An Emergency Situation Detection System for Ambient Assisted Living

Chamara Sandeepa*, Charuka Moremada†, Nadeeka Dissanayaka‡, Tharindu Gamage§ Madhusanka Liyanage¶‖
*†‡§Department of Electrical and Information Engineering, University of Ruhuna, Galle, Sri Lanka
¶School of Computer Science, University College Dublin, Ireland
‖Centre for Wireless Communications, University of Oulu, Finland
Email: *cpsandeepa@gmail.com, †charuka.kavinda@yahoo.com, ‡nadeekasumududissanayaka@gmail.com,
§tharindu@eie.ruh.ac.lk, ¶madhusanka@ucd.ie, ‖madhusanka.liyanage@oulu.fi

Abstract—**This paper proposes "An Emergency Situation Detection System for Ambient Assisted Living (AAL)", to support elderly people and patients with chronic conditions and potential health-related emergencies to live independently. It implements an Internet of Things (IoT) network that continuously monitors the health conditions of these people. The network includes mobile phones, to transmit the data generated by the IoT sensors to the cloud server. Especially, the paper proposes the 3rd party unknown mobile relays instead of dedicated gateways as opposed to many existing solutions for IoT healthcare applications. The wireless communication technology used to provide the connectivity between the sensor nodes and mobile relays is Bluetooth Low Energy (BLE). To establish a secure end-to-end connectivity between low power IoT sensor nodes and cloud servers, the paper proposes several techniques. After the medical data transmission to the cloud server, it is responsible for emergency detection and alert generation accordingly. The type of emergency is not limited to a specific health issue, but new emergency situations can be defined and added to the proposed system. Ultimately, the interested parties such as family members, caretakers and doctors receive these alerts. The development of a prototype of the system as a part of the work using commercial off-the-shelf devices verifies the validity of the proposing system and evaluates the performance advantage over the existing systems.**

Index Terms—**Internet of Things, Bluetooth Low Energy, Mobile Relays, Advanced Encryption Standard, Emergency Detection, Mobility, Power Efficiency**

I. Introduction

Recent advancements in IoT based techniques have made a significant influence on modern technological applications including continuous (e.g. patients) monitoring and healthcare [1]. The improvements in these techniques would be very helpful for elderly people and people with chronic conditions. Often, they should be monitored continuously and their health condition should be analyzed based on the monitored data. If there is any situation where immediate attention or medical care is needed whenever they are in a position that no nearby support is available, these people would be at high risk and may sometimes face a life-threatening situation [2]. With this paper, we are introducing an IoT based solution for remote patient monitoring. The solution can be identified as a cost effective, flexible, mobility supported and comparatively power efficient way to establish a solid patient monitoring.

Patients with chronic disease conditions and elderly people can get support from BLE wearable sensors to keep track of their health [3]. These BLE sensors facilitate communication with the user's mobile phones. It is very important to continuously transmit the data from the sensor to the mobile, to keep track of their health conditions. But the elderly people may often forget to keep their mobile phones always charged. Also, during an emergency situation, the user's mobile phone might be turned off or might get damaged. Healthcare devices using dedicated gateways cannot transmit the data in such instance. Also, when the healthcare devices are out of range from the gateway, the continuous monitoring is halted due to connection loss. Further, the devices that communicate through generic BLE protocol cannot establish a connection and transmit the person's health data to an unknown gateway device in proximity in such a connection loss.

Our Contribution

Therefore, this paper proposes a secure, third party mobile relay-based solution to overcome the above issues, which are critical during an emergency situation. The paper contributes to followings:

- Proposes a protocol to successfully establish communication with a third party mobile relay and perform end-to-end secure communication with a cloud server.
- Implements a prototype of the emergency situation detection system with off-the-shelf BLE modules for the use case of heart rate monitoring by transmitting data to a remote server.
- Addresses the issues associated with real-time and stored data transmission from the IoT device.
- Discusses and experimentally analyzes the problems associated with the generic BLE communication channel implementation when transmitting data over a distance.
- Proposes a solution of handovering for relay mobiles to address the detected problems with BLE.

The remainder of this paper is organized as follows: Section II provides the related work. Section III contains the proposed architecture. The details of the proposed protocol are given by Section IV. The implementation details is presented in section V. Section VI contains the experiments and results. Finally, conclusions and future research directions are presented in section VII.

II. Related Work

The work presented in [4] [5] introduces a mobile-based relay assistance system for the establishment of a secure End to End (E2E) connection between low-power IoT sensors and cloud servers without using any specific and dedicated gateway. This work proposes a basic prototype to accomplish the communication task and the E2E connection establishment is done through a secure AES-CCM encryption technique.

The solution in [6] contains the main server and several other sensing servers which are acting as gateways. It describes the IoT sensor networks middleware to perform sensor data translations. As the system is not a cost-effective solution and due to its poor scalability, it is not a much feasible solution for IoT applications.

The work, related with dedicated gateways presented in [7] describes the implementation of smart e-health gateways, named as UT Gates, at the edge of healthcare IoT in clinical environments.

In addition to the cloud processing as in general case, they suggest the local data processing through the Smart Gateways. This step helps to create a fast response and avoids latency but it may be vulnerable to security problems such as the possibility of implementing malicious gateways that could eavesdrop on the patient's data. Moreover, this system may not support the mobility-related aspects due to cost and it is hard to provide universal connectivity in the external environments with the proposed system.

In [8], the authors introduce an open BLE platform (custom-designed beacon platform nRF24Cheep) and open source development of the BLE physical and Medium Access Control (MAC) layer in order to provide the capabilities of changing the communication stack. The Contiki OS port is provided for the new platform.

In [9], authors propose an end-to-end security scheme for mobility enabled healthcare Internet of Things (IoT). Their scheme mainly consists 3 characteristics as, (i) a secure and efficient end-user authentication and authorization architecture which is based on the certificate based DTLS handshake, (ii) secure end-to-end communication developed on session resumption, and (iii) robust mobility implemented using the interconnected smart gateways.

In [10], a model named iConfig is proposed for managing IoT devices in smart cities. This has an edge-driven platform that has mainly addressed the three major issues in current IoT management (i.e. registration, configuration and management).

Under the [11], they propose a scheme named Collaborative on Demand Wi-Fi Sharing (COWS) and in this case, they propose a system to enable the Wi-Fi roaming facilities for users.

But this system is not fully compatible with resource-constrained devices such as those that have power limitations with Wi-Fi.

Other than [4], all the other related works use a dedicated gateway for data transmission.

Moreover, the proposed solution in [4] supports a single relay for real-time data transmission only. If there is no relay nearby during the incident, the model proposed in [4] will not work. Therefore, a relay-based system is needed and it is suggested in our work.

III. Proposed Architecture

This system is about implementing an IoT based remote patient monitoring and caring system which facilitates maximum mobility and flexibility for users. The proposing system is similar to a fog computing approach [12], with third party mobile relays. An overview of the proposed system architecture is shown in Fig. 1.

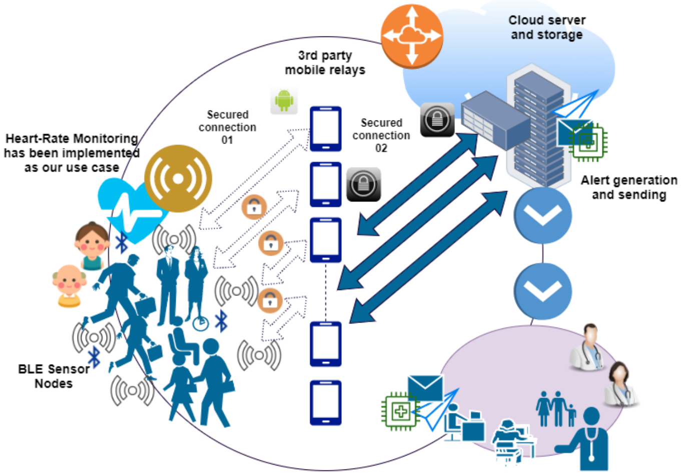

Fig. 1. Proposed system architecture

As indicated by Fig.1, the whole architecture consists of four main components. At one end of the architecture, the network consists of BLE based sensor nodes with low power consumption. This part of the system is responsible for gathering the required patient data. For the use case scenarios, we selected the heart-rate and fall detection.

The data generated from the sensors are forwarded into a 3rd party unknown mobile relay. In this case, the BLE sensor node selects a specific mobile relay before sending the information and the selection procedure is mentioned under the protocol section. Furthermore, each 3rd party mobile relay has a mobile application which enables and controls its connectivity with the network. At the mobile stations, there is no data processing or storing work, but the mobile can attach its location information with the transmitting data, to get the approximate patient location.

In the next step, the mobile relay sends data to the cloud server via its internet connection accordingly. In this case, a secure socket communication establishes between the mobile relay and the cloud server.

The server performs data processing, data storing and emergency situations detection. After the detection of any emergency, the server sends notifications to the registered carers of the patients as SMS and emails. Other than the alerts, the system also supports real-time data monitoring and location tracking services for the carers of the patient. Therefore, the patient's data is available for the carers to view via a web application.

IV. Protocol

A. Single mobile relay node BLE connection

The message flow of the protocol for a single mobile in the relay can be illustrated as the figure 2.

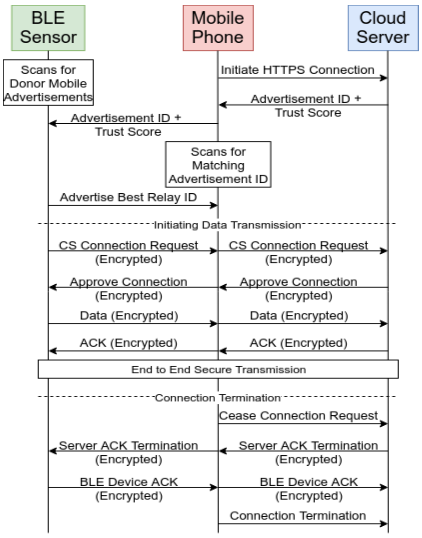
BLE Sensor
Mobile Phone
Cloud Server
Scans for Donor Mobile Advertisements
Initiate HTTPS Connection
Advertisement ID + Trust Score
Advertisement ID + Trust Score
Scans for Matching Advertisement ID
Advertise Best Relay ID
Initiating Data Transmission
CS Connection Request (Encrypted)
CS Connection Request (Encrypted)
Approve Connection (Encrypted)
Approve Connection (Encrypted)
Data (Encrypted)
Data (Encrypted)
ACK (Encrypted)
ACK (Encrypted)
End to End Secure Transmission
Connection Termination
Cease Connection Request
Server ACK Termination (Encrypted)
Server ACK Termination (Encrypted)
BLE Device ACK (Encrypted)
BLE Device ACK (Encrypted)
Connection Termination

Fig. 2. Message flow of the proposed protocol

1) *Phase1:* The donor mobile relay node connects with the Cloud Server(CS) via HTTPS connection request by the mobile app. Upon successful authentication, the CS issues an advertisement ID and a trust score. The advertisement ID is a plain text string which is used by the mobile to advertise itself. The trust score is an integer value about the mobile donor's success in the previous data transmissions.
2) *Phase2:* The mobile relay node starts advertising the received advertisement ID along with the trust score. Meanwhile, this mobile is also scanning for an advertisement from a BLE sensor advertising with the same ID. A BLE sensor device which is scanning for mobile relays can get the advertisement ID from the mobile relay node and start advertising itself with that same ID.
   In case of multiple mobile devices in proximity, the sensor node can select the mobile device with the best Received Signal Strength Indicator (RSSI) value and the best trust score. Then, the mobile app can establish a connection with a sensor if a match is found with its ID.
3) *Phase3:* After the connection establishment with the mobile relay node, the BLE sensor can initiate a connection request from the CS. The mobile relay would forward this request to the cloud server. The server can then validate this request and approve the connection.
4) *Phase4:* After the connection approval, the BLE sensor initiates the transmission of data to the mobile relay. These data packets may contain a timestamp and the data is encrypted so that the mobile relays cannot eavesdrop the user's data. Once a fixed amount of data is transferred, the sensor expects an encrypted acknowledgment from the CS. If the acknowledgment is sent, then the data transmission is continued. Otherwise, the sensor terminates the connection with the relay and reports this session to the CS in the next successful data transmission.
5) *Phase5:* The mobile donor can set the maximum threshold of data that a sensor can consume, so if the threshold value is reached, it can request to cease the connection from the CS. The CS then sends the last acknowledgment message to the BLE sensor and the sensor terminates the connection with the relay node. In this case, the sensor discards the session information as this is a legitimate session termination. To send more data, the sensor can restart from phase 2 and start scanning for nearby mobile relay devices.

B. BLE Multi connect

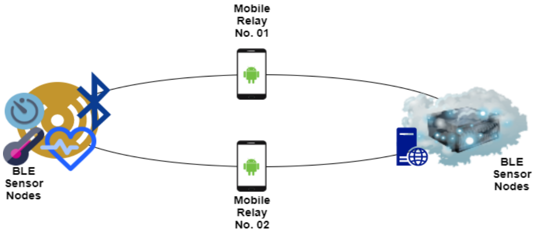

Fig. 3. Multiple mobile relay node connection

The transmission of real-time data and previously stored data together by a single mobile relay node may cause problems such as latency. Here, the stored data is the data that is generated by the sensors when the device is not connected to a nearby mobile relay. The paper proposes an expansion of the same protocol to transmit real-time and stored data separately by two mobile relays as indicated in Fig. 3. Here, the IoT device follows a procedure such that one mobile gateway is dedicated for transmitting the real-time data and another for the transmission of any stored data.

In order to achieve this, there should at least two donor mobile devices. The real-time data has more priority and hence the selection of the mobile relay device for real-time data transmission is based on the best RSSI value and the trust score. To transmit the stored data, the IoT device selects the second best mobile.

V. Implementation

In order to implement the protocol, the selected use case scenario is the emergency situation detection system for people with problems in the heart. This involves sensors that can detect heart problems, a BLE device that can transmit the data, a relay mobile that supports BLE and a cloud server that performs operations such as user registrations, authentication, storage, detection of emergency situations and performing alert

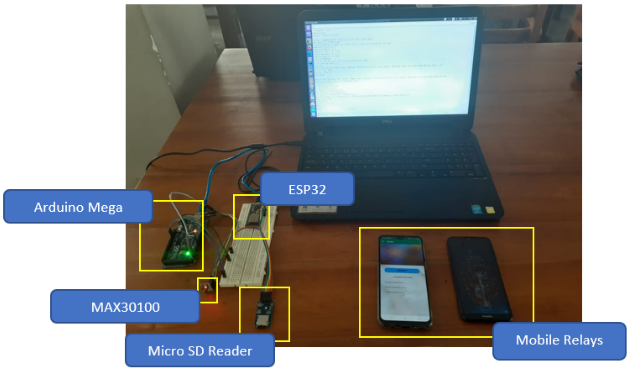

Fig. 4. Prototype implementation

generation. The basic experimental setup which is implemented with the research work is shown in Fig. 4.

A micro-controller unit named ESP32, which is powered with BLE 4.2, is used to establish the communication pathway. The mobile phones used for the implementation were Samsung Galaxy M20 and Samsung Galaxy A20 with Android 9 Pie system and OPPO A37 running Android 5.1 Lollipop. The Android Studio 3 libraries were used for the mobile application. For the server-side implementation, Java Spring Boot framework is used. The database was implemented with PostgreSQL and for the time-series data storage, TimescaleDB extension of the PostgreSQL is used. The heart rate is obtained from a heart rate sensor model named MAX30100.

TABLE I
CONFIGURATION SETTINGS FOR ESP32

Attribute	Configured values
Transmission power	-21dBm
Number of BLE services	1
Number of BLE characteristics	2
Maximum packet size	244 bytes
Maximum app memory	3.5 MB

When a communication failure occurs or until the BLE sensor establishes a connection with the mobile relay, the data that is captured by the sensors are stored in an external flash memory connected to the ESP32. An external flash for the ESP32 was needed since ESP32 overall flash memory is 4 MB and a considerable amount of this space (3.5 MB) is allocated for the storage of the running application. The remaining amount of space would not be sufficient to store information generated over a long period of time.

The ESP32 supports FreeRTOS and thus the parallel operation of multiple threads is possible from its dual-core CPU having Xtensa LX6 microprocessor. This can be used to establish the parallel operation of transferring real-time data from one thread and transferring stored data from another thread. It is more feasible than using a single thread as the real-time data should be given more priority over old stored data.

Before beginning the data transmission, the device has to be registered by the user. For that, the remote server issues an IoT device ID to the patient and it should be saved within the BLE device's memory. The server is able to distinguish each device and the patient associated with it for each data sent when attached with this saved device ID.

After that, ESP32 continuously scans for the mobile relays within the close proximity of the sensor. Meanwhile, it saves the data generated from the heart rate sensors in the flash memory. When a single mobile device is discovered, it connects with the device directly and starts transferring data. In the presence of multiple mobile relay nodes, it selects the best mobile relay node based on the RSSI value and before connecting, it repeats the scanning multiple times to verify the availability of the best relay node.

Once the connection is established, the data is encrypted by AES encryption and transmitted to the mobile app. From the app, this data is directed to the server. We have also appended location details of the mobile relay anonymously to detect the approximate location of the patient because rather than implementing it in the sensor, it is energy efficient as more energy will be drained to operate a dedicated GPS sensor along with the BLE device. The front end web user interface is made from the Angular framework.

In the front end web application, it is required to perform user registration prior to the data transmission and is implemented as follows. A person should first undergo initial registration as a general user and he or she is able to select the role as a patient, donor or a carer. The donors are the third party mobile users who contribute to the mobile relay. The carer is someone who receives notifications about the patient's health status. Carers will be able to receive these notifications and to view real-time health information about the patient after the patient gives permission for that. A user can perform any of these roles including all three. The real-time data from the patient can be rendered in a graph and viewed by both patient and the carers.

The remote server back end of our system is implemented using Java Spring Boot as a set of Representational State Transfer (REST) Application Program Interfaces (API).

We used web sockets to transfer data from the mobile app to the server. This data is continuously stored in the database. For each user session, the patient status is continuously monitored based on various criteria. The monitoring criterion can vary according to the patient's healthcare device sensors. First, we selected heart rate detection as the use-case for the implemented system and implemented a criterion based on upper and lower threshold heart Beats Per Minute (BPM) where the lower is threshold is 60 BPM and the upper is 100 BPM. To test another use case, we used a device with an accelerometer sensor ADXL345 to detect falls as well. Therefore, new devices can be added with different sensors that need any other criteria, according to the requirements of the patients or carers. If the emergency situation detection module detected any emergency situation it sends a request to the notification API to forward email and SMS notifications to patients and to pre-registered carers. Email notifications are sent via Gmail to the registered carers of a patient. To send SMS notifications, we used a cloud communication service named Twilio.

VI. Experiments and Results

To address several issues associated with the IoT device and BLE while designing the system, we conducted two experiments and integrated their results with the system.

A. Data processing time under single core and dual core processing

In this experiment, we observed the data processing times under the single and dual-core conditions of the ESP32 module. Under the single-core mode, both the stored data and real-time data are processed through a single-core and transmitted accordingly. But in contrast, two cores split the real-time data processing and stored data processing under the dual-core mode. For the experiment, we have selected 10 generated data samples with the sizes of, 5 kB, 10 kB, 20 kB, 30 kB, 50 kB, 75 kB, 100 kB, 150 kB, 300 kB and 500 kB. Then each data sample is processed 30 times in the ESP32 module and the average processing times were calculated. Here, we observed a significant improvement in processing time when two cores are utilized for the data processing function in ESP32 as indicated in the following Fig. 5.

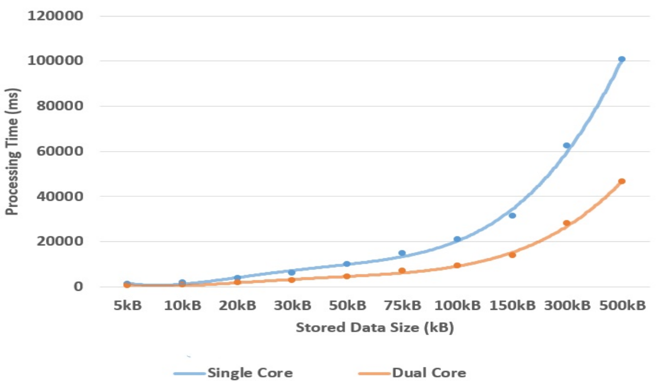

Fig. 5. Stored data size vs processing time of data in a single core and dual cores

Moreover the achieved improvement can also be summarized as in Table III, as follows.

As seen from the Table II table over 50% plus transmission time improvement can be achieved in each case through the utilization of two cores for the data processing ESP32.

B. Data transmission losses with distance

This experiment intends to detect the transmission losses related to transmission distances. First, data is sent to the mobile phone from the ESP32 continuously and for every 1000 points of data, ESP32 sends an acknowledgment to the mobile. Meanwhile mobile also keeps track of the number of packets received. With higher inter-frame delays, the mobile phone was able to exactly synchronize this acknowledgment with its count. But when the delay is lower, we observed that the counts did not match each other. The observed data is indicated in Fig.6 and it is clear that the transmission disparity increases with the increasing distance of transmission. It shows a clear variation of transmission success rate variation when the mobile is traveling away from the BLE sensor node ESP32. Moreover, we can clearly identify that more dissimilarities occur when the inter-frame delay is decreasing. Therefore, we can expect more reliable data transmission under sufficiently larger inter-frame delays.

TABLE II
Transmission Time Improvement under multi-connect

Sample Size	Transmission Time Improvement
5kB	53.33%
10kB	55.93%
20kB	55.34%
30kB	54.91%
50kB	53.67%
75kB	53.01%
100kB	55.36%
150kB	55.68%
300kB	54.95%
500kB	53.59%

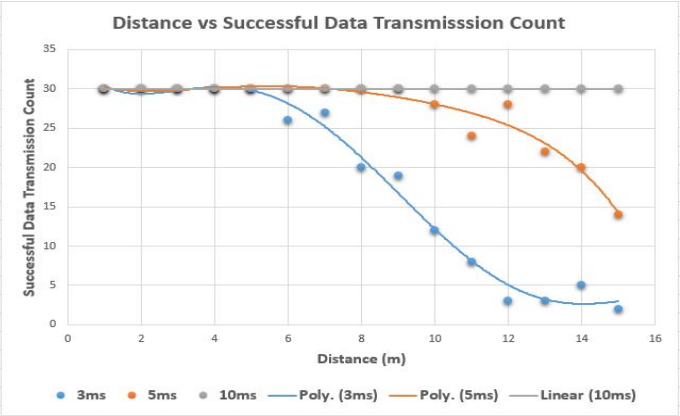

Fig. 6. Distance vs successful data transmission count

Thus, when there are multiple relays nearby, the best mobile donor should be selected who is in the closest proximity and it is implemented in our work. Also, it shows the importance of implementing a handover mechanism to mitigate the issue of data transmission loss, when the previously selected mobile is moving away from the BLE sensor. In such a situation, IoT sensor scans and connects to the new best mobile as described in the protocol description.

The handover mechanism can be illustrated as in Fig. 7. The BLE sensor can perform the handover from one mobile to another if the current host donor mobile goes away from the sensor than a threshold distance. In this case, it is halfway of 15 m distance that we have experimented. From Fig. 8, it can be observed that the successful data transmission possibility is significantly increased with the mobile handover because the distance between the BLE sensor and the mobile would not increase beyond the threshold distance when handovering is done.

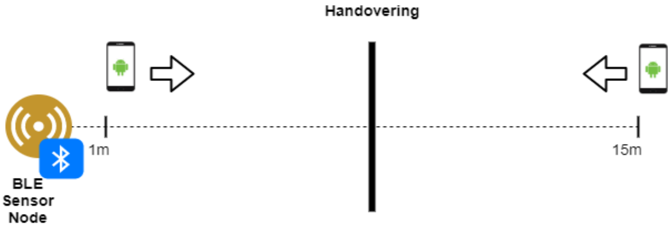

Fig. 7. Handover of mobile relays at a threshold distance

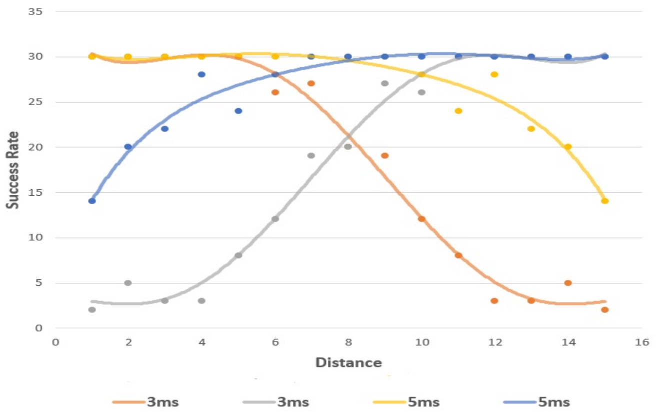

Fig. 8. Distance vs successful data transmission count when handovering under 3ms and 5ms inter frame gaps

Table III shows the added features of proposed architecture in compared with existing solutions. With the table it is clear that the proposed system is a unique solution and addresses many problems existing with similar proposals and implementations.

TABLE III
COMPARISON OF PROPOSED SOLUTION WITH THE EXISTING PERTINENT WORKS

Characteristic	Ref. [7]	Ref. [8]	Ref. [9]	Ref. [10]	Ref. [4]	**Our Proposal**
BLE Support	–	✓	–	✓	✓	✓
Support for Third party Relay	–	–	–	✓	✓	✓
E2E Encryption	✓	✓	✓	✓	✓	✓
Automatic Relay Handover	✓	–	✓	–	–	✓
Multi Connect	–	✓	–	–	–	✓
Transmission of Real-time data	✓	✓	✓	✓	✓	✓
Store and Forwarding of data	✓	✓	✓	–	–	✓
Support for Load Balancing	–	–	✓	–	–	✓

VII. Conclusion

IoT has the potential to offer an extensive support for AAL to continuously monitor patients and detect emergencies. Most of the wearable IoT devices use dedicated gateways for data transmission. However, elderly people may often forget to keep their mobile phones always charged and also the user's mobile phone might be turned off or might get damaged during an emergency. To mitigate these issues this paper proposed a BLE relay-based emergency situation detection system for AAL. The proposed solution extended the features of existing solutions by adding new capabilities such as multi-connect, automatic handovering, storage, forwarding data and load balancing.

We have performed several experiments to get insights on the performance of the proposed system. It was detected that multi-core processing of real-time and stored data separately is a better solution than processing both together sequentially. The experiments revealed that the distance of the mobile from the sensor is important due to increase in the reliability of data with closer proximity. Therefore, it can be concluded that a data transmission handover mechanism in the BLE sensor node from a distant mobile relay to a closer one is important when implementing such systems. For the future work, we intend to extend the work to develop machine learning algorithms to detect emergencies and anomalies.

Acknowledgement

This work is party supported by European Union in RESPONSE 5G (Grant No: 789658) and Academy of Finland in 6Genesis (grant no. 318927) and Secure Connect projects.

References

[1] G. Marques, "Ambient Assisted Living and Internet of Things," in *Harnessing the Internet of Everything (IoE) for Accelerated Innovation Opportunities*. IGI Global, 2019, pp. 100–115.

[2] M. Liyanage, A. Braeken, P. Kumar, and M. Ylianttila, *IoT Security: Advances in Authentication*. John Wiley & Sons, 2020.

[3] B. Farahani, F. Firouzi, and K. Chakrabarty, "Healthcare IoT," in *Intelligent Internet of Things*. Springer, 2020, pp. 515–545.

[4] P. Porambage, A. Manzoor, M. Liyanage, A. Gurtov, and M. Ylianttila, "Managing Mobile Relays for Secure E2E Connectivity of Low-Power IoT Devices," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2019, pp. 1–7.

[5] A. Manzoor, P. Porambage, M. Liyanage, M. Ylianttila, and A. Gurtov, "Mobile Relay Architecture for Low-power IoT Devices," in *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 2018, pp. 14–16.

[6] J.-W. Yoon, Y.-k. Ku, C.-S. Nam, and D.-R. Shin, "Sensor network middleware for distributed and heterogeneous environments," in *2009 International Conference on New Trends in Information and Service Science*. IEEE, 2009, pp. 979–982.

[7] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting Smart E-Health Gateways at the Edge of Healthcare Internet-of-Things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[8] S. Raza, P. Misra, Z. He, and T. Voigt, "Building the Internet of Things with Bluetooth Smart," *Ad Hoc Networks*, vol. 57, pp. 19–31, 2017.

[9] S. R. Moosavi, T. N. Gia, E. Nigussie, A. M. Rahmani, S. Virtanen, H. Tenhunen, and J. Isoaho, "End-to-End Security Scheme for Mobility Enabled Healthcare Internet of Things," *Future Generation Computer Systems*, vol. 64, pp. 108–124, 2016.

[10] M. Haus, A. Y. Ding, and J. Ott, "Managing Iot at the Edge: The Case for BLE Beacons," in *Proceedings of the 3rd Workshop on Experiences with the Design and Implementation of Smart Objects*. ACM, 2017, pp. 41–46.

[11] H. Wirtz, T. Zimmermann, M. Serror, and K. Wehrle, "Collaborative on-demand wi-fi sharing," in *2015 IEEE 40th Conference on Local Computer Networks (LCN)*. IEEE, 2015, pp. 19–27.

[12] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.