IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 13, 2018, accepted June 11, 2018, date of publication June 15, 2018, date of current version July 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2848119

Fog Computing Approach for Mobility Support in Internet-of-Things Systems

TUAN NGUYEN GIA[1], (Student Member, IEEE), AMIR M. RAHMANI[2,3], (Senior Member, IEEE), TOMI WESTERLUND[1], (Member, IEEE), PASI LILJEBERG[1], (Member, IEEE), AND HANNU TENHUNEN[1], (Member, IEEE)

[1]Department of Future Technologies, University of Turku, 20500 Turku, Finland
[2]Department of Computer Science, University of California at Irvine, Irvine, CA 92697, USA
[3]Institute of Computer Technology, TU Wien, 1040 Wien, Austria

Corresponding author: Tuan Nguyen Gia (tunggi@utu.fi)

This work was supported in part by the Academy of Finland, in part by the Nokia Foundation, in part by the University of Turku Graduate School, and in part by the Finnish Foundation for Technology Promotion.

ABSTRACT Handover mechanism for mobility support in a remote real-time streaming Internet-of-Things (IoT) system was proposed in this paper. The handover mechanism serves to keep the connection between sensor nodes and a gateway with a low latency. The handover mechanism also attentively considers oscillating nodes which often occur in many streaming IoT systems. By leveraging the strategic position of smart gateways and Fog computing in a real-time streaming IoT system, sensor nodes' loads were alleviated whereas advanced services, like push notification and local data storage, were provided. The paper discussed and analyzed metrics for the handover mechanism based on Wi-Fi. In addition, a complete remote real-time health monitoring IoT system was implemented for experiments. The results from evaluating our mobility handover mechanism for mobility support shows that the latency of switching from one gateway to another is 10%–50% less than other state-of-the-art mobility support systems. The results show that the proposed handover mechanism is a very promising approach for mobility support in both Fog computing and IoT systems.

INDEX TERMS Mobility, fog computing, IoT, health monitoring, handover, latency, energy efficiency.

I. INTRODUCTION

Internet-of-Things (IoT) [1]–[3] can be described as a worldwide network where humans and objects from different disciplines in both physical and virtual world can be interconnected and interact with each other. IoT is considered a key enabler to address problems in many fields ranging from healthcare to smart spaces and transportation. Remote monitoring IoT-based systems often use wireless sensor network to collect and transfer data to the Cloud where the data is retrieved in real-time via terminals such as a web browser or mobile applications [4]–[9]. Wireless protocols such as Wi-Fi, classic Bluetooth, LoRAWAN, Bluetooth Low Energy (BLE), nRF, or IEEE 802.15.4 are commonly applied in many applications [10]–[12]. For example, environment monitoring for agriculture often uses low data rate wireless protocols such as LoRaWAN or 6LoWPAN because information of environments such as temperature and humidity does not change rapidly. In contrast, remote real-time health monitoring applications demanding high-fidelity multi-channel bio-signals often use high data rates wireless protocols such as Wi-Fi or IEEE 802.11ah [13].

Although the conventional IoT systems [14], [15] have shown some advantages such global data access and real-time monitoring, they still have several limitations in terms of latency, reliability, communication bandwidth, and accessibility. In these systems, conventional gateways merely receive data from sensor nodes and forward the data to the Cloud. There has been a growing tendency towards the three-layer architecture applying Fog computing which is a convergence network of interconnected and distributed smart gateways. The three-layer sensor-Fog-Cloud architecture provides a proper solution for mentioned limitations [16]–[18]. Fog is capable of reducing the burdens of the Cloud and tendering variety of services such as geographical distribution,

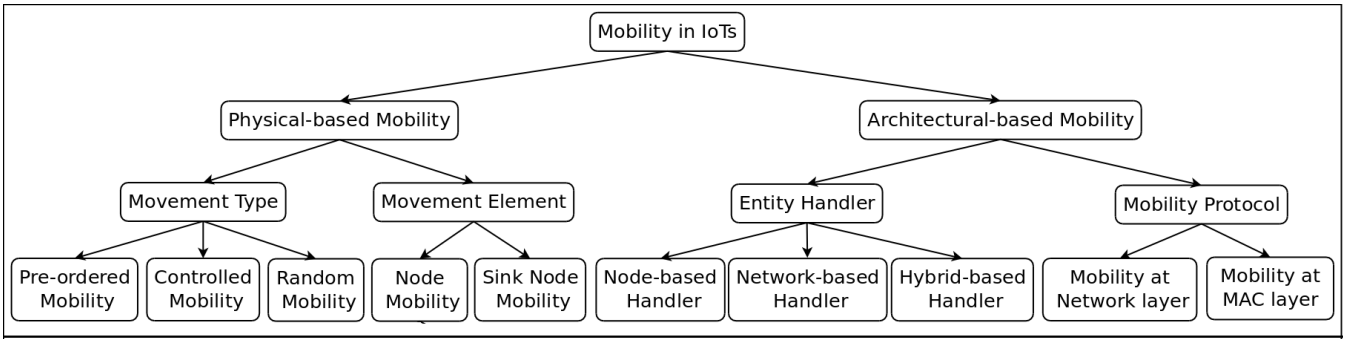

FIGURE 1. Mobility in IoT.

location awareness, and real-time interaction. As shown in [19] and [20], Fog enables low-power consumption at sensor nodes as well as bandwidth savings (from sensors to the Cloud) for data-intensive applications. Fog have been applied in many systems [20]–[24] to solve existing challenges.

Mobility support is a key requirement for many real-time IoT systems as missed or delayed data during mobility can lead to severe consequences. In order to support mobility, an IoT system needs to be equipped with a handover or hand-off mechanism which is responsible for de-registering a sensor node from a source access-point and registering it to a new access-point seamlessly. It is a challenging task to implement an advanced handover mechanism for full mobility support in critical domains such as healthcare [25], [26] due to strict requirements of security, latency, network coverage, and reliability [27]. This issue becomes much more challenging for fog-assisted IoT systems because smart gateways at the edge provide distributed storage and Fog services. When handling mobility, a handover mechanism needs to effectively cooperate with Fog services to update and synchronize the distributed storage.

Currently, existing Fog-based methods [28]–[30] cannot completely solve problems in Fog-based systems especially for high data rate applications. For instance, mobility management cannot be guaranteed when the connection between Fog and the Cloud is interrupted. Oscillating nodes which move back-and-forth between gateways during a short time period are not considered. The node oscillation is critical to the handover mechanism as it can cause overloading the gateways. In some cases, it might cause the connectivity interruption between other sensor nodes and gateways. In this paper, we propose a mobility support approach for Wi-Fi-based real-time IoT health monitoring systems through an efficient handover mechanism. Exploiting the proposed approach, objects/persons can be remotely monitored in real-time without any interruption in the mobility. Our approach addresses primary types of mobility together with a node oscillation phenomenon. The main contributions of this work are summarized as follows:

- Novel handover mechanism for remote real-time monitoring with a negligible latency overhead.
- Real-time notification services for emergency or other irregular situations such as a dead node.
- Light-weight solution to address node oscillation.
- Analysis of the handover mechanism characteristics, particularly latency, through a hardware-software prototype.

The remainder of the paper is organized as follows: Section 2 covers background and motivation. In section 3, metrics in handover mechanism are presented. In section 4, impact factors on mobility support are discussed. Section 5 presents the proposed handover mechanism. Section 6 presents the test-bed setup. Section 7 covers implementation of the proposed system. Section 8 presents evaluation the proposed system. Section 9 covers discussion. Finally, Section 10 concludes the work.

II. BACKGROUND AND MOTIVATION

Mobility in IoT systems can be hierarchically classified into primary mobility types shown in Fig. 1. In order to provide an elaborated view of mobility, each type is discussed in this section with proper details.

Movement type can be categorized into random, predefined, and controlled classes. Dealing with random mobility is the most challenging because mobility parameters of the random mobility such as moving paths, destination points, and movement duration are unknown. When a handover mechanism can handle the random mobility, it can also control other movement types.

Movement elements can be categorized further into sink node movement [31], [32] and sensor node movement. Among these movements, dealing with the sink node movement is more complicated because it causes changes in the network topology and the network's coverage areas. Fortunately, sink nodes or gateways (access-points) in applications in different fields such as manufacturing industry, education, and healthcare centers are often fixed in particular places because several costs (e.g., setup, management and maintenance) can be reduced while maintaining the high quality of services. Smart-phone-based sink nodes or access-points

are used in some systems [33], [34] but they are not widely applied. In such systems, the quality of services cannot be guaranteed when the gateway's battery level goes low. In practice, most of the mobility cases are caused by the sensor node movement. Therefore, we focus on the sensor node movement in the paper. It is noted that access-point and gateway are interchangeable terms in this paper.

According to Raja and Su [35], node mobility can be categorized into weak and strong mobility. Weak mobility is primarily caused by hardware failures or the depletion of battery. If weak mobility is not detected in time, the connectivity of sensor nodes will be disrupted. Strong mobility occurs when a node moves from a gateway to another one in the same network by intention or external interactions such as wind, water, or rain. From another viewpoint [36], micro and macro mobility are two categories of node mobility. Micro mobility arises when a sensor node moves from a gateway to another one in the same network. Macro mobility occurs when a node moves from one network to another network. It is recognized that a single viewpoint among mentioned alternatives cannot cover all cases of mobility. In this paper, we consider both strong mobility and micro mobility as node mobility while weak mobility is considered as a malfunction case. In the paper, weak mobility discussed in [35] is not considered as a type of node mobility because a static node may deplete their battery or crash. However, dead devices and malfunction cases due to hardware failures or the depletion of the battery are considered in this work for avoiding the discontinuation of services.

In most of the cases, dealing with healthcare applications requirements (e.g. latency and quality of bio-signals) are often more challenging than the requirements in other fields such as farming. For example, additional efforts are required for mobility support in e-health applications due to strict requirements of medical systems such as critical response time [37]. Accordingly, many examples and discussed applications in this paper are related to healthcare.

Due to the demand for mobility awareness in remote health monitoring systems, many approaches have been recently proposed. In this context, González-Valenzuela *et al.* [38] present a mobility support approach for in-home health monitoring systems using wearable sensors. In their approach, continuous monitoring of in-home patients is facilitated via an efficient hand-off protocol. In [39]–[41], Jara *et al.* propose a mobility support solution based on 6LoWPAN protocol for in-hospital health monitoring systems. By deploying sink nodes and gateways in their proposed architecture, intra-mobility and fault tolerance are also supported. In [42], Fotouhi *et al.* present a mobility support solution for wireless sensor network (WSN) and wireless body sensor networks. The approach uses the sensor velocity and the received signal strength (RSS) as vital parameters for the handover mechanism. One shortcoming of their approach is the overhead of the presented continuous message exchange algorithm which causes transmission overhead and high power consumption.

The discussed approaches show several benefits such as reasonable handover latency in the context of healthcare, however, they are not designed for fog-enabled IoT systems. More precisely, distributed storage and push notifications cannot be maintained or updated during mobility in the aforementioned approaches.

For dealing with mobility in smart cities, Francesco *et al.* [26] propose a mobility management method using follow-me Cloud-Cloudlet [43] in Fog-based radio access networks. In this approach, the handover for mobility support is triggered when a user moves between transportation infrastructures of smart cities (e.g., bus, train, etc.). As this technique is designed for city-scale mobility support, the mobility mechanism in this approach happens through the Cloud infrastructure rather then gateway-to-gateway data/control handover. The approach necessitates a large volume of data exchange between the edge and the Cloud and also under-utilizes the benefits of Fog computing (e.g., Internet connection to the Cloud is needed for mobility support). However, this approach is not efficient for mobility support in infrastructures such as hospitals, nursing homes, etc. where users' movement happens within the premises (i.e., between gateways) and is expected to be more frequent compared to city-scale travels. Such short-scale scenario calls for more efficient local mobility support mechanisms. Bittencourt *et al.* [28] address scheduling issues during mobility in the Fog layer. However, they do not provide a fog-based handover mechanism for mobility support.

In this paper, Wi-Fi is focused due to the following reasons: i) In the continuous e-health monitoring systems such as multi-channel ECG, EMG, and EEG monitoring, high transmission data rates are the prerequisites to achieve the high quality of signals. For example, each sensor node often collects about 90 kbps, 190 kbps, and 96 kbps for 8-channel ECG, 8-channel EMG, and 24-channel EEG applications, respectively [44]. Comparing to the other popular wireless communication protocols such as classic Bluetooth, Bluetooth Low Energy (BLE), IEEE 802.15.4, Wi-Fi supports much higher data rate and throughput. For example, data rates of IEEE 802.11b are up to 11 Mbps while data rates supported by other protocols such as Zigbee, 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks), and BLE are about 250 kbps [45]. In practice [46], these protocols merely support a data rate up to 160 kbps. ii) Wi-Fi supports multiple connection simultaneously whilst BLE and classic Bluetooth cannot support. iii) Wi-Fi-based systems are ubiquitously applied in many fields. Therefore, a solution for mobility issues of these systems can play a large contribution to the society.

The rationale behind this work is the demand for the design and implementation of mobility aware service with a robust handover mechanism customized for IoT systems based on Wi-Fi. In detailed, the proposed approach will focus on real-time remote health monitoring IoT systems

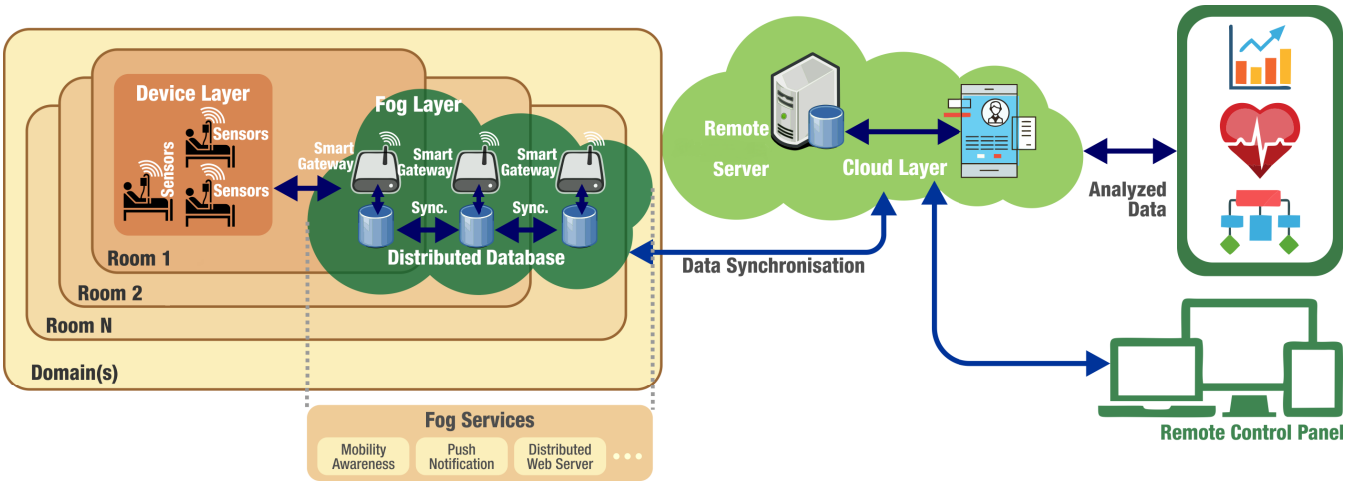

FIGURE 2. Remote real-time monitoring IoT system.

gathering a large amount of data, such as the scenario shown in Fig. 2. The Fog-based system shown in Fig. 2 has 3 main layers including a layer of sensor nodes, a layer of smart gateways with Fog computing and a layer of the Cloud and terminals. Sensor nodes collect contextual and e-health data such as ECG, EMG, room temperature, humidity and transmit the data to smart gateways for distributed storage, processing, and analysis. The processed or raw data is then transmitted to the Cloud for global storage and further processing. End-users such as medical doctors can access to real-time data via a mobile application or a web browser. The mobility-aware service needs to support different movement types (pre-defined, controlled, and random type), and node mobility. In addition, the burdens of sensor nodes cannot be increased when the system deals with mobility. To address these targets, the handover mechanism should be completely implemented at the Fog. Furthermore, the handover mechanism should reduce the handover latency to fulfill time requirements of critical applications such as real-time remote health monitoring IoT systems.

III. METRICS IN HANDOVER MECHANISM

Handover mechanisms often rely on one or several metrics such as Received Signal Strength Indicator (RSSI), velocity of objects and Link Quality Indicator (LQI) for making handover processes. These metrics are discussed in detail as follows:

Received Signal Strength Indicator (RSSI) indicates the signal power of a message received by a node. RSSI is one of the most popular metrics used in handover mechanisms [47], [48]. In optimal cases, RSSI can be used for directly estimating the distance between a sender and a receiver. However, it is not a simple task to calculate the distance in practice when merely relying on RSSI because it is not linear and it is affected by interference from the surrounding environment [49]. Therefore, surrounding environments, context, and network deployment must be attentively considered when building a handover mechanism based on RSSI.

General handover approaches are often based on the best RSSI value and a threshold value [49]. RSSI values of a sensor node towards two or more gateways are compared when the node moves to an overlapping area which is an area covered by two adjacent gateways. Correspondingly, the stronger RSSI value indicates that the node may be close to one gateway and it is likely to move to that gateway. Therefore, the node is instructed to connect and register to that gateway. This approach has advantages of simplicity but it has several drawbacks such as instability and inaccuracy in many cases. For example, it is not reasonable to directly compare RSSI values when an overlapped area is covered by an indoor gateway and an outdoor gateway. In order to overcome some of the mentioned drawbacks, another approach uses a threshold value for deciding an instant moment to register to a new gateway. When an RSSI value of a sensor node towards a gateway is smaller than a threshold value, the node starts to look for other gateways via active or passive scanning discussed in Section IV. If the RSSI value from the scanning is larger than the threshold value, it registers with the gateway corresponding to this RSSI value. Although this approach provides some advantages, there are several disadvantages. For instance, a node may continuously search for a gateway when it does not find an RSSI value larger than the threshold value. Accordingly, it causes a large overhead of network transmission and energy consumption. Hence, RSSI should not be used as a standalone metric for assessing link quality or qualifying handover mechanism [50].

Link Quality Indicator (LQI): In addition to RSSI, LQI can be used for handover mechanisms [51] as the second parameter. LQI is based on signal-to-noise ratio and indicates the quality of each received packet via average correlation values. In general, the LQI value depends on the distance between a sensor node and a gateway. When the distance increases, the LQI value decreases, and vice versa.

Similar to RSSI, the LQI value is influenced by the surrounding environment. The usage of LQI is similar to the discussed RSSI based approaches.

Signal to Interference Plus Noise Ratio (SINR) can be calculated by dividing the sum of the interference power from all interfering signals and the power of background noise. SINR can be considered as one of the most proper metrics for assessing link quality [50], [52]. However, it is difficult to retrieve an accurate SINR due to interference from unknown devices.

Packet Delivery Ratio (PDR) is the ratio between the number of received packets at a receiver and the number of sent packets. It can be approximately estimated by utilizing the history of PDR or by counting the number of received packets in a short period of time [53]. PDR is commonly used as a metric for calculating the best route and transmission rate. In many handover mechanisms, PDR is used alongside with RSSI or LQI for providing appropriate handover decisions and assessing link quality [50].

Bit Error Rate (BER): represents the ratio of error bits towards received bits during a certain time window. However, this metric is not often used in handover mechanisms because it is not simple to measure BER where a pseudo-random data sequence transmission must be considered during measurements [54].

Velocity is used alongside with RSSI or LQI in handover mechanisms [49]. When the velocity of a sensor node increases, a handover latency proportionally rises [55]. In general, it is not simple to capture the speed of a sensor node in an instant time. In order to measure the speed of a sensor node, other technologies such as dual loop detector, or magnetic sensor [56] should be implemented in the sensor node. Correspondingly, it causes large energy consumption. Despite the difficulties, velocity is used as a supplementary metric in many handover mechanisms for improving handover decisions. Fortunately, the node velocity in some applications does not vary dramatically and can be estimated. For example in healthcare, the speed of a sensor node attached to a patient is approximately 1-2 m/s in normal cases [49].

Moving Direction: It is an advantage for a handover mechanism when the movement direction of a sensor node is detected, as it can be used to predict the next destination gateway. As a result, overheads of network transmission caused by broadcasting or multicasting from the source gateway to other gateways can be avoided. The movement direction of a sensor node can be possibly estimated via methods such as the triangulation [57], [58], angle of arrival [59], and the time of arrival [60].

Global Position: Possibly, sensor nodes are equipped with global positioning systems. Corresponding, a map of sensor nodes can be tracked and a handover mechanism can use GPS values (global locations) for performing its handover decisions. However, GPS has several major drawbacks: (i) when a GPS device enters indoor or underground areas, GPS signals get blocked easily, (ii) GPS signals are highly influenced by interference when a GPS device is located near tall buildings, (iii) continuously collecting GPS signals costs high energy consumption [61]. Therefore, it is not commonly used in handover mechanisms for mobility support in many IoTs systems.

IV. IMPACT FACTORS ON MOBILITY SUPPORT

In order to provide a comprehensive view of a handover mechanism, factors impacting on mobility support in IoT systems using the 802.11 technology are discussed. These factors are mobility scenarios, handshaking messages for 802.11 connection, and network deployment.

A. MOBILITY SCENARIOS

With the purpose of achieving an accurate and precise handover mechanism, we categorize health monitoring related mobility into two scenarios: (i) node mobility between in-door or outdoor locations, (ii) node mobility between in-door and outdoor locations. In the first scenario, vital metrics (e.g. RSSI and LQI) can be directly used for the handover mechanism. In the second scenario, these parameters must be recalculated by adding effects from the surrounding environment such as temperature and interference signals. For example, a temperature of a hospital room is usually stable. In contrast, out-door temperature varies depending on particular geographical locations and weather conditions. According to Xu *et al.* [62], RSSI varies approximately 5.0 dBm for a change of 10 Centigrade. The differences between two adjacent contexts (indoor and outdoor) are complementary by offsets. These offset values must be periodically updated due to potentially rapid changes in surrounding environments.

B. MESSAGE HANDSHAKING IN 802.11 CONNECTION

When a Wi-Fi client wants to connect to a network, it must register to a Wi-Fi access-point or a gateway. The registering process consists of several request and response messages shown in Fig. 3. First, the client searches for nearby access-points via a passive or active scanning. Particularly in

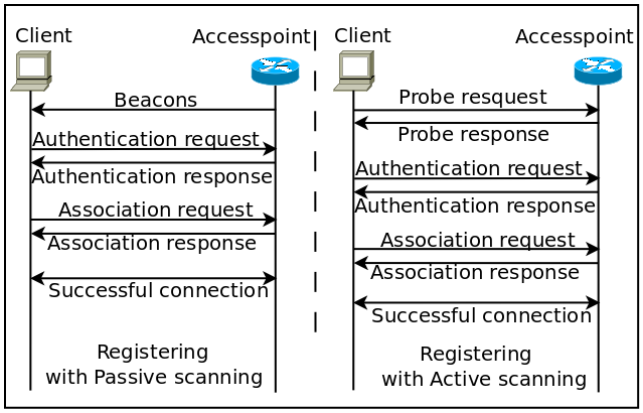

FIGURE 3. Wi-Fi connection.

the passive scanning, the client listens to beacon frames which are periodically sent by the access-points. In the active scanning, the client sends probe requests to nearby access-points and waits for probe responses from these access-points. After receiving beacon frames or probe responses, the client has detailed information of these access-points such as SSID, capability information and supported data rates. Based on the information, the client can choose the most suitable access-point to associate with. In order to achieve a successful connection, the client must fulfill network security requirements. For instance, a client must exchange the correct WPA2 key with an access-point to connect to a Wi-Fi network which is secured with a WPA2-personal type. In order to provide security information, the client sends an authentication request frame and waits for an authentication response frame from the access-point. The number of the authentication request and response frames depends on network security types. For example, it requires a couple of authentication request frames and an authentication response frame in an open access network while it exchanges two couples of those frames in a network with WEP security. In other complex mechanisms like 802.1X/EAP, the number of exchanged authentication frames is higher. After authentication steps are completed, the client can associate with the access-point by sending an association request frame. The client is registered to the access-point when it receives an association response frame from the access-point. These registering steps are expensive in terms of latency and energy consumption; especially in case of deregistering a sensor node from one access-point and registering the node to a new access-point during mobility.

C. GATEWAY DEPLOYMENT

In order to maintain a continuous connection between a device and a network, the device must be located inside the network coverage areas. In most of the cases, adjacent gateways have some overlapping areas. In this paper, we propose an arrangement for adjacent gateways as shown in Fig. 5. In the setting, there are four zones including personal zone, weak zone, sensitive zone and shared zone. To maintain the consistency for the whole article, gateways in the following discussions are similar in terms of type, model, and specification such as in-door smart gateways based on Pandaboard devices [63].

1) PERSONAL ZONE

The personal zone, shown in Fig 5, has the best values of metrics (i.e. RSSI value and link quality indicator) among all zones. In the personal zone, a connection between a sensor node and a gateway is maintained without any interruption in most of the cases. Therefore, it is unnecessary to run the handover algorithm. In some cases such as hardware failure (e.g. malfunction node or gateway) or the depletion of the power supply, the connection can be interrupted or disconnected. In order to deal with such cases, an investigation service implemented in Fog checks both hardware failure (e.g. malfunction node or gateway) and a status of the connection between sensor nodes and gateways. When a gateway does not receive any data from a sensor node during a short time period (e.g. about 5-10 s), the service sends some pre-defined signals (e.g. ''status'' signals) to a node and waits for responses. In the configuration of sensor nodes, when a sensor node receives a specific signal or a command (e.g. ''status'' signals) from an associated gateway, it will reply to the gateway with a specific message such as ''alive node'' or ''low battery level''. If there is no response from the sensor node, ''double checking'' method is performed by continuously sending 3 more signals in every 3 s. If there is still no response, the service invokes the notification service to inform about the malfunction node to network administrators. The investigation service is applied to all sensor nodes in all zones.

2) SHARED ZONE

The shared zone, shown in Fig. 5, is the center area of the overlapping area between two or several gateways. In this area, RSSI, link quality, and other metrics values of a sensor node towards these gateways are almost similar. The handover mechanism starts when a sensor node moves to this zone and it is likely to pass by the middle line ''AB'' shown in Fig. 6, with the opposite direction towards its connected gateway.

3) WEAK ZONE

In the weak zone, shown in Fig. 5, all radio-related parameters are worse than those radio-related parameters in the personal zone and the shared zone. Fortunately, the weak zone is located in the outermost area of the coverage area. Therefore, when a node moves to the weak zone, it already passed through the shared zone where the handover mechanism is actually triggered and the sensor node is already associated with a new gateway. The weak zone is important in detecting a relative position of a sensor node in a gateway's coverage area and confirming the connection status of a sensor node. Particularly, when a sensor node located in the weak zone, a gateway, which the sensor node is used to associate with before triggering the handover mechanism, informs adjacent gateways about the disconnection by messages. When the adjacent gateways receive the messages, they will update their ''neighboring'' tables which contain the information of connections between adjacent gateways and sensor nodes. In some cases, when the overlapping area of two adjacent gateways is very small, the weak zone is used for triggering the handover mechanism. Fortunately, these cases can be avoided by properly defining zones' areas.

4) SENSITIVE ZONE

The sensitive zone shown in Fig. 5 is a special case of the weak zone. The sensitive zone is an overlapping area of weak zones of several adjacent gateways. When a sensor node located in this zone, its status is recorded in a ''sensitive zone'' table. In this case, corresponding gateways are

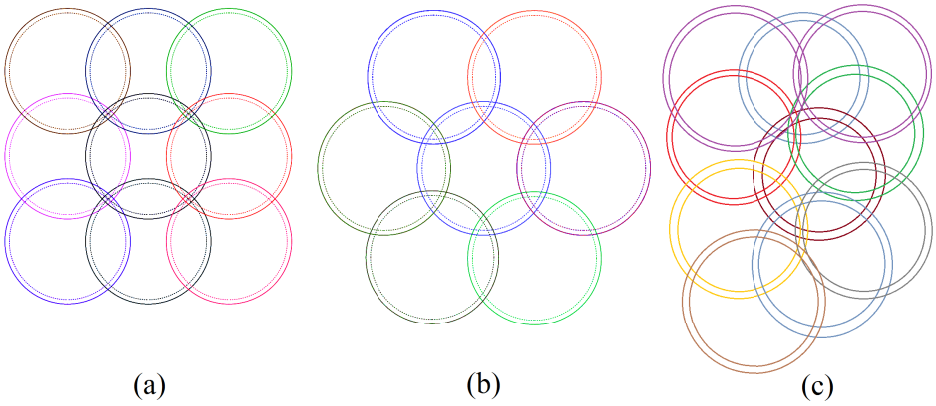

FIGURE 4. Gateway topology (a) Square topology (b) Hexagon topology (c) Random topology.

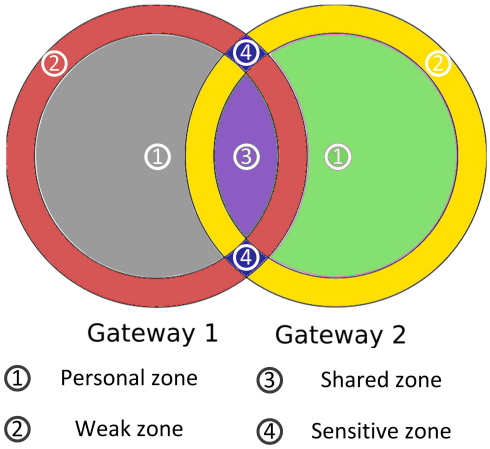

FIGURE 5. Setting up two adjacent gateways.

informed via messages by the handover service. In addition, the handover service will associate the sensor node with a gateway which the sensor node is likely to move to.

In a viewpoint of a network of gateways, adjacent gateways can be located as a square topology, a hexagon topology, or a random topology, shown in Fig. 4. In practice, a random topology is the most popular among three topologies whilst square and hexagon topologies merely occur in well-organized networks such as in a corporation or an institute. Therefore, the paper primarily focuses on a random topology. When the mobility algorithm can support mobility in a random topology, it is definitely able to support mobility in other topologies as well. However, with the purpose of providing a comprehensive view of the handover service in Fog, the handover service is evaluated with square, hexagon, and random topologies. Among the mentioned topologies, the random topology is the worst one in terms of mobility management because it has many disadvantages (e.g. undefined coverage areas and undefined overlapping areas between gateways) that do not exist in the square and hexagon topologies. Except for the information that the handover service is likely to be triggered at the shared zone in most of the cases, there is no specific pattern for triggering the handover mechanism in the random topology.

D. AREA OF GATEWAY'S ZONES

As mentioned above, each gateway has its own personal, shared, weak and sensitive zones. Depending on a particular network topology and a distance between two adjacent gateways, the area of these zones can be flexibly defined for reducing undesirable issues such as incorrect handover triggering or missing mobility events. For example, when the shared zone of two adjacent gateways is small (e.g. 1-2 square meters), a possibility of missing a mobility event may be high. In this case, a sensor node already passes through the shared zone while the system may not react in time and the handover mechanism is not triggered properly. The issues become more severe in case of an oscillating node. For example, the number of handover triggering times in such an oscillation event increases dramatically. As a result, it causes large overheads for the system performance and can cause serious problems such as missing mobility cases. For example, other simultaneous mobility cases cannot be handled properly because most of the system resources are occupied by a process of handling the oscillating node. In contrast, when the shared zone is very large, the personal zones of the gateways become smaller. Corresponding, the number of handover triggering times may increase dramatically. Therefore, it is important to specify all zones' areas precisely. These areas can be calculated by the formulas below whose parameters are shown in Fig. 6.

Angles α, β, γ, and δ in Fig. 6 are calculated by the following formulas:

$$cos(\alpha/2) = \frac{|O_1 - O_2|}{2R_1}; \quad cos(\gamma/2) = \frac{|O_1 - O_2|}{2r_1}$$
$$cos(\beta/2) = \frac{|O_1 - O_2|}{2R_2}; \quad cos(\delta/2) = \frac{|O_1 - O_2|}{2r_2}$$

where $|O_1 - O_2|$: distance between two adjacent gateways
R_1, R_2: radius of a whole coverage area of gateway 1 and gateway 2, respectively

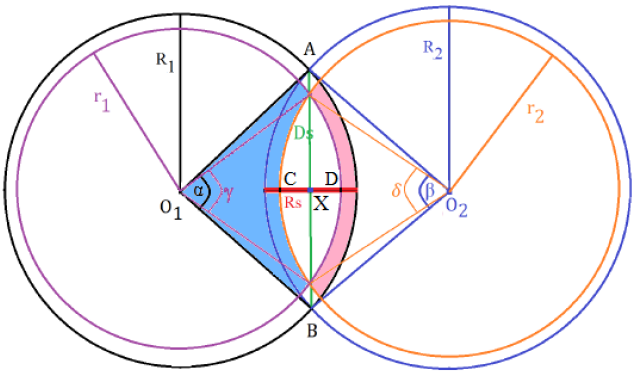

FIGURE 6. Areas of two adjacent gateways.

r_1, r_2: soft radius of coverage area of gateway 1 and gateway 2, respectively

The radius R_1 and R_2 are retrieved by scanning the maximum actual radius of coverage area of a gateway. The soft radius r_1 and r_2 are software-based values defined based on the radius R_1 and the radius R_2. For example, if the radius R_1 and the radius R_2 are 20 meters and 22 meters, the soft radius r_1 and the soft radius r_2 can be set as 18 meters and 20 meters, respectively. These soft radiuses can be flexibly defined and it is recommended that they should be slightly lesser or larger than the radius R_1 and the radius R_2.

1) AREA OF TWO DIFFERENT ADJACENT GATEWAYS

The shared zone area of two adjacent gateways (A_{Sha}) includes two adjacent parts $A_{Sha}(O_1)$ and $A_{Sha}(O_2)$.

$$A_{Sha} = A_{Sha}(O_1) + A_{Sha}(O_2) = \frac{r_1^2}{2} * \left(\frac{\pi * \gamma}{180} - sin(\gamma)\right) + \frac{r_2^2}{2} * \left(\frac{\pi * \delta}{180} - sin(\delta)\right)$$

The area of a sensitive zone (A_{Sen}) including an area of two separate zones shown in Fig. 6 is calculated by the following equation:

$$A_{Sen} = \left(\frac{\pi * \alpha}{360}\left(R_1^2 - r_1^2\right)\right) + \left(\frac{r_1^2}{2} * \left(\frac{\pi * \gamma}{180} - sin(\gamma)\right)\right) - \left(\frac{R_1^2}{2} * \left(\frac{\pi * \alpha}{180} - sin(\alpha)\right)\right) + \left(\frac{\pi * \beta}{360}\left(R_2^2 - r_2^2\right)\right) + \left(\frac{r_2^2}{2} * \left(\frac{\pi * \delta}{180} - sin(\delta)\right)\right) - \left(\frac{R_2^2}{2} * \left(\frac{\pi * \beta}{180} - sin(\beta)\right)\right)$$

Gateway 1's weak zone ($A_W(O_1)$) and gateway 2's weak zone ($A_W(O_2)$) are calculated as below:

$$A_W(O_1) = \pi * \left(R_1^2 - r_1^2\right) - \sum_{n=1}^{k} A_{Sen} - \sum_{n=0}^{k} A_{Sen\_overlapped}$$

$$A_W(O_2) = \pi * \left(R_2^2 - r_2^2\right) - \sum_{n=1}^{k} A_{Sen} - \sum_{n=0}^{k} A_{Sen\_overlapped}$$

where n: a minimal number of adjacent gateways
m: a number of Sensitive areas are overlapped
k: a number of all adjacent gateways
$A_{Sen\_overlapped}$: the area where sensitive zone's areas of gateway 1 and 2 are overlapped with sensitive zone area of gateway 1 and another adjacent gateway when there are more than two adjacent gateways.

In practice, a possibility of having $A_{Sen\_overlapped}$ is low but it may happen. Therefore, $A_{Sen\_overlapped}$ must be included in the formulas. In case that $A_{Sen\_overlapped}$ exists, its area is really small.

In order to provide detailed information related to the weak zone, an apart area of the weak zone of gateway 1 named ($aA_W(O_1)$) which is a pink area shown in Fig. 6 is calculated by the below equation:

$$aA_W(O_1) = A_{Pink} = \frac{\pi * \alpha}{360} * \left(R_1^2 - r_1^2\right) - A_{Sen}$$

Similarly, an apart area of a weak zone of gateway 2 ($aA_W(O_2)$)is calculated by:

$$aA_W(O_2) = \frac{\pi * \beta}{360} * \left(R_2^2 - r_2^2\right) - A_{Sen}$$

Personal zone area of gateway 1 ($A_P(O_1)$) and personal zone area of gateway 2 ($A_P(O_2)$) are:

$$A_P(O_1) = r_1^2 * \pi - \sum_{n=1}^{k} A_{Sha} - \sum_{m=0}^{k} A_{Sha\_overlapped}$$

$$A_P(O_2) = r_2^2 * \pi - \sum_{n=1}^{k} A_{Sha} - \sum_{m=0}^{k} A_{Sha\_overlapped}$$

where n: a minimal number of adjacent gateways
m: a number of shared areas are overlapped
k: a number of all adjacent gateways
$A_{Sha\_overlapped}$: area where a shared area of gateway 1 and gateway 2 is overlapped with a shared area of gateway 1 and another gateway when there are more than 2 adjacent gateway

In addition, an area which is a blue area in Fig. 6, is important. The area named as A_{Blue} is calculated as below:

$$A_{Blue} = r_1^2 * \frac{\pi * \alpha}{360} - A_{Sha}$$

2) AREA OF TWO IDENTICAL ADJACENT GATEWAYS

When two gateways are identical, in terms of brand and model, we have: $\alpha = \beta$; $\gamma = \delta$; $r_1 = r_2 = r$; $R_1 = R_2 = R$. The above formulas for calculating areas can be simplified:
Shared area:

$$A_{Sha} = r^2 * \left(\frac{\pi}{180} * (\gamma) - sin(\gamma)\right)$$

Sensitive area:

$$A_{Sen} = \left(\frac{\pi * \alpha}{180}\left(R^2 - r^2\right)\right) + \left(r^2 * \left(\frac{\pi * \gamma}{180} - sin(\gamma)\right)\right) - \left(R^2 * \left(\frac{\pi * \alpha}{180} - sin(\alpha)\right)\right) = R^2 * sin(\alpha) + \frac{\pi * r^2}{180} * (\gamma - \alpha) - r^2 * sin(\gamma)$$

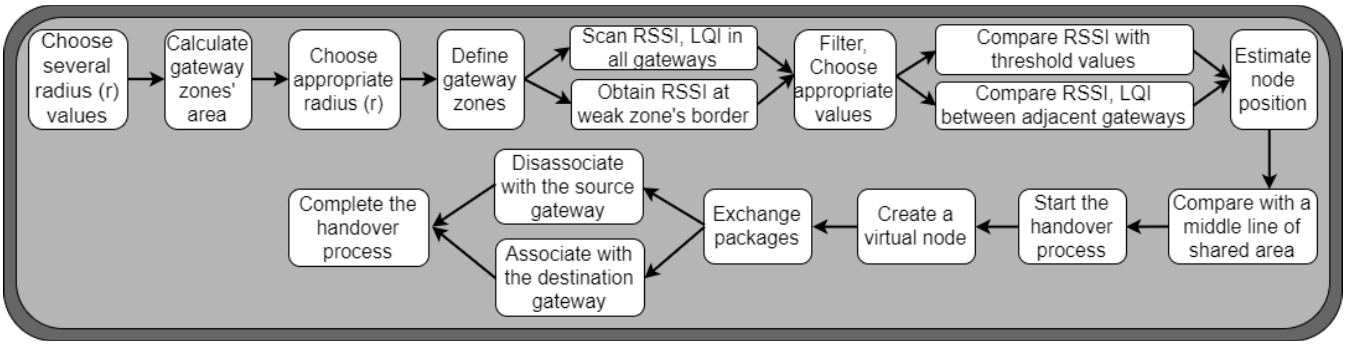

FIGURE 7. Mobility handover mechanism.

Weak area:

$$A_W = \pi * \left(R^2 - r^2\right) - n * A_{Sen} - \sum_{m=0}^{k} A_{Sen\_overlap}$$

where n: a number of all adjacent gateways
m: a number of sensitive areas are overlapped
An apart of weak area (a pink area in Fig. 6):

$$aA_W = \frac{\pi * \alpha}{360} * \left(R^2 - r^2\right) - A_{Sen}$$

Personal area:

$$A_P = r^2 * \pi - \sum_{n=1}^{k} A_{Sha} - \sum_{m=0}^{k} A_{Sha\_overlapped}$$

where n: a minimal number of adjacent gateways
m: a number of Sensitive areas are overlapped
k: a number of all adjacent gateways
A_{Blue} in this case is identical to A_{Blue} area in a general case.

$$A_{Blue} = r^2 * \frac{\pi * \alpha}{360} - A_{Sha}$$

V. THE HANDOVER MECHANISM

The main target of the proposed handover mechanism is to achieve both energy efficiency of sensor nodes and a seamless mobility with the minimized handover latency. The mechanism is based on a combination of several methods such as signal strength and quality measurement (RSSI, link connection quality), multilevel thresholds, and frame injection. Although RSSI is one of the most important metrics for a handover mechanism, it should not be used as a standalone metric. Therefore, other metrics such as link connection quality, the number of connected nodes per each gateway, a bandwidth utilization rate are used in conjunction with RSSI. Using these supplementary metrics does not cost extra overhead while they are helpful to achieve better handover decisions.

In this paper, all gateways are identical in terms of coverage area, geographical location type (indoor gateways), and specification. In case that gateways are dissimilar such as an indoor gateway and an outdoor gateway, or heterogeneous gateways from several providers, offset values must be used for precise calculations in the handover mechanism. Offset values are calculated by comparing metrics collected from those gateways in different environments and contexts. In this paper, a gateway, which a sensor node associates with and is likely to move away from, is named as a source gateway. In contrast, a gateway to which a sensor node is likely to move to is named as a destination gateway.

The proposed mobility handover mechanism flow having 16 blocks is shown in Fig. 7. In the following paragraphs, some blocks are explained in detailed whilst other blocks are briefly discussed.

A. DEFINING GATEWAY ZONES AND SCANNING RSSI, LQI IN ALL GATEWAYS

Before the gateway zones are defined, an appropriate radius (r) must be chosen because it has a significant impact on all zones' areas. When the radius (r) is larger, the weak zone will be smaller and vice versa. When the zones and their areas are not properly defined, the quality of the handover mechanism such as efficiency and preciseness can be reduced. For achieving good results, shared zone's area and personal zone's area should be large enough and these areas should be equivalent to a large portion of the whole coverage area of a gateway. These zones' areas depend on both distances, including a distance between a gateway and its weak zone's border, and distance between two the adjacent gateways. Fortunately, even in a random topology, the distance between two adjacent gateways is static and it can be measured easily. Therefore, the distance between a gateway and its weak zone's border is considered. To find an appropriate distance, equations presented in section IV are applied. Results from the formulas provide some piece of evidence (e.g. a ratio of the shared area and the whole coverage area of a gateway) for finding several most suitable candidates.

B. OBTAINING RSSI AT WEAK ZONE'S BORDER, FILTERING AND CHOOSING APPROPRIATE VALUES

When the distance between a gateway and its weak zone's border is decided, threshold values (e.g. RSSI) at the weak zone's border can be obtained via the scanning method.

In order to avoid corrupted values during the scanning, the gateway scans 10 times and chooses appropriate values.

C. COMPARING RSSI WITH THRESHOLD VALUES, COMPARING RSSI, LQI BETWEEN ADJACENT GATEWAYS AND ESTIMATING NODE POSITION

Moving sensor nodes are regularly checked by comparing a set of metrics values such as RSSI and threshold values of the associated gateways. These values are obtained via scanning processes. A scanning interval between scanning processes can be flexibly defined or edited depending on particular applications. In our application, a short interval is preferred for enabling fast response to movements of sensor nodes. In the proposed mechanism, all gateways perform the scanning process simultaneously for achieving high accuracy in estimating the position of sensor nodes. In order to avoid corrupted data, each scanning process has 3 scanning rounds without any delay between the rounds. Results from the scanning process are filtered and stored in a scanning table of the gateway. Values belonging to the same category from the scanning table are compared with each other and with values from the previous scanning process. Inappropriate values are possibly eliminated. The filtered data is multicasted to adjacent gateways which the gateway shares some overlapping areas. Correspondingly, each gateway has several RSSI and LQI values from its own scanning and adjacent gateways' scanning after each scanning process. These values are used for estimating the position of sensor nodes. For example, each gateway has its own weak zone having RSSI values from -75 dBm to -65 dBm. If RSSI values of a sensor node measured by two adjacent gateways are -55 dBm and -62 dBm, then the sensor node must be in the shared zone of these two gateways, being closer to the gateway having -55 dBm from the scanning.

D. COMPARING WITH A MIDDLE LINE OF THE SHARED AREA AND STARTING THE HANDOVER PROCESS

The handover process is triggered when a sensor node located in the shared area of two adjacent gateways and it is likely to pass through the middle line of the shared area, see Fig. 6, line AB. The middle line's RSSI values are set when RSSI values of the sensor node towards these adjacent gateways are equal. During the handover process, the data in the database of the source gateway is also sent to the destination gateway. During mobility, when a sensor node is still associated with the source gateway, the collected e-health data is sent to the source gateway which immediately forwards the data to the destination gateway. This method helps to avoid missing data during mobility.

As mentioned, link quality also plays an important role in the handover mechanism and quality of service. If link quality is worse than some pre-defined requirements (e.g. 70%), Fog sends a notification to a system administrator. Depending on particular gateways and the condition of the surrounding context (e.g. interference), the pre-defined LQI requirements can be different. However, it is recommended that LQI value should be high for achieving a high level of QoS.

E. CREATING A VIRTUAL NODE, DISASSOCIATING AND ASSOCIATING WITH THE SOURCE AND DESTINATION GATEWAY, RESPECTIVELY

For maintaining the connection between a sensor node and its system network during mobility, the moving sensor node must be deregistered by the source gateway and registered by the destination gateway because a node cannot be associated with more than one gateway. In order to perform these tasks, an advanced method of creating a virtual node is used. As mentioned, the handover mechanism is triggered when a moving node is in the shared area of two gateways. Correspondingly, the MAC address of the moving node can be collected by these gateways. Based on the MAC address, the destination gateway creates a virtual node which is used as a representative of the moving node. The virtual node registers itself with the destination gateway by exchanging messages described in Section IV. In the proposed handover mechanism, the exchanging of messages is performed via the packet injection method. Particularly, the virtual node starts by injecting probe request packets and it waits for the probe response packet. When it receives the response packet, it continues to inject other packets (i.e. authentication request, association request). Depending on a Wi-Fi configuration, the number of exchanged messages varies. Importantly, during the registration of the virtual node with the destination gateway, the moving node maintains its registration with the source gateway. Therefore, all data sent by the moving node can be collected without any interruption and the handover latency is minimal. When the virtual node has just been registered with the destination node, the moving node is simultaneously deregistered from the source gateway. As a result, the moving node is already registered with the destination gateway and it can transmit data to the destination gateway without any delay.

F. OSCILLATION EVENT HANDLING

During mobility, oscillating event is always considered via a mechanism that checks the disassociating and associating time. When the time periods of the most two recent events are short and less than a pre-defined threshold, the sensor node is detected as a pre-oscillating node. When the handover mechanism confirms that the pre-oscillating node only moves in the shared area, the sensor node is detected as an oscillating node. The pre-defined threshold can be set based on shared zone area of two gateways and the movement speed of the sensor node. For example, if a shared zone area of two gateways is about 10 m^2, a distance (CD line in Fig. 6) should be about 3.5 m. In addition, the sensor node attached to a patient often moves with an average speed less than 2 m/s. Based on the information, the threshold value can be approximately 3 seconds. Correspondingly, for maximum distance which a sensor node can move is 3 m (i.e. this is a multiplying result of 1.5 s and 2 m/s) for a single way from the source

gateway to the destination gateway. This threshold can be flexibly changed depending on particular applications. For example, in other environments such as a factory where sensor nodes attached to vehicles can move with a faster speed, the threshold value should be smaller.

When an oscillating node is detected, the handover mechanism compares several parameters of two adjacent gateways including information in oscillation tables, RSSI tables and RSSI values of a line AB shown in Fig. 6. For example, the handover uses line AB as a vertical border of two gateways in this case. If a sensor node located on the left side of the border in a longer time period than the right sight of the border, the left gateway will be chosen for remaining the association with the sensor node. It is unnecessary to perform the handover mechanism if the sensor node moves within the shared zone and remains the moving pattern.

VI. TESTBED SETUP

For evaluating mobility support and other services of Fog, the system architecture shown in Fig. 2 was implemented. The system consists of medical sensor nodes, smart e-health gateways with Fog computing, a remote server and end-user terminals (e.g. mobile applications and browsers).

In the implementation, four setups, shown in Fig. 8, based on square, hexagon and random topologies are applied. In the first three setups (shown in Fig. 8(a), Fig. 8(b) and Fig. 8(c)) 6 gateways are deployed while in the fourth setup Fig. 8(d), 7 gateways are deployed. It is noted that a gateway "G6" is not shown in Fig. 8(b) due to the limited width of the presented paper. By adding one more gateway (G7) into the network shown in Fig. 8(d), the network becomes much more complex and the number of shared areas between adjacent gateways increases dramatically. This setup is used for evaluating the efficiency of the handover mechanism in a dense network where several areas of many gateways are overlapped.

In each experiment, 5 sensor nodes are used in which two of them move freely without any pattern from a gateway

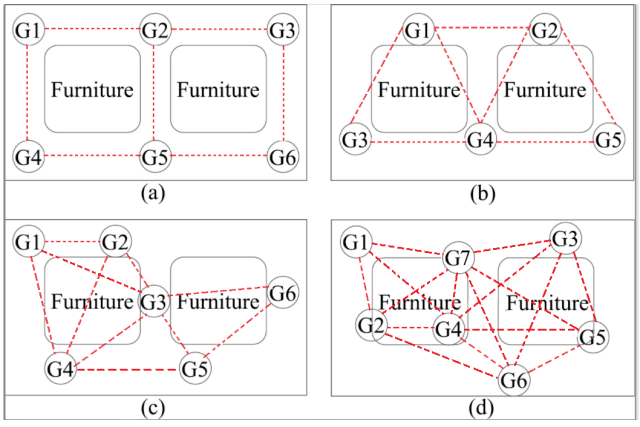

FIGURE 8. Gateway placement in a room (a) Square topology (b) Hexagon topology (c) Random topology (setup 1) (d) Random topology (setup 2) (- -) indicates adjacent gateways having overlapping areas.

to another simultaneously. With the purpose of analyzing impacts of the software-based radius (r) and the distance between gateways on the proposed algorithm, two groups of configurations (i.e. group 1 and group 2) are applied. The first group including 3 configurations (i.e. Conf 1(a), Conf 1(b) and Conf 1(c)) is applied to both square and hexagon topologies while the second group including 12 configurations (i.e. from Conf 2(a) to Conf 2(l)) is applied for the random topology. These configurations help to reveal the relationship between several parameters (i.e. the software-based radius (r), the actual radius (R), the distance between adjacent gateways) and areas of gateways zones.

The radius of the coverage area of a gateway in practice can be about 17 m or a bit further (e.g. 25 m) depending on particular gateways. In the experiments, an actual radius of coverage area of gateways is around 18 m. However, it is difficult for achieving the same experimented environment (e.g. noise, interference and wireless transmission conditions) when deploying many gateways with their actual coverage areas. Therefore, three parameters including the actual radius (R) of the whole coverage area of a gateway, the software-based radius (r) counting from a gateway to its weak zone's border and the distance between two gateways are scaled down three times. In the experiments, the radius (R) is 6 m after scaling down. The RSSI values and other radio-related parameters (e.g. LQI) applied in the experiments are measured according to these scaled points and distances.

VII. IMPLEMENTATION

For performing experiments, a complete remote real-time health monitoring IoT system is built. The system consists of several sensor nodes, Fog-assisted smart gateways, a Cloud server, and an end-user terminal. The implementation of the system is presented as below

A. SENSOR NODE IMPLEMENTATION

This work focuses on mobility support. Therefore, sensor nodes are built from general purposes devices. In our implementation, two sets of devices are used as sensor nodes. The first set includes Arduino Mega [64], ADS1299 [44], ESP8266 [65] and sensors. Arduino Mega is equipped with 16 MHz ATmega1280 micro-controller, 8 Kb SRAM, 4 Kb EEPROM and 128 Kb Flash memory. ADS1299 is a low-noise and multichannel device produced by Texas Instruments for acquiring medical data (e.g. ECG, EMG, EEG) with a high data rate up to 16k samples per second per channel. ADS1299 enables scalable medical systems with small size, low power, and a reasonable overall cost. ESP8266 is a low-cost Wi-Fi chip with a full TCP/IP stack. Several medical and environmental sensors such as SpO2, heart rate, temperature and humidity sensors are utilized. Integration of these devices creates a sensor node capable of acquiring data (multi-channel ECG, medical signals, and contextual data) with high data rates and transmitting the data in real-time to the smart gateway via Wi-Fi. However, this

set is only used in the final experiment for showing real-time ECG data during mobility. Another simple set consisting of Arduino Uno [66] and Wi-Fi shield [67] is used as a sensor node in most of the experiments to reduce complexity.

Arduino Uno is equipped with 16MHz ATmega328P, 2Kb SRAM, 1Kb EEPROM and 32Kb SRAM. It generates data and transmits the generated data to the gateway with high data rates via the Wi-Fi shield. Correspondingly, the mobility support capability could be successfully verified by comparing generated data with data received from an end-user browser. In this paper, the node is set up to merely perform the primary tasks of collecting and sending data to the smart gateway while other tasks such as data processing, data analytics, and mobility support are implemented in the Fog layer of smart gateways.

B. GATEWAY IMPLEMENTATION

A smart gateway [10] consists of Pandaboard [63] and a 300Mbps wireless USB adapter [68]. The Pandaboard is low cost, low power platform based on OMAP4430 processors. Pandaboard is equipped with a dual-core 1.2 GHz CPU, 384 MHz GPU, Ethernet, wireless chip-set (Bluetooth and 80211), and a set of I/O ports. In addition, the board supports up to 32GB SDHC card. Correspondingly, different operating systems (Windows, Ubuntu, Android) and databases (MySQL, MongoDB, PostgreSQL) can be installed in the Pandaboard for managing the gateway and enhancing gateway's services.

In our implementation, the MySQL database is used for storing medical and context data received from sensor nodes and recording vital information used for Fog services such as the push notification service and visualization of real-time ECG data. Accordingly, the database is persistently maintained and real-time updated.

Furthermore, due to the limited capacity of the distributed database, it is purged every 30 minutes after receiving a confirmation of the synchronization from Cloud. As mentioned, the gateway with its distributed database can act as a local web server when the connection between Fog and Cloud is disrupted. In this case, when the distributed database runs out of available storage capacity, the new incoming data overwrites the old one. Fortunately, in general, the disconnection usually does not last for a long period of time because when the disconnection occurs, the push notification service is triggered to inform network administrators in real-time. While the gateway acts as a local web server, it will send responses either in XML or JSON format as requested and leave all rendering tasks to the client.

We implemented a parallel notification method in both Fog and Cloud. In general, the notification service is primarily implemented in Cloud whilst in a few cases, it is run at Fog. By applying this method, all emergency cases can be notified whereas it does not cost significant resources in Fog. For implementing the notification service on a client-side, an Android application, which can communicate with both smart gateways and Cloud, is developed. When the notification service is triggered, the Android application receives real-time push-messages. In addition, a web browser can be also used as a client for visualizing real-time e-health data.

In our implementation, the Ubuntu operating system is used in smart gateways because Ubuntu not only manages hardware resources and Fog services but also provides useful daemon services, libraries and applicable tools such as the firewall. For example, Uncomplicated Firewall (UFW) [69] in Ubuntu can be used for constructing accessibility rules such as protocols blocking and ports blocking. In our implementation, all unnecessary ports and protocols are blocked except for ones used by Fog services. However, applying firewall does not guarantee a high level of security. We also applied an end-to-end security scheme for healthcare IoT mobility proposed in [70].

In our implementation, all gateways are configured to have the same Service Set Identifier (SSID). Thanks to this setup, the configuration of sensor nodes is kept intact during mobility. Correspondingly, high power consumption and latency caused by reconfiguring sensor nodes during mobility can be partly avoided. In the paper, RSSI and LQI are periodically collected via a scanning method which is constructed by utilizing *iw, iwlist, iwconfig* packages and API provided in Ubuntu OS.

In order to construct virtual nodes, 300 Mbps wireless USB adapters are used in the system in which each adapter is attached to a gateway. Due to the simple configuration of the adapters, it is not challenging to integrate these adapters into the system. When receiving instructions from the handover mechanism, the adapter at a destination gateway acts as a representative of a virtual node by utilizing its actual hardware for performing registering tasks mentioned earlier. In our implementation, registration between a virtual node and a destination gateway is performed by a packet injection method. We implemented the method with the Libtins library [71] which is a high-level, multi-platform C++ network packet sniffing and crafting library. The library is open-source and supports popular protocols such as IEEE 802.11, IEEE 802.3, IEEE 802.1q, Ethernet, ARP, IP, IPv6UDP, and TCP. In addition, the library is reliable because it has been tested with 624 unit tests.

There are two approaches of utilizing the actual hardware (adapter) for implementing packet injection during node registration. In a simple approach, a mobility buffer with the first-come-first-served strategy or a mobility buffer with an arbiter can be used. When a node is detected as a moving node, it is added to the mobility buffer and waits for its turn. In case of using the first-come-first-served strategy, the first moving node is always given the right to use the actual hardware. When the buffer is used with the arbiter, the higher priority node has the right to use the actual hardware. In our implementation, zero is the highest priority. A priority of a node is decided by the arbiter via a mechanism based on time and RSSI. Accordingly, when the RSSI value of the second node is less than a pre-defined threshold,

the second node is set with the highest priority and it is given the right to use the actual hardware. Although the mobility handling in these methods is based on the mobility buffer, it is possible to support mobility for approximately 10 moving nodes while fulfilling latency requirements of real-time health monitoring. For a complex approach, instead of using the mobility buffer, threading (multi-threading) is applied. The advantages of this method are asynchronous and non-blocking behavior. Each sensor node is handled by a single thread. Correspondingly, it is possible to handle mobile health monitoring for the large number of moving nodes simultaneously. However, it is difficult to deal with debugging. For testing and assessing QoS (mobility support), we implemented the second approach using multi-threading.

VIII. EVALUATION

In this section, several experiments have been carried out. These experiments are explained in details as follows:

A. GATEWAY'S SIGNAL LEVEL

In the experiments, RSSI and LQI of a sensor node in different positions towards a single gateway are measured. Each distance has been experimented with for 10 times and average values are reported. All of these experiments are carried out in the same environment (i.e. located in the same single warehouse's room and affected by the same interference noise). Results are shown in Fig. 9. The results indicate that the RSSI and LQI values do not decrease linearly when the distance between the sensor node and the gateway increases linearly. When the sensor node is far away from the gateway, the RSSI and LQI values are low. Therefore, it is recommended that RSSI and LQI values of a gateway's coverage border must be measured. If the values at the border show low quality of signals (e.g. LQI less than 60%), system administrators need to use a lower value for the radius R to avoid low-quality signals and transmission loss. In our experiments, when LQI is less than 60%, there are lost packages in transmission. Therefore, 60% LQI is used as the threshold for defining the radius R. For instance, if the actual radius R of the coverage area is 18 m and the LQI value at the border is less than 60%, the radius R used in the configurations and the handover mechanism should be around 16 m for achieving 70% LQI. The threshold values (e.g. 60% LQI) are flexibly defined by system administrators depending on particular applications and environments.

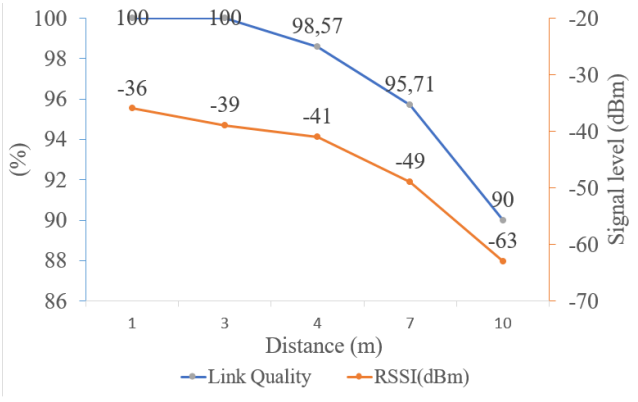

FIGURE 9. Gateway's signal level and link quality.

B. IMPACT OF MOBILITY SUPPORT AND FOG SERVICES ON THE SYSTEM LATENCY

Based on our knowledge, the current state-of-the-art real-time continuous e-health monitoring IoT systems based on Fog computing do not support mobility completely. Therefore, we would like to propose the IoT system with fully mobility support based on Fog computing. Although it is unfair to compare between the systems with and without the mobility support, it is valuable to provide an overview of the impact of the proposed algorithm on latency.

We evaluated the impact of the mobility support and Fog services on the system latency by constructing a health monitoring IoT system based on Fog computing. In details, the system is setup with three different cases. The first one is a typical IoT system without mobility support. In this configuration, each gateway has a distinct SSID. The second configuration is similar to the first one except that all gateways have the same SSID and overlapping areas. The third configuration is an upgraded version of the second one with the mobility support service. In all cases, only the first three setups shown in Fig. 8 are applied. For fair comparisons, metrics (radius R, software-based r, distance between gateways) in each case are the same. There are 10 experiments in each case and average values are reported. The results of handover latency are shown in Fig. 10. Results indicate that the proposed system with the handover mechanism for complete mobility support reduces the system latency for reestablishing/remaining the connection between sensor nodes and a gateway during mobility dramatically. The handover mechanism helps to save approximately 98% and 95% comparing to the system without mobility support

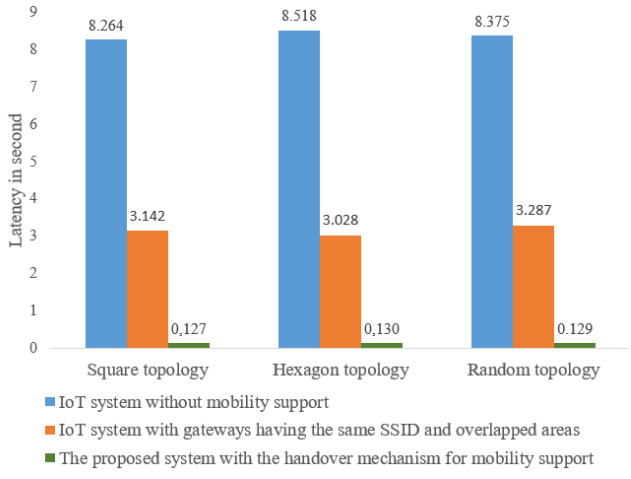

FIGURE 10. Latency of IoT systems with different configurations.

TABLE 1. Area of gateway zones in different configurations in case of a single adjacent gateway.

	R (m)	r (m)	Distance between two gateways (O_1O2) (m)	Shared zone (A_{Sha}) (m^2)	Sensitive zone (A_{Sen}) (m^2)	Weak zone (A_W) (m^2)	A_{Pink} area (i) (m^2)	Personal zone (A_P) (m^2)	A_{Blue} area (ii) (m^2)
Conf 1(a)	6	5.25	8	11.635	1.048	25.458	6.048	74.954	11.5467
Conf 1(b)	6	5.25	9.5	3.0110	1.60365	24.9035	3.9419	83.5791	15.1048
Conf 1(c)	6	5.5	8.25	13.7126	0.4870	17.5770	4.1862	81.3204	10.8731
Conf 2(a)	6	5	7.5	11.332	1.677	32.879	8.174	67.2070	11.0588
Conf 2(b)	6	5.25	7.5	15.1757	0.9303	25.5768	6.6268	71.4143	9.5109
Conf 2(c)	6	5.5	7.5	19.4680	0.4084	17.6556	4.7416	75.5651	7.6257
Conf 2(d)	6	5.75	7.5	24.1987	0.1010	9.1273	2.5299	79.6701	5.4141
Conf 2(e)	6	5	8.25	6.7179	2.0263	32.5311	6.9139	71.8218	13.6009
Conf 2(f)	6	5.25	8.25	9.9727	1.1157	25.3914	5.7418	76.6173	12.4288
Conf 2(g)	6	5.5	8.25	13.7126	0.4870	17.5770	4.1862	81.3204	10.8731
Conf 2(h)	6	5.75	8.25	17.9172	0.1199	9.1085	2.2675	85.9516	8.9544
Conf 2(i)	6	5	9	2.9362	2.51722	32.0402	5.4328	75.6035	15.1320
Conf 2(j)	6	5.25	9	5.4906	1.3675	25.1396	4.7305	81.0995	14.4297
Conf 2(k)	6	5.5	9	8.599	0.5912	17.4728	3.5644	86.4341	13.2636
Conf 2(l)	6	5.75	9	12.2176	0.1444	9.0839	1.9785	91.6512	11.6777

(i) pink area and (ii) light blue area in Fig. 6
total coverage area of a gateway in call cases: 113.097 m^2

and the system in the second configuration, respectively. In addition, the system latency in the second configuration is equal to a half of the latency of the first configuration. Therefore, it is recommended that if the system cannot be configured or equipped with the handover mechanism, its gateways should be configured to have the same SSID to reduce the system latency for reestablishing the connection between sensor nodes and gateways. In addition, results show that a topology type such as square, hexagon, and random topology does not affect the handover latency. In these experiments, a few cases of abnormal values have occurred. For example, handover latency in a hexagon topology in the first case once reached up to 10.42 s. Some of the reasons for having such a high latency are that some exchanged packages may be lost or incoming packages at the gateway are corrupted during handshaking. Corresponding, the sensor node and the gateway must send many packages for handshaking that causes the increase of latency. Although abnormality does not occur in other cases in the experiments, this issue may happen anytime in any cases. In the experiments, the surrounding noise sources are not considered. However, all of the experiments are done in the same warehouse room. Therefore, these results are all affected by the same surrounding noise sources. The effects of the noise sources are much larger if the noise sources have similar frequencies as the sensor nodes.'

C. THE RELATIONSHIP OF THE SOFTWARE-BASED RADIUS R, AREAS OF DIFFERENT ZONES AND THE DISTANCE BETWEEN ADJACENT GATEWAYS

As mentioned, 2 groups of different configurations are applied in the experiments. Areas of gateways' zones in each configuration are calculated based on the formula set presented in Section IV and results are shown in Table 1. The results reveal the information of the relationship of the software-based radius r and areas of different zones and the distance between adjacent gateways. Particularly, results from the first three configurations of group 1(i.e. from Conf 1(a) to Conf 1(c)) provide some general information about the relationship. Based on the results from these configurations, we know that these configurations might not be the most optimal because the shared zone areas are very small. Currently, there are no specific requirements for zones' areas. Depending on particular applications, zones and their areas can be flexibly set. However, it is recommended that shared zones' areas should be large enough for avoiding the missed handover triggering cases while the personal area should be large for avoiding overheads of utilizing resources for unnecessary handover triggering. Based on our experiments and results discussed in the following paragraph, it is recommended that the shared zone's area should be greater than a value calculated by a formula: $value = speed\_of\_sensor\_node^2 * 2/3$.

We carried out more than 100 experiments for achieving the requirements of minimum areas of the shared zone. The random topology shown in Fig. 8 is applied for these experiments. In these experiments, there are 10 different cases and each case is carried out for 10 times. In each case, a sensor node moves freely from a gateway into an adjacent gateway. As mentioned, the handover mechanism is triggered in a shared zone and it relies on the scanning interval. In order to have correct and fair measurements, several parameters including short scanning interval of 0.1 s, distance of 8 m between two adjacent gateways, radius (R) of 6 m, and movement speed of 5m/s are applied for all experiments. The recommended minimum value of personal zone's area calculated by the formula above is 16.6666 m^2. The shared zone area is changed by increasing the software-based radius (r) a value of 0.1 m starting from 5 m. Correspondingly, the latter case has a larger shared zone area than the previous case. Results of these experiments are shown in Table 2.

Results from Table 2 show that when the shared area is larger than the recommended minimum area of the personal

TABLE 2. Experimental case.

Experimental case	r (m)	Shared zone (A_{Sha}) (m^2)	Weak zone (A_W) (m^2)	Sensitive zone (A_S) (m^2)	Personal zone (A_P) (m^2)	total shared zone and pink area (i)	Triggering times in shared zone vs in pink zone (i)	Missed Triggering time in both shared amd pink zones (i)
Case 1	5	8.1750	30.7601	1.8987	62.1897	15.528	4-6	0
Case 2	5.1	9.5012	28.3325	1.5259	62.7102	16.3774	5-5	0
Case 3	5.2	10.9048	25.7546	1.1969	63.1388	17.2437	6-4	0
Case 4	5.3	12.3843	23.0295	0.9102	63.4785	18.1269	7-3	0
Case 5	5.4	13.9385	20.1594	0.6645	63.7317	19.0269	7-3	0
Case 6	5.5	15.5662	17.1467	0.4587	63.9006	19.9436	8-2	0
Case 7	5.6	17.2666	13.9931	0.2919	63.9870	20.8772	10-0	0
Case 8	5.7	19.0388	10.7003	0.1633	63.9925	21.8276	10-0	0
Case 9	5.8	20.8822	7.2697	0.0722	63.9186	22.7948	10-0	0
Case 10	5.9	22.7960	3.7025	0.0179	63.7666	23.7789	10-0	0

(i) pink area and (ii) light blue area in Fig. 6
total coverage area of a gateway in call cases: 113.097 m^2

zone, all mobility events are triggered in the personal zone. In contrast, when the share zone area is smaller than the recommended minimum area, some mobility events are triggered in a pink zone which is apart of the weak zone shown in Fig. 6. Although there is no difference between triggering the handover mechanism in the shared zone and in the pink zone in terms of handover latency, triggering the handover mechanism in the shared zone is still expected especially in case of low quality of signals (e.g. LQI and RSSI). In these cases, the handover mechanism may not be triggered correctly in the weak zone whilst it is highly possible to trigger the handover mechanism correctly in the shared zone because the quality of signals in the shared zone is often high due to the close proximity to the gateway. In addition, the shared zone is the main zone for triggering the handover mechanism while the weak zone can be used as a backup zone for the handover mechanism. For example, when a mobility case is missed in the shared zone, there is still a chance and time for triggering the handover mechanism in the weak zone.

D. VERIFYING THE ACCURACY OF THE HANDOVER MECHANISM

For verifying the accuracy of the handover mechanism, some complex cases of mobility are applied. In these cases, a sensor node moves from the personal zone of a source gateway to the weak zone of a destination gateway with a direction of 60 degrees counterclockwise measured from the line-of-sight line between these gateways. In the moving path, the sensor node passes the middle line of the shared zone of two gateways. However, the duration and path distance which the sensor node has been located in the right part of the shared zone is very short. Due to the short scanning interval (i.e. 0.1 s - 0.3 s). This case is detected and the handover mechanism is triggered successfully. Another tough case is a case that a sensor node moves within the sensitive zone and it passes the middle line of this zone. In our experiments, this case (i.e. moving within the sensitive zone) have not been experimented with the handover mechanism because these sensitive zone areas shown in Table 2 are very small. Based on our experiments, a case that a sensor node moves within the sensitive zones of two adjacent gateways is seldom and it can be avoided when setting up the configuration suitably. In this case, the sensitive area is often really small (e.g. less than 1 m^2). In critical applications, a new gateway can be added in between these adjacent gateways for avoiding mentioned "tough" cases. Although this method is not recommended due to wasting resources, it helps to avoid tough cases above because the sensitive zones will be overlapped with the shared or the personal zone of the new gateway.

In case of Conf 2(a) and Conf 2(d), when the distance of gateways is 25% larger than the radius R, and the software-based radius r increases about 12.5%, the shared zone area and the personal zone area increase about 11.3% and 11%, respectively whilst the weak zone area and the A_{Blue} area decrease 21% and 5%, respectively. Results of the comparison between Conf 2(d) and each of two configurations (i.e. Conf 2(b) and Conf 2(c)) have the same pattern as the comparison one from Conf 2(d) and Conf 2(a). In case that the distance between two gateways is about 25% larger than the radius R and the gateway only has an adjacent gateway sharing some overlapping areas, Conf 2(d) is better than Conf 2(a) because the shared zone and the personal zone of Conf 2(d) are larger. There are no specific requirements for the shared zone areas and other zones' areas. Depending on particular applications, the shared zone area is differently set. For example, the shared zone area should be large for applications in which the sensor node moves with a high speed. Based on the above experiments, a large personal area and a small weak zone area together with an appropriate shared zone area are the most suitable option for the handover mechanism. When the personal zone is small, a possibility to trigger the handover mechanism during a movement of a sensor node is higher. The most important target of the system is to keep the connection between sensor nodes and the system while reducing the number of handover triggering times as much as possible.

Results from the comparison between different pairs in a group of 4 configurations (i.e. Conf 2(e), Conf 2(f), Conf 2(g), Conf 2(h)) have the same pattern as the results from the group of Conf 2(a), Conf 2(b), Conf 2(c) and Conf 2(d), respectively. Similarly, it is valid for pairs of another group (i.e. Conf 2(i), Conf 2(j), Conf 2(k), Conf 2(l)). It can be inferred that small changes in a software-based radius r can cause dramatically impacts on different zones' areas. It can be concluded the shared zone area increases and the weak zone area decreases when increasing the software-based radius r regardless of a distance between two gateways.

Results from Conf 2(d), Conf 2(h) and Conf 2(l) indicate that when the distance between two gateways are smaller and the same software-based radius r is used, the shared zone area increases and the personal zone area decreases. Although the conf 2(l) is the best configuration in its group (i.e. Conf 2(i), Conf 2(j), Conf 2(k), and Conf 2(l)), it is still not the optimal configuration because its shared zone area is still small (i.e. 12.2176 m^2). As mentioned, the handover mechanism is triggered when a sensor node passes the middle line AB of the shared area. It implies that the system only has about 6 m^2 to complete the handover mechanism. In this case, if the movement of the sensor is high (e.g. 8 m/s), the sensor node only needs about 250 ms to pass the shared area. In most cases, the handover latency is less than this time. However, in some special cases (e.g. many lost packages), the latency of the handover mechanism may be higher. Therefore, depending on the distance between gateways, the software-based radius r should be carefully chosen. Among all configurations, Conf 2(d) seems to be the best one since its shared zone area is large while its personal zone area still occupies the large portion of the whole coverage area of a gateway.

E. LATENCY, RELIABILITY EVALUATION AND COMPARISON WITH OTHER STATE-OF-THE-ART WORKS

For evaluating the latency of data synchronization between distributed databases, various data sets having different sizes are applied. The result shown in Table 3 displays that the synchronization latency is low in most of the cases and it is not proportionally linear with regard to data size. Correspondingly, the latency of data synchronization does not have a significant impact on the total latency of the system during mobility.

TABLE 3. Handover latency of transmitting data between distributed database of gateways.

Data size (Byte)	Latency (ms)
10	2.176
50	2.321
100	2.787
500	3.275
1000	3.524
5000	4.462

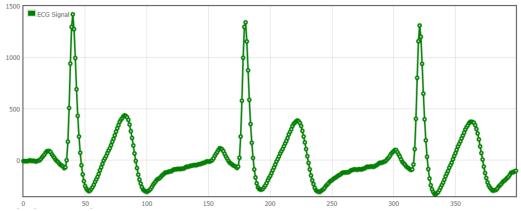

FIGURE 11. Graphical ECG waveforms at a remote browser.

For assessing the reliability of the handover mechanism, more than 50 mobility cases including node oscillation are tested. In most of the cases, the connection between sensor nodes and the system is maintained without any interruption. Fig. 11 shows ECG waveforms at an end-user browser when connecting to Fog's web service during mobility. A user is a 30 year-old male volunteer. The result shows that real-time monitoring with the high quality of signals can be guaranteed with Fog's services during mobility. In rare cases, the push notification is triggered when the handover mechanism cannot handle unexpected situations.

Finally, we compared our proposed method for mobility support with the recent state-of-the-art works for mobility support. Results are shown in Table 4. Results show that our method is the most efficient in terms of handover latency among all mentioned works and the proposed handover method does not cause overloads of sensor nodes. Correspondingly, sensor node's battery cycle time does not decrease. In addition, our method concerns oscillating nodes during mobility whilst others do not consider that attentively.

IX. DISCUSSION

For detecting a position of a particular sensor node, several adjacent gateways have to scan the RSSI values and exchange the collected values with each other. This may over-utilize network bandwidth. Fortunately, the system performance does not decrease due to a large network bandwidth (40 Mbps - 54 Mbps) of Wi-Fi.

The paper primarily focuses on mobility support and the handover mechanism. Therefore, for reducing complexity, several software packages provided in Ubuntu (e.g. "iw," "iwlist" and "iwconfig") are applied for scanning RSSI and LQI. However, these packages are not optimal in terms of latency for scanning RSSI. In some cases, when the number of sensor nodes including both sensor nodes belonging and not belonging to the system is numerous, results from scanning parameters from sensor nodes will be large and some of the nodes may not appear in the result list. We recommend that other state-of-the-art methods for obtaining RSSI and LQI should be used.

In general, the method of injecting wireless packages may cause some severe issues related to security and gateways' performance if it is misused. For example, by using the package injection method, the Wi-Fi network can be hacked. In details, a hacker can use a Wi-Fi-based device for scanning MAC address of other devices using Wi-Fi around

TABLE 4. A comparison between available handover mechanisms for health monitoring systems.

Approach	Handover management	Movement type	Metrics in Handover mechanism	Oscillating node management	Handover latency (ms)
Valenzula *et al.* [38]	Hybrid (focus on node side)	Random	RSSI	No	moderate(*)
Jara *et al.* [39]–[41]	Network-based handler	Random	RSSI and movement direction	No	173.5
Fotouhi *et al.* [42]	Hybrid-based handler (focus on node side)	Random	RSSI, velocity,hop number, traffic load, energy level, and LQI	No	130-200
Silva *et al.* [37]	Network-based handler	Random	LQI	No	>220
Our proposed mechanism	Network-based handler	Random	RSSI, LQI, velocity	Yes	127.5

(*) Information related to handover latency cannot retrieve from the paper. But the handover latency should be moderate since many messages/packages must be exchanged during the handover process [38]

his/her geographical location. Then, the hacker can use found MAC addresses for injecting several types of packages to the network. Correspondingly, there are two severe consequences. Firstly, the network channels are fully occupied. Therefore, other devices cannot connect or associate with the network gateway. Secondly, if a hacker injects disassociation packages, gateways will disassociate with real devices. Then, for re-establishing the connection with the network, the real devices have to re-associate with gateways via exchanging packages. At this moment, the hacker will be a man in the middle to monitor all packages exchanged between the real devices and the network gateways. In our experiments, we occasionally tracked information of adjacent Wi-Fi-based devices and we are able to collect their transmitted data.

If the injection method does not precisely inject packages in time, there is no guarantee that the connectivity between a sensor node and smart gateways is maintained with low latency even though the handover mechanism is successfully triggered. Therefore, it is recommended that the injection method has to be carefully designed and implemented.

In practice, the measurements related to latency are mostly relative because they rely on different parameters such as network channels, interference of different radio sources, and transmission conditions. Similarly, applying the proposed handover method in different places may provide different handover latency. Therefore, a network administrator needs to consider mentioned parameters for achieving a high quality of service.

Although the proposed handover mechanism does not intensively use broadcasting, it often uses multicasting between adjacent gateways. When the number of connected devices is large (e.g. 100 devices) and they are moving simultaneously, the system performance may decrease.

It can be seen in Fig. 10 that the handover mechanism and the system latency for maintaining the connection between sensor nodes and gateways are not dependent on the network topology. In addition, the hexagon topology provides the largest coverage areas among all mentioned topologies when the same number of gateways and the same configuration are applied. Based on our experiments, the hexagon topology is the best option for setting up a new network of gateways. Although is difficult to build such a hexagon network in practice, it is recommended that network administrators should apply the hexagon topology if it is possible.

X. CONCLUSION

In the paper, we proposed the handover mechanism for complete mobility support in a remote real-time streaming IoT system. The handover mechanism helped to remain the connection between the sensor nodes and the system with the low latency. The handover mechanism also attentively considered oscillating nodes which often occur in many streaming IoT systems. By leveraging the strategic position of smart gateways and Fog computing in a real-time streaming IoT system, sensor nodes' loads were alleviated whereas advanced services (e.g. push notification and local data storage) were provided. The paper discussed and analyzed popular metrics for the handover mechanism based on Wi-Fi. In addition, the complete remote real-time e-health monitoring IoT system was implemented for experiments. The results from evaluating our mobility handover mechanism for mobility support shows that the latency of switching from one gateway to another is 10% - 50% smaller than other state-of-the-art mobility support systems. The results show that the proposed handover mechanism is a very promising approach for mobility support in both Fog computing and IoT systems.

REFERENCES

[1] *European Commission Information Society, Internet of Things Strategic Research Roadmap*, document, 2009, pp. 118–173. [Online]. Available: http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2011.pdf

[2] L. DaXu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Inf.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[3] S. Li, L. Da Xu, and S. Zhao, "The Internet of Things: A survey," *Inf. Syst. Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[4] T. N. Gia *et al.*, "IoT-based fall detection system with energy efficient sensor nodes," in *Proc. Nordic Circuits Syst. Conf. (NORCAS)*, Nov. 2016, pp. 1–6.

[5] S. R. Moosavi *et al.*, "Session resumption-based end-to-end security for healthcare Internet-of-Things," in *Proc. IEEE Int. Conf. (CIT/IUCC/DASC/PICOM)*, Oct. 2015, pp. 581–588.

[6] T. N. Gia *et al.*, "Fault tolerant and scalable IoT-based architecture for health monitoring," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Apr. 2015, pp. 1–6.

[7] S. R. Moosavi *et al.*, "SEA: A secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways," *Procedia Comput. Sci.*, vol. 52, pp. 452–459, Jan. 2015.

[8] T. N. Gia *et al.*, "Energy efficient wearable sensor node for IoT-based fall detection systems," *Microprocess. Microsyst.*, vol. 56, pp. 34–46, Feb. 2018.

[9] T. N. Gia *et al.*, "IoT-based continuous glucose monitoring system: A feasibility study," *Procedia Comput. Sci.*, vol. 109, pp. 327–334, Jan. 2017.

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

[11] V. K. Sarker, M. Jiang, T. N. Gia, A. Anzanpour, A. M. Rahmani, and P. Liljeberg, "Portable multipurpose bio-signal acquisition and wireless streaming device for wearables," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Mar. 2017, pp. 1–6.

[12] M. Ali *et al.*, "Autonomous patient/home health monitoring powered by energy harvesting," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–7.

[13] M. Jiang *et al.*, "IoT-based remote facial expression monitoring system with SEMG signal," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Apr. 2016, pp. 1–6.

[14] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "IoT gateway: Bridging-wireless sensor networks into Internet of Things," in *Proc. IEEE/IFIP 8th Int. Conf. Embedded Ubiquitous Comput. (EUC)*, Dec. 2010, pp. 347–352.

[15] I. Grønbæk, "Architecture for the Internet of Things (IoT): API and inter-connect," in *Proc. 2nd Int. Conf. Sensor Technol. Appl. (SENSORCOMM)*, Aug. 2008, pp. 802–807.

[16] A. M. Rahmani *et al.*, "Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Generat. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2017.

[17] A. M. Rahmani, P. Liljeberg, J.-S. Preden, and A. Jantsch, *Fog Computing in the Internet of Things: Intelligence at the Edge*. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-57639-8.

[18] B. Negash *et al.*, "Leveraging fog computing for healthcare IoT," in *Fog Computing in the Internet of Things*. Cham, Switzerland: Springer, 2018, pp. 145–169, doi: 10.1007/978-3-319-57639-8.

[19] T. N. Gia, M. Jiang, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare internet of things: A case study on ECG feature extraction," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT)*, Oct. 2015, pp. 356–363.

[20] T. N. Gia *et al.*, "Low-cost fog-assisted health-care IoT system with energy-efficient sensor nodes," in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, 2017, pp. 1765–1770.

[21] M. Ahmad, M. B. Amin, S. Hussain, B. H. Kang, T. Cheong, and S. Lee, "Health fog: A novel framework for health and wellness applications," *J. Supercomput.*, vol. 72, no. 10, pp. 3677–3695, 2016.

[22] C. S. Nandyala and H.-K. Kim, "From cloud to fog and IoT-based real-time U-healthcare monitoring for smart homes and hospitals," *Atlantic*, vol. 10, no. 2, pp. 187–196, 2016.

[23] T. N. Gia *et al.*, "Fog computing in body sensor networks: An energy efficient approach," in *Proc. IEEE Int. Body Sensor Netw. Conf. (BSN)*, Jun. 2015, pp. 1–7.

[24] I. Tcarenko, T. N. Gia, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Energy-efficient IoT-enabled fall detection system with messenger-based notification," in *Proc. Int. Conf. Wireless Mobile Commun. Healthcare*, 2016, pp. 19–26.

[25] P. Kulkarni and Y. Öztürk, "Requirements and design spaces of mobile medical care," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 11, no. 3, pp. 12–30, 2007.

[26] M. D. Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Trans. Sensor Netw.*, vol. 8, pp. 1–7, 2011, Art. no. 7.

[27] R. Mulligan and H. M. Ammari, "Coverage in wireless sensor networks: A survey," *Netw. Protocols Algorithms*, vol. 2, no. 2, pp. 27–53, 2010.

[28] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 26–35, Mar./Apr. 2017.

[29] Y.-S. C. Chen and T. Yi-Ting, "A mobility management using follow-me cloud-cloudlet in fog-computing-based RANs for smart cities," *Sensors*, vol. 18, no. 2, p. 428, 2018.

[30] L. F. Bittencourt, M. M. Lopes, I. Petri, and O. F. Rana, "Towards virtual machine migration in fog computing," in *Proc. 10th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput. (3PGCIC)*, 2015, pp. 1–8.

[31] Z. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in *Proc. 38th Annu. Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2005, pp. 287a–297a, doi: 10.1109/HICSS.2005.259.

[32] H. S. Kim, T. F. Abdelzaher, and W. H. Kwon, "Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst.*, New York, NY, USA: ACM, 2003, pp. 193–204, doi: 10.1145/958491.958515.

[33] S. Yang and M. Gerla, "Personal gateway in mobile health monitoring," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM Workshops)*, Mar. 2011, pp. 636–641.

[34] N. Nawka, A. K. Maguliri, D. Sharma, and P. Saluja, "SESGARH: A scalable extensible smart-phone based mobile gateway and application for remote health monitoring," in *Proc. IEEE 5th Int. Conf. Internet Multimedia Syst. Archit. Appl. (IMSAA)*, Dec. 2011, pp. 1–6.

[35] A. Raja and X. Su, "Mobility handling in MAC for wireless ad hoc networks," *Wireless Commun. Mobile Comput.*, vol. 9, no. 3, pp. 303–311, Mar. 2009.

[36] R. S. Koodli and C. E. Perkins, *Mobile Inter-Networking With IPv6*. Hoboken, NJ, USA: Wiley, 2007, pp. 27–53, doi: 10.1002/9780470126486.

[37] R. Silva, J. S. Silva, and F. Boavida, "Mobility in wireless sensor networks—Survey and proposal," *Comput. Commun.*, vol. 52, pp. 1–20, Oct. 2014.

[38] S. González-Valenzuela, M. Chen, and V. C. M. Leung, "Mobility support for health monitoring at home using wearable sensors," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 4, pp. 539–549, Jul. 2011.

[39] A. Jara, M. Zamora, and A. Skarmeta, "An initial approach to support mobility in hospital wireless sensor networks based on 6LoWPAN (HWSN6)," *J. Wireless Mobile Netw., Ubiquitous Comput., Depend. Appl.*, vol. 1, nos. 2–3, pp. 107–122, 2010.

[40] A. J. Jara, M. A. Zamora, and A. F. G. Skarmeta, "HWSN6: Hospital wireless sensor networks based on 6LoWPAN technology: Mobility and fault tolerance management," in *Proc. Int. Conf. Comput. Sci. Eng.*, vol. 2, 2009, pp. 879–884.

[41] A. Jara, M. Zamora, and A. Skarmeta, "Intra-mobility for hospital wireless sensor networks based on 6LoWPAN," in *Proc. 6th Int. Conf. Wireless Mobile Commun.*, 2010, pp. 389–394.

[42] H. Fotouhi, M. Alves, M. Zuniga Zamalloa, and A. Koubaa, "Reliable and fast hand-offs in low-power wireless networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 11, pp. 2620–2633, Nov. 2014.

[43] M. Haseeb, A. Ahsan, and A. W. Malik, "Cloud to cloudlet—An intelligent recommendation system for efficient resources management: Mobile cache," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, 2016, pp. 40–45.

[44] *ADS1299*. Accessed: May 2016. [Online]. Available: http://www.ti.com/product/ADS1299

[45] F. Touati and R. Tabish, "U-healthcare system: State-of-the-art review and challenges," *J. Med. Syst.*, vol. 37, no. 3, p. 9949, 2013.

[46] T. N. Gia, N. K. Thanigaivelan, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Customizing 6LoWPAN networks towards Internet-of-Things based ubiquitous healthcare systems," in *Proc. NORCHIP*, 2014, pp. 1–6.

[47] T. Rappaport, *Wireless Communications: Principles and Practice*, vol. 2. 1996.

[48] K. Atalah, E. Macías, and A. Suárez, "A proactive horizontal handover algorithm based on RSSI supported by a new gradient predictor," *Ubiquitous Comput. Commun. J.*, vol. 3, pp. 77–88, 2008, doi: 10.1016/j.compind.2014.09.002.

[49] J. M. L. P. Caldeira, J. J. P. C. Rodrigues, and P. Lorenz, "Intra-mobility support solutions for healthcare wireless sensor networks–handover issues," *IEEE Sensors J.*, vol. 13, no. 11, pp. 4339–4348, Nov. 2013.

[50] A. Vlavianos, L. K. Law, I. Broustis, S. V. Krishnamurthy, and K. Faloutsos, "Assessing link quality in IEEE 802.11 wireless networks: Which is the right metric?" in *Proc. 19th IEEE Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 2008, pp. 1–6.

[51] K. Srinivasan and P. Levis, "RSSI is under appreciated," in *Proc. Workshop Embedded Netw. Sensors*, 2006, pp. 1–5.

[52] C. Reis, R. Mahajan, M. Rodrig, M. Rodrig, and J. Zahorjan, "Measurement—Based models of delivery and interference in static wireless networks," in *Proc. ACM SIGCOMM*, 2006, pp. 51–62.

[53] C. Guo, J. Zhou, P. Pawelczak, and R. Hekmat, "Improving packet delivery ratio estimation for indoor ad hoc and wireless sensor networks," in *Proc. 6th IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2009, pp. 1–5.

[54] G. Breed, "Bit error rate: Fundamental concepts and measurement issues," *High Frequency Electron. Tutorial*, vol. 2, no. 1, pp. 46–47, 2003.

[55] J. M. L. P. Caldeira *et al.*, "Impact of sensor nodes scaling and velocity on handover mechanisms for healthcare wireless sensor networks with mobility support," *Comput. Ind.*, vol. 69, pp. 92–104, May 2015, doi: 10.1016/j.compind.2014.09.002.

[56] C. Pelczar, K. Sung, J. Kim, and B. Jang, "Vehicle speed measurement using wireless sensor nodes," in *Proc. IEEE Int. Conf. Veh. Electron. Safety*, Sep. 2008, pp. 195–198.
[57] S. Čapkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad hoc networks," *Cluster Comput.*, vol. 5, no. 2, pp. s157–167, 2002.
[58] L. Wang and Q. Xu, "GPS-free localization algorithm for wireless sensor networks," *Sensors*, vol. 10, no. 6, pp. 5899–5926, 2010.
[59] K. Ashton, "That Internet of Things thing," *RFiD J.*, vol. 22, no. 7, pp. 97–114, 2009.
[60] J. J. Caffery and G. L. Stüber, "Overview of radiolocation in CDMA cellular systems," *IEEE Commun. Mag.*, vol. 36, no. 4, pp. 38–45, Apr. 1998.
[61] M. Lin and W. J. Hsu, "Mining GPS data for mobility patterns: A survey," *Pervasive Mobile Comput.*, vol. 12, pp. 1–16, Jun. 2014.
[62] L. Xu, F. Yang, Y. Jiang, L. Zhang, C. Feng, and N. Bao, "Variation of received signal strength in wireless sensor network," in *Proc. 3rd Int. Conf. Adv. Comput. Control (ICACC)*, vol. 3, Jan. 2011, pp. 151–154.
[63] *Pandaboard*. Accessed: Apr. 2018. [Online]. Available: https://elinux.org/PandaBoard
[64] *Arduino Mega*. Accessed: May 2016. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardMega
[65] *Esp8266*. Accessed: May 2016. [Online]. Available: http://www.esp8266.com/
[66] *Arduino Uno*. Accessed: May 2016. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno
[67] *WiFi Shield for Arduino UNO*. Accessed: May 2016. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoWiFiShield
[68] *ESP8266*. Accessed: May 2016. [Online]. Available: http://www.netis-systems.com/en/Products/Wireless%20USB%20Adapters/861.html
[69] T. A. Beardsley and J. Qian, "The TCP split handshake: Practical effects on modern network equipment," *Netw. Protocols Algorithms*, vol. 2, no. 1, pp. 197–217, 2010.
[70] S. R. Moosavi *et al.*, "End-to-end security scheme for mobility enabled healthcare internet of things," *Future Generat. Comput. Syst.*, vol. 64, pp. 108–124, Nov. 2016.
[71] M. Fontanini, *Libtins—Packet Crafting and Sniffing Library*. Accessed: May 2016. [Online]. Available: http://libtins.github.io/

TUAN NGUYEN GIA received the master's (Tech.) degree from the University of Turku, Turku, Finland, in 2014. He is currently with the Internet-of-Things for Healthcare (IoT4Health) Research Group, Future Technologies Department, University of Turku. His research field is fog computing, Internet-of-Things, healthcare, FPGA, and autonomous system. He received a 4-year funded position in the University of Turku and a grant scholarship from Nokia Foundation for an efficient and excellent researcher in 2015. In addition, he received a grant scholarship for excellent research from the Finnish Foundation for Technology Promotion in 2016.

AMIR M. RAHMANI received the M.Sc. degree from the Department of ECE, University of Tehran, Iran, in 2009, the Ph.D. degree from the Department of IT, University of Turku, Finland, in 2012, and the M.B.A. degree jointly from the Turku School of Economics and the European Institute of Innovation & Technology Digital, in 2014. He is currently a Marie Curie Global Fellow with the University of California at Irvine, Irvine, USA, and TU Wien, Austria. He is also an Adjunct Professor (Docent) in embedded parallel and distributed computing with the University of Turku, Finland. His work spans self-aware computing, runtime resource management for systems-on-chip and resource-constrained IoT devices, wearable sensor design, and fog computing. He has authored over 150 peer-reviewed publications.

TOMI WESTERLUND received the Ph.D. (Tech.) degree from the University of Turku, Turku, Finland, in 2008. He joined the Department of Information Technology, University of Turku, as a Senior Researcher, and in 2015 became a University Research Fellow. Since 2013, he has been a Visiting Scholar with Fudan University, Shanghai, China. His current research interest is Internet of Things (IoT); how we can utilize IoT technology to provide better services and improve the quality of life. With that in mind, the main application areas for his research are smart agriculture, smart cities, and health technology.

PASI LILJEBERG received the M.Sc. and Ph.D. degrees in electronics and information technology from the University of Turku, Turku, Finland, in 1999 and 2005, respectively. He received an Adjunct Professorship in embedded computing architectures in 2010. He is currently a Professor with the University of Turku in the field of embedded systems and Internet of Things. He has authored over 270 peer-reviewed publications. At the moment his research is focused on Internet of Things, fog computing, biomedical engineering, self-aware systems, and approximation computing and health technology. In that context, he has established and leading the Internet-of-Things for Healthcare (IoT4Health) Research Group.

HANNU TENHUNEN (M'86) received the Diploma degree from the Helsinki University of Technology, Finland, 1982, and the Ph.D. degree from Cornell University, Ithaca, NY, USA, 1986. In 1985, he joined the Signal Processing Laboratory, Tampere University of Technology, Finland, as an Associate Professor, and later served as a Professor and the Department Director. Since 1992, he has been a Professor with the KTH Royal Institute of Technology, Sweden, where he also served as the Dean. He has over 600 reviewed publications and 16 patents internationality.