Future Generation Computer Systems 107 (2020) 274–285

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Software-defined application-specific traffic management for wireless body area networks

Khalid Hasan [a], Khandakar Ahmed [b], Kamanashis Biswas [c,a], Md. Saiful Islam [a], Omid Ameri Sianaki [d,*]

[a] Griffith University, Gold Coast, Australia
[b] Victoria University, Melbourne, Australia
[c] Australian Catholic University, NSW, Australia
[d] Victoria University Business School, Melbourne, Australia

ARTICLE INFO

Article history:
Received 28 August 2019
Received in revised form 5 January 2020
Accepted 28 January 2020
Available online 31 January 2020

Keywords:
WBAN
Application classification
SDN
SBD
SDWBAN

ABSTRACT

Wireless body area networks (WBANs) are usually used to collect and monitor health-related information for both critical and non-critical patients. However, the traditional WBAN communication framework is unable to guarantee the successful delivery of critical information due to a lack of administrative control and priority support for emergency data. To overcome these issues, this paper proposes a novel software-defined networking (SDN)-based WBAN (SDWBAN) framework for application-specific traffic management. An application classification algorithm and a packet flow mechanism are developed by incorporating SDN principles with WBAN to effectively manage complex and critical traffic in the network. Furthermore, a Sector-Based Distance (SBD) protocol is designed and utilized to facilitate the SDWBAN communication framework. Finally, the proposed SDWBAN framework is evaluated through the CASTALIA simulator in terms of Packet Delivery Ratio (PDR) and latency. The experimental outcomes show that the proposed system achieves high throughput and low latency for emergency traffic in SDWBANs.

1. Introduction

The unprecedented development of information and communication technologies in every sphere of life, especially in medical sectors, has given us more secure and seamless healthcare services. The sensing and monitoring capabilities of tiny electronic devices known as ‘sensors’ enable the real-time collection, storage and processing of various health data. Such capabilities have made a significant contribution to monitoring and diagnosing patients not only in hospitals, but also in elderly homes, private homes, remote areas and so on. The wireless communication framework of sensors in patient monitoring offers a flexible and almost no infrastructure deployment in the network where the sensors can be attached to or implanted inside the human body for monitoring physiological parameters such as blood pressure, heart rate, glucose level, and temperature [1]. This communication network is known as WBAN (see Table 1 for list of abbreviations). Due to the diverse traffic pattern of heterogeneous WBAN applications, it is critical to deal with WBAN of various applications. Moreover, the successful deployment of WBANs is very challenging in terms of utilizing appropriate technology, maintaining strict security regulations, network architecture, traffic engineering, data and QoS management. Furthermore, WBANs are also subject to many additional challenges such as, environmental challenges, physical layer challenges, media access control (MAC) layer challenges, security challenges and transport challenges. However, extensive research has been conducted to address these issues and challenges at different application levels and it is evident that a robust communication architecture with flexible, scalable and more dynamic control over WBAN operations is urgently needed to improve security and efficiency in managing data from various applications. An optimum solution to many of the challenges of WBANs could be achieved by the emerging SDN paradigm [2].

The traditional architecture of WBANs is subject to poor QoS due to a complex management system, poor resource utilization, inability for dynamic reconfiguration, inefficient traffic management and security vulnerabilities [3]. These issues are resolved in [4], where the authors claim that deploying a new application or prioritizing a new application does not require any changes in the data plane since changes in the controller can easily be made with minimal effort through a piece of coding.

* Corresponding author.
E-mail addresses: khalid.hasan@griffithuni.edu.au (K. Hasan), khandakar.ahmed@vu.edu.au (K. Ahmed), kamanashis.biswas@acu.edu.au (K. Biswas), saiful.islam@griffith.edu.au (Md. Saiful Islam), omid.amerisianaki@vu.edu.au (O. Ameri Sianaki).

https://doi.org/10.1016/j.future.2020.01.052
0167-739X/

Table 1
List of Abbreviations.

Abbreviation	*Elaboration*	*Abbreviation*	*Elaboration*
WBAN	Wireless Body Area Network	SH	Sector Head
QoS	Quality of Service	BS	Body Sensor
SDN	Software Defined Networking	RTT	Round Trip Time
SDWBAN	SDN-based WBANs	TOF	Time of Flight
SBD	Sector-Based Distance	LOS	Line of Sight
PDR	Packet Delivery Rate	NLOS	Non-Line of Sight
MAC	Media Access Control	TDMA	Time Division Multiple Access
IoT	Internet of Things	SDESW	SDN enabled Switch

From a management perspective, static applications supporting the architecture of WBANs allow very little or no administrative control over the network. Moreover, the total management of the network becomes cumbersome when multiple application specific sensors from various vendors need to be installed in a WBAN. Since vendors employ different wireless transmission modules in the sensor platform, sensors from various vendors do not interoperate with each other [5]. In most cases, WBAN operators need to stick to one specific vendor device.

In traditional WBANs, there is no administrative control on data traffic as it is dependent on access mechanisms defined by different standards. Control over various types of traffic i.e. high priority, low priority is very important. This study shows how emergency applications retain low latency and high PDR in the presence of a high volume of data traffic. With the rapid advancement of SDN technology, SDN is now being considered for wireless sensor networks (WSNs), industrial automation, smart grids, healthcare and so on. Of these various applications, our primary goal is to leverage the benefits of both SDN and the Internet of Things (IoT) by tightly coupling these two emerging technologies in healthcare systems. With SDN, the network can be configured, monitored and controlled using a set of SDN controllers regardless of vendor specific network devices. The SDN controllers can deploy multiple packet dissemination schemes through managed SDN-enabled switches (SDESW) based on traffic demand [6]. In a nutshell, SDN has the potential to support complex networks of various applications.

In this paper, we propose and implement an SDWBAN framework for heterogeneous WBAN applications by managing both normal and emergency data traffic. The SDWBAN framework implements a cluster-based routing approach to route data packets from sensor nodes to the destination. More precisely, we devise a modified version of the SBD routing protocol to facilitate the packet communication model at layer 3 [7]. In the application layer, we also develop an application module named SDWBAN, which adopts a packet dissemination model from our previous work [3] to accommodate emergency and normal data traffic. Finally, we implement the proposed framework using the CASTALIA simulator [8] and analyse the performance of the framework in terms of PDR and latency. The contributions of this paper include the following:

- A flexible and scalable SDWBAN framework that provides dynamic control over the network with growing number of applications (traffic management).
- An efficient application classification algorithm to support various applications in WBANs.
- A modified version of the SBD routing protocol to facilitate the SDWBAN communication framework.
- Implementation and evaluation of the proposed framework using the CASTALIA simulator.

The rest of the paper is organized as follows: Section 2 presents the related works. A contrasting architectural description of traditional a WBAN and SDWBAN is presented in Section 3. Section 4 describes the SBD analysis for SDWBAN. Section 5 provides the implementation details of the SDWBAN framework and Section 6 discusses the performance analysis of the framework. Finally, Section 7 concludes the paper.

2. Related works

Extensive research has been conducted on various aspects of the development of WBANs such as enhancement of the MAC layer, traffic modelling, and energy efficiency. Here, we provide a brief overview of some of the recent research.

To manage the increased traffic load, in [9], the authors proposed a cloud-assisted WBAN architecture based on the disease-centric patient group (DPG) formation process among the WBANs with a specific disease type. However, in a real-life situation, the cluster formation of a DPG is troublesome as patients with heterogeneous applications have the liberty of mobility. Virtual queue-based priority queueing was used in [10] to support the critical data of WBAN. The study shows that using the load balancing priority queuing mechanism is effective in transmitting critical traffic with minimum delay. However, the work lacks a proper analysis of the load balancing mechanism and the process of severity measurement for the data packets received from a remote location.

One of the important attributes of WBANs, reliability, is evaluated for two different body states i.e., standing, and running in [11]. The simulation results show that using adaptive transmit power mechanism, the PDR can be improved for a variety of body postures. However, the experimental setup was restricted to a limited number of WBAN nodes and the inclusion of the emergency application was avoided. In inter-WBAN interference environment, a health criticality index has been proposed in [12] to prioritize WBAN traffic. Time slots with superior channel conditions are used in [13] to ensure transmission priority for critical traffic in WBANs. It is shown that the proposed technique achieves a notable improvement in terms of QoS and energy efficiency.

A modification of the superframe structure of IEEE 802.15.6 is proposed in [14] to deal with emergency data, where the average delay and throughput is improved by allocating dedicated channels for emergency and non-emergency data. However, the major limitation is that the issue of accommodating various applications is not considered in the simulation and inefficiency could be a significant concern in utilizing resources.

One primary approach of incorporating SDN into healthcare is proposed in [15] which utilizes a centralized controller for a health surveillance application with the help of a software-defined robot. However, the architecture lacks a detailed description of SDN functionalities and priority-based data traffic management i.e. emergency data. Another attempt to incorporate SDN into healthcare is described in [16] to securely monitor patients who demonstrate wandering behaviour. However, the study is limited to patient tracking and therefore overlooks the implementation of WBANs on a large scale. An SDN-based control system is introduced in [17] to manage emergency alerts in a smart city environment. It is shown that by modifying the data routes of emergency and normal traffic, emergency resources can be made available in the locations of the emergency event.

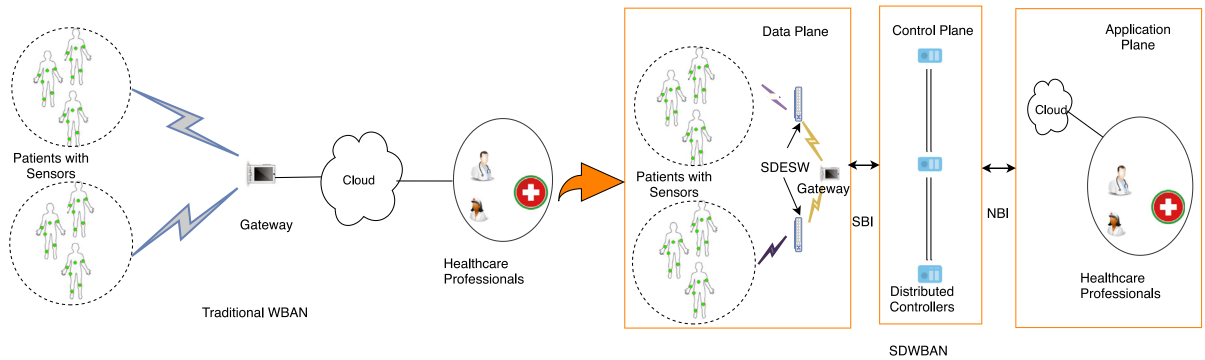


Fig. 1. Traditional WBAN vs SDWBAN.

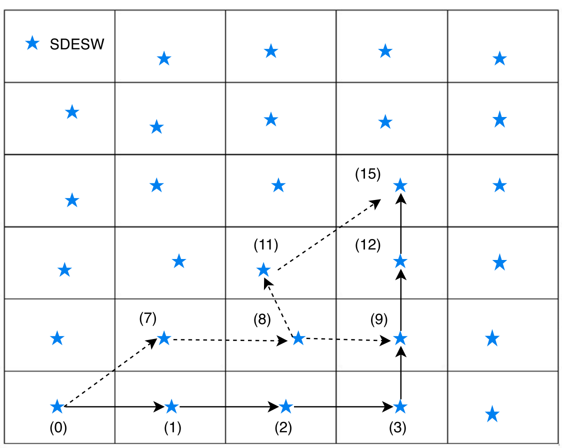


Fig. 2. Route selection in SBD protocol.

Based on the existing literature, it is observed that the relevant works have numerous limitations. These limitations can be highlighted as administrative control over emergency and normal data, the DPG clustering technique, the elaboration of SDN functionalities, and the large number of implementations of WBANs. In order to overcome these limitations, we design a robust communication framework named SDWBAN to facilitate real-time patient monitoring in healthcare services to signify the importance of the study. The proposed model utilizes cluster-based SBD routing and prioritizes emergency data packets over normal traffic based on the application classification algorithm.

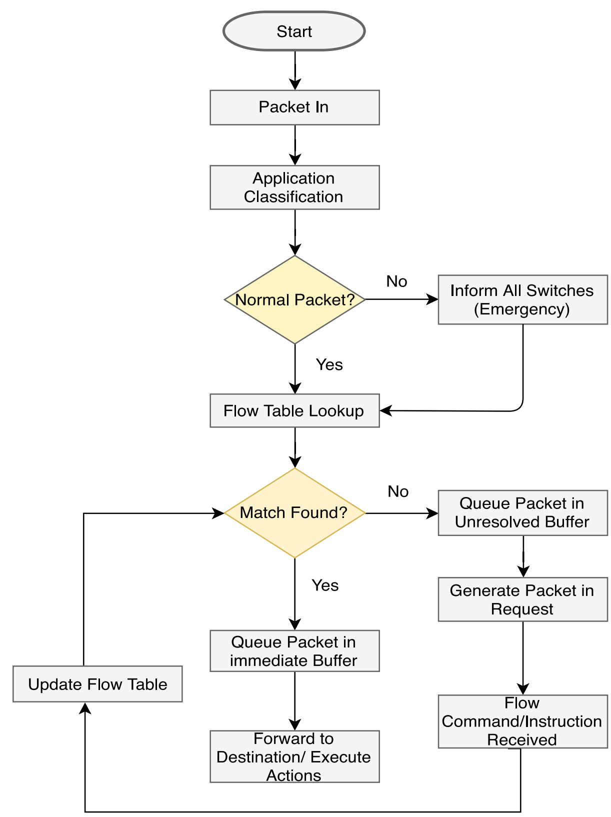


Fig. 3. Packet flow diagram.

3. Traditional WBAN vs SDWBAN architecture

In this section, we present a short overview of the working principles of the traditional WBAN architecture and SDWBAN architecture. In addition, we also present an architectural view of traditional WBAN vs SDWBAN to demonstrate the transformation from the conventional system to the proposed system as illustrated in Fig. 1.

3.1. Traditional WBANs

In general, WBAN architecture is a three layer communication system where all body sensors (BS) are located in the first layer. The BS senses the physiological data according to predefined command and periodically transmits the information to a gateway. The gateway resides at the second layer of the hierarchy and the communication interface between gateway and senors utilizes short-range technologies such as Bluetooth, ZigBee, IEEE 802.15.6 etc to transmit and receive information. Any resourceful device such as a personal digital assistant (PDA) or a mobile phone may work as a gateway. After collecting the physiological data from the BS, the gateway sends the data to the next layer using long-range technologies such as WiFi, WiMax, LTE, LTE-A etc. The cloud system works at layer three which enables the healthcare providers to access and monitor patients of concern.

3.2. SDN-based WBANs (SDWBAN)

In contrast to traditional systems, SDN is a new networking paradigm, which separates the control plane from the data plane. Decoupling the control plane from the data plane offers operators the flexibility of working on a centralized control programme instead of numerous multi-vendor, network devices to implement their favourite policies [2]. The control plane is a software-based controller whereas network devices work as simple packet forwarding devices in the data plane that can be programmed via

Packet ID	Source ID	Destination ID	Data Packet

Fig. 4. The SDWBAN_gen_pkt format.

Packet ID	Source ID	Destination ID	Application ID	Application Name	Data Packet

Fig. 5. The SDWBAN_app_pkt format.

an open interface e.g., forwarding and control element separation (ForCES) [18] and OpenFlow [19]. In SDN, the controller machine creates packet-forwarding rules for any changes in the network topology, a connection initiated by end hosts, shifts in traffic loads or messages from other controllers. The controller drives these rules towards programmable switches where the necessary functionalities are implemented. These features of SDN allow the easier deployment of new protocols and applications.

There have been very few works [20] on integrating SDN into WBANs and a standard architecture for SDWBANs is yet to be defined. In our previous work [3], a novel architecture for SDWBANs was proposed that consists of three planes, namely the data plane, control plane and application plane. The data plane consists of BSs, gateways and SDESWs, where the SDESWs receive the control information from the control plane. In the data plane, a group of patients attached to sensors may form a cluster and share a common SDESW. The BSs transmit physiological data at regular intervals to the associated SDESW to reach the destination. On the other hand, the control plane consists of multiple distributed controllers that can be in operation in a network. For east–west communication, these distributed controllers can be inter-connected so that in the case of the failure of any controller, another nearby controller can support the SDESWs. In the application plane, the management authority can define and install various WBAN applications to monitor patients. The communication interface between the application and the control plane is referred to as the northbound interface (NBI) while the southbound interface (SBI) works as an interface between control and data planes. A detailed explanation of SDWBAN communication architecture including the packet dissemination model can be found in [3].

The proposed architecture presents a multiple number of SDESWs, gateways and controllers. In the case of system failure, the network can be supported by neighbouring nodes. For instance, if a controller fails, another controller is required to cater for a greater number of SDEWSs. This might cause a little delay in re-establishing the communication between the controller and SDESWs. Furthermore, if an SDESW fails while forwarding a packet to its destination, the next available SDESW can serve this purpose and forward it to its destination in a multi-hop fashion. In the case of the failure of SDESW, WBAN sensors become orphaned and then they need to associate themselves with the next available SDESW based on the rank number.

4. SBD for SDWBAN

To support our SDWBAN framework, we propose a modified version of the SBD routing protocol [7]. The sector-based routing divides the network into multiple sectors with a sector head (SH) in each sector. The SH works as an SDESW where the SDN functionalities are implemented to retrieve control information from the controllers. Based on the control information, the SDESW routes the data packets to the appropriate destination. The SDESW is a static node which resides in the vicinity of the patient's bed or in a room and multiple sensors from a sector can share a common SDESW. However, the association with the SDESW is not fixed as it changes when the patients move around in the neighbourhood. This feature supports mobility issues of WBAN. The SBD routing function is split into rounds that consist of two phases: the learning phase and relaying phase. In the following sub-section we describe both in detail.

4.1. Learning phase

In this phase, BSs associate themselves with the corresponding SDESW. The BSs periodically receive a beacon from the neighbouring SDESWs. As they receive the beacon periodically, in the learning phase they can associate themselves with the appropriate SDESW. The BS-SDESW association begins with receiving beacons from nearby SDESWs as defined in Algorithm 1. Each BS assesses the received signal strength indicator (RSSI) and organizes them into a vector $v(SDESW_i, RSSI_i)$, where $RSSI_i \geq RSSI_{i+1}$. However, RSSI does not guarantee to find the closest SDESW with which to be associated. Other environmental factors such as noise, fading, and attenuation are also important parameters to be taken into account. Therefore, we consider round trip time (RTT) to select the associated SDESW in the proposed model. To compute RTT, each BS sends a request packet to all nearby SDESWs and asks for immediate acknowledgement. Once the acknowledgement is received, the BS calculates the distance of the corresponding SDESW using the time of flight (TOF) principle. The BS then assigns a rank number for each nearby SDESW based on the RSSI and TOF values and associates itself with the appropriate SDESW.

Algorithm 1: Beacon Dissemination Process

Input: Beacon frame
Output: Null

```
Network layer packet construction,
  beacon_ctr_pk
// Assigning packet type and source id
Packet_tp = 1 in network layer packet
Source_id = Self_Net_Add
  and set -1 as destination in the beacon_ctr_pk
Broadcast beacon_ctr_pk
to Mac Layer (beacon_ctr_pk, Broadcast_Net_Add)
```

Algorithm 2: The SDESW Selection Procedure

Input: rank, SDESW_IDs
Output: Null

```
Sort SDESW_IDs in descending order
  based on rank
Create network layer packet assoc_ctr_pk
Select dst_SDESW = top element from SDESW_IDs
Set Self_Net_Add as source, dst_SDESW
  as destination and Packet_type = 2 to assoc_ctr_pk
Unicast association request to the closest SDESW
to Mac Layer (assoc_ctr_pk)
```

The TOF can be calculated by the following equation [21]:

$$T_{TOF} = \frac{T_{RTT} - T_{TPP}}{2} \tag{1}$$

Here, T_{TPP} denotes the time to process packet. The distance between two nodes can be calculated as follows:

$$d_{RTT} = T_{TOF} \times c \tag{2}$$

Here, c is the speed of light. According to [22], Eq. (2) can be further extended by incorporating the faultiness factors involved in distance measurement as follows:

$$d_{RTT} = T_{TOF} \times c + \varepsilon_{RTT}^{LOS} + \varepsilon_{RTT}^{NLOS} \tag{3}$$

In real environments, particularly in the case of WBANs, sensors can be located on different parts of the body. Moreover,

Packet ID	Source ID	Destination ID	Application ID	Application Name	Priority ID	Data Packet

(a)

Packet ID	Source ID	Destination ID	Application ID	Application Name	Priority ID	Source Switch ID	Destination Controller ID

(b)

Packet ID	Source ID	Destination ID	Application ID	Application Name	Source Switch ID	Destination Controller ID	Actions
							Forward/Drop/ Unicast/Broadcast

(c)

Fig. 6. (a) SDWBAN_app_pkt with priority ID (b) SDWBAN_Ctrl_pkt with packet header (c) SDWBAN_Cntrl_pkt_cmd.

due to the various postures and movements of the body, signal propagation from sensors faces obstacles such as various objects located in the patient's surroundings and sometimes even patient body. Ultimately, the issue of line of Sight (LOS) and Non-line of Sight (NLOS) occurs. As can be seen, the above equation includes two fault components: ε_{RTT}^{LOS} and ε_{RTT}^{NLOS}. The ε_{RTT}^{LOS} refers to the LOS setting whereas ε_{RTT}^{NLOS} occurs due to a ranging in NLOS settings. The faultiness of the multipath effect (i.e. the obstruction of signals) of ε_{RTT}^{NLOS} can be reduced using the empirical approach described in [23] in order to calculate the accurate d_{RTT}. On the other hand, hardware related noise and uncertainties especially jitter contributes to the error component ε_{RTT}^{LOS}. Therefore, considering the jitter effect, T_{TOF} can be calculated as [24]:

$$T_{TOF} = \frac{T_{RTT} - (J_{t1} + J_{c1} + T_{TPP} + J_{c2} + J_{t2})}{2} \tag{4}$$

For the accurate calculation of the time between sending the initial packet and receiving the acknowledgement packet by the sender node, two timestamps are used. One contains the jitters $J_{t0}, J_{c0}, J_{t3}, and J_{c3}$. Similarly, on the other end, two timestamps are used to calculate the time between receiving a packet and sending the first bit of the acknowledgement where the jitter values are $J_{t1}, J_{c1}, J_{t2}, and J_{c2}$. The measured T_{RTT} is estimated as follows:

$$T_{RTT} = J_{t0} + J_{c0} + TOF_R + TOF_A + J_{t3} + J_{c3} + T_{TPP} \tag{5}$$

In Eqs. (4) & (5), TOF_R = TOF for the request packet, TOF_A = TOF for the acknowledgement packet, J_{tK} = jitter caused by the clock of the transceiver, $[K = 0, 1, 2, 3]$, and J_{cK} = jitter caused by the clock of the microcontroller.

On the basis of RSSI and TOF values, the BSs calculate the rank of the corresponding SDESWs using the following equation:

$$rank_i = \left(\left(\frac{RSSI_i}{max_{i=1}^{M}(RSSI_i)} \right) + \left(\frac{d_{RTTi}}{max_{i=1}^{M}(d_{RTTi})} \right) \right)^{-1} \tag{6}$$

Now, every BS sends an association request to the SDESW which has the highest rank in its list. Eq. (6) finds the highest rank for the node with the maximum RSSI and minimum d_{RTT} between two nodes. The inverse operation of finding the maximum rank between two nodes occur because, the node with the minimum distance will receive the highest rank value. Algorithm 2 presents the steps involved in this process. Upon receiving the association request from the BSs, SDESW creates a list of BSs and assigns the conflict free time division multiple access (TDMA) frame slot to the BSs.

4.2. Relaying phase

After receiving the data from the associated BS, the SDESW forwards the data packet to the destination in a multi-hop fashion. The associated SDESW knows the co-ordinate of the neighbouring SDESWs and selects a neighbouring node that is the minimum distance to the destination. However, if the destination node is within the communication range of the SDESW, it can transmit the data packet directly. The next hop selection process in relaying phase follows the steps presented in Algorithm 3.

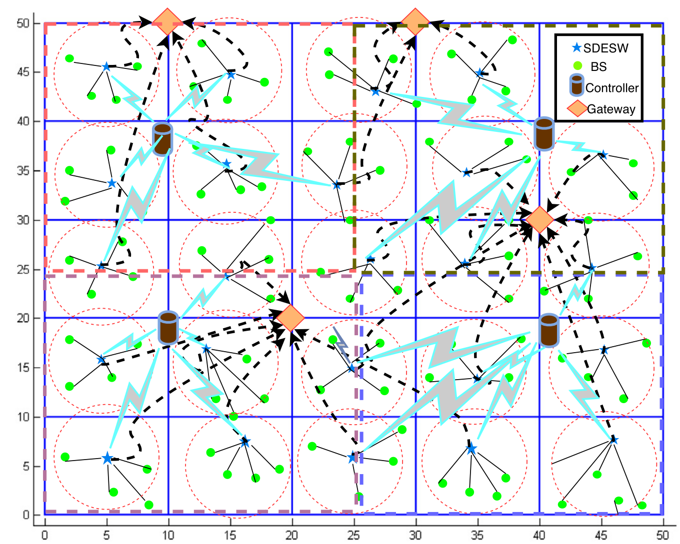


Fig. 7. Implementation Scenario.

According to Algorithm 3, all the neighbouring SDESWs (including the associated SDESW) calculate the Euclidean distance to the gateway. The associated SDESW sorts all the distances and then forwards the packet to its immediate neighbouring SDESW which is the minimum distance to the gateway. This process continues until the packet reaches its destination.

Fig. 2 illustrates the packet forwarding process from the associated SDESW to the gateway. As shown in the figure, sector (0) indicates the associated SDESW's location whereas the destination node is located in sector (15). There can be different routes to send the packet to the destination, as shown in the figure with the solid and dotted lines. It can be seen that, the possible routes for data transmission in the given scenario are as follows:

$(i)(0) \rightarrow (1) \rightarrow (2) \rightarrow (3) \rightarrow (9) \rightarrow (12) \rightarrow (15)$,
$(ii)(0) \rightarrow (7) \rightarrow (8) \rightarrow (11) \rightarrow (15)$,
$(iii)(0) \rightarrow (7) \rightarrow (8) \rightarrow (9) \rightarrow (12) \rightarrow (15)$.

However, according to Algorithm 3, the packet will be forwarded through the shortest possible route calculated on the basis of Euclidean distance. In this case, the shortest path is: $(0) \rightarrow (7) \rightarrow (8) \rightarrow (11) \rightarrow (15)$ and all packets will be forwarded through this path as long as the route is available.

Algorithm 3: Next Hop Selection Process

Input: Target switch $SDESW_i$ where $SDESW_i \in [1 \dots s]$
Output: Next hop $SDESW_k$, where $SDESW_k \in [1 \dots s]$

1: **procedure** FINDING(position of destination Gateway (GW) and current co-ordinate in the grid.)
2: $destCo \leftarrow$ Co-ordinate of GW, (x, y, z)
3: $CurCo \leftarrow$ Co-ordinate of associated $SDESW$, (x_0, y_0, z_0)
4: $Co\_Neigh_i \leftarrow$ Co-ordinate of neighbouring $SDESW(x_i, y_i, z_i, \dots x_n, y_n, z_n)$ where, $i = 1, 2, 3, \dots n$
5: Calculate distance, *t_dist* from all neighbouring *SDESWs* to *GW* by Euclidean system
6: Sort *t_dist* in ascending order into a vector, $t\_dist = [d_1, d_2, d_3, \dots d_n]$
7: Select minimum distance from *t_dist* and corresponding *SDESW*
8: **if** *destCo* $\neq$ co-ordinate of selected *SDESW* **then**
9: Return $SDESW_k$
10: Move the packet to $SDESW_k$
11: Repeat step 4 to 8
12: **else**
13: Move the packet to the *GW*
14: **end if**
15: **end procedure**

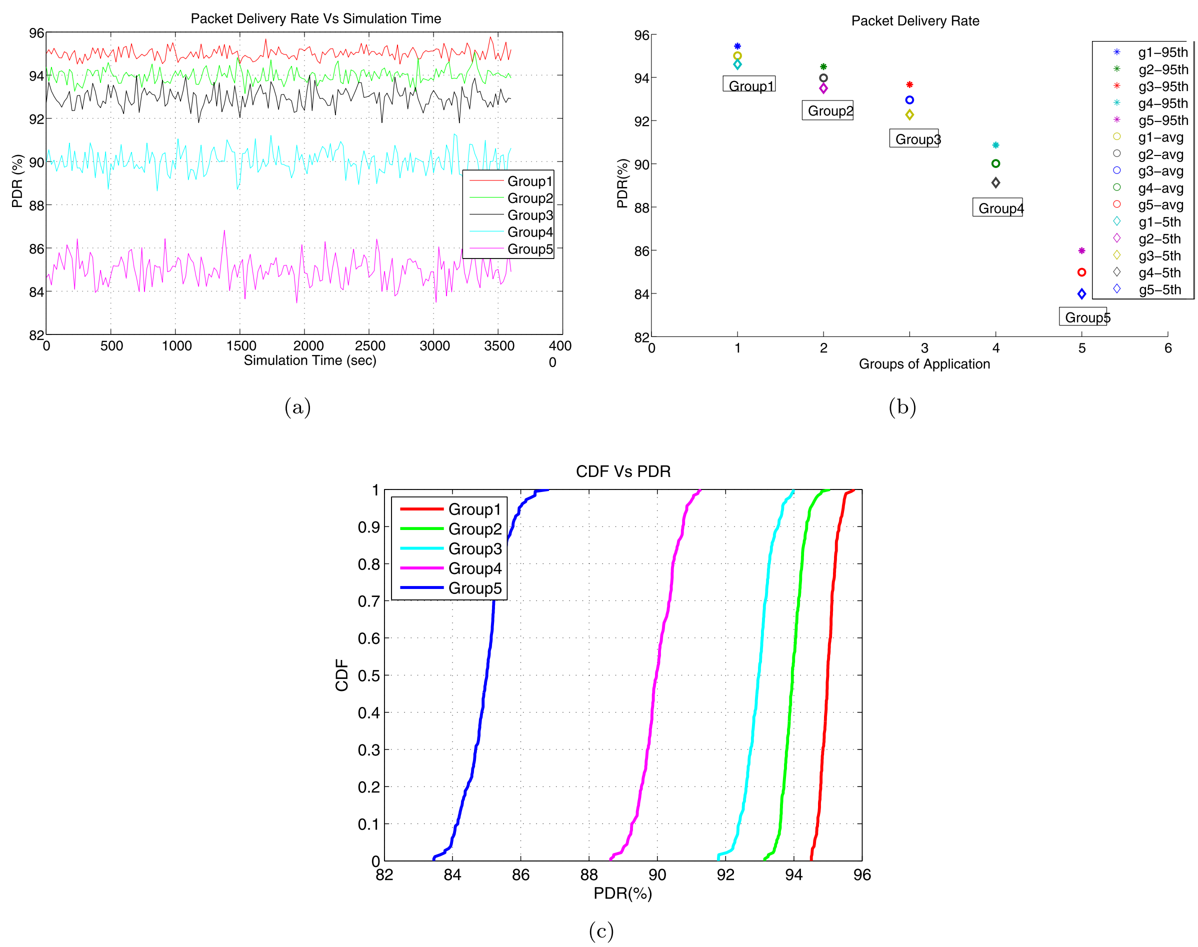


Fig. 8. (a) PDR VS Simulation Time, (b) PDR VS Group of Application (95_AVG_5th Percentile graph), (c) CDF of PDR.

5. The SDWBAN communication framework

In this section, we describe the communication model of our SDWBAN framework, which includes an application classification algorithm and packet flow mechanism.

5.1. Communication model

As clusters form in different sectors, the BSs located in every sector send data to their corresponding SDESWs. The SDESW then classifies the received data packet according to the predefined application IDs and checks if it is a normal data packet or an emergency application. Then the data packet is assigned an application and priority-based ID for processing. It should be noted that emergency data packets are given priority over normal data packets. In the case of an emergency data packets, all SDESWs are also informed immediately regarding the emergency packets by the controller. After application classification and prioritization, the flow table look-up task is performed. If there is a match found at SDESW, packets are queued in the immediate buffer and then forwarded to the destination. However, if no match is found in the flow table, that particular packet is sent to the queue in the unresolved buffer. The SDESW then generates a packet_in request to the controller requesting for a flow command or asking instructions for the unmatched packet. The controller sends new flow commands or instructions in reply to the packet_in request. Note that the flow commands can be modelled in such a way that SDESW can unicast, multicast or broadcast the packet based on the WBAN applications. The flow chart shown in Fig. 3 demonstrates the packet flow process through the SDESW.

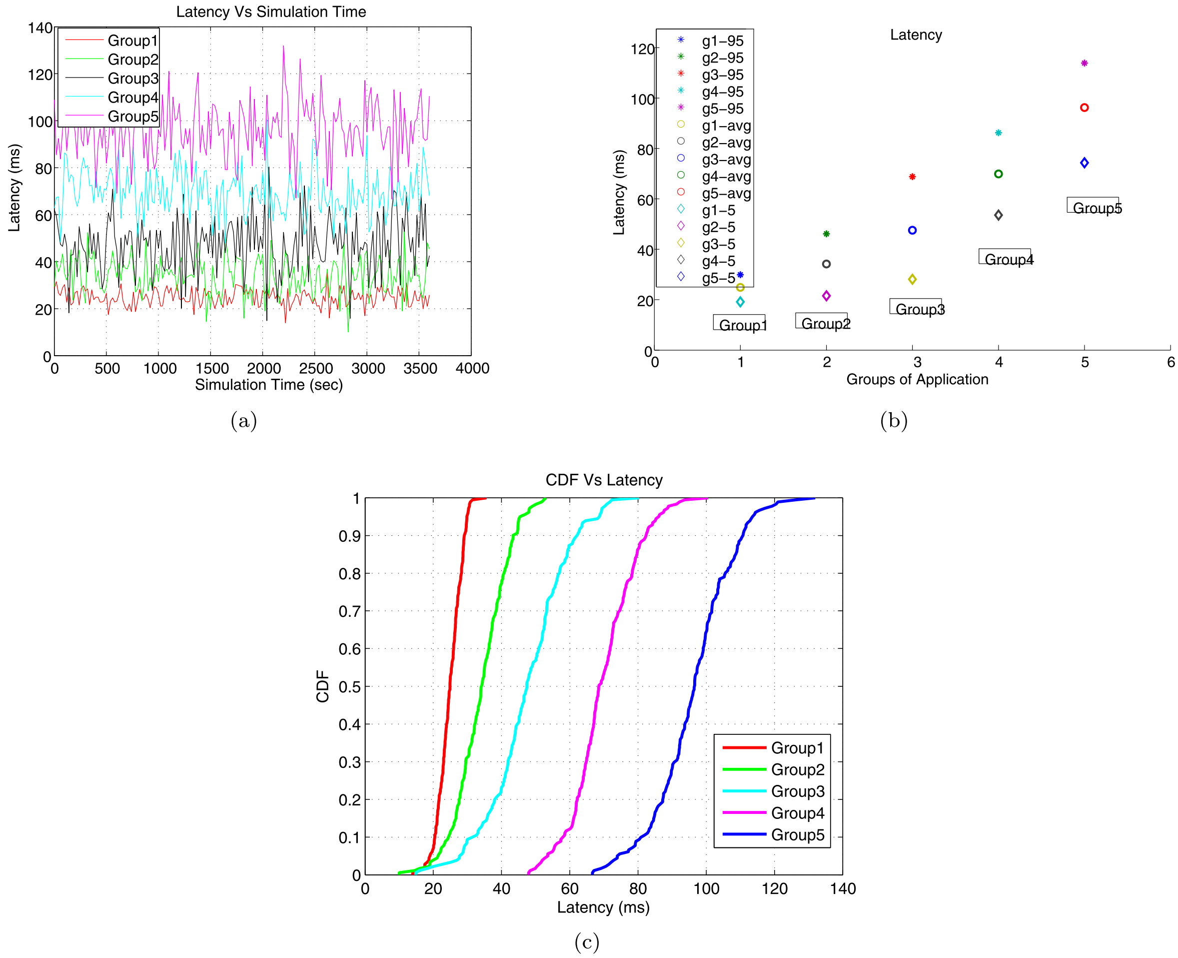


Fig. 9. (a) Latency VS Simulation Time, (b) Latency VS Group of Application (95_AVG_5th Percentile graph), (c) CDF of Latency.

5.2. Application classification module

Usually, the BSs unicast data packets to their corresponding switches. Fig. 4 presents the format of a unicast packet at the application layer. Upon arrival of the SDWBAN_app_pkt, the switch identifies the corresponding application of the packet using the application classification module (Algorithm 4).

At this stage, the application name and ID are added to the incoming packet as shown in Fig. 5. After classifying the packet according to the application name and ID, the processing priority is checked, and a priority-ID is added to the packet as illustrated in Fig. 6(a). Then, the switch performs flow table look-up task. When a flow is found, the related actions corresponding to this flow are performed. In the case of flow miss, the switch sends the packet to the queue of an unresolved buffer. After this, a control packet is generated, which is called the "packet_in_request" and is sent to the controller only with the packet header fields (see Fig. 6(b)). The controller receives the packet and replies with the "packet_out_response" which contains the flow command to be implemented by the switch. Fig. 6(c) presents the SDWBAN_app_pkt including new flow command attached by the controller.

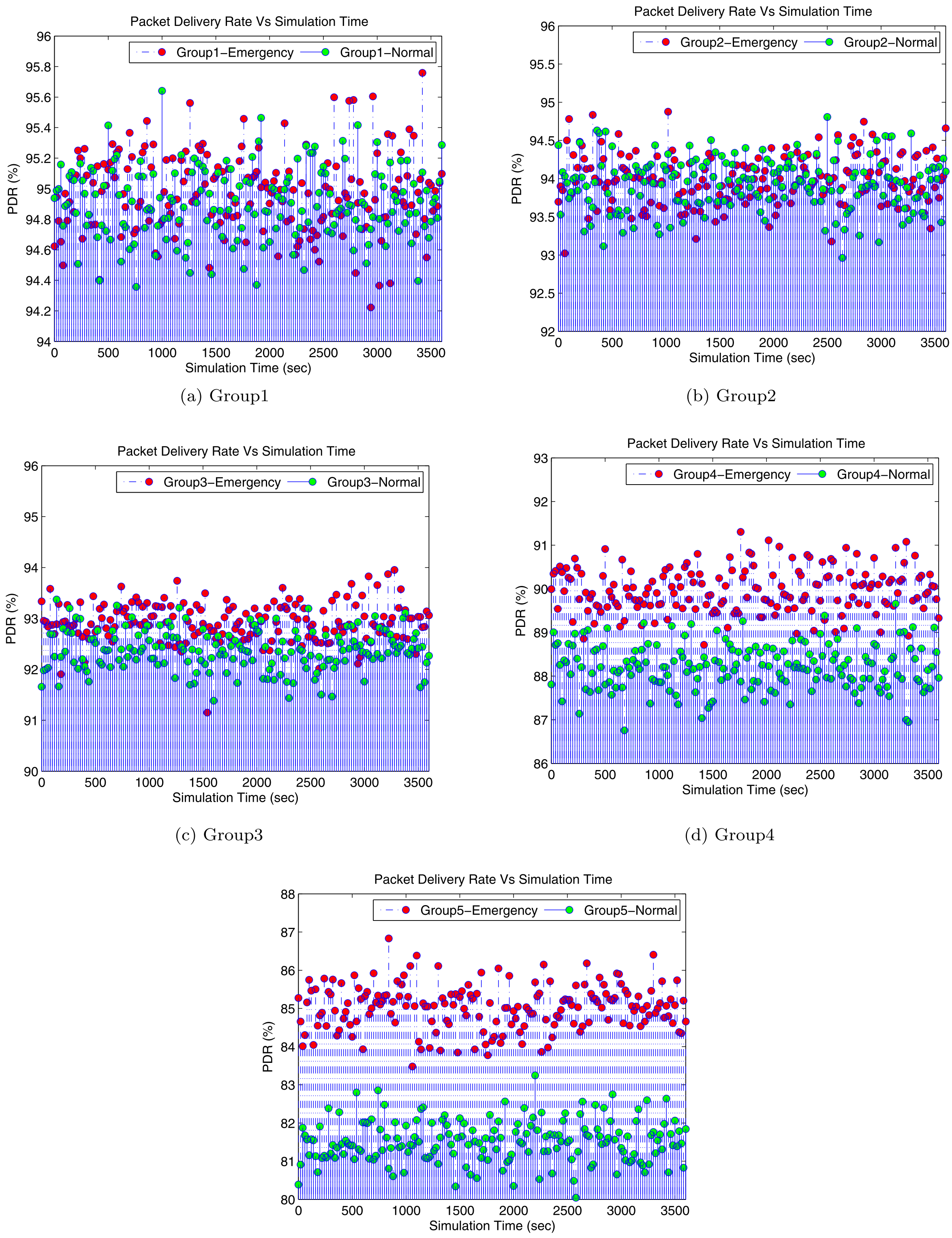
6. Performance analysis

This section provides the implementation scenario and discusses the experimental outcomes in terms of PDR and delay.

6.1. Implementation scenario

We simulate the proposed SDWBAN architecture in a rectangular grid of a 5 × 5 building block model where each block represents a sector. The network field size is 75 × 75 m^2, where BSs are deployed randomly in the whole network area. These BSs associate themselves with an appropriate SDESW and thus form a number of clusters. We deploy the SDESWs in a static fashion in each sector of the 5 × 5 grid and the gateways are positioned in such a way that multiple SDESWs can share a single gateway. The distributed controller is statically deployed in a 2 × 2 grid, where a single distributed controller supports multiple SDESWs from multiple sectors. Fig. 7 depicts the implementation scenario where, the dotted circle demonstrates the cluster under SDESW and the dotted arrow represents the destination gateway. The simulation parameters are given in Table 2. The path loss exponent has different values based on the environment. Typically, the value of the exponent ranges from 2 to 4. As we have considered an indoor wireless environment, the measured path loss exponent value in the indoor propagation environment is selected as 2.4. Anderson et al. further explained indoor wireless propagation characteristics in [25].

Our work has potential application in various sectors, in particular, the healthcare sector, where sensors can be deployed in an elderly home attached to a patient's body for the purpose of measuring physiological parameters. Patients equipped with WBAN sensors in the same room could form a cluster and transmit data to the nearby SDESW. Various kinds of WBAN application sensors such as ECG, blood pressure, and temperature



(a) Group1

(b) Group2

(c) Group3

(d) Group4

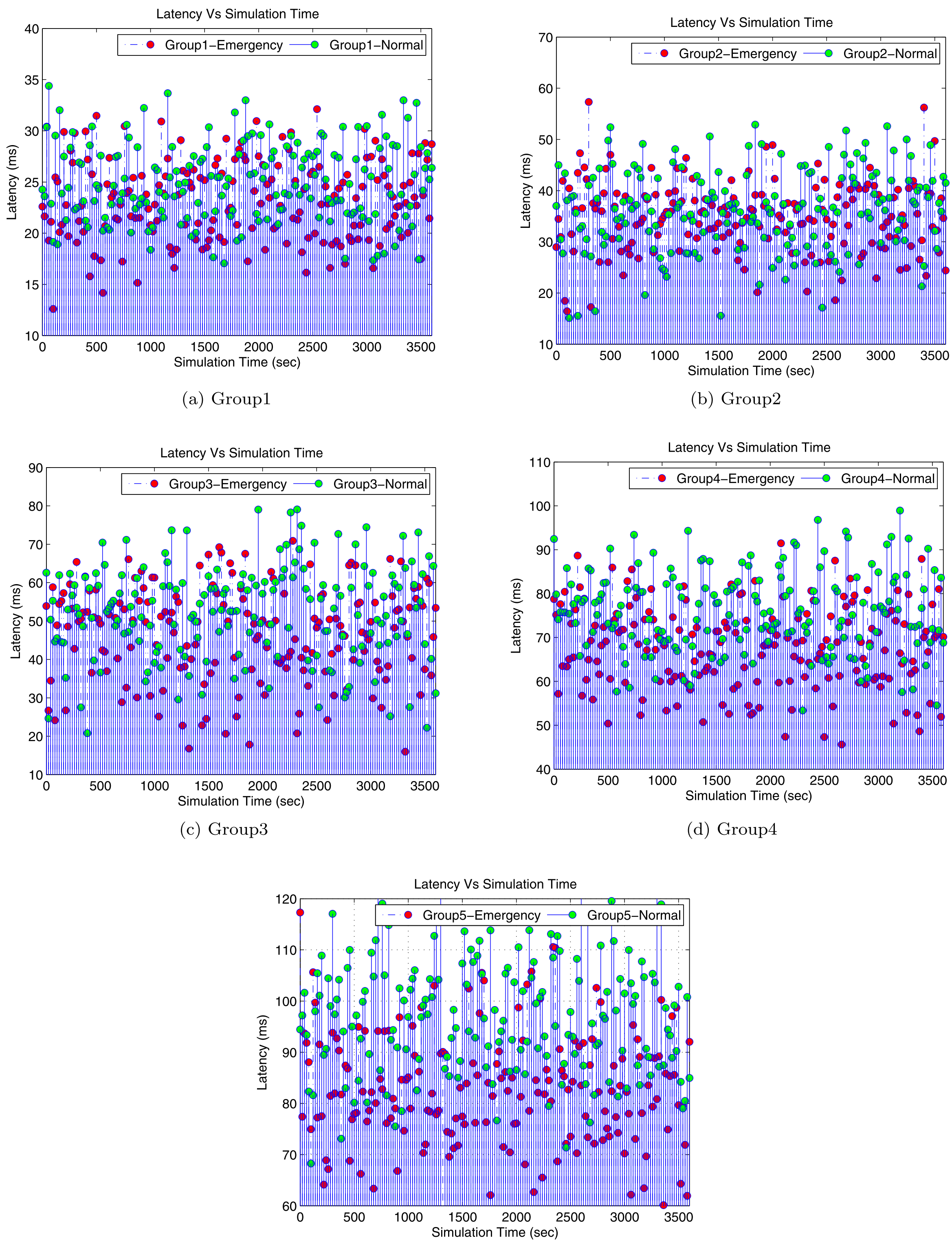
(e) Group5

Fig. 10. PDR VS Simulation Time (Group 1–5) Scenario 2.

etc., generate valuable physiological data periodically. Depending on the physiological condition of the patients, the applications can be categorized as either normal or emergency applications. For instance, a particular application could be categorized as an emergency application, when the measured parameter seems to be abnormal based on a pre-defined threshold value.

6.2. Results and discussion

The simulation is built on a Castalia-3.2 version [8] on Ubuntu 16.04.4 where the Castalia simulator is extended by adding two modules namely SDWBAN and SBD routing. We run the experiment for 100 iterations to realize the output in a number of ways



(a) Group1

(b) Group2

(c) Group3

(d) Group4

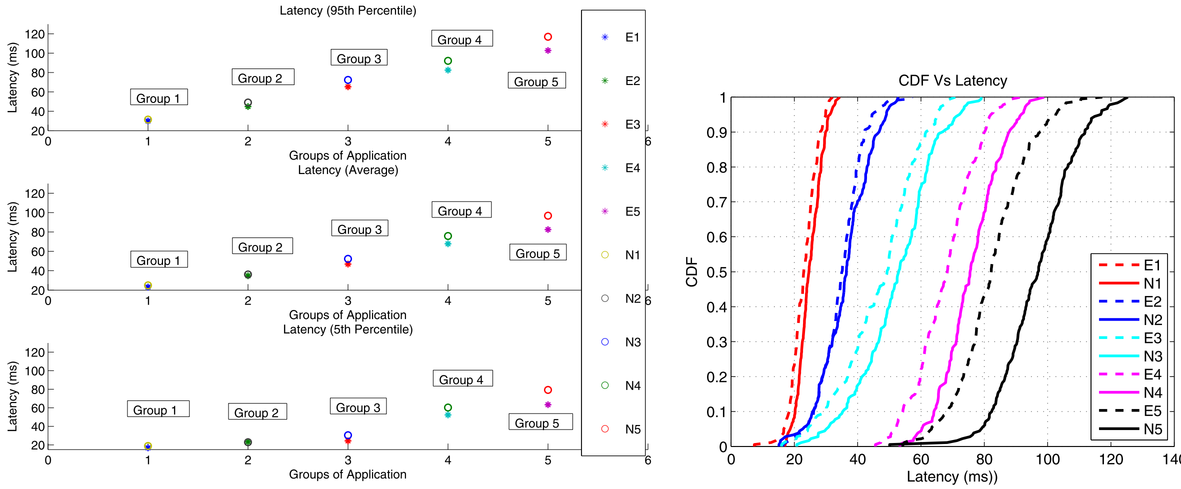
(e) Group5

Fig. 11. Latency VS Simulation Time (Group 1–5) Scenario 2.

such as average, median and 95% confidence interval. We experiment and analyse the PDR and latency against a varying number of applications. The PDR provides the percentage of successful packet deliveries for the proposed SDWBAN framework for a different number of applications. On the other hand, latency shows the time difference between packet generation and resolution by an application. The PDR can be defined as follows:

$$PDR = \frac{\text{No. of Packets Resolved}}{\text{No. of Packets Transmitted}} \tag{7}$$

In our simulation work, we use four different kinds of WBAN applications, namely blood flow, blood pressure, body temperature, and blood pH. We clone the physiological data transmission



(a) (b)

Fig. 12. (a) Latency VS Group of Application (95th percentile, Average & 5th Percentile), (b) CDF of Latency for each group.

Algorithm 4: Application Classification Module

Input: application_ID
Output: Null

```
1: procedure AppClassification(
   SDWBAN_gen_pkt)
2: classification_result ←
   classify_pkt(SDWBAN_gen_pkt)
3:    if classification_result ≠ normal then
4:       inform_emergency()
5:    end if
6: lookup_result ← lookup_flow_table
   (classification_result)
7:    if match = True then
8:       queue_pkt_immediate_buffer()
9:       execute_actions()
10:   else
11:      queue_pkt_unresolved_buffer()
12:      pkt_in_req
13:      received_instruction()
14:      update_flow_table()
15:      Go To Step 7
16:   end if
17: end procedure
```

Table 2
Simulation parameters.

Parameter	Value	Parameter	Value
Radio range (BS, SDESW, Controller)	∼8 m, ∼20 m, ∼20 m	Reference distance (d_0)	1 m
Transmission power (SDESW, BS)	0 dBm, −10 dBm	Simulation area	75 × 75 m^2
Data rate, Modulation type, Bits Per symbol, Bandwidth	250 Kbps, PSK, 4, 20 MHz,	Number of BS, Gateway	100, 4
Noise bandwidth, Noise floor, Sensitivity	194 MHz, −100 dBm, −95 dBm	BS density	4 nodes/225 m^2
Free Space Path Loss exponent	2.4	Total SDESW	25 (1 node per sector)
Initial Average Path Loss ($PL(d_0)$)	55 dB	Total Controller	4 (2 × 2 grid)
Gaussian Zero-Mean Random Variable (X_α)	4.0	Number of clusters	25

system of these applications into our simulation. These sensors generate packets at different sampling intervals. For instance, the sample inter-arrival time of the body temperature sensor is 5 s whereas the sample inter-arrival time of the blood pressure sensor is 0.01 s [26]. Throughout groups 1 to 5, we assign different application IDs to each group so that the system treats all these different IDs as a new application.

6.2.1. Scenario 1

In the first scenario, we generate traffic for a different number of applications and observe the PDR against the simulation time. We group the applications according to Table 3.

Scenario1_PDR: We run the simulation for 60 min and observe the PDR of different application groups at different time interval. During the simulation, we keep the packet generation rate the same for each application group. As we can see from Fig. 8(a), the PDR decreases over time as the number of applications increases.

Table 3
Scenario 1 Group.

Group	No. of applications
Group1	1
Group2	5
Group3	10
Group4	15
Group5	20

Significant differences can be observed from the PDR of groups 4 and 5, as in group 4 the PDR is around 90% whereas for group 5 the PDR reduces to 84%. This is because, as the number of applications increases, SDESW to the controller communication increases and thus traffic load. Consequently, this results in more packets being dropped in the largest number of applications, i.e., group 5. For further analysis, we also present the average

Table 4
Scenario 2 Group.

Group	Normal applications	Emergency applications
Group1	1	1
Group2	2	3
Group3	4	6
Group4	7	8
Group5	10	10

PDR of each application group with the 95th and 5th percentile in Fig. 8(b) and the cumulative distribution function (CDF) of the PDR is presented in Fig. 8(c). Both figures summarize the results of Fig. 8(a) and indicate that the packet dropping rate increases as the network becomes more complex with more applications. For instance, at the 90th percentile point of Fig. 8(c), the PDR of the group 1 is more than 95% and the PDR of group 5 is above 85%. This result demonstrates that any packet of group 1 at the 90th percentile has a 95% probability of being delivered successfully whereas it has a 5% probability of being dropped. Similarly, any packet of group 5 at the 90th percentile has an 85% probability of being delivered successfully but a 15% probability of being dropped.

Scenario1_Latency: In this part, we carry out an analysis to measure the latency of each application group through the same experimental setup used in the PDR analysis. It can be seen that as the number of applications grows, latency increases (Fig. 9). This is due to the increased number of control packets between SDESW and the controller for a different number of applications. In Fig. 9, the latency of group 1, 2 and 3 ranges roughly from 20 ms to 60 ms while the latency of groups 4 and 5 results in between 60 to 120 ms. More distinguishable points can be observed in Fig. 9(b) & (c). The CDF of the latency in Fig. 9(c) at the 90th percentile point exhibits latency of 30 ms for group 1 whereas 110 ms of latency occurs in the case of group 5. This further demonstrates that any packet of group 1 has a 90% probability o being delivered within 30 ms whereas 10% of the time, it may not be delivered within 30 ms. Similarly, any packet of group 5 has a 90% probability of being delivered within 110 ms whereas 10% of the time, the packets may not reach within 110 ms.

6.2.2. Scenario2

In this part of the experiment, we introduce emergency and normal applications. Therefore, we group the applications in a heterogeneous style where there is at least one normal and one emergency application in each group. We increase the number of applications in various groups and we allocate more emergency applications to visualize the priority of the emergency applications based on our proposed algorithm. We group the applications according to Table 4.

Scenario2_PDR: We again run the simulation for 60 min and analyse the PDR for different groups of applications. Fig. 10 depicts the PDR of various heterogeneous application groups over the simulation period. The Fig. 10(a) & (b) show the PDR versus the simulation time results, where the significant difference between the emergency and normal applications cannot be interpreted clearly as the number of both the emergency and normal applications is low. This does not incur a quantifiable amount of traffic between controller and SDESW communication. Therefore, the PDR in the case of group 1 and group 2 output a similar percentage. However, it can be seen from Fig. 10(c–e) that the PDR of the emergency applications is always higher than normal applications. This is because whenever an emergency application is detected at the SDESW, the emergency data packet is resolved immediately due to its high priority. Although the PDR decreases with an increase in the number of applications in each group, the PDR of the emergency applications is always higher than normal applications. The experiment outcomes prove that incorporating SDN into WBANs makes the network more flexible in terms of giving priority to life-critical traffic even if the number of applications increases over time.

Scenario2_Latency: Here, we analyse the latency of all application groups as in the previous experiment. First, we inspect the latency of emergency and normal applications over the simulation time. Fig. 11 shows the latency incurred over 60-min time frame. It can be seen that the latency of normal and emergency applications remain the same for group 1 and group 2 as shown in Fig. 11. However, for groups 3 to 5, the latency of normal applications starts to rise whereas the latency of emergency applications reduces. In order to conduct a clearer analysis of our results, we also plot the average latency, 95th & 5th percentile and CDF as presented in Fig. 12(a) & (b). Application group 5 in Fig. 12(b) shows that at the 90th percentile point, emergency applications (E5) experience around 98 ms latency while normal applications (N5) experience around 112 ms. This further indicates that 90% of the time, data packets of emergency application are delivered within 98 ms while 10% of the time packets may not be delivered within 98 ms. Similarly, 90% of the time data packets from normal applications have the probability of reaching their destination within 112 ms while 10% of the time, they may not reach the destination within this time frame. The result obtained for scenario 2 supports the design goals of the proposed SDWBAN framework as it shows that the latency of emergency traffic is less than that for normal traffic.

7. Conclusion and future work

In this paper, we proposed an SDWBAN framework that provides administrative controls on incoming data traffic in order to prioritize sensitive data over normal data in healthcare applications. Further, an application classification algorithm and a modified version of the SBD protocol are employed to implement a data prioritization policy and to ensure the efficient routing of data packets from source to destination nodes. Finally, the proposed framework is implemented in a Castalia simulator and the impacts of the SDN controller to resolve WBAN traffic are analysed for a variety of application groups. It can be seen that when the complexity in the network increases with large number of flow requests from SDESW, the data packets from normal applications are dropped by approximately 18.49% for group 5. On the contrary, the PDR of emergency applications for the same group is reduced by 14.93% which illustrates a 3.56% improvement in the PDR. A similar level of performance improvement in latency is observed for emergency applications compared to normal applications. The improvement in emergency applications is due to the fact that the controller is still capable of processing the emergency data on a first-most priority basis. The outcomes of this implementation support our proposed framework of SDWBAN. In our implementation, we have arbitrarily chosen four controllers to support the SDWBAN framework. However, there can be an optimized relationship in the number of WBAN sensors, SDESWs and controllers. The optimization will enable us to design a SDWBAN which is more effective and economical. In future work, we will incorporate an optimization mechanism for our proposed SDWBAN framework. Moreover, we aim to incorporate a variable payload size with a varying packet transmission rate to experiment with a true heterogeneous WBAN scenario.

Declaration of competing interest

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

References

[1] A. Darwish, A.E. Hassanien, Wearable and implantable wireless sensor network solutions for healthcare monitoring, Sensors 11 (6) (2011) 5561–5595, http://dx.doi.org/10.3390/s110605561.

[2] J.H. Cox, J. Chung, S. Donovan, J. Ivey, R.J. Clark, G. Riley, H.L. Owen, Advancing software-defined networks: A survey, IEEE Access 5 (2017) 25487–25526, http://dx.doi.org/10.1109/ACCESS.2017.2762291.

[3] K. Hasan, X. Wu, K. Biswas, K. Ahmed, A novel framework for software defined wireless body area network, in: 2018 8th International Conference on Intelligent Systems, Modelling and Simulation, ISMS, 2018, pp. 114–119, http://dx.doi.org/10.1109/ISMS.2018.00031.

[4] L. Hu, M. Qiu, J. Song, M.S. Hossain, A. Ghoneim, Software defined healthcare networks, IEEE Wirel. Commun. 22 (6) (2015) 67–75, http://dx.doi.org/10.1109/MWC.2015.7368826.

[5] A. Drescher, A survey of software-defined wireless networks, Tech. Rep., Dept. Comput. Sci. Eng., Washington Univ. St. Louis, St. Louis, MO, USA, 2014, pp. 1–15.

[6] Z. Guo, M. Su, Y. Xu, Z. Duan, L. Wang, S. Hui, H.J. Chao, Improving the performance of load balancing in software-defined networks through load variance-based synchronization, Comput. Netw. 68 (2014) 95–109, http://dx.doi.org/10.1016/j.comnet.2013.12.004.

[7] K. Ahmed, M.A. Gregory, Optimized TDMA based distance routing for data centric storage, in: 2012 IEEE 3rd International Conference on Networked Embedded Systems for Every Application, NESEA, 2012, pp. 1–7, http://dx.doi.org/10.1109/NESEA.2012.6474019.

[8] A. Boulis, Castalia 3.2, User's Manual, National ICT Australia Ltd, Australia, 2011.

[9] S. Misra, A. Samanta, Traffic-aware efficient mapping of wireless body area networks to health cloud service providers in critical emergency situations, IEEE Trans. Mob. Comput. 17 (12) (2018) 2968–2981, http://dx.doi.org/10.1109/TMC.2018.2822279.

[10] M. Ambigavathi, D. Sridharan, Energy efficient and load balanced priority queue algorithm for wireless body area network, Future Gener. Comput. Syst. 88 (2018) 586–593, http://dx.doi.org/10.1016/j.future.2018.05.044.

[11] E. Sarra, T. Ezzedine, Performance improvement of the wireless body area network (WBAN), in: 2016 IEEE 18th International Conference on E-Health Networking, Applications and Services (Healthcom), 2016, pp. 1–6, http://dx.doi.org/10.1109/HealthCom.2016.7749509.

[12] X. Yuan, C. Li, Q. Ye, K. Zhang, N. Cheng, N. Zhang, X. Shen, Performance analysis of IEEE 802.15. 6-based coexisting mobile WBANs with prioritized traffic and dynamic interference, IEEE Trans. Wireless Commun. 17 (8) (2018) 5637–5652, http://dx.doi.org/10.1109/TWC.2018.2848223.

[13] G. Sun, K. Wang, H. Yu, X. Du, M. Guizani, Priority-based medium access control for wireless body area networks with high-performance design, IEEE Internet Things J. 6 (3) (2019) 5363–5375, http://dx.doi.org/10.1109/JIOT.2019.2900661.

[14] F. Ullah, A.H. Abdullah, M.M. Arshad, A. Aliyu, Emergency data handling medium access control protocol for wireless body area network, in: 2017 6th ICT International Student Project Conference, ICT-ISPC, 2017, pp. 1–4, http://dx.doi.org/10.1109/ICT-ISPC.2017.8075314.

[15] L. Hu, M. Qiu, J. Song, M.S. Hossain, A. Ghoneim, Software defined healthcare networks, IEEE Wirel. Commun. 22 (6) (2015) 67–75, http://dx.doi.org/10.1109/MWC.2015.7368826.

[16] V. Varadharajan, U. Tupakula, K. Karmakar, Secure monitoring of patients with wandering behavior in hospital environments, IEEE Access 6 (2018) 11523–11533, http://dx.doi.org/10.1109/ACCESS.2017.2773647.

[17] A. Rego, L. Garcia, S. Sendra, J. Lloret, Software defined network-based control system for an efficient traffic management for emergency situations in smart cities, Future Gener. Comput. Syst. 88 (2018) 243–253, http://dx.doi.org/10.1016/j.future.2018.05.054.

[18] A. Doria, J.H. Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, J. Halpern, ForwardIng and Control Element Separation (ForCES) Protocol Specification, Tech. Rep., 2010.

[19] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: enabling innovation in campus networks, ACM SIGCOMM Comput. Commun. Rev. 38 (2) (2008) 69–74, http://dx.doi.org/10.1145/1355734.1355746.

[20] M. Al Shayokh, A. Abeshu, G. Satrya, M. Nugroho, Efficient and secure data delivery in software defined WBAN for virtual hospital, in: 2016 International Conference on Control, Electronics, Renewable Energy and Communications, ICCEREC, IEEE, 2016, pp. 12–16, http://dx.doi.org/10.1109/ICCEREC.2016.7814973.

[21] S. Adler, S. Pfeiffer, H. Will, T. Hillebrandt, J. Schiller, Measuring the distance between wireless sensor nodes with standard hardware, in: 2012 9th Workshop on Positioning, Navigation and Communication, 2012, pp. 114–119, http://dx.doi.org/10.1109/WPNC.2012.6268749.

[22] A. Bahillo, S. Mazuelas, R.M. Lorenzo, P. Fernandez, J. Prieto, R.J. Duran, E.J. Abril, Hybrid RSS-RTT localization scheme for indoor wireless networks, EURASIP J. Adv. Signal Process. 2010 (1) (2010) 126082, http://dx.doi.org/10.1155/2010/126082.

[23] H. Hashemi, The indoor radio propagation channel, Proc. IEEE 81 (7) (1993) 943–968, http://dx.doi.org/10.1109/5.231342.

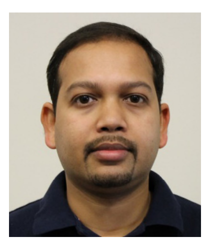
[24] S. Adler, S. Pfeiffer, H. Will, T. Hillebrandt, J. Schiller, Measuring the distance between wireless sensor nodes with standard hardware, in: 2012 9th Workshop on Positioning, Navigation and Communication, 2012, pp. 114–119, http://dx.doi.org/10.1109/WPNC.2012.6268749.

[25] J.B. Andersen, T.S. Rappaport, S. Yoshida, Propagation measurements and models for wireless communications channels, IEEE Commun. Mag. 33 (1) (1995) 42–49, http://dx.doi.org/10.1109/35.339880.

[26] J.Y. Khan, M.R. Yuce, G. Bulger, B. Harding, Wireless body area network (WBAN) design techniques and performance evaluation, J. Med. Syst. 36 (3) (2012) 1441–1457, http://dx.doi.org/10.1007/s10916-010-9605-x.

Khalid Hasan is a Ph.D. candidate at School of Information and Communication Technology, Griffith University, Australia. He received his Master of Science (Radio Communication) from Aalto University, Finland in 2014. He received his Bachelor of Engineering (Communication) from International Islamic University Malaysia in 2010. His current research interests include Wireless Body Area Network, Software Defined Networking, Blockchain and Internet of Things Applications.

Khandakar Ahmed (SM'11 - M'15) is currently a Lecturer with the Discipline of IT, College of Engineering and Science, Victoria University. He received the Ph.D. degree from RMIT University, Melbourne, Australia and M.Sc. in Network and e-Business Centred Computing (NeBCC) under the joint consortia of University of Reading, UK; Aristotle University of Thessaloniki, Greece and Charles III University of Madrid (UC3M), Spain. Khandakar has extensive industry engagement as a CI in multiple research projects related to the Internet-of-Things, smart cities, machine learning, cybersecurity and health informatics. He has published more than 50+ refereed scholar articles.



Kamanashis Biswas is currently working as a lecturer in Information Technology in the Peter Faber Business School (Sydney), Faculty of Law and Business. He received his Ph.D. degree from School of Information and Communication Technology, Griffith University in 2016. He received his masters in Security Engineering from Blekinge Institute of Technology, Sweden in 2007. His research interests include cryptography, IDS, energy efficient and secure routing, and clustering in WSNs.

Md. Saiful Islam is a Lecturer in the School of Information and Communication Technology, Griffith University, Australia. He has finished his Ph.D. in Computer Science and Software Engineering from Swinburne University of Technology, Australia in February, 2014. He has received his B.Sc. (Hons) and M.S. degree in Computer Science and Engineering from University of Dhaka, Bangladesh, in 2005 and 2007, respectively. His current research interests are in the areas of database usability, spatial data management and big data analytics.

Omid Ameri Sianaki received the B.Sc. degree in Mechanical Engineering and M.Sc. degree in Industrial Management (Operational Research) from Islamic Azad University, Tehran, Iran in 2000 and 2007, respectively. He received his Ph.D. degree from School of Information Systems, Curtin University, Perth, Australia in 2016. Currently, he is a lecturer and a Programme Convenor for Master of Applied Information Technology at Victoria University Sydney. Dr. Ameri is a Unit Convenor for two Research Project Part A and B units in Master of IT programme at Victoria University Sydney. His current research interests are in Big Data Analytics and Blockchain.