raw ECG data are sent by the wearable sensor node to the router for data processing and heart rate computation. The heart rate data are then transferred to the remotely located gateway via LoRa and cloud server. Fig. 11 demonstrates the 30-min heart rate data on the Web-based GUI. With the proposed hybrid edge architecture system, fundamental data processing tasks are distributed to the router, which helps to reduce the burdensome workload on the cloud server. Long-range connected health monitoring is also realized, which reduces the need for multiple gateway infrastructure. Local health data visualization, storage, and cloud connection are achieved with the help of the IoT gateway.

2) Emergency Situations: In an emergency situation, such as sudden cardiac arrest, immediate emergency medical services are required. An alarm system that can provide fastest response is essential. In our system, an emergency alarm will be triggered if the router does not detect a heartbeat within 2 s, that is, when the heart rate is below 30 bpm or cardiac arrest occurs. The maximum response time is the 2-s threshold in addition to the BLE transmission delay from the sensor node to the local router.

In this case study, we demonstrate the advantages of using this edge architecture to realize long-range connected healthcare monitoring. The health sensor node can also connect with the gateway via BLE network that supports a higher data rate than LoRa. Since the gateway has a better computation power, more advanced functionalities of the gateway can be utilized. For instance, heart rate variability (HRV) can be analyzed and processed at the gateway. PPG and ECG biosignals can be monitored and analyzed in real time together at the gateway to gain a better understanding of personal health conditions.

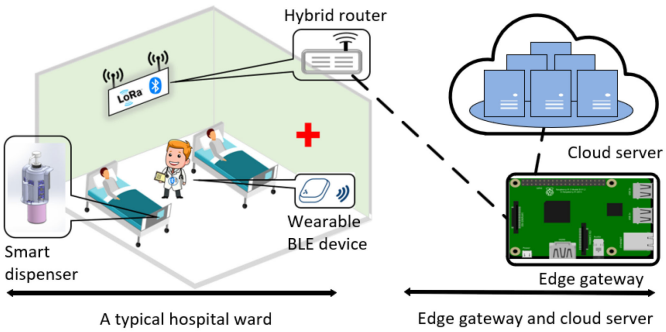


Fig. 12. Contact tracing and hand hygiene activities monitoring in smart hospital applications.

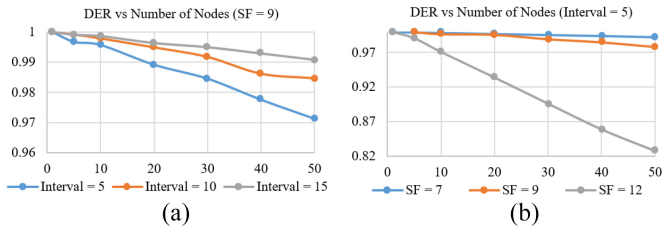


Fig. 13. (a) DER versus number of nodes with fixed spreading factor. (b) DER versus number of nodes with 5 min transmitting interval.

VI. Case Study 3—Smart Hospital for Contact Tracing and Hand Hygiene Monitoring

A. System Implementation

The third case study is a smart hospital application targeting contact tracing and hand hygiene monitoring. The system architecture with single ward scenario is presented in Fig. 12 [35], which consists of wearable BLE devices, two smart hand-washing dispensers, a hybrid router, an IoT gateway, and a cloud server. Within a hospital ward, when a health professional wearing a wearable BLE device enters the room, his/her ID information and present time can be constantly monitored and recorded by smart dispensers as well as any hand-washing activities. The monitored information will be forwarded to the gateway and cloud server. Therefore, the health professional's close contacts as well as places he/she has been to can be recorded for contact tracing purposes. The hand-washing information recorded by smart dispensers can also be utilized to improve hand hygiene compliance rates.

Our edge architecture can play important roles in reliable wireless data collection and edge computing tasks for this smart hospital application. In a multiple wards scenario, each ward can be regarded as an isolated edge system. Only the router collects, manages, and forwards any data from dispensers and wearable IDs to the gateway via a private LoRa network. The private network only covers targeted hospital buildings, which is secure, reliable and can protect better the privacy of healthcare works as all the data are encrypted during the transmission and only accessible by authorized people. Data storage, analysis, security, and cloud connection functions can be performed to support the contact tracing and hand hygiene monitoring. Such usage of the proposed edge architecture reduces the requirement of a gateway in each ward, improves the network coverage, and supports several edge computing tasks.

B. Experimental Results and Discussion

An open-source LoRa network simulator named LoRaSim [36] is utilized to test the network scalability. In this simulator, users can define the number of nodes, average sending interval, simulation time, and LoRa modem configuration to simulate the data extraction rate (DER). DER is defined as the ratio of received messages to transmitted messages over a period of time, which can be used to present the network scalability and packet delivery rate. Here, we configure the LoRa modem the same as case study 1 with different transmitting intervals, number of nodes, and SFs. The results are presented in Fig. 13. As can be seen from Fig.13(a), the DER increases as the transmitting interval increases and decreases with the increasing number of nodes. The DER is still higher than 0.97 when there are 50 LoRa nodes. Higher SF increases the time of air, which will result in higher packet collision and lower DER. With SF 12, the DER reduces significantly to approximately 0.82 compared to SF 7 and 9 as demonstrated in Fig.13(b).

A further field test involving ten LoRa routers (SF 9, CR 4/7, and BW 125 kHz) to simulate a 10-ward hospital environment has been tested with the gateway. Routers are placed in a building where the furthest one is 130 m from the gateway. Each router is configured to transmit data to the gateway at an interval of 5 min. Over a period of 10 h, 1200 LoRa packets are transmitted, and 1190 packets are successfully received. The DER is 0.991. From the above results, the proposed edge architecture is cable of covering a small-sized hospital and providing a reliable network solution. Since the router is responsible for each ward's data collection and providing any feedback if necessary, the BLE network coverage is greatly enhanced by LoRa and the system's response time can be further shortened by local data processing capabilities at the router and the gateway. Furthermore, the private and encrypted LoRa network does not rely on any commercial LoRa networks, which can address better the privacy and security concerns in the smart hospital settings.

VII. Conclusion

In this article, we have introduced the design of an edge architecture consisting of a hybrid router and a gateway that can be used for various IoT applications. The IoT gateway has been designed to support multiple protocols and perform some higher level edge computing tasks, for instance, local data storage, cloud connection, data processing, local UI, and data filtering. The hybrid router can extend the range of short-range wireless protocols such as BLE to a remote location via a LoRa network. It can also perform some fundamental edge tasks, for example, preliminary data filtering, data storage, and prompt response. The system has been verified and studied with three practical IoT applications to demonstrate its feasibility. Such a system architecture can be integrated

with various IoT applications to facilitate the system development and deployment process, improve the IoT heterogeneity, extend the range of wireless protocols, and also provide a better QoS for edge-based applications.

The existing system utilizes hybrid routers and an edge gateway with different edge computing tasks. The router consumes low power, which has great mobility and can be installed outdoors and indoors with battery power supply. It is also feasible to utilize the gateway device to perform the tasks of the router. However, the gateway requires mains power supply in the building and its installation location is not as flexible as the router. Although the proposed system was only studied with three IoT applications in this article, its applicability is not limited to them. The system can be applied with other IoT sensing applications by installing appropriate software and hardware components. In the future, we plan to integrate more long-range (e.g., SigFox and NB-IoT) and short-range [e.g., ZigBee and near field communication (NFC)] wireless protocols into the edge architecture, and evaluate their performance for different IoT applications. The current security modules focus on decryption/encryption of wireless data, and access control of the edge gateway, we plan to integrate more edge tasks with advanced security measures, such as secure authentication functions for better protecting, collecting and processing the health data at the edge.

ACKNOWLEDGMENT

The authors would like to thank Mohammed AlDujaili for preparing the initial prototype of the IoT gateway.

REFERENCES

[1] E. A. Rogers and E. Junga, *Intelligent Efficiency Technology and Market Assessment*, American Council for an Energy-Efficient Economy, Washington, DC, USA, 2017.

[2] A. Gaur, B. W. Scotney, G. P. Parr, and S. I. McClean, "Smart city architecture and its applications based on IoT," in *Proc. ANT/SEIT*, 2015, pp. 1089–1094.

[3] P. Gope and T. Hwang, "BSN-Care: A secure IoT-based modern healthcare system using body sensor network," *IEEE Sensors J.*, vol. 16, no. 5, pp. 1368–1376, Mar. 2016.

[4] T. Wu, F. Wu, C. Qiu, J.-M. Redoute, and M. R. Yuce, "A rigid-flex wearable health monitoring sensor patch for IoT-connected healthcare applications," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6932–6945, Aug. 2020.

[5] P. Gope, Y. Gheraibia, S. Kabir, and B. Sikdar, "A secure IoT-based modern healthcare system with fault-tolerant decision making process," *IEEE J. Biomed. Health Inform.*, early access, Jan. 1, 2020, doi: 10.1109/JBHI.2020.3007488.

[6] N. Suma, S. R. Samson, S. Saranya, G. Shanmugapriya, and R. Subhashri, "IoT based smart agriculture monitoring system," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 5, no. 2, pp. 177–181, 2017.

[7] A. Alghamdi and S. Shetty, "Survey toward a smart campus using the Internet of Things," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud (FiCloud)*, 2016, pp. 235–239.

[8] C.-H. Chen, M.-Y. Lin, and C.-C. Liu, "Edge computing gateway of the industrial Internet of Things using multiple collaborative microcontrollers," *IEEE Netw.*, vol. 32, no. 1, pp. 24–32, Jan./Feb. 2018.

[9] P. Pace, G. Aloi, R. Gravina, G. Caliciuri, G. Fortino, and A. Liotta, "An edge-based architecture to support efficient applications for healthcare industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 481–489, Jan. 2019.

[10] A.-M. Rahmani *et al.*, "Smart e-health gateway: Bringing intelligence to Internet-of-Things based ubiquitous healthcare systems," in *Proc. 12th Annu. IEEE Consum. Commun. Netw. Conf. (CCNC)*, 2015, pp. 826–834.

[11] 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.*, vol. 78, pp. 641–658, Jan. 2018.

[12] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[13] P. Gope, J. Lee, R.-H. Hsu, and T. Q. S. Quek, "Anonymous communications for secure device-to-device-aided fog computing: Architecture, challenges, and solutions," *IEEE Consum. Electron. Mag.*, vol. 8, no. 3, pp. 10–16, May 2019.

[14] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[15] L. Greco, G. Percannella, P. Ritrovato, F. Tortorella, and M. Vento, "Trends in IoT based solutions for health care: Moving AI to the edge," *Pattern Recognit. Lett.*, vol. 135, pp. 346–353, Jul. 2020.

[16] F. Wu, T. Wu, and M. Yuce, "An Internet-of-Things (IoT) network system for connected safety and health monitoring applications," *Sensors*, vol. 19, no. 1, p. 21, 2019.

[17] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: Architecture, key technologies, applications and open issues," *J. Netw. Comput. Appl.*, vol. 98, pp. 27–42, Nov. 2017.

[18] M. Al-Rakhami *et al.*, "A lightweight and cost effective edge intelligence architecture based on containerization technology," *World Wide Web*, vol. 23, no. 1, pp. 1341–1360, 2020.

[19] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data*, 2015, pp. 37–42.

[20] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[21] M. Chen, W. Li, G. Fortino, Y. Hao, L. Hu, and I. Humar, "A dynamic service migration mechanism in edge cognitive computing," *ACM Trans. Internet Technol.*, vol. 19, no. 2, pp. 1–15, 2019.

[22] F. A. Kraemer, A. E. Braten, N. Tamkittikhun, and D. Palma, "Fog computing in healthcare—A review and discussion," *IEEE Access*, vol. 5, pp. 9206–9222, 2017.

[23] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, 2018.

[24] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues," *Comput. Netw.*, vol. 144, pp. 17–39, Oct. 2018.

[25] X. Zhang, M. Zhang, F. Meng, Y. Qiao, S. Xu, and S. Hour, "A low-power wide-area network information monitoring system by combining NB-IoT and LoRa," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 590–598, Feb. 2019.

[26] L. Davoli, L. Belli, A. Cilfone, and G. Ferrari, "From micro to macro IoT: Challenges and solutions in the integration of IEEE 802.15.4/802.11 and sub-GHz technologies," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 784–793, Apr. 2018.

[27] J. Rubio-Aparicio, F. Cerdan-Cartagena, J. Suardiaz-Muro, and J. Ybarra-Moreno, "Design and implementation of a mixed IoT LPWAN network architecture," *Sensors*, vol. 19, no. 3, p. 675, 2019.

[28] P. Fraga-Lamas *et al.*, "Design and experimental validation of a LoRaWAN fog computing based architecture for IoT enabled smart campus applications," *Sensors*, vol. 19, no. 15, p. 3287, 2019.

[29] P. Verma and S. K. Sood, "Fog assisted-IoT enabled patient health monitoring in smart homes," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1789–1796, Jun. 2018.

[30] E. Gioia, P. Passaro, and M. Petracca, "AMBER: An advanced gateway solution to support heterogeneous IoT technologies," in *Proc. 24th Int. Conf. Softw. Telecommun. Comput. Netw. (SoftCOM)*, 2016, pp. 1–5.

[31] A. Nugur, M. Pipattanasomporn, M. Kuzlu, and S. Rahman, "Design and development of an IoT gateway for smart building applications," *IEEE Internet Things J.*, early access, Dec. 7, 2018, doi: 10.1109/JIOT.2018.2885652.

[32] R. K. Pathinarupothi, P. Durga, and E. S. Rangan, "IoT-based smart edge for global health: Remote monitoring with severity detection and alerts transmission," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2449–2462, Apr. 2019.

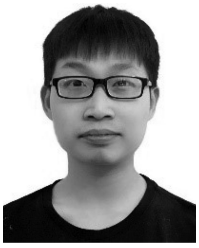
[33] A. Boicea, F. Radulescu, and L. I. Agapin, "MongoDB vs Oracle–Database comparison," in *Proc. 3rd Int. Conf. Emerg. Intell. Data Web Technol.*, 2012, pp. 330–335.

[34] F. Wu, J.-M. Redouté, and M. R. Yuce, "We-safe: A self-powered wearable IoT sensor network for safety applications based on LoRA," *IEEE Access*, vol. 6, pp. 40846–40853, 2018.

[35] F. Wu *et al.*, "An autonomous hand hygiene tracking sensor system for prevention of hospital associated infections," *IEEE Sensors J.*, early access, Nov. 30, 2020, doi: 10.1109/JSEN.2020.3041331.
[36] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRA low-power wide-area networks scale?" in *Proc. 19th ACM Int. Conf. Model. Anal. Simulat. Wireless Mobile Syst.*, 2016, pp. 59–67.

Fan Wu (Member, IEEE) received the B.E. and Ph.D. degrees from Monash University, Melbourne, VIC, Australia, in 2015 and 2020, respectively.

He is currently a Research Fellow with Monash University, where he was a Research Assistant with the Engineering Department from 2015 to 2017. His main areas of research interest are wireless sensor networks, wearable sensors, energy harvesting, triboelectric nanogenerator, and IoT innovations.



Chunkai Qiu (Graduate Student Member, IEEE) received the B.E. degree from Monash University, Melbourne, VIC, Australia, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Systems Engineering.

His main area of research interest is wearable devices and triboelectric nanogenerator.

Taiyang Wu (Member, IEEE) received the B.E. degree from Southeast University, Nanjing, China, in 2014, and the Ph.D. degree from the Department of Electrical and Computer Systems Engineering, Monash University, Melbourne, VIC, Australia, in 2019.

His main areas of research interest are wireless sensor network, wearable biomedical sensors, IoT-connected healthcare applications, and energy harvesting techniques.

Mehmet Rasit Yuce (Senior Member, IEEE) received the M.S. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2001, and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, NC, USA, in December 2004.

He is an Associate Professor with the Department of Electrical and Computer Systems Engineering, Monash University, Melbourne, VIC, Australia. His research interests include wearable devices, Internet-of-Things for healthcare, wireless implantable telemetry, wireless body area network, bio-sensors, integrated circuit technology for wireless, biomedical, and RF applications. He has published more than 180 technical articles in the above areas.

Dr. Yuce received the NASA Group Achievement Award in 2007 for developing an SOI transceiver. He received the Best Journal Paper Award in 2014 from the IEEE Microwave Theory and Techniques Society. He received a Research Excellence Award in the Faculty of Engineering and Built Environment, University of Newcastle in 2010. He is a Topical Editor for IEEE SENSORS JOURNAL, the Editor-in-Chief for *Sensors*, and a guest editor for several IEEE journals.