TUCS

Tuan Nguyen Gia

Design for Energy-Efficient and Reliable Fog-Assisted Healthcare IoT Systems

Turku Centre *for* Computer Science

TUCS Dissertations
No 235, December 2018

Design for Energy-Efficient and Reliable Fog-Assisted Healthcare IoT Systems

Tuan Nguyen Gia

To be presented, with the permission of the Faculty of Science and Engineering of the University of Turku, for public criticism in Auditorium XXI of Agora on December 10, 2018, at 12 noon.

University of Turku
Department of Future Technologies
20014 TURUN YLIOPISTO, FINLAND

2018

Supervisors

Professor Pasi Liljeberg
Department of Future Technologies, University of Turku
Finland

Adjunct Professor Amir M. Rahmani
Department of Future Technologies, University of Turku, Finland
Marie Curie Global Fellow
University of California Irvine, USA and TU Wien, Austria

Adjunct Professor Tomi Westerlund
Department of Future Technologies, University of Turku
Finland

Professor Hannu Tenhunen
Department of Future Technologies, University of Turku
Finland

Reviewers

Associate Professor Manuel Roveri
Department of Electronics, Information and Bioengineering
Polytechnic University of Milan, Italy

Professor Enzo Mingozzi
Department of Information Engineering
University of Pisa, Italy

Opponent

Associate Professor Mario Di Francesco
Department of Computer Science
Aalto University
Finland

ISBN 978-952-12-3773-7
ISSN 1239-1883
The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin Originality Check service

Abstract

Cardiovascular disease and diabetes are two of the most dangerous diseases as they are the leading causes of death in all ages. Unfortunately, they cannot be completely cured with the current knowledge and existing technologies. However, they can be effectively managed by applying methods of continuous health monitoring. Nonetheless, it is difficult to achieve a high quality of healthcare with the current health monitoring systems which often have several limitations such as non-mobility support, energy inefficiency, and an insufficiency of advanced services. Therefore, this thesis presents a Fog computing approach focusing on four main tracks, and proposes it as a solution to the existing limitations. In the first track, the main goal is to introduce Fog computing and Fog services into remote health monitoring systems in order to enhance the quality of healthcare.

In the second track, a Fog approach providing mobility support in a real-time health monitoring IoT system is proposed. The handover mechanism run by Fog-assisted smart gateways helps to maintain the connection between sensor nodes and the gateways with a minimized latency. Results show that the handover latency of the proposed Fog approach is 10%-50% less than other state-of-the-art mobility support approaches.

In the third track, the designs of four energy-efficient health monitoring IoT systems are discussed and developed. Each energy-efficient system and its sensor nodes are designed to serve a specific purpose such as glucose monitoring, ECG monitoring, or fall detection; with the exception of the fourth system which is an advanced and combined system for simultaneously monitoring many diseases such as diabetes and cardiovascular disease. Results show that these sensor nodes can continuously work, depending on the application, up to 70-155 hours when using a 1000 mAh lithium battery.

The fourth track mentioned above, provides a Fog-assisted remote health monitoring IoT system for diabetic patients with cardiovascular disease. Via several proposed algorithms such as QT interval extraction, activity status categorization, and fall detection algorithms, the system can process data and detect abnormalities in real-time. Results show that the proposed system using Fog services is a promising approach for improving the treatment of diabetic patients with cardiovascular disease.

Tiivistelmä

Sydän- ja verisuonitaudit sekä diabetes ovat johtavia kuolinsyitä kaikissa ikäryhmissä jonka takia niitä pidetään yleisesti vakavina sairauksina. Valitettavasti sydän -ja verisuonitauteja tai diabetesta ei voida täysin parantaa nykyisen tietotaidon ja olemassa olevan teknologian avulla. Näitä sairauksia voidaan kuitenkin tehokkaasti hoitaa soveltamalla jatkuvan terveydenseurannan menetelmiä. Terveydenhuollon korkeaa laatua on kuitenkin vaikeaa saavuttaa nykyisten terveydentilanseurantajärjestelmien avulla, sillä monesti nämä seurantajärjestelmät sisältävät useita puutteita. Puutteet järjestelmissä liittyvät liikkuvuuden rajoittuneisuuteen, energiatehokkuuden puutteeseen ja kehittyneiden palveluiden riittämättömyyteen. Tässä tutkielmassa esitetään neljän pääluvun avulla ehdotuksia ratkaisuiksi edellä esitetyille puutteille. Ensimmäinen kappale käsittelee sumutietojenkäsittelyn- ja sumupalveluiden käyttöä etäterveyden seurantajärjestelmissä terveydenhuollon laadun parantamiseksi.

Toinen kappale käsittelee sumutietoverkkoa käyttävää järjestelmää, joka tukee liikkuvuutta ja reaaliaikaista lähestymistapaa terveystilanteen seurannassa. Sumu-avusteisten älykkäiden yhdyskäytävien ylläpitämä luovutusmekanismi auttaa ylläpitämään yhteyttä anturisolmujen ja yhdyskäytävien välillä, joiden välisessä tiedonsiirrossa on vähäinen viive. Tulokset osoittavat, että ehdotetun sumuteknologiaan perustuvan lähestymistavan kanavanvaihdon viive on 10% - 50% vähemmän verrattuna nykyiseen, liikkuvuuden mahdollistavaan, huipputeknologiaan.

Kolmas kappale käsittelee ja kehittää eteenpäin neljää energiatehokasta terveydenseurantamenetelmää. Kolme energiatehokasta järjestelmää sekö niiden anturisolmut, on suunniteltu palvelemaan erityisesti veren glukoosipitoisuutta, EKG-valvontaa tai kaatumisen havaitsemista. Neljännessä järjestelmässä on yhdistetty useiden sairauksien, kuten diabeteksen ja sydän- ja verisuonitautien samanaikainen seuranta. Tulokset osoittavat, että tämä järjestelmä voi yhtäjaksoisesti toimia, käytetystä sovelluksesta riippuen, jopa 70-155 tuntia, käytettäessä 1000 mAh: n litiumparistoa.

Neljäs kappale käsittelee sydän- ja verisuonitautia sairastavien diabeetikoiden sumuavusteisen etäterveyden seuranta-IoT-järjestelmää. Useiden ehdotettujen algoritmien, kuten QT-intervallin poiston, aktiivisuustasojen luokittelun ja kaatumisen tunnistamisalgoritmien kautta, järjestelmä voi käsitellä tietoja ja havaita epänormaaleja tekijöitä reaaliaikaisesti. Tulokset osoittavat, että ehdotettu järjestelmä, jossa käytetään sumupalveluita, on lupaava lähestymistapa diabetesta sairastavien sydän- ja verisuonitautipotilaiden hoidon laadun parantamiseen.

Acknowledgements

It was a difficult decision to refuse the great offers from many global leading companies to choose a PhD journey. However, it was one of the wisest decisions. During my PhD, I have gained much more than I could think. My PhD journey is going to end and the work would not been possible without the help and support from many persons.

First of all, I would like to express my appreciation and thank my supervisors Prof. Pasi Liljeberg, Adjunct Prof. Amir M. Rahmani, Adjunct Prof. Tomi Westerlund and Prof. Hannu Tehnunen. I would like to specially thank Prof. Pasi Liljeberg for guiding, helping and supporting me as much as he could. I want to special thank Dr. Rahmani and Dr. Westerlund for helping me to improve my research skills and deal with problems during my PhD journey.

I would like to thank my co-authors and colleagues for their efforts, helping and kindness. Particularly, I would like to thank Mr. Victor Kathan Sarker, Mr. Igor Tcarenko, Mr. Behailu Negash, Mr. Iman Azimi, Mr. Arman Anzanpour, Ms. Sanaz Rahimi Moosavi, Ms. Mai Ali, Adj Prof. Ethiopia Nigussie, and Dr. Imed Ben Dhaou. Especially, I would like to express my appreciation to Ms. Mingzhe Jiang for helpful discussions, gratefulness, friendly accompaniment and magnificent help during my PhD. I would like to thank all of my friends, friendly persons, and helpful staffs for supporting and helping me during my PhD time.

I would like to thank Prof. Enzo Mingozzi and Assoc Pro. Manuel Roveri for accepting to be the reviewers of my thesis. I would like to express my appreciation to Assoc Pro. Manuel Roveri for providing the detailed comments and considering my thesis as a honored dissertation. I would like additionally to thank Assoc Prof. Mario Di Francesco for accepting to be an opponent of my dissertation.

I would like to thank for the financial support from Turku Graduate School from 2015 to 2018. Additionally, I want to thank Nokia Foundation, the Finnish Foundation for Technology Promotion, IEEE for providing financial support and travelling grants.

Last but not least, I would like to thank my family in Vietnam. Especially, I would like to thank my grand mother Le Thi Cho, my father Nguyen

Gia Hiep, my mother Nguyen Thi Thu Huong and my sister Nguyen Gia Quynh Nhu for supporting, understanding and encouraging me from my bachelor to my PhD. Without their teaching, guiding, sympathizing and advising, I could not complete my research.

List of original publications

This work discussed in this thesis is based on the following original publications:

Articles included in the thesis

1. Paper I
   Tuan Nguyen Gia, Mingzhe Jiang, Amir-Mohammad Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. "Fog computing in healthcare internet of things: A case study on ECG feature extraction." In Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 2015 IEEE International Conference on, pp. 356-363. IEEE, 2015.

2. Paper II
   Tuan Nguyen Gia, Mingzhe Jiang, Victor Kathan Sarker, Amir M. Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen, "Low-cost fog-assisted health-care IoT system with energy-efficient sensor nodes." in Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International, pp. 1765-1770. IEEE, 2017.

3. Paper III
   Rahmani, Amir M., **Tuan Nguyen Gia**, Behailu Negash, Arman Anzanpour, Iman Azimi, Mingzhe Jiang, and Pasi Liljeberg. "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach." Future Generation Computer Systems 78 (2018): 641-658.

4. Publication IV
   Tuan Nguyen Gia, Amir M. Rahmani,Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. "Fog Computing Approach for Mobility Support in Internet-of-Things Systems Mobility." In IEEE Access, pp. 36064 - 36082, 2018, Doi: 10.1109/ACCESS.2018.2848119.

5. Paper V
Tuan Nguyen Gia, Mai Ali, Imed Ben Dhaou, Amir M. Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. "IoT-based continuous glucose monitoring system: A feasibility study." Procedia Computer Science 109 (2017): 327-334.

6. Paper VI
Tuan Nguyen Gia, Victor Kathan Sarker, Igor Tcarenko, Amir M. Rahmani, Tomi Westerlund,Pasi Liljeberg, Hannu Tenhunen, "Energy Efficient Wearable Sensor Node for IoT-based Fall Detection Systems" in Microprocessors and Microsystems, 56, pp.34-46.

7. Paper VII
Tuan Nguyen Gia, Imed Ben Dhaou, Mai Ali, Amir M. Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. "Energy Efficient Fog-assisted IoT System for Monitoring Diabetic Patients with Cardiovascular Disease." In Future Generation Computer Systems, 2019 (Accepted)

Papers not included in the thesis

8. Paper VIII
Tuan Nguyen Gia, Amir-Mohammad Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. "Fault tolerant and scalable IoT-based architecture for health monitoring." In Sensors Applications Symposium (SAS), 2015 IEEE, pp. 1-6. IEEE, 2015.

9. Paper IX
Rahimi Moosavi, Sanaz, **Tuan Nguyen Gia**, Ethiopia Nigussie, Amir M. Rahmani, Seppo Virtanen, Hannu Tenhunen, and Jouni Isoaho. "End-to-end security scheme for mobility enabled healthcare Internet of Things." Future Generation Computer Systems 64 (2016): 108-124.

10. Paper X
Tuan Nguyen Gia, Mingzhe Jiang, "Exploiting Fog computing in health monitoring" in Buyya, Srirama (Ed): Fog and Edge Computing: Principles and Paradigms, Wiley STM, USA, 2018

11. Paper XI
Tuan Nguyen Gia, Mingzhe Jiang, Amir- Mohammad Rahmani, Tomi Westerlund, Kunal Mankodiya, Pasi Liljeberg, Hannu Tenhunen, "Fog Computing in Body Sensor Networks: An Energy Efficient Approach," in Proc. of IEEE International Body Sensor Networks Conference (BSN'15), 2015, USA.

12. Paper XII
Tuan Nguyen Gia, Syed M.A.H. Jafri, Masoud Daneshtalab, Ahmed Hemani, Sergei Dytckov, Juha Plosila, and Hannu Tenhunen, “FIST: A Framework to Interleave Spiking neural networks on CGRAs”, In 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, 751–758, IEEE, 2015

13. Paper XIII
Amir. M. Rahmani, Nanda Kumar Thanigaivelan, **Tuan Nguyen Gia**, Jose Granados, Behailu Negash, Pasi Liljeberg, and Hannu Tenhunen, ”Smart e-Health Gateway: Bringing Intelligence to Internet-of-Things Based Ubiquitous Healthcare Systems,” in Proc. of Annual IEEE Consumer Communications and Networking Conference (CCNC’15), pp. 826-834, 2015, USA.

14. Paper XIV
Sanaz Rahimi Moosavi, **Tuan Nguyen Gia**, Ethiopia Nigussie, Amir-Mohammad Rahmani, Seppo Virtanen, Jouni Isoaho, and Hannu Tenhunen, “Session Resumption-Based End-to-End Security for Healthcare Internet-of-Things”, in Proc. of IEEE International Conference on Computer and Information Technology (CIT2015), 2015, UK

15. Paper XV
Sanaz Rahimi Moosavi,**Tuan Nguyen Gia**, Amir-Mohammad Rahmani, Ethiopia Nigussie, Seppo Virtanen, Jouni Isoaho, and Hannu Tenhunen, “SEA: A Secure and Efficient Authentication and Authorization Architecture for IoT-Based Healthcare Using Smart Gateways”, in Proc. of 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), 2015, UK

16. Paper XVI
Igor Tcarenko,**Tuan Nguyen Gia**, Amir-Mohammad Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen, “ Energy-Efficient IoT-Enabled Fall Detection System with Messenger-Based Notification”

17. Paper XVII
Mingzhe Jiang, **Tuan Nguyen Gia**, Arman Anzanpour, Amir-M. Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen, “ IoT-based Remote Facial Expression Monitoring System with sEMG Signal” In Sensors Applications Symposium (SAS) 2016 IEEE, IEEE, pp. 1-6, 2016.

18. Paper XVIII
Victor Kathan Sarker, Mingzhe Jiang, **Tuan Nguyen Gia**, Arman

Anzanpour, Amir-M. Rahmani, Pasi Liljeberg, and Hannu Tenhunen, “Portable Multipurpose Bio-signal Acquisition and Wireless Streaming Device for Wearables”, USA, 2017

19. Paper XIX
    Imed Ben Dhaou, **Tuan Nguyen Gia**, Pasi Liljeberg, and Hannu Tenhunen, “Low-Latency Hardware Architecture for Cipher-Based Message Authentication Code,” in ISCAS 2017, Baltimore, MD, USA, 2017

20. Paper XX
    Negash, Behailu, **Tuan Nguyen Gia**, Arman Anzanpour, Iman Azimi, Mingzhe Jiang, Tomi Westerlund, Amir M. Rahmani, Pasi Liljeberg, and Hannu Tenhunen. "Leveraging Fog Computing for Healthcare IoT." In Fog Computing in the Internet of Things, pp. 145-169. Springer International Publishing, 2018.

21. Paper XXI
    Mai Ali, **Tuan Nguyen Gia**, Abd-Elhamid Taha, Amir M. Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. "Autonomous Patient/Home Health Monitoring powered by Energy Harvesting." In GLOBECOM 2017-2017 IEEE Global Communications Conference, pp. 1-7. IEEE, 2017.

22. Paper XXII
    Tuan Nguyen Gia, Nanda Kumar Thanigaivelan, Amir-Mohammad Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. "Customizing 6LoWPAN networks towards Internet-of-Things based ubiquitous healthcare systems." In NORCHIP, pp. 1-6. 2014.

23. Paper XXIII
    Jafri Syed MAH, **Tuan Nguyen Gia**, Sergei Dytckov, Masoud Daneshtalab, Ahmed Hemani, Juha Plosila, and Hannu Tenhunen. "NeuroCGRA: A CGRA with support for neural networks." In High Performance Computing & Simulation (HPCS), 2014 International Conference on, pp. 506-511. IEEE, 2014.

Abbreviations

3GPP	The 3rd Generation Partnership Project
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
AHA	American Heart Association
ANSI	American National Standards Institute
AES	Advanced Encryption Standard
ECG	Electrocardiogram
EEG	Electroencephalography
e-health	healthcare practice supported by electronic processes and communication
EMG	Electromyography
EMRs	Electronic Medical Records
HL7	Health Level Seven International
I2C	Inter-Integrated Circuit
ICU	Intensive Care Unit
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IPv6	Internet Protocol version 6
OS	Operating System
SpO2	Blood Oxygen Saturation

SPI	Serial Peripheral Interface
WBAN	Wireless Body Area Networks
WBASN	Wireless Body Area Sensor Network
WBSN	Wireless Body Sensor Networks
WSN	Wireless Sensor Networks

Contents

Part I

Research Summary

Chapter 1

Introduction

This chapter presents an overview of the most common high-risk diseases such as cardiovascular disease, diabetes, and falls which are the reason for many death and serious injuries around the world. In addition, it discusses the possible method and the current state-of-the-art technologies to deal with these diseases. Finally, the chapter presents the thesis outline. This chapter consists of four sections: (i) high-risk diseases and pervasive healthcare, (ii) Internet-of-Things and Fog computing, (iii) research questions and the target platform, and (iv) thesis outline.

1.1 High-risk Disease and Pervasive Healthcare

Cardiovascular disease is a dangerous disease and the leading cause of death in all ages. Approximately 83.6 million American people have heart-related diseases and 610 thousand people living in the US die every year [1]. It is projected that the number of people with heart diseases will dramatically increase. Cardiovascular disease is a medical term denoting many heart-related diseases with the most common being Coronary Artery Disease (CAD), heart attack, heart failure, arrhythmia, and stroke. For instance, 33 million people had a stroke in 2010, and of these it was the first stroke for 16.9 million people [2]. In [2, 3], the authors show the benefits of continuous health monitoring systems for dealing with heart-related diseases. For instance, these systems help to enable early detection of deterioration, enhance diagnosis, and improve the efficiency of the prescription process. Atrial fibrillation, in particular, seems to be the primary cause of strokes but it is often undiscovered due to the asymptomatic and irregular nature of the disease [2]. Continuous Electrocardiogram (ECG) monitoring systems have increased the possibility of identifying atrial fibrillation in patients.

Diabetes occurs when insulin cannot be adequately produced by the pancreas or the produced insulin cannot be effectively used by the body.

Insulin is a hormone in the body which regulates the amount of glucose in the blood [4]. Diabetes was ranked 7$^{\text{th}}$ in the list of the most dangerous disease in the world in 2014 [5]. In 2015, diabetes directly caused 1.6 million deaths. In addition, it is one of the primary causes of stroke, kidney damage, eye diseases, heart attack, heart failure, lung failure and other severe diseases [6].

Diabetes can be categorized into three types including type 1, type 2, and an exceptional type named gestational diabetes. In general, type 1 and type 2 diabetes are more dangerous than gestational diabetes that occurs during pregnancy and disappears after the child is born. Type 1 diabetes is the most severe among all diabetes types because the cause of type 1 is undiscovered and it cannot be prevented with the existing knowledge [4]. The number of type 1 diabetes cases is much less than type 2 diabetes cases, which often occur in adults due to overweight and lack of activity. Nonetheless, any person can develop diabetes regardless of gender, age and health status. Currently, the number of children with diabetes is increasing. For example, around 200 thousand Americans under 20 years old have diabetes [7]. Based on the statistics, the number of people with diabetes has significantly increased from 108 million in 1990 to around 422 million in 2014 [4, 5]. This number is projected to escalate by two or three times in the next 15 years [4]. These numbers indicate an alarming state of health. Therefore, healthcare and solutions for issues related to diabetes have to be carefully considered. Although diabetes cannot be cured with existing knowledge, it can be continuously monitored for on-time actions from responsible persons such as caregivers. The dangerous consequences of diabetes especially, can be lessened or avoided by properly controlling the blood glucose level.

Falls cannot be underestimated because they can cause serious injuries such as head injuries, brain damages, and bone fracture [8, 9]. More than 2.8 million adults who are over 65 years old have fall injuries annually. In addition, falls are an indirect cause of reducing the quality of life because people who have fallen often have a fear of falling which makes them lose their confidence and want to avoid physical activities [10, 11]. Although problems after fall are significant, only a half of the fall cases are reported. The hazard consequences of a fall can be cured easier or become less severe when a fall case is reported and treated in real-time.

Cardiovascular disease, diabetes, falls, and ageing have direct and indirect linkages. Old people who are over 65 years old are likely to fall more often and the number of these people having diabetes and cardiovascular disease is higher than in younger groups. Diabetes is one of the risk factors for cardiovascular disease and falls [4,12,13]. Heart repolarization and other cardiac arrhythmia's may occur when the blood glucose goes below certain levels such as 60 mg/dl. These diseases cannot be underestimated since they can cause sudden deaths. [14, 15]. Based on the statistics, three quarters of 65-years-old people with diabetes die from heart-related diseases [13].

One of the widely-used methods to scope for cardiovascular disease and diabetes is to continuously monitor bio-signals such as electrocardiogram (ECG), blood oxygen saturation (SpO2), and blood glucose. Currently, these signals are often collected and analyzed a few times per day by a nurse or a physiology specialist. However, this method has several drawbacks such as inconvenience, high possibility of inaccurate disease analysis, high costs, and low possibility of detecting emergencies in time. A particular problem is that, a diabetic patient cannot eat anything for a few hours before a blood sample is taken. To some extent, it might be inaccurate to provide a treatment procedure for diabetes based on the glucose information from the latest blood sample because blood glucose levels depend on the time of day of the blood withdrawal. In the current health monitoring protocols, each nurse can merely collect and analyze e-health signals from a patient at a specific moment. This can lead to a waste labor resources which is one of the reasons for high hospital's costs. In particular, 555 billion US dollars have been spent treating cardiovascular disease in which healthcare labor costs can be up to 50 percent of the hospitals' total expenses [16, 17]. High hospital costs reduce the possibility of providing healthcare delivery to low income and developing countries where non-communicable diseases such as diabetes and cardiovascular are increasing dramatically [18]. In addition, emergencies cannot be detected in real-time by the existing monitoring protocols; accordingly, this can cause serious consequences such as death.

Currently, hospitals often apply their own medical systems to identify patients and record patients' data. As a result, it is challenging for external doctors to access patients' data. In cases of emergency, this drawback becomes more serious. For instance, unconscious patients involved in car accidents may die due to the injected of medicines to which they have allergies. Such unforeseen cases can be avoided when patients' data such as medicine allergies, history of medicine usage and current health status is globally accessible in real-time. Based on the statistics, more than 100 thousand Americans die due to medical errors such as the wrong provision of medicine, prescriptions with the wrong dosages, the inaccessibility of the history of medicine usage and allergies, and prescriptions with the wrong treatment series [19].

To summarize, it is crucial to have a reliable and enhanced health monitoring system which overcomes the above mentioned disadvantages of the current health motoring approach. In addition, the system should be able to provide advanced services for improving the quality of healthcare.

1.2 Internet-of-Things and Fog Computing

Internet-of-Things (IoT) is a prominent technology for solving the problems mentioned. IoT can be defined as an infrastructure or platform where virtual and physical objects are interconnected and intercommunicated with each other [20]. IoT consists of many advanced technologies such as wireless sensor networks (WSN), wireless body sensor networks (WBSN), sensing and Cloud computing. This makes IoT the most ubiquitous and preferred choice in many applications and fields such as entertainment, education, and healthcare. In general, an IoT system shown in Figure 1.1 often comprises three main components such as sensor nodes, gateways, and a back-end system. Sensor nodes are used to collect and transmit data wirelessly to gateways; these act as a hub to receive data from multiple nodes and forward the data to Cloud servers via the Internet. Via an IoT system, data can be globally monitored in real-time. For instance, medical doctors can access real-time ECG, blood glucose levels, or motion-related data in both graphical and textual forms via a laptop browser or a mobile application [21,22,23,24]. In addition, the system can provide the necessary services such as data analysis and global data storage [25]. Medical doctors can access a patient's historical data and information such as allergies, diseases, and blood type. Lastly, IoT systems can use international standard formats for storing data and identifying users. Consequently, external doctors with proper rights can access a patient's data easily and globally at anytime. By applying IoT systems to healthcare, costs can be significantly reduced and the inadequacy problems of medical caregivers in many countries (e.g., developing countries) can be partially solved. For instance, caregivers do not need to visit patients physically to monitor the patient's health status and one doctor can be responsible in general cases for many patients.

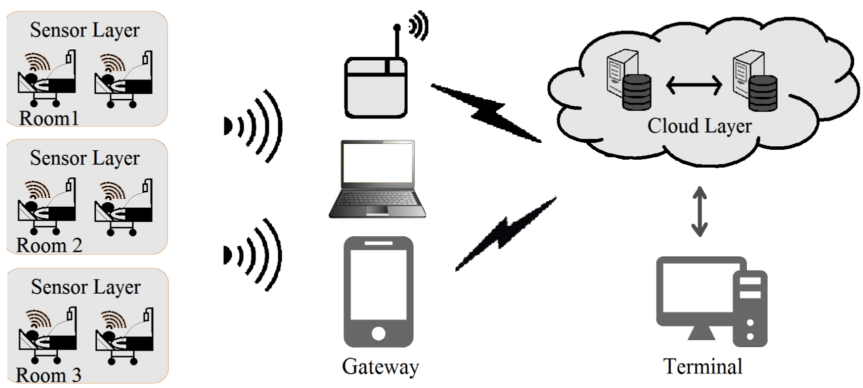


Figure 1.1: Typical architecture of an IoT system for health monitoring

In order to proffer high quality healthcare services, a healthcare IoT system must fulfill many strict requirements such as reliability, interoperability, security, low latency, and energy efficiency. When the system is not qualified for these prerequisites, it can cause serious consequences. For instance, delay in monitoring can cause an inaccuracy in the disease analysis and diagnosis. In addition, delay in detecting and informing serious abnormalities such as stroke and heart attack can cause danger to a patient. When the system's security is not guaranteed, the patient's data can be lost and even the patient's life can be endangered. An IoT-based insulin pumper, in particular, can be hacked and controlled within a range of 100 meters [26]. When some services are interrupted or they do not work properly, the quality of services cannot be maintained. Currently, the IoT systems for healthcare merely fulfill one or some of the requirements. It is challenging to combine all the services in complete harmony so as to address the objectives because of the trade-off relationship between the requirements and the limitations of the existing technologies.

For example, a high quality of data cannot be achieved when low resolutions and data rates are applied. However, when high resolutions and data rates are used, they can cause high energy consumption and the high latency of data transceiving. For instance, EMG signals should be collected with a data rate of 500 samples/s or higher to achieve high-quality signals [25]. In order to improve the accuracy of disease diagnosis and analysis, e-health signals and contextual data have to be considered because e-health data varies in different contexts. For example, heart rates during standing and running are different and an ambient temperature affects the pulse rate variability in young adults [27]. Nonetheless, obtaining all these data may increase the energy consumption of the sensor nodes.

One solution for coping with these problems mentioned is to apply advanced technologies and a new system architecture which can enhance sensor nodes, gateways, and services. The new architecture should be evolved from the existing IoT system architecture to save costs (i.e., replacement and deployment costs). Fog computing, which is the extra layer between conventional gateways and Cloud servers, is a suitable candidate. Fog computing can be described as a convergence network of smart gateways in which the gateways are interconnected and can intercommunicate with each other [28,29]. Fog computing empowers the edge of the network to enable the capability of location awareness, geographical distribution, interoperability, online analytic and other augmented, smart features. For instance, Fog computing helps to save network bandwidth by reducing the volume of the data transmitted over the network (e.g., data compression) while maintaining the high quality of services. In addition by bringing the Cloud computing paradigm to the edge of the network, Fog computing provides a means to address the unsupported fundamental features of Cloud computing.

1.3 Research Questions and Thesis Structure

Although Fog computing is able to provide some enhancements to IoT systems, the question still remains: "Is Fog computing applicable to healthcare, especially to the existing healthcare systems in order to augment the quality of the services?". When addressing this question, some challenges arise.

- Is it feasible to enhance the quality of service in healthcare IoT systems by leveraging Fog computing at the edge of the network?
- Is it possible to support mobility in healthcare IoT systems without diminishing quality of service by utilizing Fog computing?
- Is a dramatic improvement in the energy efficiency of the system achievable, especially the energy efficiency of wearable sensor nodes by combining energy saving methods and Fog computing?
- Is it possible to monitor and analyze diabetes, cardiovascular disease, and human falls simultaneously via an IoT system without infringing healthcare requirements of latency, security, and high-quality signals?

This thesis aims to answer the above questions by proposing a design for energy-efficient and reliable Fog-assisted healthcare Iot systems. The proposed design is built by considering various aspects of hardware design, system architecture, Fog services, and algorithms. Figure 1.2 presents an overview of the research work encompassed in this thesis. The four primary tracks in this thesis are Fog approach for enhancing the quality of service in health monitoring IoT systems, mobility support, energy efficiency, and Fog-assisted IoT systems for monitoring diabetics with cardiovascular disease. These tracks are discussed in the followings:

This thesis presents a Fog-assisted architecture for healthcare IoT systems. The thesis particularly presents a Fog layer which is an extra layer between traditional gateways and Cloud servers. The proposed architecture consists of sensor nodes, Fog-assisted smart gateways, Cloud servers, and end-user terminals. By exploiting the proposed architecture, advanced Fog services can be provided in order to improve the quality of healthcare service. In addition, the thesis introduces advanced Fog services consisting of local distributed database, push notification, categorization, channel management, data compression, security, mobility support and interoperability which are published in Paper I, II, and III. In this thesis, this track is presented in Chapter 4.

Mobility support is a necessary feature of health monitoring IoT systems because a patient can change location at anytime. When a system does not support mobility, data cannot be continuously monitored. However, it is challenging to support mobility in healthcare due to the strict requirements

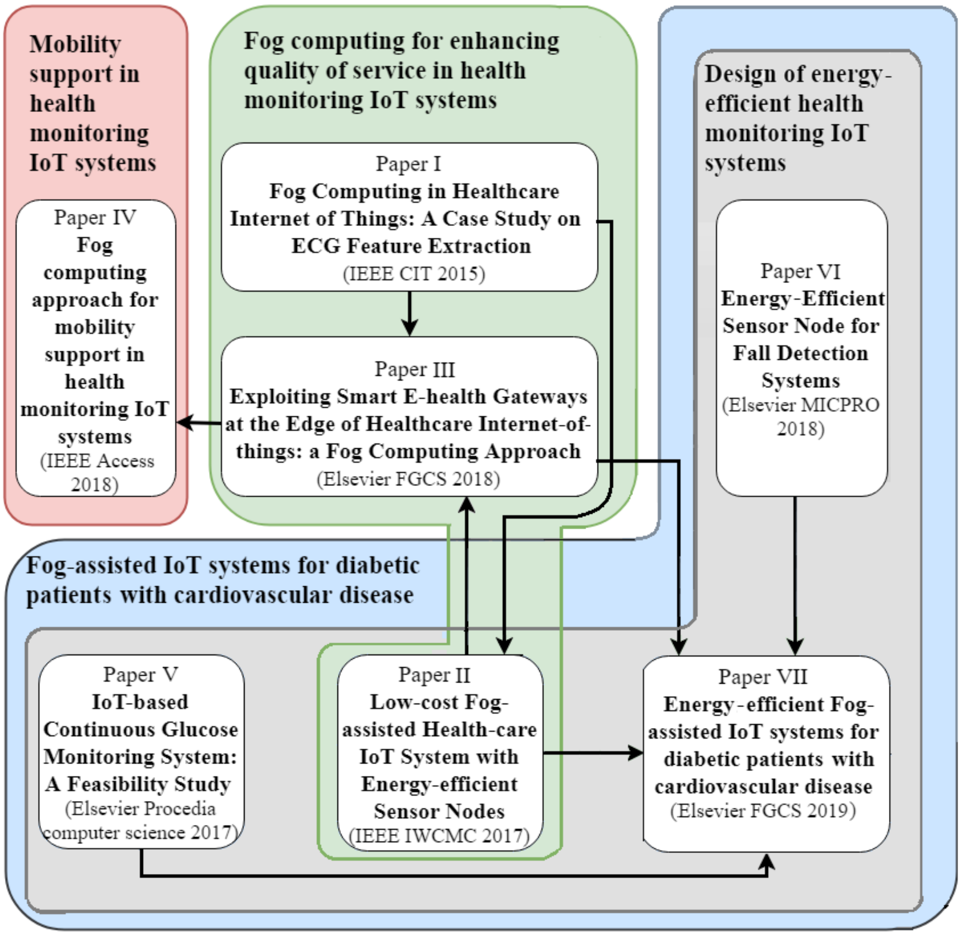


Figure 1.2: Illustration of paper cohesion

of latency. The mobility support issue becomes more challenging when dealing with Fog-assisted IoT health monitoring systems because Fog-services must be properly maintained during mobility. In this case, a mobility support mechanism not only considers mobility aspects (e.g. disconnection and re-connection) but also guarantees the continuation and stability of Fog services. In order to deal with issue, the thesis proposes a Fog approach for mobility support in health monitoring IoT systems which was published in Paper IV. In this thesis, this approach is presented in Chapter 5.

Sensor nodes for collecting bio-signals are often implanted or wearable. Therefore, their size and weight including the battery should be small and light in order to avoid interference with the user's daily activity. With the current state-of-the-art battery technology, small and light batteries cannot have a large capacity. Correspondingly, sensor nodes cannot continuously operate and provide high-quality signals over a long period of time. For instance, ECG sensor nodes can only work up to 13.6 hours with a 1700 mAh battery [30]. When sensor nodes do not function, the whole monitoring system cannot operate. In contrast, other parts of the system (i.e., gateways and a back-end system) do not have issues of sensor nodes. Hence, enhancing the energy efficiency of sensor nodes is one of the ultimate goals. To address this target, this thesis proposes energy-efficient Fog-assisted IoT healthcare systems for ECG monitoring, fall detection, and diabetic patients with cardiovascular disease. These systems consist of ultralow-power wearable sensor nodes and Fog-assisted smart gateways which help to save energy consumption of the wearable sensor nodes. The sensor nodes provide a high quality of signals and can work from 141 to 155 hours with a 1000 mAh battery depending on the systems published in Paper II, V, VI, and VII. These sensor nodes are small, light-weight and can be used without interfering with the user's daily activity. In these systems, the primary energy consumption sources of sensor nodes such as sampling rate, communication bus interface, transmission protocol, and transmission rate are investigated in order to achieve a high level of energy efficiency. In addition, many energy saving techniques related to both hardware and software are customized and applied. In this thesis, these energy-efficient Fog-assisted IoT health monitoring systems are presented in Chapter 6.

As mentioned previously, diabetes, cardiovascular disease, and falls are often related problems. A solution for dealing with these diseases is to monitor blood glucose level, ECG, and motion-related data in real-time. When abnormalities occur, the system should inform caregivers in real-time. In order to target the solution, this thesis proposes an advanced Fog-assisted system for diabetic patients with cardiovascular disease. The proposed system is combined and customized from several health monitoring systems (e.g., an ECG and contextual data monitoring system, a glucose monitoring system, a fall detection system, a system for diabetic patients with cardiovascular disease) published in Paper II, V, VI, and VII. The proposed system can monitor many types of data including bio-signals (e.g., ECG, blood glucose, body temperature) and contextual data (i.e., air quality and room temperature). In addition, the system can process the data in real-time via algorithms (e.g., fall detection, ECG feature extraction, and activity status categorization) and send push messages to caregivers in real-time to provide information on abnormalities such as high blood glucose, falls, or an abnormal ECG waveform. In this thesis, the proposed system is presented in Chapter 7.

Chapter 2

Related work

Many types of health monitoring IoT-based systems have been proposed for both hospital and home, such as, bio-signals based and camera-based systems. However, bio-signals based systems are more ubiquitous since they have more advantages; for instance, these bio-signals based systems do not limit, in most of the cases, a patient's movement. In addition, they can accurately monitor all primary vital signs (i.e., body temperature, pulse rate, respiration rate, and blood pressure) which are necessary for disease analysis. In bio-signals based IoT systems, bio-signals are collected via wearable or implanted sensors attached or implanted into the body. The acquired signals are wirelessly transmitted to a gateway which then sends the raw or pre-processed data to Cloud for storing, global accessing, and further analysis. Most of the commonly used bio-signals are ECG, body temperature, SpO2, blood pressure, blood glucose, heart rate, and respiration rate as they can be used to assess the body's most basic functions and diagnose common diseases such as cardiovascular disease, and diabetes. However, other bio-signals such as EMG and EEG can be targeted depending on the applications. For instance, EMG is mainly used in pain assessment applications while EEG is used for brain-related applications. Hence, the thesis focuses on bio-signals based IoT systems for health monitoring. The state-of-the-art IoT-related approaches for fall detection, mobility support, energy efficiency enhancement, and cardiovascular disease monitoring are discussed as follows:

2.1 Fall detection device and system

Many approaches have been proposed for detecting a human fall. For instance, a smart watch can be used to collect acceleration rates from a user's wrist and send the acquired data via Bluetooth Low Energy (BLE) to a smartphone. In the smartphone, the data is processed with a fall detection algorithm. When a fall case is detected, the phone uses 3G/4G to send push notification messages to Cloud servers which forward the messages to caregivers [31].

In [32], the authors used wearable sensors equipped with low-power micro-electromechanical systems (MEMS) for collecting acceleration from the waist and transmitting the collected data to a mobile mote or a base station via radio frequency (RF). The system can estimate a fall location based on the RF signal strength. The received data is processed with a simple fall detection algorithm.

Acceleration is often used in many fall detection applications. For instance, three-dimension (3-D) acceleration is collected via an ADXL345 sensor equipped in a small, light wearable device [33]. The collected data is transmitted wirelessly to a gateway for further processing with a fall detection algorithm. In another example, a GSM-based device collecting 3-D acceleration is used for fall detection [34]. The device processes the collected data with a fall detection algorithm. When a fall case is detected, it sends an SMS (short message service) message containing a universal resource locator (URL) link to a caregiver who checks the fall location by accessing the URL link. Another system based on 3-D acceleration is a fall detection IoT system for elderly people [35]. The system consists of sensor devices, a smart gateway, and Cloud services. The sensor device uses an ARM 32-bit Cortex-M3 processor for achieving high computational performance. In addition, a sensor device is equipped with a MEMS sensor (i.e., LSM6DS0) to acquire 3-D acceleration. The sensor device runs Contiki operating system (OS) to manage resources efficiently. Furthermore, the sensor device uses an IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) protocol for transmitting the collected data to a smart gateway which processes the data and trains the model. As a result, events including fall or ASL events are classified. When a fall event is detected, the gateway sends a notification message to a caregiver.

In other fall detection applications, 3-D acceleration is used together with 3-D angular velocity. For example, a complex fall detection algorithm based on the kNN algorithm and sliding window is applied and run at a gateway based on smartphone [36]. The fall detection algorithm uses 3-D acceleration and 3-D angular velocity which are collected from a custom-made vest and sent to a gateway via BLE. When a fall case is detected, the gateway sends an alert message and the fall location to responsible people via SMS messages. Another fall detection system also uses 3-D acceleration and 3-D angular velocity [37]. The motion-related data is collected by a device equipped with MEMS sensors (i.e., MMA7431 and ITG3200) and transmitted to a computer via Zigbee. In the computer, the data is processed for detecting a fall.

In [24, 38, 39], general purpose devices such as TelosW, Arduino Fio and LilyPad collect motion-related data (e.g., acceleration or angular velocity) from MEMS sensors. The devices then transmit the collected data via a wireless protocol such as Bluetooth or BLE to a gateway for further processing with a fall detection algorithm. As a result, a fall case can be detected and information about the case is sent to a caregiver via a real-time notification message.

2.2 Glucose monitoring systems

Many efforts have been devoted to develop remote glucose monitoring systems. For example, a continuous glucose monitoring system has been proposed for patients in intensive care units (ICUs) [40]. The system consists of glucose sensors, glucose clients and the hospital's servers. A glucose sensor collects blood glucose data every six hours and transmits the data to a glucose client via a wire cable. The glucose client, which is a personal digital assistant (PDA) device, forwards the glucose data via a Health Level-7 (HL7) message to the hospital's servers. Medical doctors can access the glucose data via a bedside device. Another proposal is an implantable glucose monitoring system [41]. Glucose is collected via an implanted sensor which transmits the data to an external unit placed on the skin. The data is then transmitted to a smartphone via BLE. Other research work has proposed a glucose monitoring system for diabetic patients using an implanted glucose sensor which can stay inside the body or under the skin for a long period of time, i.e. up to 180 days [42]. Glucose data is collected and transmitted every two minutes to an external unit placed on the skin.

Recently, IoT-based systems for blood glucose monitoring have been introduced. A blood glucose monitoring IoT system is presented in [43]. The system consists of glucose sensors, gateways, Cloud servers, and applications. A glucose sensor collects blood glucose and sends the data via Zigbee to a gateway which is a combination of Arduino Uno, Zigbee, and a computer. At the gateway, data can be locally stored or sent to Cloud servers. Medical caregivers can access the glucose data via a web-page or an application running on the computer. In another piece of research work, a blood glucose monitoring system for type 2 diabetic patients has been proposed [44]. This system consists of blood glucose sensors, gateways, and Cloud servers. Patient glucose can be collected via a glucose sensor or a blood glucose analyzer which forwards the data to a gateway. The gateway then sends the collected data to Cloud servers which can be accessed by a medical doctor anywhere and at anytime. In another glucose monitoring IoT system [45], glucose is collected from a non-invasive opto-physiological sensor connected to a TelosB device which sends the acquired data to a gateway via 6LoWPAN. The data is then forwarded to Cloud servers for remote access. In [46], the authors propose a glucose monitoring and management IoT-based system. The system encompasses sensor devices, gateways, servers, and terminal applications. Blood glucose is acquired and transmitted by a sensor device to a gateway via 6LoWPAN. The gateway then forwards the data to Cloud servers for storing and remote access. In addition, the system supports RFID for checking a patient profile and a caregiver authority. Only an authorized person can access the collected data via a web-page.

2.3 ECG monitoring systems

Many approaches have been proposed for remotely monitoring ECG in real-time. In [21, 22], remote ECG monitoring systems are presented. The systems consist of wearable sensor nodes, smart gateways, Cloud servers, and terminal applications. A sensor node was built from a low-power wireless micro-controller (TI CC2538) and an analog front-end (ADS1292). The sensor node collects and transmits 2-channel ECG data to a smart gateway via 6LoWPAN. The sensor node is managed by Contiki OS, which efficiently controls the tasks. The smart gateway was constructed by a combination of TI CC2238, SmartRF06 board, and Pandaboard forwards the collected data to Cloud servers for remote access. In addition, the smart gateway proffers some services such as local data storage and data tunneling. A caregiver can remotely access ECG in real-time via a web-page.

In other research, the authors have proposed a remote health monitoring IoT system [47]. The system encompasses wireless sensor nodes, smart gateways, a back-end system consisting of Cloud servers, and end-user terminals. The sensor node collects not only e-health data (e.g., body temperature and ECG) but also contextual data such as room temperature and humidity. The collected data can be sent to a smart gateway via one of the protocols such as BLE, 6LoWPAN, and Wi-Fi. At the smart gateway, the data is processed, compressed, and locally stored in a distributed database. In addition, the smart gateway offers some services for detecting abnormalities (e.g., high or low heart rate) and sending a push notification message to the responsible persons in real-time. Caregivers such as medical doctors can remotely access contextual and e-health data in both textual and graphical forms.

A remote ECG monitoring IoT system is introduced [48]. The system includes sensor nodes, a conventional gateway, Cloud servers, and end-user applications. A sensor node is built from Arduino UNO, AD8232, and Raspberry Pi. The sensor node collects and sends 2-channel ECG to Rasberry Pi which transmits the data to a gateway. The gateway then sends the data to Cloud servers. An end-user can remotely access the real-time ECG data via a web page.

An ECG monitoring IoT-cloud based system consisting of sensor nodes, a traditional gateway, and Cloud servers is presented in [49]. Sensor node acquires ECG via an AD8323 sensor, then transmit the data to a traditional gateway. At the gateway, the data is forwarded to Cloud servers for storing and processing such as data cleaning and analysis. Real-time data be accessed via GUI shown in a web page.

In further research, the authors have introduced a remote ECG monitoring system for cardiovascular disease [50]. The system has three main layers consisting of sensing, transport, application layers. The sensing layer with wireless sensor nodes is responsible for collecting and transmitting ECG and SpO2 to the transport layer which is built from a computer or a PDA device. An end-user can access real-time e-health data via a graphical user interface (GUI) which retrieves the data from remote Cloud servers.

In [51], the authors present a remote ECG monitoring system consisting of sensor nodes, a local controller, a router, Cloud servers, and a remote controller. The sensor node includes both sensing and actuating nodes. The sensing node collects 3-D acceleration and ECG, then transmits the data to a local controller via RF while the actuating node is responsible for receiving commands from the local controller supporting local data storage. An end-user such as a caregiver can use the remote controller to access the real-time e-health data and control the actuating nodes.

2.4 Mobility support for remote health monitoring systems

A mobility awareness approach for remote health monitoring system is proposed in [52]. The approach is based on an efficient hand-over protocol and a wireless body sensor node network. The approach helps to dramatically reduce the lost packet rate when a patient moves at a low speed such as 0.5 m/s. Results show that the performance of the approach is influenced by the wireless body area sensor network (WBASN) coordinator's position.

In other research [53, 54, 55], the authors present mobility awareness approaches for remote health monitoring 6LoWPAN-based systems in hospitals. These approaches help to maintain the connection between a sensor node and the remote monitoring system without infringing the latency requirements. In these approaches, Neighbour Discovery messages are used to avoid any additional exchange of messages. Correspondingly, the workload of the sensor node can be alleviated in order to save energy consumption. In addition, these approaches support fault tolerance.

A mobility support approach for remote health monitoring ZigBee-based systems is proposed in [56]. The approach using a Zigbee mobile manager helps to reduce management costs and enables the patient movement between different places covered by a fixed ZigBee network.

A design framework for mobility support in wearable health monitoring systems is presented in [57]. The framework is based on a multi-layer architecture consisting of hardware, communication, low-level processing, semantic processing, system operation, and front-end layers. The framework can be used as a starting point for providing the full mobility support in wireless body sensor networks. The framework proposes to consider patient context constraints during mobility for achieving an accurate evaluation. In addition, the framework provides conceptual recommendations on a three-tier architecture system consisting of sensor nodes, gateways, and a back-end part.

Chapter 3

Bio-signals and collecting approaches

In remote health monitoring IoT applications, in order to achieve real-time data streaming and a high quality of data, requirements of latency and data rate must be fulfilled. Disease analysis and diagnosis might be inaccurate when these requirements are not completely satisfied. Each bio-signal has its own requirements of latency and data rate which are shown in Table 3.1 [58]. In some critical or particular applications which require the high quality of data and accuracy of analysis, a minimum sampling rate requirement can be stricter than the commonly used values shown in Table 3.1. For instance, the ECG of a baby should be acquired with a data rate of at least 500 samples/s whilst the ECG of an adult can be obtained with a data rate of 125 samples/s [59].

Table 3.1: Requirements of latency and data

Bio-signal	Latency	Data rate
Acceleration	300 ms	20 samples/s
Angular Velocity	300 ms	20 samples/s
Body Temperature	1 s	1 sample/s
Blood Glucose	several minutes	1 sample/ 10 minutes
Blood Pressure	1s	1 sample/s
ECG	250 ms per channel	125 samples/s per channel
EEG	350 ms per channel	240 samples/s per channel
EMG	15.6 ms per channel	500 samples/s per channel
Pulse Oximeter	1 s	1 sample/s
Heart Rate	1s	1 sample/s

Table 3.1 shows that ECG, EMG, and EEG must be acquired with high sampling rates and the latency requirements of these signals are strict to

a few hundreds of milliseconds. Therefore, dealing with these signals is more challenging than other signals such as blood glucose whose maximum latency can be in minutes. When the number of sensor nodes and the number of monitoring channels increase, the bandwidth requirement is raised proportionally. For instance, the minimum bandwidth of a sensor node for successfully monitoring 9-channel ECG is equal or higher than 18 kbps.

3.1 Bio-signals

Depending on the remote health monitoring applications, particular bio-signals such as ECG, EMG, EEG, EOG, blood glucose, and body temperature can be collected. This thesis simply focuses on bio-signals which are primarily used by our proposed remote health monitoring systems shown in the later chapters. Detailed information about the bio-signals are discussed in the following:

3.1.1 ECG

Electrocardiography (ECG) interchangeable with electrocardiogram (EKG) was firstly introduced by Einthoven in the years between 1890-1900 [60]. ECG can be defined as a measurement recording electrical activities in the heart during a certain time period. ECG is often used by medical experts to detect cardiovascular disease. An example of an ECG cycle is shown in Figure 3.1.

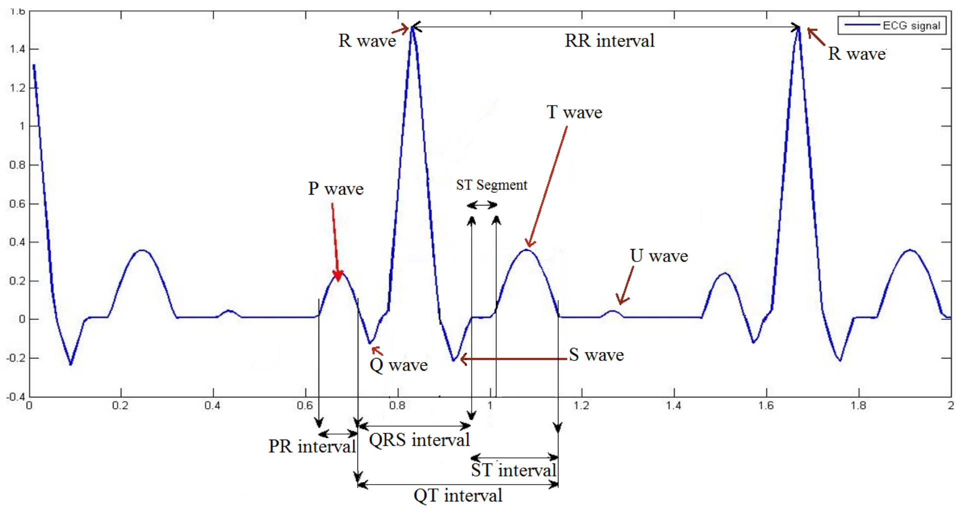


Figure 3.1: An ECG waveform [61]

ECG consists of several waves such as Q, R, S, P, T, and U. Each wave indicates the distinct activities in a particular area of the heart. Some waves such as Q, R, S appear clearly in every ECG cycle while some other waves such as U wave may not appear.

P wave

The P wave reflects atrial depolarization. Its amplitude is often between 0.05 and 0.25 mV. The P wave is often a small and smooth wave and during a sinus rhythm, the P wave is always positive in lead II. The P wave may represent two humps such as depolarization of the right and left atrium because the atria are not simultaneously depolarizing. The P wave duration is often less than 80 ms [61]. Two primary types of abnormal P-wave are P mitrale and P pulmonale which are the consequence of left and right atrial enlargement, respectively.

PR interval

The PR interval is measured from the onset of the P wave to the onset of the QRS complex. The PR interval represents the interval between the start of atrial depolarization and the start of ventricular depolarization. The normal PR interval is between 120 and 200 ms [61].

QRS complex

The QRS complex reflects the ventricular depolarization. QRS duration is between 80 and 100 ms. When QRS duration is larger than 120 ms, it indicates that a patient may have a bundle branch block, hyperkalemia, or pre-excitation. The amplitude of the R wave in the QRS complex should be less than 2.6 mV in V5 and V6 while the amplitude of the R wave in lead I, II, III should be less than 2 mV.

T wave

The T wave reflects ventricular repolarization. The T wave of an adult is often positive in most of the leads. In some cases, a negative T wave called an inverted T wave occurs. The inverted T wave is concordant with the QRS complex. The T wave's amplitude is the largest in lead V2-V3. However, the T wave's amplitude should be less than 0.8 and 1 mV for women and men in lead V2-V3, respectively. In lead II, the positive T wave is often less than 0.6 mV. The T wave appears after the occurrence of the QRS complex. When the T wave's amplitude is higher than the mentioned values, it indicates that it is highly possibly that the patient has hyperkalemia.

QT interval

The QT interval reflects the total time for depolarization and repolarization. When the QT interval is prolonged, it indicates that there is a highly possibly that the patient has a life-threatening ventricular arrhythmia. The QT interval is the opposite of the heart rate. The QT interval decreases at high heart rates and vice versa. The QT interval can be used to calculate a corrected QT duration. There are different formulas to calculate corrected QT. Currently, Bazett's QTc (QTcB) is used as a clinical standard formula which is expressed as follows:

$$QTc = QT/(\sqrt{RR})$$

where QT: QT interval (s)
RR: RR interval (s)

ST segment

The ST segment reflects the early part of the ventricular repolarization. The normal ST segment is often flat and isoelectric.

U wave

The U wave occasionally appears in an ECG wave. The U wave is the most visible in leads V2-V3 and more prominent during a slow heart rate. The amplitude of the U wave is often equal to one-third of the T wave amplitude. A negative U wave is rare. The U wave is a strong indicator of hypertension and ischemic heart disease.

ECG monitoring system

ECG is often measured via electrodes attached to body skin at specific locations. Depending on the number of electrodes, ECG monitoring systems are specifically named such as 3-lead, 5-lead and 12-lead ECG systems. These systems are noninvasive recording medical equipment widely used in many healthcare centers. A 3-lead ECG monitoring system is often used in pre-hospital care while other ECG systems are often preferred in hospitals since they can provide more detailed information about ECG. The electrode placements of 3-lead and 12-lead ECG monitoring systems are shown in Figure 3.2 and 3.3, respectively.

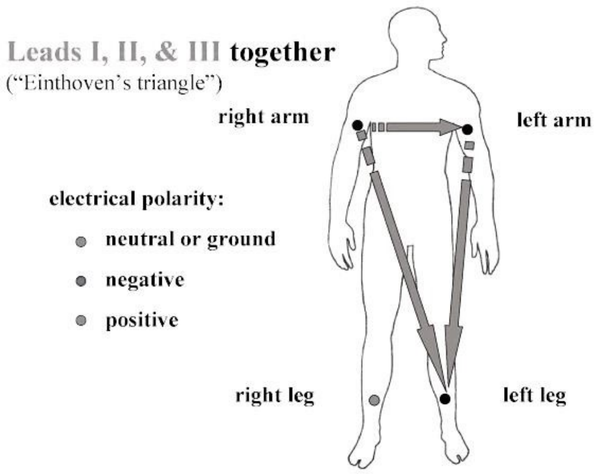


Figure 3.2: The 3-lead ECG monitoring system [62]

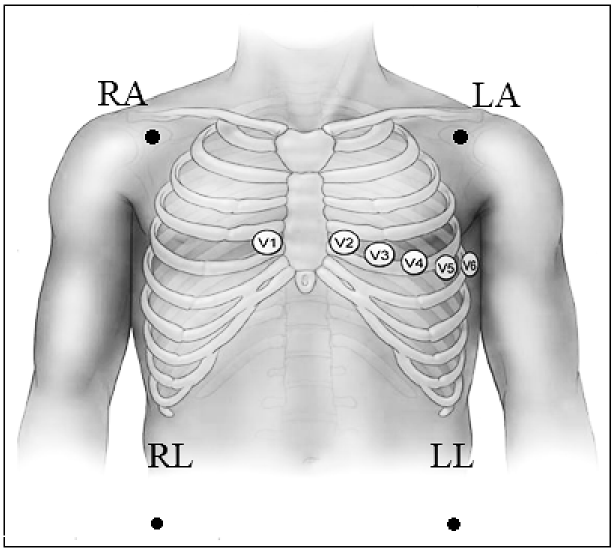


Figure 3.3: The 12-lead ECG monitoring system [63]

In order to avoid or lessen the effects of surrounding noise sources which negatively impact on the quality of ECG signals, it is recommended that all electrical devices around ECG monitoring systems should be powered off during the monitoring process; this helps to avoid or lessen for instance the strong electromagnetic energy emitted by the cellular phone.

3.1.2 Heart rate

Heart rate can be expressed as the number of heartbeats per minute (bpm). The heart rate is one of the vital signs indicating the status of the body's primary functions. The heart rate can vary depending on physical activities such as sleeping, resting, walking, running or body-building. In general, when the intensity level of physical activities increases, the heart rate is raised. The heart rate particularly varies according to the volume of oxygen needed during each activity. For instance, the normal heart rate range during sleeping is between 40 and 50 bpm whilst the normal heart rate of a 40-year-old person during running is between 89 and 125 bpm [64]. In addition, the heart rate can vary according to the air temperature, emotions, body size, and meditations. There are many studies of heart rate [59,65,66]. Some show that the normal heart rate of an adult during resting is between 50 and 90 bpm whilst American Heart Association (AHA) mentions that the normal heart rate range of an adult is 60-100 bpm. However, the range of the normal heart rate defined by the AHA is more commonly used in many health monitoring applications in hospitals. Therefore, this range proposed by the AHA is also used in this thesis. When the heart rhythm is abnormal, this is referred to as arrhythmia. The primary types of arrhythmia are atrial fibrillation, ventricular fibrillation, heart block, bradycardia, and tachycardia. When a heart rate is lower than 60 bpm or higher than 100 bpm during resting, it is called bradycardia or tachycardia, respectively [67]. Arrhythmia can occur in all ages but elderly people are more likely to have atrial fibrillation which is one of the primary causes of stroke.

Table 3.2: Target heart rate based on age during exercise

Agel	Target Heart Rate zone (50-85%)	Predicted maximum heart rate
20	100-170	200
25	98-166	195
30	95-162	190
35	93-157	185
40	90-153	180
45	88-149	175
50	85-145	170
55	83-140	165
60	80-136	160
65	78-132	155
70	75-128	150
75	73-123	145
80	70-119	140

There have been many efforts to measure and calculate the maximum heart rate of a person [68, 69, 70]. The formula by Fox and Haskell is commonly used. The formula is expressed as follows:

$$HRmax = 220 - age$$

Based on medical experiments, the target heart rate is often equal to 50-85% of the maximum heart rate [71]. Results of these heart rate types are shown in Table 3.2. The results indicate that the target heart rate and the maximum heart rate reduce with age.

3.1.3 Acceleration, Angular velocity, fall detection

Acceleration and angular velocity can be explained as the changing rate of velocity and angular position of an object with respect to time. Three dimensional (3-D) acceleration and 3-D angular velocity can be used to detect human falls since a sudden fall can cause a dramatic change of acceleration and angular velocity. In particular, the sum vector magnitude (SVM) of 3-D acceleration and the sum vector magnitude of 3-D angular velocity remain at 1 g and 0 deg/s during resting, respectively. When a person falls, these values rise rapidly to a peak value then drop, as shown in Figure 3.4. By applying threshold-based methods, a human fall can be properly detected. Threshold values specially, are set in between normal values (i.e., 1 g and 0 deg/s) and peak values (e.g., 1.7 g and 1.8 deg/s). These sum vector magnitude values can be calculated by the following formulas.

$$SVM_i = \sqrt{{x_i}^2 + {y_i}^2 + {z_i}^2} \tag{1}$$

$$\Phi = \arctan\left(\frac{\sqrt{y_i^2 + z_i^2}}{x_i}\right) * \frac{180}{\Pi} \tag{2}$$

$$DSVM_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \tag{3}$$

SVM: Sum vector magnitude
i: sample number
x,y,z : accelerometer value or gyroscope value of x, y, z axis
Φ : the angle between y-axis and vertical direction
$DSVM$: Differential sum vector magnitude

In order to collect both 3-D acceleration and 3-D angular velocity, accelerometer and gyroscope sensors are used and placed on particular locations on the body such as the wrist, chest, or waist.

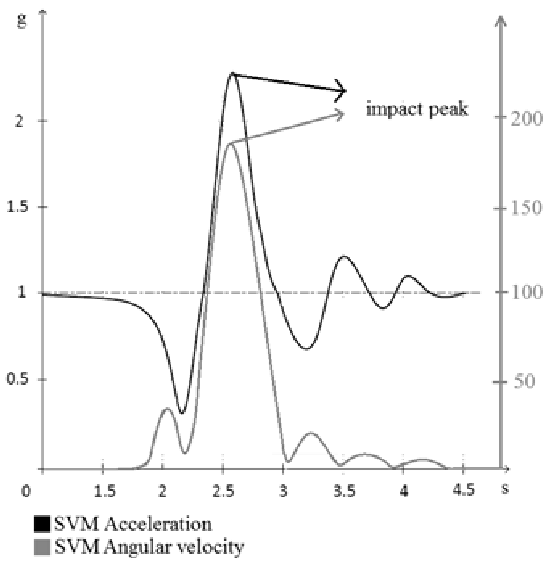


Figure 3.4: Acceleration and angular velocity change during a fall

3.1.4 Glucose and Blood glucose level

Glucose is produced, stored and released by the liver depending on the need of the body. Glucose is important for the body as it is the primary energy source of cellular respiration, the brain and other parts of the body.

The blood glucose level is the amount of glucose in human blood. The blood glucose level is regulated by the body as a part of metabolic homeostasis [72]. When the blood glucose level is low, glucagon is released by alpha cells in the pancreas. Correspondingly, glucose is released by the liver into the blood. In contrast, insulin is released by beta cells in the pancreas when the blood glucose level is high. Accordingly, glucose from the blood is taken in by fat cells.

The blood glucose level can vary depending on the time of the day and on the meals eaten. For instance, the blood glucose level is often low in the morning and before the first meal of the day. In contrast, the blood glucose level increases after meals. The blood glucose level can also vary depending on certain drugs [73].

Instances of a persistently high blood glucose level over a period of time is referred to as hyperglycemia. In contrast, a persistently low blood glucose level over a period of time is referred to as hypoglycemia. Diabetes mellitus, which is often called as diabetes, is a dangerous disease in which high blood glucose levels occur over a long period of time. Diabetes is one of the primary causes of many severe and life-threatening diseases such as cardiovascular disease, stroke, and kidney failure.

The three main types of diabetes are type 1, type 2 and gestational diabetes [72]. Type 1 diabetes can be expressed as a loss of beta cells in the pancreas, which leads to a deficiency of insulin. Type 2 diabetes is a type of insulin resistance. When a person has this type of diabetes, his or her body does not use insulin properly. In type 2 diabetes, sufficient insulin cannot be created to maintain the blood glucose at the appropriate level. Excessive body weight and insufficient exercise are considered to be two of the most common reasons for developing type 2 diabetes. Gestational diabetes occurs when a pregnant woman has a high blood glucose level. Unlike other types of diabetes, gestational diabetes often disappears after the baby is born. However, it is recommended that women with gestational diabetes need to be continuously monitored by caregivers.

3.1.5 Body temperature

The human body temperature known as normothermia or euthermia is one of the vital signs of life. The normal human body temperature range is between 36.5–37.5 degrees Celsius [74]. In most cases, the body temperature is 37 degrees Celsius. Body temperature varies according to many aspects such as age, infection, activity level, and emotional level. Body temperature has a relationship with many severe diseases. For instance, a human body temperature of under 36.5 degrees Celsius is associated with increased mortality and organ failure in cases of patients with severe sepsis [75].

Body temperature can be measured at different places on the body such as in the rectum, in the mouth, under the arm, in the ear, and in the nose. The measurement location may affect the measurement result. In addition, the measurement can be taken via different methods or tools such as a medical thermometer or temperature sensor. In this thesis, temperature sensors are used because they can be integrated into a wireless sensor node and provide accurate values.

Some concepts related to body temperature are fever, hyperthermia, hypothermia, and basal body temperature. A person has a fever when the body temperature is higher than his or her own normal body temperature. For instance, when an early morning temperature of a person is 38 degrees Celsius which is higher than his/her normal temperature (i.e., 37.2 degrees Celsius), this person might have a fever.

A person has hyperthermia when the body dissipates less heat than the produced or absorbed heat. Hyperthermia with a temperature of 40 degrees Celsius is severe and needs to be immediately cared for by medical professionals. When hyperthermia occurs, it can cause fatigue, headache, confusion and many other symptoms.

Hypothermia occurs when the body temperature is below the normal body temperature about 1-2 degrees Celsius. A person with hypothermia often feels cold.

Basal body temperature is the lowest body temperature maintained during sleeping or resting. The basal body temperature is often measured immediately when a person wakes up. Although this method cannot provide an actual basal body temperature accurately, it is accepted by healthcare centers.

3.2 Wireless protocols in health monitoring IoT systems

In health monitoring IoT systems, sensor nodes collect and transmit data via a wireless protocol (e.g., 6LoWPAN, nRF, or BLE) which has several advantages and disadvantages. Depending on the application such as glucose or EEG monitoring, a specific protocol is chosen. For instance, 48-channel EEG systems often use Wi-Fi as a primary wireless communication protocol as it needs to transmit a large amount of data whilst glucose monitoring systems can use BLE to save energy consumption. Information about widely used wireless protocols in health monitoring IoT systems is shown in Table 3.3 [58]. In this chapter, a focus is not placed on some long-range protocols such as 3G, 4G, and 5G because they are not often used for Fog-assisted smart gateways.

Generally, there are two perspectives to the categorization of wireless protocols. From the first perspective, wireless protocols for healthcare applications are categorized into short and long-range protocols whilst from the second, wireless protocols for health monitoring are categorized into low and high bandwidth/data rate protocols. Short-range wireless protocols are preferred because they are more energy efficient than long-range protocols when applying a similar data rate. For instance, Wi-Fi is more preferable than 4G when both protocols are available. Similarly, low data rate protocols are preferred because they are more energy efficient than high data rate protocols.

3.3 Overview of e-health sensors

E-health sensors (devices) can be categorized into different types such as implanted, wearable or digestive sensors. However, this thesis only focuses on wearable and implanted sensors which are widely used in many remote health monitoring IoT systems including our proposed systems [76, 77, 78]. For instance, implanted glucose sensors can be used for blood glucose level monitoring while wearable sensors such as accelerometer and gyroscope are

Table 3.3: Target heart rate based on age during exercise

Technology	Frequency	Data Rate	Distance	Peak Current	Network Topology
Zigbee	868/ 915 MHz, 2.4 GHz	250 Kbp	10-200 m	< 20 mA	Star/ Mesh
Classic Bluetooth	2.4 GHz	1-3 Mbps	10-100 m	< 40 mA	Scannet net/Star
Bluetooth 4.0 (BLE)	2.4 GHz	1/2 Mpbs	10-100m	< 18 mA	Star/ Mesh
Lora	433/ 868/ 915 Mhz	27 Kbps	22 Km	< 39 mA	Star
nRF	2.5 GHz	250 Kbps, 1/ 2 Mbps	1-1000 m	< 17 mA	Star/ Mesh
Wi-Fi	2.4 /5 GHz	11/ 54/ 450 Mbps	50 m	< 330 mA	Star
MICS	402-405 MHz	500 Kbps	2 m	2.7-35 mA	Star

used for motion tracking applications.

Wearable sensors

Wearable sensors (devices) can be used for collecting different bio-signals such as ECG, blood pressure, body temperature, heart rate, and motion-related data [76, 77, 78]. Many efforts have been dedicated to develop these wearable sensors. For instance, Emotive presents a wearable device for multi-channel EEG monitoring in which each channel can support a data rate of 128 samples/s. The device can work for up to 4 hours and transmits the collected data via BLE. Sarker *et al.* proposes a wearable device for collecting ECG and EMG. The device is small, lightweight and equipped with a Bluetooth communication chip. In [79], the authors present a wearable device for fall detection. The wearable device equipped with GSM can send information about a fall case to a caregiver in real-time. Depending on the applications, a specific wearable sensor or a group of sensors can be integrated into a wearable device. For example, our sensor nodes for fall detection, ECG monitoring, and body temperature consists of an analog front-end device, temperature sensors, a 3-D accelerometer, and a 3-D gyroscope.

Wearable devices are usually powered by a battery. Therefore, it is required that wearable devices have to be energy efficient for operating over a long period of time. The working duration of a wearable device can vary according to the battery capacity, the data rate of signals and the transmission rate. For example, with the same battery, a wearable device for heart rate monitoring can function longer than a wearable device for ECG monitoring because ECG has to be acquired at a high data rate (125-250 samples/s) while the heart rate can be collected with a data rate of 1 sample/s. In addition, wearable devices have to be small and lightweight to avoid interfering with the user's daily activities.

In remote health monitoring systems, data collected from wearable sensors are transmitted to a gateway via one of the wireless communication protocols such as Bluetooth Low Energy (BLE), Wi-Fi, nRF, Zigbee, or 6LoWPAN. As mentioned, the choice of a wireless protocol depends on healthcare applications.

Implanted sensor

Currently, implanted sensors (devices) are widely used in many health monitoring applications such as core body temperature monitoring and glucose monitoring. Due to the tiny size and long working duration, implanted sensors can be used in specific applications where wearable and other sensors cannot be deployed or it is difficult to collect bio-signals. For instance, the traditional method of testing blood glucose level needs a blood sample from a patient. This method may have some limitations. For example, a blood sample at the collected time cannot accurately reflect the glucose patterns affiliated with daily activities. In addition, it may cause some pain and panic for a patient, especially for a child. Implanted sensors can help to overcome these limitations. Implanted sensors are often equipped with one or several types of energy harvesting such as thermal and radio frequency energy harvesting [80, 81]. Implanted sensors often use the medical implant communication service which operates at a frequency band between 401 and 406 MHz and supports bi-directional communication.

Chapter 4

Fog computing for enhancing quality of service

Many IoT-based systems for remote health monitoring have been proposed [21, 25]. In these IoT systems, the 3-layer architecture including sensor nodes, gateways, and a back-end layer is often used. Sensor nodes can consist of e-health nodes, contextual sensor nodes or both. E-health sensor nodes collect e-health and e-health-related data such as ECG, heart rate, body temperature, blood glucose, 3-D acceleration, and 3-D gyroscope while contextual sensor nodes acquire information about surrounding environments and contextual statuses such as room temperature, humidity, and air quality. It is recommended that both e-health and contextual data should be collected since they help to improve the accuracy of disease analysis and diagnosis. For instance, heart rate can vary due to the air quality. The heart rate variability, in particular, can be large when the air is polluted [82, 83]. However, some of the health monitoring systems merely collect e-health data without considering contextual data. These sensor nodes send the collected data to a conventional gateway which simply forwards the data to Cloud servers for storing and further processing. Although the 3-layer architecture helps to overcome some challenges of the tradition health monitoring systems, it still has some limitations. For instance, many systems cannot support mobility, local data storage, and fault tolerance. Accordingly, the monitoring service can be interrupted when a gateway does not function or patient movement occurs. One proper solution for such challenges is to add an extra layer called a Fog layer between smart gateways and the Cloud layer. Fog can be defined as a convergent network of smart gateways where gateways are interconnected and can intercommunicate with each other. Each smart gateway in a Fog layer has information about other gateways in the network such as gateway ID and gateway status (e.g., active or inactive). A Fog layer not only helps to overcome the existing limitations but also prof-

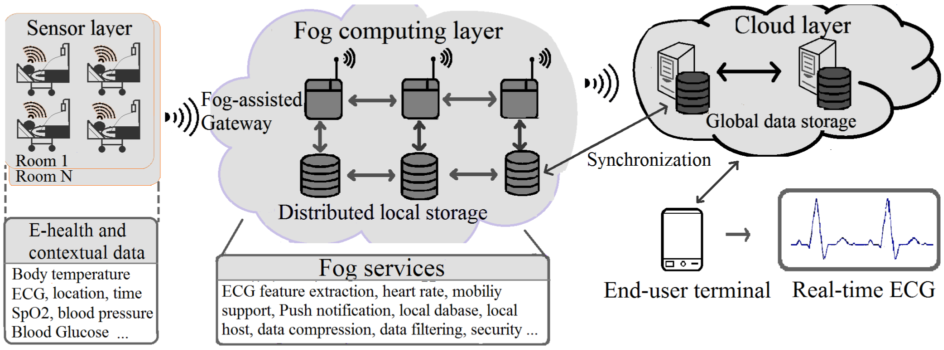


Figure 4.1: Fog-assisted IoT system for health monitoring

fers advanced services to enhance the quality of healthcare. For instance, a Fog layer provides local distributed data storage and location awareness. Fog can help to dramatically reduce the volume of transmitted data in order to save network bandwidth between smart gateways and Cloud servers. In addition, Fog can provide interoperability to support sensor nodes equipped with different wireless communication protocols. Furthermore, Fog proffers real-time online analysis at smart gateways and real-time notification of the emergency. Paper I introduces a Fog-based 4-layer architecture system for health monitoring, as shown in Figure 4.1. The Fog-based architecture is used as the primary architecture for the proposed energy-efficient and reliable healthcare IoT systems. The proposed systems provide the advanced Fog services shown in Figure 4.2. These Fog services are located on the top of the software stack of smart gateways. These proposed Fog services are discussed in detail in the following:

Distributed Databases

The distributed database of Fog-assisted smart gateways helps to maintain a high quality of service. For instance, the distributed database helps to maintain a continuous real-time monitoring when a connection between smart gateways and Cloud servers is temporarily interrupted. In this case, e-health and other data in the distributed database can be directly accessed by authorized persons or devices. The distributed database is involved in many Fog services such as security, channel management, categorization service, and mobility support. The distributed database can be considered as one of the most important elements of Fog services.

Each Fog-assisted smart gateway has its own distributed database consisting of static database and temporary database. The static database stores parameters and configurations used by algorithms and mechanisms. For instance, sensor nodes' id, sensor nodes' MAC address, users' id, and

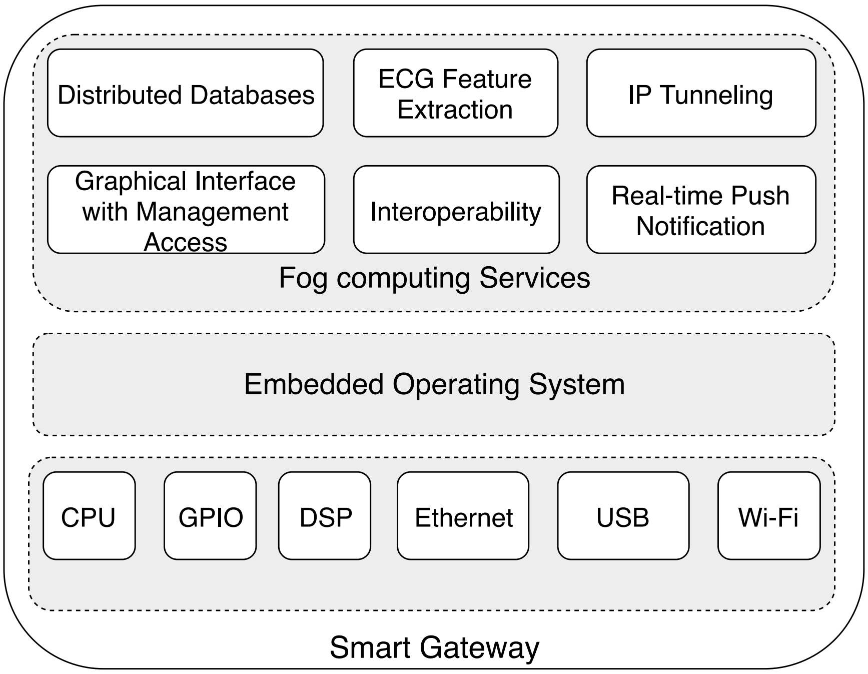


Figure 4.2: Operational gateway structure

users' password are stored in the static database. In most cases, static data is kept intact except for modification by the system administrators, other Fog services or an update from the system. For instance, after a user changes a password in a graphical interface, the system updates the password table in the static database. In another example, one part of the static database of a Fog-assisted smart gateway such as "connected device" table is updated and synchronized with the similar table in a static database of adjacent smart gateways when a mobility support service occurs. The temporary database stores e-health and contextual data such as ECG, blood glucose level, body temperature, motion-related data (i.e., 3-D acceleration, and 3-D angular velocity), room temperature, humidity, and air quality. The temporary database only stores new coming data during a short period of time and the oldest data will be replaced by the new coming data. The data of this database is always forwarded and stored on Cloud servers. The type of the distributed database can be SQL or NoSQL depending on the applications while the maximum supported volume size of the database can vary depending on a structure and an operating system of Fog-assisted smart gateways.

ECG Feature Extraction

As mentioned previously, ECG consists of different waves such as P, Q, R, S, T, U waves. Each wave can be used to diagnose the functionality of specific areas of the heart. Therefore, it is necessary to extract the ECG features. Depending on the applications, more focus can be placed on specific ECG features. For instance, prominent U wave and prolonged QRS duration indicate hypokalemia or P wave can be used to detect valvular heart disease. One of the most concerning ECG features is the heart rate because it is a risk factor for many cardiovascular diseases and it can be used for many applications. For instance, training individuals should reduce the workout intensity level when their heart rate is much higher than the accepted values (i.e., the target heart rate) for a long period of time.

An ECG feature extraction service is implemented at Fog-assisted smart gateways in order to extract heart rate, P wave, and T wave. The extraction service uses an ECG feature extraction template shown in Figure 4.3. The template consists of several stages such as movement artifact removal, wavelet transformation, threshold estimation, P wave, and T wave detection. The movement artifact removal stage includes band-pass filters and moving average filters to remove noise from surrounding environments (e.g., 50 Hz noise from the local power-line in Europe). The filtered data from the movement artifact removal stage is used as inputs for wavelet transformation that decomposes the ECG into another waveform presenting details of signals and trends as a time function. Daubechies-4 wavelet transformation is used because it is suitable for extracting P-wave and T wave while a computation latency is not large in terms of milliseconds. For instance, it takes 101 ms for successfully running the ECG extraction algorithm using Daubechies-4 with an input of 1000 samples on a Pandaboard-based gateway [84]. Therefore, the strict latency requirements of real-time e-health monitoring system can be fulfilled. Several thresholds for R, P, and T wave are estimated based on the results of the wavelet transformation stage. Thresholds for R wave are higher in terms of milliVolt than thresholds of P and T wave. For instance, 1 mV can be used as a threshold value for R peaks in lead I while 0.08 mV and 0.1 mV can be used as threshold values for P and T wave in lead I, respectively. These threshold values can vary depending on the ECG leads. Based on the threshold values, R peak, R-R interval, P, and T wave are detected. From the R-R interval, heart rate can be calculated via the following formula:

$$Heart\ rate = \frac{60}{R - R\ interval}$$

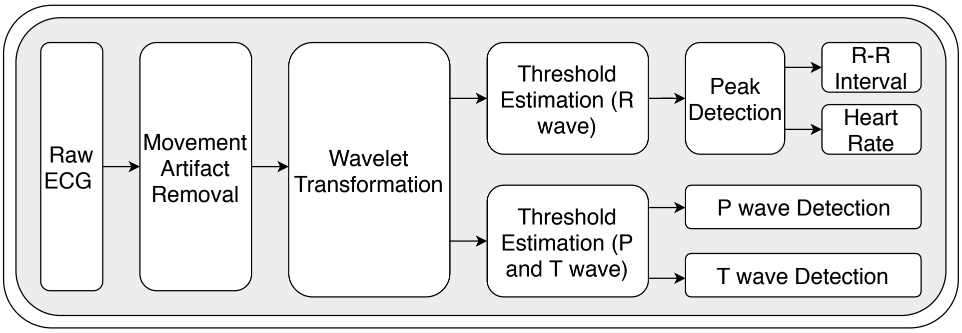


Figure 4.3: ECG feature extraction template

Wavelet transformation in the proposed ECG feature extraction can help to efficiently utilize network bandwidth between Fog-assisted smart gateways and Cloud servers. The number of data samples reduces by a half for each discrete wavelet transform level. Instead of sending raw ECG data, the data after the wavelet transformation stage and the coefficient values can be sent from smart gateways to Cloud servers. This method can help to save 40-80% network bandwidth depending on wavelet transformation types and levels but it may increase a system latency. The higher wavelet transform level data is processed with, the higher error possibility in an inverse transformation process may occur. Therefore, wavelet transform types and levels must be appropriately chosen depending on the applications.

Graphical user interface with management access

When a user such as a caregiver is using the same local network as the system network, he or she can access patient data via a graphical interface implemented at a smart gateway's Fog services. In this case, the graphical interface directly retrieves real-time data from Fog-assisted smart gateways. This helps to reduce the latency of data transmission from Fog-assisted smart gateways to Cloud servers and traversing back from Cloud servers to smart gateways. As a result, real-time data with the lowest latency can be monitored and fast responses can be given by local caregivers who are close to patients. Furthermore, the graphical user interface at Fog-assisted smart gateways helps to maintain the real-time data visualization when the connection between Fog-assisted smart gateways and Cloud servers is interrupted. The data shown in the interface is different depending on the user's authority set by system administrators. In order to access to data shown in the user interface, an authorized user needs to provide username, password, and a confirmation from his or her phone. This access management method helps to increase the security level. Although a user (e.g., a patient) can monitor his or her own data via the graphical interface, it is not recommended because patients can overreact when they see their disease or health status.

Real-time push notification

The push notification service is responsible for informing the responsible persons (i.e., caregivers) about abnormalities in real-time for on-time responses such as first-aid treatments. The push notification service can be triggered when the heart rate or ECG signals are abnormal (i.e., long duration of P wave or high amplitude of T wave). In addition, when the internal temperature of a smart gateway is higher than a predefined threshold or the smart gateway does not receive any incoming data from sensor nodes during a certain time period, push notification messages are sent to system administrators. The content and the priority level of the push message vary depending on specific events. For instance, when the heart rate is higher than 80 bpm, a message with priority level 1 is sent. When the heart rate of the same person increases higher than 120 bpm, a message with priority level 3 is sent. Depending on the applications, the push notification can be implemented and triggered at Fog-assisted smart gateways, Cloud servers, or both. In the proposed systems, the push notification service is implemented and triggered at both Fog-assisted smart gateways and Cloud servers because this helps to maintain stability of the push notification service when the connection between Fog-assisted smart gateways and Cloud servers is interrupted.

Interoperability

Interoperability is the capability of supporting various sensor nodes which are equipped with different sensors and wireless communication protocols. Fog-assisted smart gateways are designed to support sensor nodes using 6LoWPAN, Wi-Fi, Bluetooth, nRF, LoraWan, ZigBee regardless of the producers such as Zolertia or Zigduino. Briefly, several different threads are created and run on an embedded operating system installed at a Fog-assisted smart gateway in which each thread is used for a sensor node type such as 6LoWPAN-based or Wi-Fi-based sensor nodes. When a sensor node type is not used, a thread is deleted in order to save gateway resources. The process of creating and deleting threads is managed by a system administrator or an autonomous program implemented at the smart gateway. The program regularly checks a table of connected devices and the connection protocols from the gateway's database. When it detects that a wireless connection protocol is unused but a thread for this protocol still exists, it deletes the thread.

IP tunneling

This service is used for connecting 6LoWPAN with IPv4/IPv6. The service was implemented at the smart gateways' Fog services by a combination of "gogoc" services and a router advertisement daemon service. The services are used for querying a tunneling between a server and the smart gateway are "gogoc" services, while the router advertisement daemon service is used for listening to router solicitations and sending router advertisements. As regards the advertisements, hosts can configure their address and choose a default router.

Categorization service

Categorization service helps to classify end-users' devices (i.e. local and external devices). Local devices use a local network and are geographically close to smart gateways whilst all other devices are external devices. The categorization service regularly scans the connected devices and updates a table of the connected devices. A scan interval is often small in order to ensure the table of the connected device is updated on time. When a device tries to access data, the categorization service checks the table. If information of the device is not in the table, it is categorized as an external connected device and vice versa. When the local devices access data, data is directly retrieved from Fog-assisted smart gateways . In contrast, data is retrieved from Cloud servers in case of external devices. This mechanism helps to reduce the latency overhead of data travelling via Cloud servers in case of local devices.

Channel managing

An nRF protocol has a working frequency range from 2.4 to 2.525 GHz which is classified into 125 channels. It is required to manage these channels to avoid channel conflict which leads to incorrect data at a receiving side (e.g., gateways). An advanced Fog service called a channel managing service is proposed and implemented at Fog-assisted smart gateways of the proposed systems. The service combines a channel conflict avoidance mechanism, push notification, and a local storage database. The conflict avoidance mechanism checks available channels by scanning all free-to-use channels and compares them with used channels stored in the local database. When a new sensor node is connected to the system, an available channel will be assigned for that node. In some cases, channels of the health monitoring systems conflict with channels of other systems. Therefore, the mechanism checks and verifies channels regularly. When it detects a channel conflict, it sends push notification messages to the system administrators and tries to assign a newly available channel to the sensor node.

Data filtering

In order to provide the a high quality of data, a data filtering service is run by a Fog-assisted smart gateway. Data consisting of e-health and contextual data is filtered to remove noise and incorrect values. Bandpass filters and filtering algorithms are applied to remove 50 Hz noise and abnormal values (e.g., a heart rate of 500 bpm or 20 g acceleration). Principally, a filtering algorithm compares a value with its neighbor values and an acceptable range (i.e, 30-200 bpm for a human heart rate). When the value is totally different from its neighbor values and the acceptable range, it is removed.

Data compression

In order to save the network bandwidth and reduce the latency of data transferring, data is compressed before being sent to Cloud servers or between Fog-assisted smart gateways. Two types of data compression algorithms are lossy and lossless compression. In most of the cases, lossless compression methods are often applied at the Fog-assisted smart gateways because the decompressed data is the same as the compressed data. However, lossless compression algorithms often run complex computational algorithms which can cause an increase in energy consumption and latency. Therefore, the specific choice of lossless compression methods or algorithms running at the smart gateways' Fog services is important. The lossless compression algorithm not only reduces the data volume significantly but also has to guarantee a low latency to fulfill the latency requirements of a real-time health monitoring system. A LZW lossless compression algorithm is applied in the proposed systems [85]. The algorithm has a compression rate of 10:1 and the compression latency is small. The compression rate can increase when the data volume is larger. The LZW algorithm helps to reduce the data volume from 8400 Bytes to 808 Bytes and the transmission latency by 80%.

Security

Security is a vital issue in IoT systems on the ground that in invulnerable systems, sensitive data or actuators can be occupied or controlled by unauthorized persons. Consequently, this can cause undesired negative consequences such as loss of trust and property. In healthcare IoT systems, security is even more important since unsecured systems can cause serious issues. For instance, disease diagnosis can be incorrect because e-health data in unsecured monitoring systems can be edited by unauthorized persons. In the worse case, it can cause a danger to the patient's life. For example, an insulin level injected into the patient blood from an insulin pump can be hacked within 100 meters [26]. When the insulin level in the blood is much higher than a recommended level, it can lead to death. Therefore, security must be diligently maintained. In order to provide some levels of security in the smart gateways' Fog layer, IPtables and IPFW which regulate sets of rules for managing incoming and outgoing network packages are applied and implemented at Fog-assisted smart gateways. They can be configured to block and only allow specific communication ports in the network. Although they provide some benefits, they cannot completely protect the system. For example, the connection between sensor nodes and smart gateways cannot be protected. In order to secure the system completely, these tools must be used together with other robust security tools or methods [86, 87, 88]. Nevertheless, the conventional security methods often require processing power and resources (e.g., memory). Therefore, it cannot be suitable for resource-constrained sensor nodes. The proposed systems applies a light-weight data encryption mechanism where the data is encrypted at sensor nodes and the encrypted data is decrypted at smart gateways. Results show that the encryption mechanism is suitable for Fog-based real-time monitoring IoT systems as the latency of the systems with the data encryption mechanism is only approximately 210 μs higher than the systems without the mechanism. Energy consumption of a sensor node when applying the mechanism is 1.6 mW higher than conventional sensor nodes which do not apply the mechanism. Recently, we have proposed advanced end-to-end efficient and secure authentication and authorization methods for healthcare IoT systems [86, 87, 88]. These methods help to secure systems whilst they do not cause a large increase in system latency and sensor nodes' energy consumption.

To summarize up, applying a Fog computing concept to healthcare IoT systems helps to overcome the existing limitations of these systems and significantly improve the quality of healthcare services. Correspondingly, the patients' health is taken care of while their daily activities are not disturbed. The proposed Fog services can be applied for many IoT systems used in different fields and areas.

Chapter 5

Mobility support in IoT systems

A case of delayed or lost data cannot be neglected in remote monitoring IoT systems because it can lead to undesired consequences such as an inaccurate analysis or decision. One of the reasons causing such a case is mobility. An IoT system often mainly consists of several gateways to cover the whole geographical area used because each gateway in the system only covers a specific area. When a movement distance is large and out of the covered range of a gateway, the system needs to switch the connection of a moving person to another gateway which covers the area to which the person is going. If the system is not equipped with a handover mechanism, the switch latency is high and it breaches the latency requirement of the system. In order to minimize the switch latency, IoT systems must have a handover or hand-off mechanism to support mobility. For example, when a patient moves from one place covered by a source gateway to another place covered by a destination gateway, a handover mechanism de-registers the sensor node attached to a human body from the source gateway and registers the sensor node to the destination gateway. The handover mechanism should ensure the lowest handover latency to fulfill the requirements of real-time health monitoring systems. However, it is arduous to develop and implement a handover mechanism which supports full mobility especially in healthcare IoT systems [89, 90] because there are strict requirements (e.g., latency, reliability, and security) for these systems [91]. In the case of Fog-based healthcare IoT systems, it is much more challenging to support full mobility due to the smart gateways' Fog services. The handover mechanism must guarantee that all Fog services must not be interrupted during mobility. For instance, the mobility mechanism must update the local distributed database of smart gateways during the mobility.

Currently, conventional Fog-based approaches [92,93,94] cannot support full mobility especially in the case of high data rate applications such as real-time multi-channel ECG monitoring systems. In some cases, these approaches are not efficient in terms of latency when there is an interruption between Fog and Cloud servers. This chapter proposes an advanced handover mechanism for full mobility which supports Fog-based high data rate health monitoring IoT systems. The proposed mechanism addresses unfit or unsolved issues related to mobility support in Fog-based systems. For instance, oscillating nodes which move back and forth between locations covered by adjacent gateways during a short period of time are carefully considered in the proposed mechanism. The oscillating nodes cannot be neglected because they can lead to large network overheads or reduce the system performance.

Mobility tree

In order to provide an overview of mobility in IoT systems, this chapter presents a mobility classification tree shown in Figure 5.1.

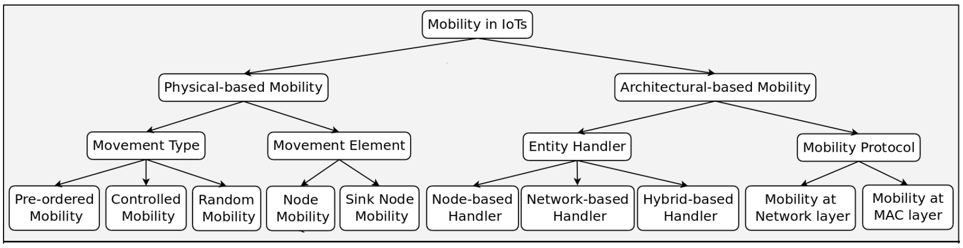


Figure 5.1: Overview of mobility

Mobility can be classified into two primary types including physical-based and architecture-based mobility. It is more arduous to deal with the architecture-based mobility than the physical-based mobility. Fortunately, in healthcare monitoring applications, the physical-based mobility primarily occurs because only the sensor nodes move while the gateways are fixed in particular places such as rooms or corridors. The physical-based mobility can be divided into two groups which are the movement type and the movement element. The movement type group encompasses pre-ordered, controlled and random mobility. The most challenging to deal with is the random mobility because many parameters (e.g., moving path, moving time, and moving destination) needed for a handover mechanism are unknown in advanced. When the system supports the random mobility, it is capable of supporting other movement types. The movement element group includes node mobility and sink node mobility in which the sensor node mobility primarily occurs. To summarize, random mobility and node mobility both occur in most of the mobility cases in healthcare centers. The proposed handover mechanism focuses on the node mobility and the random mobility.

Gateway placement

In order to cover all physical areas of a healthcare center, gateways have to be deployed in such a way that there is at least an overlapping area between two adjacent gateways. This chapter presents a placement of two adjacent gateways, as shown in Figure 5.2 and Figure 5.3, in which each gateway's area is divided into 4 zones including personal zone, weak zone, shared zone, and sensitive zone.

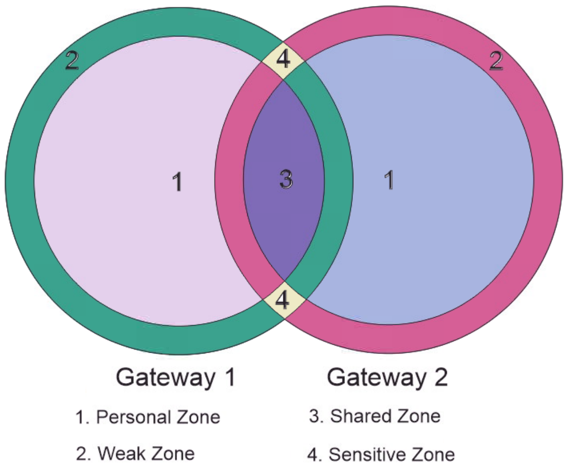


Figure 5.2: Setting up two adjacent gateways

The handover mechanism is based on radio-related values and triggered in a shared zone. Consequently, the results of the handover mechanism can be significantly affected by areas of the zones. In Paper IV, formulas for calculating all zones are explained in detail. In addition, Paper IV present different experimented cases where areas of zones vary. In most the cases, the handover mechanism is primarily triggered in the personal zone when the shared area is larger than the threshold value. In contrast, the handover mechanism is triggered in the pink area shown in Figure 5.3 in many cases when the shared zone area is smaller than the recommended threshold.

Results show that there are no specific requirements for zone areas. All zone areas can be flexibly defined based on specific applications. It is recommended that the personal zone and the shared zone should be large enough to avoid triggering missed and unnecessary handover mechanism. Paper

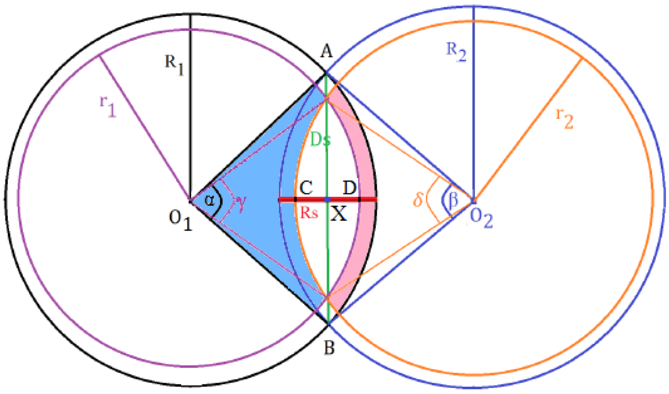


Figure 5.3: Setting up two adjacent gateways

IV recommends that the area of the shared zone should be larger than a threshold value defined by the following formula:

$$t\_v = s\_p * s\_p * 2/3$$

where t_v: threshold value
s_p: speed of sensor node (m/s)
In order to provide an overview of the proposed mobility support approach, the proposed mobility handover mechanism is discussed in the following:

Handover mechanism

The proposed handover mechanism shown in Figure 5.4 consists of 16 blocks. The following only discusses the primary blocks of the mechanism.

A. Defining gateway zones and scanning RSSI, LQI in all gateways

The soft radius, the actual radius, and the distance between two adjacent gateways are important because they directly affect the area of zones and the efficiency of the handover mechanism. The actual radius R can be defined based on the coverage radius of a gateway. Often the actual radius R is slightly smaller than the coverage radius of a gateway to avoid service interruption due to error offsets. The weak zone's area will be smaller when the soft radius r is larger. In contrast, the shared zone will be larger when the soft radius r is larger. Therefore, the soft radius r has to be to properly defined. In order to achieve good results, the areas of the shared zone and the personal zone should be sufficiently large since these areas occupy a large portion of the whole coverage area of a gateway. Fortunately, the distance between two adjacent gateways is static in time and can be easily measured.

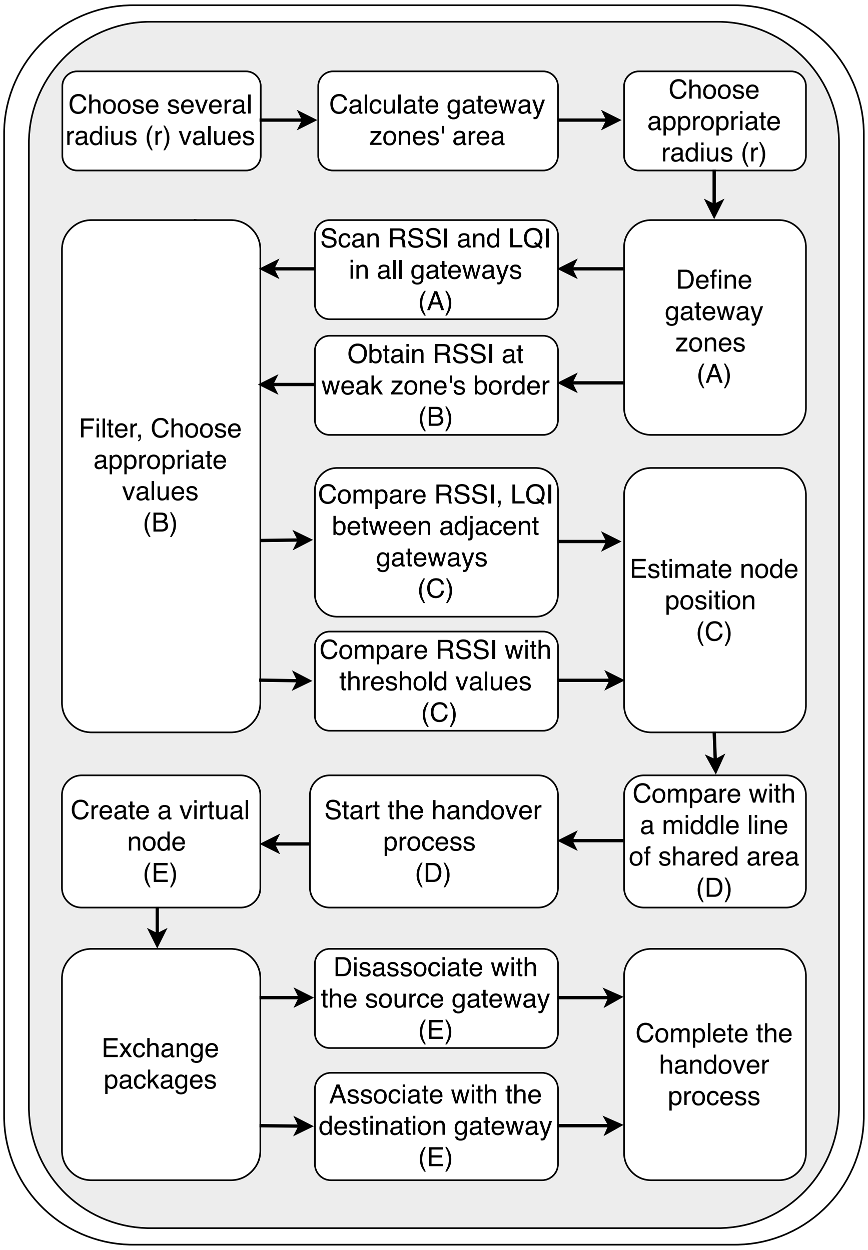


Figure 5.4: Mobility handover mechanism

B. Obtaining RSSI at weak zone's border, filtering and selecting appropriate values

After the soft radius, the actual radius and distance between gateways are defined or measured, RSSI values at the weak zone's border can easily be obtained via the scanning and eliminating mechanism. In the mechanism, the RSSI values at the weak zone's border is scanned 10 times during a short period of time and the most common value is selected.

C. Comparing RSSI with threshold values, comparing RSSI, LQI between adjacent gateways and estimating node position

The RSSI and LQI of a sensor node are collected via a scanning method in which the scanning interval can be defined. In the proposed system, the scanning interval is low for collecting data in real-time and responding on-time to the sensor node's movement. Similar to the other scanning steps, these values are often scanned my times during a short period of time. Only the most common value is used and stored in the database. These RSSI and LQI values of a sensor node between two adjacent gateways are used to estimate the position of the sensor node. For instance, RSSI values of the weak zone area are between -75 dBm and -65 dBm. When the collected RSSI values of a sensor node between two adjacent gateways are -62 dBm and -55 dBm, the sensor node might be in the shared zone of these gateways. In addition, the sensor node is closer to the gateway having the value of -55 dBm. It should be noted that all gateways discussed are similar. In the case of indoor and outdoor gateways, offset values are used to maintain the balance and similarity of the gateways.

D. Comparing with a middle line of the shared area and starting the handover process

In most of the cases, when the sensor is in the shared zone and it has just passed the middle line AB shown in Figure 5.3, the handover mechanism is triggered. To target the address, the handover mechanism compares the RSSI values of the sensor node with the RSSI values of the middle line AB. In the proposed mechanism, the RSSI values of the middle line AB are set when RSSI values of a sensor node between two adjacent gateways are equal. During the mobility process, the source gateway will send its data related to the sensor node to the destination gateway. when the transmitting occurs, the sensor node is still associated with the source gateway. Correspondingly, data will not be missed during the mobility process.

In addition, the system checks the LQI value regularly. When the LQI value is smaller than the pre-defined values such as 70%, the system triggers the push notification service to send messages to system administrators. The pre-defined LQI values can be flexibly set depending on the applications and geographical areas covered by the system. In some places where there are many noise sources, the LQI values can be set smaller.

E. Creating a virtual node, disassociating and associating with the source and destination gateway, respectively

During the mobility process, a moving sensor node must be deregistered from the source gateway and registered to the destination gateway due to the fact that a sensor node can only associate with a gateway at a single moment. However, the switching between the source and destination gateways causes large latency. In order to minimize the switching latency, a virtual sensor node is created at the destination gateway. The virtual sensor node using the MAC address of the moving sensor node registers with the destination gateway by exchanging messages required for setting up a Wi-Fi connection. The number of exchanging messages depends on the security method of Wi-Fi (i.e., WPA + AES, WEP). When the virtual node is registering with the destination gateway, the sensor node still associates with the source gateway to maintain the data transmission between the sensor node and the system. When the registration process of the virtual node completes, the sensor node deregisters the source gateway immediately. Thus, the sensor node is already registered with the destination gateway, and it can transmit the data to the gateway.

F. Oscillation event handling

The handover mechanism checks an oscillating node by inspecting the isassociating and associating time. When the interval of two continuous dissociating and associating times is less than a pre-defined threshold, the oscillating node is detected. The pre-defined threshold can be flexibly set depending on the shared zone area. When the oscillating event occurs, the sensor node maintains the association with the gateway in whose location the sensor stays longer. Accordingly, the handover mechanism is not triggered for saving the gateways' resources. In this situation, when the sensor node moves further away from a gateway, there is still enough time to trigger the handover mechanism because the scanning method interval is short.

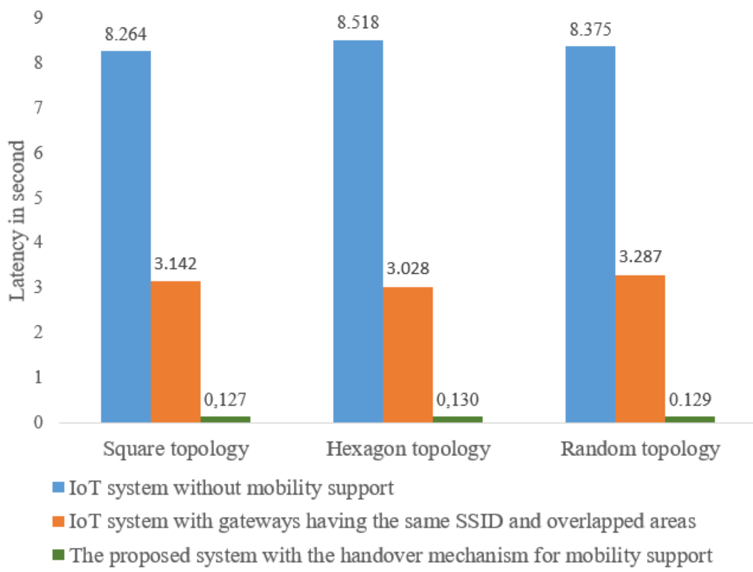


Figure 5.5: Latency of IoT systems with different configurations

Results of the experiments with the proposed handover mechanism are shown in Figure 5.5. Results show that the handover mechanism has a minimized latency to maintain a connection between a sensor node and Fog-assisted smart gateways. When compared with the other state-of-the-art approaches [52, 53, 54, 55, 95, 96], the proposed handover mechanism has the least latency. In addition, the proposed handover mechanism supports oscillating nodes whilst the other mechanisms do not properly consider oscillating nodes.

Chapter 6

Design of energy-efficient IoT systems

A remote health monitoring system cannot be considered as a reliable system when it cannot fulfill the strict requirements of energy efficiency, and latency. In Chapter 5, latency issues due to the mobility have been comprehensively discussed. In this chapter, energy efficiency issues are investigated.

Energy consumption is one of the most significant issues for battery-powered devices or systems utilizing these devices. When the energy consumption of these devices is high, it can cause serious consequences such as low-quality services, service interruption or the short lifespan of the battery pack. In health monitoring systems especially, energy inefficiency cannot be underestimated because it can lead to serious results such as incorrect disease analysis or push notification service interruption. Therefore, a wearable sensor node must be designed that will achieve a high level of energy efficiency.

This chapter presents an advanced energy-efficient health monitoring system which can be considered as a combined and customized system from four energy-efficient systems published in Paper II, V, VI, and VII. The proposed system applies Fog-based approaches for saving the energy consumption of sensor nodes while maintaining the high quality of the services. There are different types of sensor nodes in the proposed system such as e-health sensor nodes and contextual sensor nodes. The e-health sensor nodes can collect one or several bio-signals at different data rates (e.g., 125 samples/s ECG, 50 samples/s body motion-related data, and 1 sample per 5 minutes blood glucose). Contextual sensor nodes can acquire room temperature, humidity, and air quality. Depending on the applications, specific sensor nodes are used.

In order to achieve a high level of energy efficiency, both the software and hardware structure of sensors nodes must be carefully designed. For example, e-health and contextual sensor nodes have to avoid running complex data processing algorithms (e.g., ECG feature extraction based on wavelet transformation), therefore, these will instead be run by Fog-assisted smart gateways. By applying this approach, the workload of the sensor nodes can be reduced and a high level of energy efficiency can be obtained.

The sensor nodes in the proposed system consist of three primary parts: a micro-controller, sensors, and a wireless transmission part. In addition, some sensor nodes include additional parts such as a power management unit, and an energy harvesting unit. Depending on particular healthcare applications, the choice of these parts can vary.

A micro-controller is a core component in a sensor node. It performs all important tasks such as sensor controlling, and I/O interface management. Correspondingly, a micro-controller uses a greater amount of the sensor node's energy. When the micro-controller does not perform its tasks efficiently, the energy consumption of the sensor node increases dramatically. Therefore, the micro-controller must be energy-efficient in both terms of hardware and software design. The two most commonly used micro-controller families for health monitoring applications are 8-bit AVR ATmega and 32-bit ARM Cortex M0 since these micro-controller types are ultra-low power. However, these micro-controllers are not equipped with large memories (e.g., RAM and ROM). Therefore, complex algorithms cannot be implemented with these micro-controller families. Based on the experiments performed by Atmel [97], a 32-bit ARM Cortex M0 micro-controller is less efficient in terms of memory usage and energy than an 8-bit AVR ATmega micro-controller when running hardware-near functions. An ARM Cortex M0 in particular requires 33 cycles to receive a byte of data via SPI by using interrupt while an 8-bit AVR micro-controller needs merely 12 cycles to run the same task. In the same experiment, a 32-bit ARM Cortex M0 consumed 48 μA while an 8-bit AVR micro-controller expended 36.1 μA while receiving 80 kbps data via SPI. In another example, a 32-bit ARM Cortex M0 needed 192 bytes in stack memory while an 8-bit AVR Atmega micro-controller needed 70 bytes of stack memory to run a recursive 15-stage Fibonacci algorithm. In [21], the authors prove that an 8-bit AVR Atmega is able to gather high-resolution e-health data and run some simple data processing algorithms without breaching the requirements of the system latency. Hence, an 8-bit ultra-low power AVR Atmega micro-controller is an appropriate micro-controller for healthcare IoT applications. Therefore, an 8-bit ultra-low power AVR Atmega micro-controller is used in sensor nodes of the proposed system. Depending on the applications, the micro-controller is equipped with a specific circuit and runs at different frequencies.

In order to communicate with Fog-assisted smart gateways, sensor nodes use nRF which is a low-power wireless protocol supporting many-to-many communications. In another word, by applying nRF sensor nodes can connect to several Fog-assisted smart gateways simultaneously. When one of the connections is interrupted, Fog-assisted smart gateways still receive real-time data from sensor nodes. In addition, nRF supports different data rates such as 250 kbps, 1 Mbps and, 2 Mbps. Therefore, it can be suitable for different healthcare applications such as glucose monitoring, 2-channel ECG monitoring, and fall detection. It was noticed that a sensor node consumes more energy when the transmission data rate is higher. Therefore, low data rates are always considered first. In some cases when a system is updated for higher data resolutions, nRF can be reconfigured to achieve the target without the need for any other hardware. For instance, a 250 kbps data rate is suitable for 250 samples/s 2-channel ECG systems. When the system is updated to obtain 500 samples/s 8-channel ECG, 2 Mbps data rate can still be used. One of the nRF benefits is a capability of supporting many channels which correspond to specific frequencies. The nRF in particular can support 126 channels which can then be efficiently utilized by Fog services. For instance, some channels can be reserved for certain important tasks such as a confirmation or password exchange while some channels can be used for 3-way handshaking tasks or data transmitting. Last but not least, nRF can operate with a long voltage range from 1.9 V to 3.6 V. Therefore, it can be supplied with the same voltage supply as a micro-controller without the need for a voltage regulator.

Sensors play an important role because they directly affect the sensor nodes' energy consumption, signal quality, service quality, and latency. For example, missing data may occur when a sensor does not respond on-time to an instruction command sent by a micro-controller. Therefore, low-power and high-quality sensors with a fast response capability must be used in a sensor node. Sensors are often connected to a micro-controller via wire communication protocols such as I2C or SPI. However, the impact of these protocols on the energy consumption of sensor nodes has not been properly investigated. In order to explore the impact, experiments with wearable sensor nodes are carried out and presented in this chapter.

In order to investigate the energy consumption of sensor nodes used in health monitoring applications, several sensor nodes are built and experimented. A wearable sensor node for real-time glucose and body temperature monitoring is built. The sensor node collects both the blood glucose level and the body temperature simultaneously because insulin is linked to the core body temperature [98]. The data rate of the glucose sensor and the body temperature sensor is 1 sample per 10 minutes and 1 sample per 1 minute, respectively. Similar to other proposed sensor nodes, a micro-controller (i.e., AVR ATmega328P) and an nRF transceiver (i.e., nRF24L01) are used in the sensor node. The micro-controller runs at 1 MHz and is supplied with 2 V. In addition to the primary parts just mentioned, the sensor node is equipped with an energy harvesting unit and a power management unit.

Since the sensor node is attached to a human body, an energy harvesting unit can harvest both human powered and ambient energy (i.e., thermal and radio frequency energy). However, only the RF energy harvesting unit is integrated in the sensor node. The harvesting unit consists of four parts such as an antenna, matching network, RF to DC rectifier and storage element. A miniaturized printed elliptical nested fractal multiband antenna proposed in [99] covers different frequency bands such as GSM 900, 2.4 GHz (Bluetooth, Wi-Fi, 6LoWPAN, nRF), 3.2 GHz (radio-location and 3G), 3.8 GHz (LTE and 4G), and 5 GHz Wi-Fi bands. The antenna supports an omni-directional feature which allows collection of the RF signals from many directions. A matching network circuit is designed for focusing on the widely used radio frequency bands previously mentioned. In an RF to DC rectifier circuit, Schottky diodes are utilized due to the low forward voltage drop (e.g., 0.2-0.3 V). The energy harvesting unit is able to provide 2.1 V at 0 dBm input power (925 MHz RF input signals). Correspondingly, the energy harvesting unit is suitable for the sensor node which runs at 1 MHz with 2 V.

The power management unit is built from a voltage sensor which can be formed by utilizing a low-power Schmidt trigger circuit. The power management unit is used to monitor power and battery level. When the battery level is low, it sends information to the micro-controller triggering push notification messages requesting a recharge or replacement of the battery. With the information retrieved from the power management unit, a micro-controller can shutdown unnecessary or unused components.

Table 6.1: Power consumption of nRF transceiver, sensor node

Device	**Voltage supply (V)**	**Average Current (mA)**
nRF transmitter (nRF + Atmega328P)	2	0.5
Sensor node	2	1.4

Results from the experiments presented in Table 6.1 show that the sensor node is energy-efficient. By applying a 1000 mAh lithium battery, the sensor node can continuously work for up to approximately 700 hours in a daily living or working environment such as office and apartment.

An energy-efficient wearable sensor node for ECG and body temperature monitoring is built for the experiments. The sensor node encompasses ADS1292, BME280, AVR ATmega328P, and nRFL2401. As mentioned, the choice of these devices is crucial because it can dramatically affect the total energy consumption of a sensor node. ADS1292 is a low-power and low-noise analog front-end device for gathering ECG at high data rates (e.g.,

Table 6.2: Average power consumption of the health monitoring device at a data rate of 18 kbps

Mode	Voltage (V)	Average power (mW)
Idle	3	1.2
Active without AES-256	3	19.5
Active with AES-256	3	21.03

1000 16-bit samples/s). However, it is sufficient to collect ECG at a data rate of 250 samples/s while maintaining a high level of signal quality [100]. BME280 is a low-power and precise sensor for gathering both temperature and humidity. These sensors are connected to a micro-controller via SPI. The micro-controller runs at 8 MHz and is supplied with 3 V. The micro-controller sends the collected data to Fog-assisted smart gateways via an nRF protocol. For providing some levels of security, AES-256 - a light-weight encryption algorithm is applied. Results presented in Table 6.2 show that the sensor node is energy-efficient even though AES-256 is used. By using a 1000 mAh lithium button cell, the sensor node can be used for up to 155 hours in a daily working environment such as office.

In the experiments, an energy-efficient sensor node for fall detection systems is built. Similar to other sensor nodes presented above, this sensor node consists of a micro-controller (ATmega328P) and a wireless communication chip (nRF24L01). In addition, the sensor node is equipped with an MPU-9250 sensor which is a low-power and high precision sensor consisting of a 3-D accelerometer, a 3-D gyroscope, and a 3-D magnetometer. The sensor node runs at 8 MHz and collects data (e.g., acceleration, angular velocity, and compass data) via SPI with different configurations shown in Table 6.3.

Table 6.3: Scenarios setup

Configuration (Conf)	Accelerometer	Gyroscope	Magnetometer
Configuration 1	X		
Configuration 2		X	
Configuration 3			X
Configuration 4	X	X	
Configuration 5		X	X
Configuration 6	X		X
Configuration 7	X	X	X

In addition, energy consumption of the sensor node was measured with several scenarios such as different sampling rates (e.g., 50, 100, 500 samples/s), and transmission distances (i.e., 5, 10, and 20 meters). All experimental results are presented and discussed in detail in Paper VI. Results shown in Table 6.4 and Figure 6.1 indicate that watchdog timers, and SPI are more energy-efficient than other types in the same categories. In addition, the results show that in order to dramatically save energy consumption, unneeded modules such as UART, brownout, and ADC must be disable. Furthermore, the results show that energy consumption of a sensor node can be significantly saved when a transmitting unit of an nRF transceiver is primarily used whilst a receiving unit of the transceiver is disabled. In the harsh conditions (such as long communication distance, many objects in between the sensor node and a gateway), the sensor node can continuously work up to 76 hours when using a 1000 mAh lithium battery to collect motion-related data at a data rate of 50 samples/s via 1 Mbps SPI. The energy consumption of the sensor nodes shown in Table 6.5 indicates the sensor node is energy efficient. When comparing with other fall detection sensor nodes presented in the state-of-the-art works [33,34,38,101], the sensor node is more energy efficient.

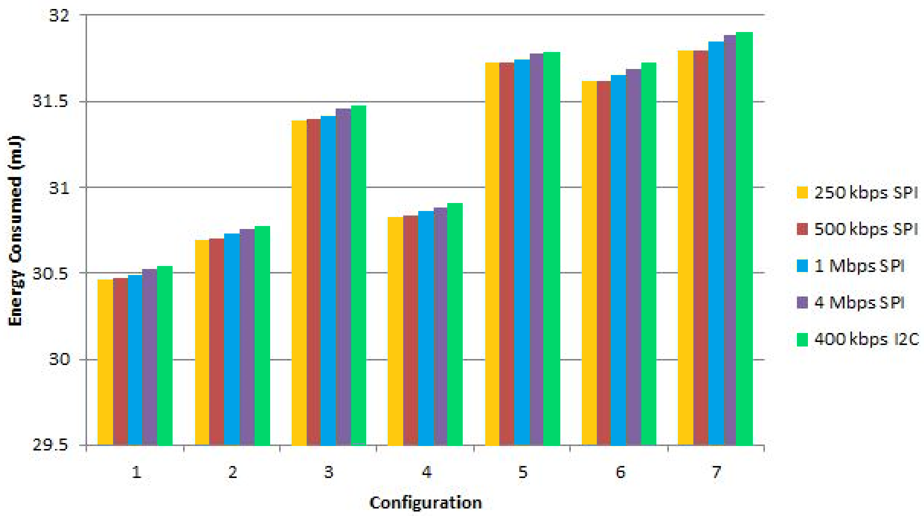


Figure 6.1: Energy consumption of sensor nodes when collecting data from multiple sensors at 50 samples/s using SPI and I2C

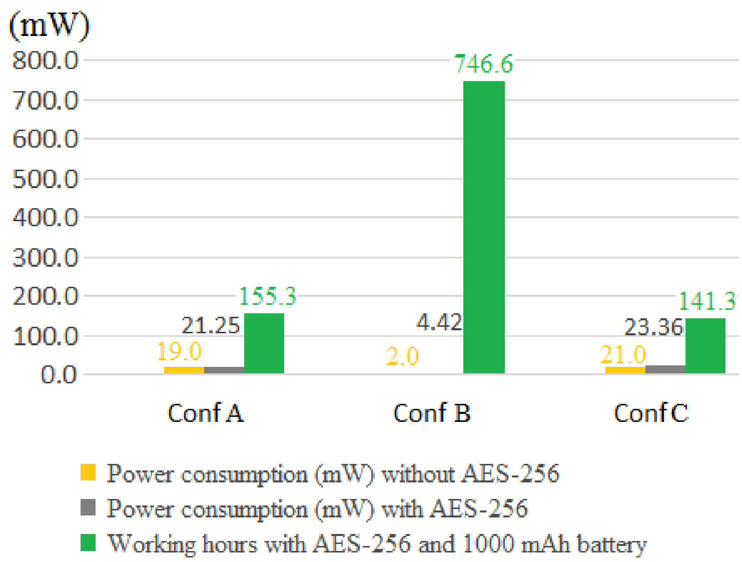
Table 6.4: Energy consumption when collecting 50 samples/s acceleration data with different techniques

	8bit timer (mJ)	16bit timer (mJ)	watchdog timer (mJ)
Conf 1	30.47	30.86	22.58
Conf 2	30.69	31.08	23.80
Conf 3	31.39	31.79	23.57
Conf 4	30.83	31.22	22.95
Conf 5	31.72	32.11	23.95
Conf 6	31.62	32.06	23.88
Conf 7	31.8	32.23	24.11

Table 6.5: Energy consumption of the sensor node when collecting 50 samples/s motion data and transmitting the data via nRF with a distance of 20 meters in different configurations

Configuration	Energy consumption (mJ)
Conf 1	30.72
Conf 2	30.95
Conf 3	32.1
Conf 4	34.53
Conf 5	35.56
Conf 6	35.35
Conf 7	36.68

In order to investigate the energy consumption of sensor nodes for healthcare applications using both low and high data rate bio-signals, a sensor node for monitoring diabetic patients with cardiovascular disease is design and implemented. The proposed e-health sensor node can be considered as an advanced and customized version of the sensor nodes presented above. Methods for energy saving applied in the sensor nodes mentioned above are reused and customized. The proposed customized sensor node can simultaneously collect several types of data such as ECG, blood glucose, body temperature, and body motion with different data rates. Similar to the aforementioned sensor nodes, these sensor nodes are equipped with an AVR ATmega328P micro-controller and an nRF24L01 transceiver. However, the micro-controller only runs at 1 MHz. In addition, the e-health sensor node is equipped with a glucose sensor, a temperature sensor (i.e., BME280), a low-power ECG analog front-end (AD8232), and a motion sensor (i.e., MPU-9250) consisting of 3-D accelerometer, 3-D gyroscope, and 3-D com-



Conf A: 50 samples/s motion-related data and 1 sample/s glucose data
Conf B: 1 sample/s body temperature data and 1 sample/s glucose data
Conf C: 1 sample/s body temperature, 1 sample/s glucose data, and
120 samples/s ECG data

Figure 6.2: Power consumption and working hours of sensor nodes in different configurations with AES-256

pass. The data rate of the glucose sensor and the body temperature sensor are 1 sample per 5 minutes and 1 sample per 2 minutes while the data rate of MPU-9250 and AD8232 is 50 samples/s and 125 samples/s, respectively. Experiments concerning the energy consumption of the sensor node were conducted using several configurations as shown in Paper VII.

Experimental results presented in Figure 6.2 show that the e-health sensor node is energy-efficient. When using a 1000 mAh lithium battery, the e-health sensor node, which runs AES-256 and collects all data (e.g., ECG, glucose, body temperature and body motion), can accurately work up to approximately 141 hours in a daily living environment such as office or apartment having electronic devices and computers. When comparing with other sensor nodes proposed in other state-of-the-art works [21,102,103,104], the proposed e-health sensor node is more energy-efficient even though the sensor node can collect different types of data simultaneously.

To summarize up, the proposed systems and sensor nodes are energy efficient. It is possible to use several proposed sensor nodes in a system since all the proposed sensor nodes have similar structures and use the same nRF protocol. Depending on the applications, a particular sensor node or a group of sensor nodes can be used.

Chapter 7

Fog-assisted IoT systems for diabetic patients with cardiovascular disease

An advanced remote health monitoring system for diabetes having cardiovascular disease is necessary due to serious consequences of diabetes and cardiovascular disease, and the large number of people having these diseases. The system must be reliable and capable of providing enhanced services such as real-time and accurate data analysis, and security. In Chapter 5 and 6, the reliability of remote health monitoring systems in terms of latency and energy efficiency have been presented. This chapter discusses enhanced Fog services and algorithms analyzing bio-signals (e.g., ECG, 3-D acceleration, 3-D gyroscope and glucose) in real-time in order to improve a quality of healthcare, particularly in diabetes, cardiovascular disease monitoring and fall detection.

This chapter shows several remote health monitoring systems presented in 4 papers (i.e., paper II, V, VI, and VII). The first three papers present monitoring systems for particular diseases such as ECG monitoring for CVD, blood glucose for diabetes, and fall detection while Paper VII proposes an advanced monitoring system for several diseases such as diabetes and cardiovascular disease. The system presented in Paper VII can be considered as the customized and augmented version of the systems presented in the other papers. These four systems use the same architecture including sensor nodes, Fog-assisted smart gateways, Cloud servers, and end-user terminals. In addition, these systems utilize the same nRF wireless communication protocol. Therefore, the advanced services related to nRF and sensor nodes of one system are also suitable for other systems. A combination of all the systems presented in the papers can create an advanced monitoring system for diabetic patients with cardiovascular disease.

The combined system is capable of providing many advanced Fog services such as distributed local data storage, interoperability, channel managing, push notification, categorization service, data compression, and an early warning score system. These services are discussed in detail in Chapter 4. In addition, the combined system can inherit the typical features of the member systems for the treatment of diabetic patients with cardiovascular disease. These features are discussed in the following paragraphs:

On feature that can be utilized is the abnormal glucose warning feature presented in Paper V where the collected blood glucose is compared with the recommended pre-defined values stored in the local distributed storage. When the collected glucose is in an abnormal range, the system sends push notification messages to caregivers. The recommended pre-defined glucose values are shown in Figure 7.1 and 7.2. These recommended values can be changed by an authorized person (e.g., a system administrator or a doctor) depending on the specific situations.

Type	Before meals	2 hours after meals	Wake up	Risk of hypoglycaemia (low blood glucose)
Healthy Person	4 to 5.9 mmol/L	under 7.8 mmol/L		under 4 mmol/L
Type I diabetes	4 to 7 mmol/L	5 to 9 mmol/L	5 to 7 mmol/L	under 4 mmol/L
Type 2 diabetes	4 to 7 mmol/L	under 8.5 mmol/L		under 4 mmol/L
Children with type 1 diabetes	4 to 7 mmol/L	5 to 9 mmol/L	4 to 7 mmol/L	under 4 mmol/L

Figure 7.1: Recommended glucose levels [105, 106]

Plasma glucose test	Normal	Prediabetes	Diabetes
Random	Below 11.1 mmol/l Below 200 mg/dl		11.1 mmol/l or more 200 mg/dl or more
Fasting	Below 6.1 mmol/l Below 108 mg/dl	6.1 to 6.9 mmol/l 128 to 125 mg/dl	7.0 mmol/l or more 126 mg/dl or more
2 hour post-prandial	Below 7.8 mmol/l Below 140 mg/dl	7.8 to 11.0 mmol/l 140 to 199 md/dl	11.1 mmol/l or more 200 mg/dl or more

Figure 7.2: Blood glucose levels in diagnosing diabetes [105, 106]

The algorithm for extracting ECG features in the system presented in Paper I can also be utilized for the combined system. In the algorithm, raw ECG data is processed with a baseline wander removal method, allowing for noise and movement artifacts to be filtered out of ECG signals via 50 Hz notch filters, and 5-15 Hz bandpass filters. The filtered ECG is then processed with a wavelet transformation mechanism based on Debauches-4 wavelet shown in Figure 7.3 in order to extract R peaks, R-R intervals, P waves, and T waves. The results of detecting R peaks and R-R intervals are

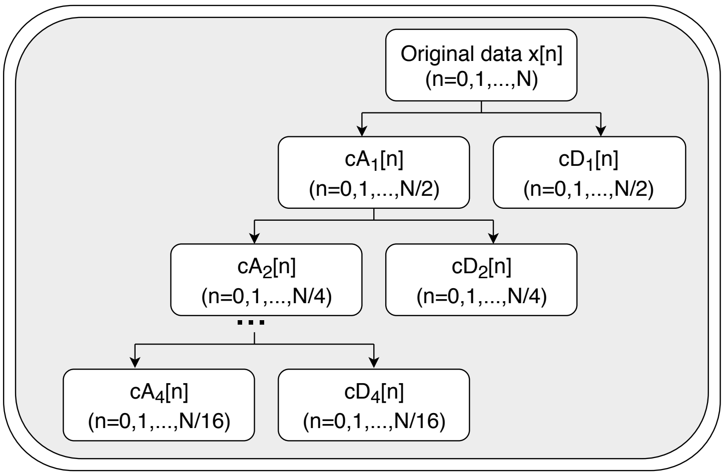


Figure 7.3: Four-level Discrete Wavelet Decomposition
cA_k: Level k approximation coefficients, cD_k: Level k detail coefficients

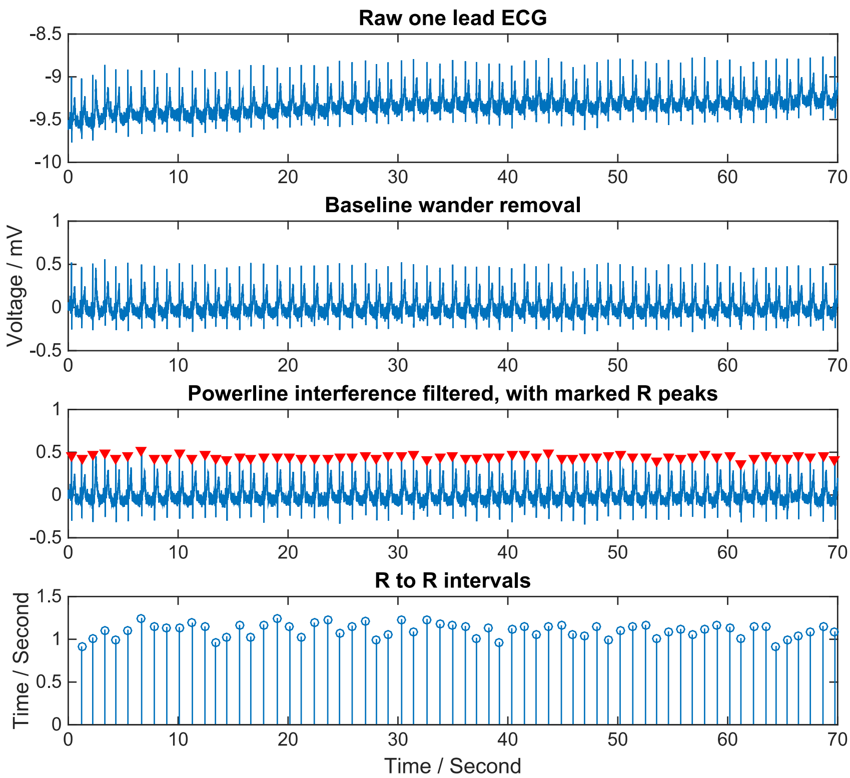


Figure 7.4: Signal processing with one lead ECG

shown in Figure 7.4. The results show that the algorithm is able to extract ECG features accurately.

In order to investigate the algorithm's sensitivity, the algorithm was run 120 times with ECG extracted from a data set (e.g., 3-D acceleration, 3-D angular velocity, and ECG) collected by a sensor node presented in Paper VII. The data set was acquired during 60 minutes from a 30-year-old healthy male person performing different activities including lying, standing, walking and running in which each activity lasts 5 minutes. In each experiment time, the algorithm extracted ECG features from 30-second ECG. The results show that the algorithm can correctly detect R peaks, R-R intervals, P waves, and T waves up to 58 times in a total of 60 experimental times including lying on the bed and standing cases. The sensitivity in these cases is about 0.966. In each incorrect extraction case, 1 P wave was missed. One of the reasons causing the incorrect detection cases was the monitored person's moving artifacts during the experiments. In walking and running cases, the algorithm successfully detected ECG features including R peaks, R-R intervals, P waves, and T waves in 25 and 2 times, respectively in which each case was carried out in 30 times. The sensitivity for the walking case and the running case is 0.83 and 0.066, respectively. The results show that the ECG feature extraction algorithm might not function correctly when a user walks fast or runs. The results imply that ECG needs to be measured alongside with a user's activity status in order to obtain a high quality of service.

In addition to ECG features mentioned above, other ECG features (e.g., QT length) are needed for an analysis of cardiovascular disease. Therefore, a QT length extraction algorithm proposed in Paper VII is utilized for the combined system. In the proposed QT length extraction algorithm, the lowest interval IP is first located in which the P wave reaches its maximum. A similar procedure is applied for calculating IP, IR, IS and IT. Finally, the QT length is calculated by applying the formula: IQT= IR+ IQ+IS+IT. The proposed QT length extraction algorithm applies the lowest interval computing algorithm, which is shown in Algorithm 1. The lowest interval computing algorithm computes the lowest interval in which a function f(x) reaches its local maximum. The lowest interval computing algorithm requires two inputs including x_i and t_i - where t_i is the instant of time t. The lowest interval computing algorithm is light-weight and fast. Therefore, it is suitable for real-time data analysis and resource-constrained devices.

The QT length extraction algorithm was run 120 times with the data set mentioned above in which each 30 times were used for a particular activity such as lying on bed, standing, walking or running. The QT lengths were successfully extracted up to 57 times in both cases of lying on bed and standing whilst there were only 26 and 2 successful times for the walking case and the running case, respectively. The sensitivity of the algorithm

Table 7.1: Formulas for calculating corrected QT interval

Algorithm	Formula
Bazett (QTcB) [108]	QTc = QT / ($\sqrt{RR}$)
Fridericia (QTcFri) [109]	QTc = QT / ($\sqrt[3]{RR}$)
Framingham (QTcFra) [110]	QTc = QT + 0.154 x (1 - RR)
Hodges (QTcH) [111]	QTc = QT + 0.00175 x ([60 / RR] - 60)
Rautaharju (QTcR) [112]	QTc = QT - 0.185 * (RR - 1) + k (k=+0.006 seconds for men and +0 seconds for women)

for the walking case, the running case and both cases of lying on bed and standing is 0.86, 0.06 and 0.95, respectively. Based on the results, it is recommended an user should not walk or run when extracting QT length. The results prove that there is a need of other algorithms to enhance a quality of service.

Algorithm 1 Algorithm to compute local maximum of a function $f(x)$

```
procedure MAX-ALGORITHM(x_i, t_i)

    if (x_{i-1} = 0 and x_i > 0) then
        I_1 ← t_i
        M ← x_i

    else if (I_1 ≠ 0 and x_i > 0) then
        M = Max(x_i, M)

    else if (I_1 ≠ 0 and x_i = 0) then
        I_2 ← t_i
        break

    return M, [I_1, I_2]
where x: value of ECG
t: specific time(s)
```

The QT interval can be used to calculate a corrected QT interval (QTc) via the state-of-the-art algorithms shown in Table 7.1. In hospitals, Bazett's QTc (QTcB) is commonly used as a stand algorithm. However, the authors in [107] show that Fridericia's QTc might be considered as the next standard algorithm replacing the current standard QTcB.

A fall detection algorithm, shown in Figure 7.5, used in the system presented in Paper VI can be utilized for the combined system. The algorithm requires inputs of 3-D acceleration and 3-D angular velocity. These motion-related signals often contain noise and movement artifact. Therefore, they have to be eliminated via 50 Hz filters and moving average filters. The filtered data is used to calculate the fall feature parameters via the formula shown in Chapter 3.

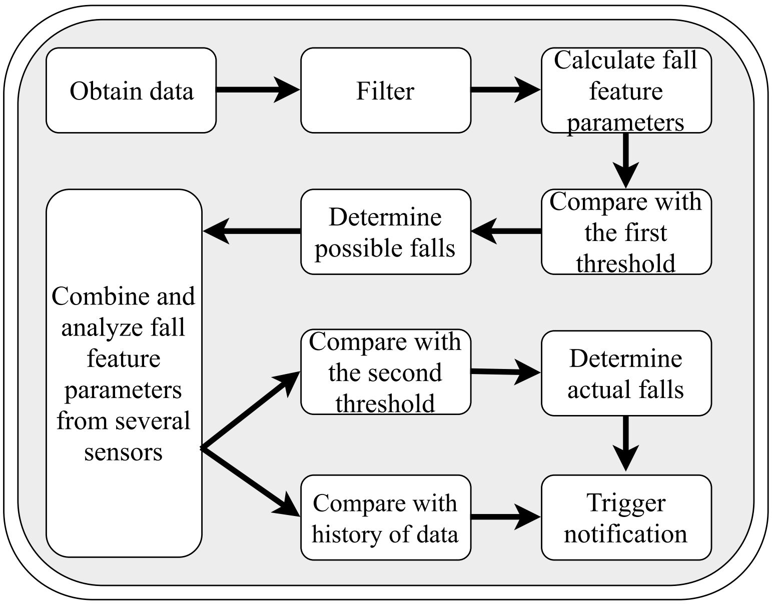


Figure 7.5: Fall detection algorithm flow

The fall feature parameters are compared with the first thresholds which are 1.6 g and 160 deg/s for SVM 3-D acceleration and SVM 3-D angular velocity, respectively. There are two possible results: (i) if all the parameters are higher than their first threshold values, they are compared with the second threshold values which are 1.9 g and 190 deg/s for SVM 3-D acceleration and SVM 3-D angular velocity, respectively. If one of the parameters is higher than the second threshold, a fall case is detected and the push notification service is triggered to inform the responsible person such as the caregiver; (ii) if one of the fall feature parameters (i.e., SVM of 3-D acceleration or SVM of 3-D angular velocity) is higher than its first threshold value while the other parameter is lower, both parameters are compared with their historical values which were collected 1 and 2 seconds before. If a similar pattern is detected, malfunctioned sensors are detected and the push notification is triggered.

In order to investigate the functionality of the fall detection algorithm, 3-D acceleration and 3-D angular velocity in two cases - with and without falling were acquired and analyzed. In each case, the data was collected during different movement activities such as standing still, standing with body movements, sitting still, sitting with body movements, and walking. Results are shown in Figure 7.6 and 7.7.

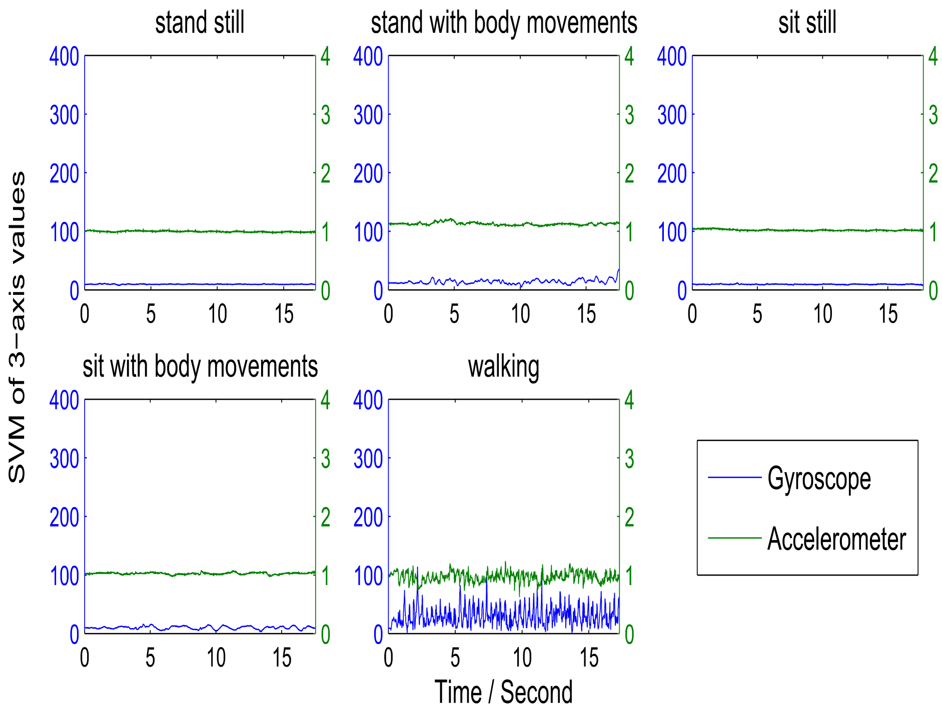


Figure 7.6: Accelerometer's and Gyroscope's data at the gateway's nRF receiver during daily activities

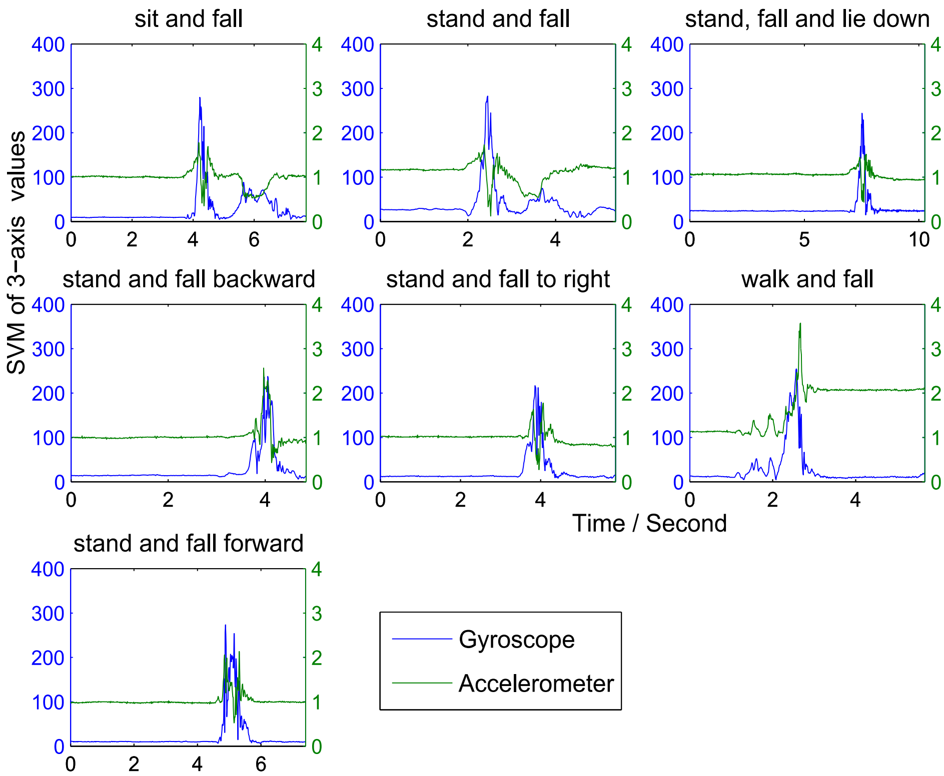


Figure 7.7: Accelerometer's and Gyroscope's data at the gateway's nRF receiver during daily activities and fall

The results show that the proposed fall detection algorithm functioned properly in different cases. In a case of where one of the sensors is malfunctioned, the algorithm still functions. In addition, the results reveal that relying on a single sensor may lead to an incorrect fall detection. For example, the 3-D accelerometer provided incorrect data in the case of a walk and fall in the experiment.

In order to evaluate a sensitivity of the fall detection algorithm, the data set mentioned above is still used. In the experimental process collecting the 60-minute data set, a user wearing the sensor node arbitrarily falls 1 to 2 times in each activity such as lying, standing, walking and running. Falls can be forward falls, backward falls or side falls. The fall detection algorithm was run 480 times in which a range window of 7.5-second 3-D acceleration and 7.5-second 3-D angular velocity was applied for each time. The algorithm accurately detected actual fall events in cases of lying on bed, standing, and walking. The algorithm detected incorrectly 12 times where there was no actual fall event but fall events were detected. One of the reasons causing the incorrect detection was a fast moving transition between activities. A sensitivity of the fall detection of algorithm in these cases is

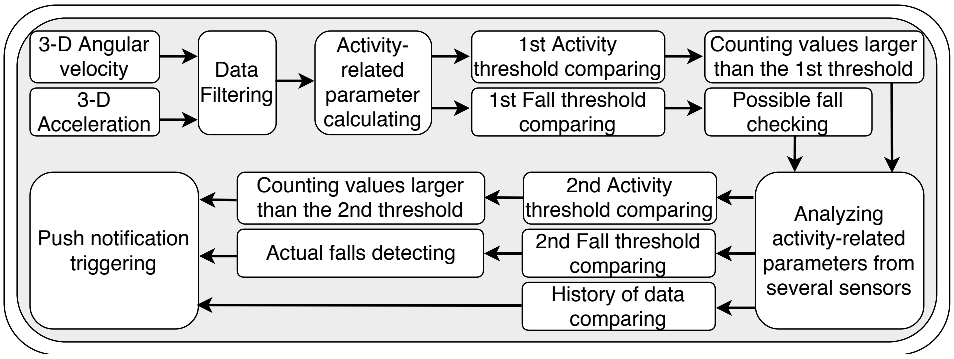


Figure 7.8: Activity status and fall detection algorithm

0.975. The algorithm detected incorrectly 62 times in a running activity case. Particularly, the algorithm detected all events including actual and not actual falls. A sensitivity of the algorithm in the running case is 0.87. The algorithm's sensitivity in the running case can vary depending on the specific type of running. The results show that the fall detection algorithm has be customized and run together with an activity status categorization algorithm in order to achieve a high quality of service.

In Paper VII, the proposed algorithm for both fall detection and activity status categorization can be considered as a customized and extended version of the previous fall algorithm shown in Paper VI. The proposed algorithm shown in Figure 7.8 can be also utilized for the combined system. Both algorithms have similar blocks such as data acquisition (3-D acceleration and 3-D angular velocity), data filtering and activity-related parameter calculating. In the customized algorithm, the activity related parameters are compared with both activity and fall threshold values. If the parameters surpass the first set of threshold values, the data is labelled as the "possibility" values and compared with the second set of threshold values and the historical data. If one of the parameters surpasses the second fall threshold value, a fall case is detected. For the activity status categorization, the number of values, which surpasses the first and second activity status categorization thresholds, is counted. In the customized algorithm, the first and second activity status categorization thresholds are 1.2 g, 1.6 g for 3-D acceleration, and 30 deg/s, 100 deg/s for 3-D angular velocity, respectively. The second activity status thresholds are set with high values for categorizing the activity status more accurately. In most of the cases, the activity parameters cannot surpass the second activity status categorization thresholds when a user is walking with a common speed of 1.4-1.6 m/s. However, the number of parameters surpassing the first activity status categorization thresholds, in this case, is large. When a user is running, the number of parameters surpassing the second activity status categorization thresholds is large. Comparing the acquired activity-related parameters with their historical values helps to improve the accuracy of the customized algorithm and detect malfunctioning sensors more accurately.

All collected data is simultaneously compared and analyzed by the system's Fog services presented in Paper VII. The results shown in Figure 7.9 and 7.10 demonstrate that the system stably works in different scenarios. The quality of ECG signals is high even when a user is walking, but when a user is running, the ECG waveform fluctuates dramatically; however, R-R peaks can still be detected. The moment when a user falls, ECG fluctuates but in a few seconds after a fall, an ECG waveform becomes appropriate again. Body temperature and blood glucose are not shown in Figure 7.9 and 7.10. They are shown in a text form in a user's interface. Data shown in these figures above is collected in the daily working environment.

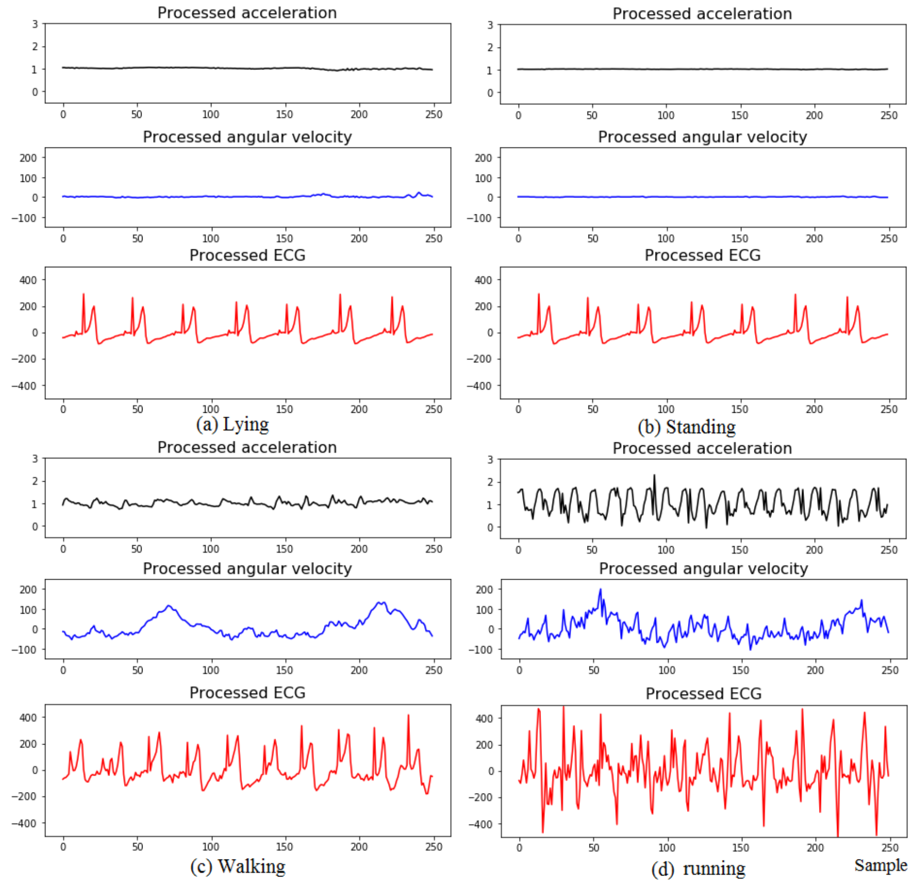
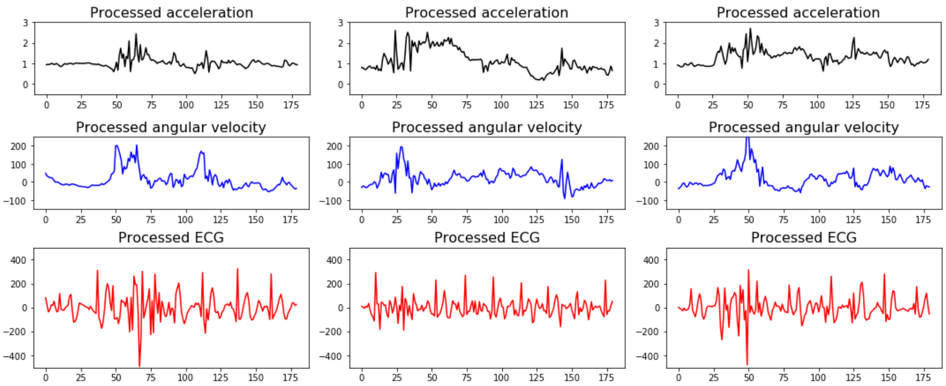


Figure 7.9: Acceleration, angular velocity and ECG in different activity status



(a) Walk and fall forward (b) Walk and fall back (c) Walk and fall aside

Figure 7.10: Acceleration, angular velocity and ECG when a user falls in different cases

The proposed algorithm for both fall detection and activity status categorization was run 480 times in which each running time operates with a range window of 7.5-second 3-D acceleration and 7.5-second 3-D angular velocity retrieved from the data set mentioned above. The algorithm accurately categorized different activities including lying on bed, standing, walking and running. In addition, the algorithm accurately detected fall events in cases of lying on bed, standing, and walking. A sensitivity of the fall detection part of the algorithm is 0.979 due to 10 incorrect detection times in which the algorithm detected 10 fall events while there was no actual fall. In case of running, a sensitivity of the fall detection part of the proposed algorithm is 0.877 because of 59 incorrect detection times.

The system proposed in Paper VII has most of the Fog services presented in Chapter 4. Accordingly, raw data and processed data can be monitored in real-time via end-user terminals such as a web browser and a mobile application. When abnormalities (i.e., high blood glucose, long QT intervals, fall, or a malfunctioned sensor) are detected, the system sends push notification messages to responsible individuals such as caregivers or system administrators. These Fog services can be utilized for the combined system. Based on the results of the experiments shown in Paper II, V, VI, and VII, the customized system which combines all the systems presented in these papers constitutes promising approaches for the treatment of diabetic patients with cardiovascular diseases. The system not only provides advanced services to enhance the quality of healthcare but also helps to improve the quality of life and reduce healthcare costs.

Chapter 8

Discussion and Conclusion

In this dissertation, we discussed and introduced Fog computing as a mean of enhancing the quality of services of the health monitoring IoT systems. Via the system implementation and verification in health monitoring case studies such as ECG monitoring and fall detection, the dissertation demonstrated that Fog computing is an appropriate approach particularly for enhancing remote health monitoring IoT systems and improving the quality of healthcare in general. By exploiting Fog computing which can be described as an extra Fog layer in between conventional gateways and Cloud servers, the health monitoring systems shown in this thesis proffered many advanced Fog services. These services include such features as distributed local data storage, ECG feature extraction, graphical interface with management access, interoperability, real-time push notification, data processing, channel managing, and security. These services will not only solve the challenges of the current health monitoring IoT systems (e.g., high latency and service interruption) but also help to improve the quality of healthcare and reduce healthcare costs. Via these services, caregivers will be able to monitor patient health remotely and respond to abnormalities such as high body temperature, and long QT intervals of an ECG wave in real-time in order to save a patient's life. In addition, these services will help to improve the accuracy of disease diagnosis via real-time analysis the current and historical data including both e-health data and contextual data. Furthermore, the services protect the information health privacy of patients, and the strict requirements of low latency has been fulfilled.

In addition to supporting advanced Fog services, an advanced remote health monitoring needs to be capable of mobility support in order to improve the quality of healthcare services. Therefore, mobility support was thoroughly discussed and considered. The mobility support issue becomes more difficult when dealing with Fog-assisted health monitoring IoT systems. It is required that a mobility support approach not only guarantees that data is continuously monitored without any interruption but also ensures that the Fog services are properly maintained. The dissertation proposed a Fog approach for mobility support in health monitoring IoT systems. The approach did not limit the patient's movement while fulfilling the latency requirements of real-time data monitoring systems. The handover mechanism in the mobility support approach carefully considered oscillating nodes which often happen in many real-time monitoring IoT systems. By exploiting the Fog approach, the burdens on the sensor nodes were reduced while enhanced Fog services such as distributed local data storage and push notification were proffered. The results from evaluating the proposed Fog approach for mobility support are promising as the latency of switching between Fog-assisted smart gateways are 10%-50% smaller than other state-of-the-art approaches.

The high quality of healthcare services cannot be maintained when remote health monitoring systems are energy inefficient. Therefore, the design of energy efficient health monitoring systems was discussed. In particular, energy efficient fall detection, ECG monitoring, and glucose monitoring systems were proposed. Each system applied the Fog computing concept at the edge of the network for shifting the burdens of sensor nodes to Fog-assisted smart gateways in order save energy consumption of sensor nodes. In addition, the sensor nodes of these sensors were designed to achieve a high level of energy efficiency while high-quality data is being collected and transmitted. Each power consumption source of the sensor nodes was specifically and carefully considered. Results from the design of energy efficient health monitoring IoT systems show that the proposed sensor nodes for ECG monitoring, fall detection, and blood glucose monitoring consume low energy and are 20-50% more energy efficient than other state-of-the-art sensor nodes. These sensor nodes could run for up to 70-155 hours depending on the particular sensor node when using a 1000 mAh lithium battery.

In order to improve the quality of healthcare particularly for human fall, diabetes and cardiovascular disease, advanced Fog-assisted health monitoring IoT systems were designed and developed. These systems could monitor contextual and e-health data such as ECG, blood glucose, body motion, body temperature, room temperature, and air quality in real-time without infringing healthcare requirements of latency, security, and high-quality signals. In addition, these systems equipped with light-weight algorithms such as QT length extraction, fall detection, and activity status categorization could accurately analyze the data in real-time. The results show that the proposed systems are reliable and they are a promising solution for overcoming the existing challenges of health monitoring systems particularly in relation to diabetes, cardiovascular disease, and human fall.

Chapter 9

Overview of Original Publications

This chapter provides an overview of the original published articles which have been included in the thesis.

9.1 Paper I: Fog Computing in Healthcare Internet of things: A Case Study on ECG Feature Extraction

In this paper, an energy-efficient IoT-based system architecture leveraging the Fog concept is proposed. A convergent network of smart gateways is specifically proposed where each gateway is equipped with Fog computing services as a core feature of the system architecture. The paper presented a complete architecture (i.e., physical and operational structures) of these smart gateways. These smart gateways can support many types of sensor nodes using different wireless communication protocols. Via the system architecture, advanced services can be deployed to improve the quality of services. For instance, the proposed algorithm for ECG feature extraction and other data processing methods can be implemented at the Fog-assisted gateway to provide real-time analyses and utilize network bandwidth efficiently. In detail, ECG features such as heart rate, P wave, and T wave can be extracted, analyzed and monitored in real-time. When these ECG features are abnormal, caregivers are informed in real-time. In addition, the system architecture and Fog services help to save the energy consumption of sensor nodes by shifting the burdens of the sensor nodes to smart gateways.

Author's contribution: The author is the main author of this paper. The author proposed and implemented the complete Fog-assisted IoT-based system for remote health monitoring. In addition, the author designed a low-latency algorithm for ECG feature detection. The author analyzed and evaluated latency and network bandwidth of the systems with and without Fog computing. Furthermore, the author augmented Fog-assisted gateway's services.

9.2 Paper II: Low-cost Fog-assisted Health-care IoT System with Energy-efficient Sensor Nodes

In this paper, a low-cost Fog-assisted IoT-based system for health monitoring is proposed. The system comprises sensor nodes, Fog-assisted smart gateways, a back-end system with terminals such as a web browser and a mobile application. A wearable sensor node collects both e-health data (e.g., body temperature and Electrocardiography) and contextual data such as room temperature and humidity. Then, it sends the data to Fog-assisted smart gateway for further processing. The proposed sensor node is wearable, low-cost and energy-efficient. When using a 1000 mAh battery, the sensor node can operate for up to 155 hours. Fog-assisted gateways combining with Cloud servers proffer many advanced services such as a categorization service, channel managing, and data processing for improving the quality of healthcare service. For instance, these services help to reduce latency and avoid data corruption during transmission. In addition, the system constantly checks the patient's heart rate which is extracted from ECG. When the heart rate is abnormal, the system will inform caregivers in real-time.

Author's contribution: The author is the main author of this paper. The author proposed and implemented the complete Fog-assisted IoT system based on nRF for remote health monitoring. The author designed and built energy-efficient wearable sensor nodes. Furthermore, the author proposed and implemented the Fog-assisted gateway and its augmented services (e.g., data processing, categorization and channel managing). The author configured the Cloud servers and built a mobile application for monitoring ECG and receiving push notification messages in real-time.

9.3 Paper III: Exploiting Smart E-health Gateways at the Edge of Healthcare Internet-of-things: a Fog Computing Approach

This paper presents a smart e-health IoT system which exploits Fog computing at the edge of the network for improving the quality of services. The paper provides proof of the concepts of healthcare services at Fog-assisted gateways such as embedded data mining, real-time data processing, syntactic interoperability, and security. The paper demonstrates that Fog computing approaches are practical solutions for challenges in IoT-based health monitoring systems. For instance, the system architecture and Fog services help to enhance the security level, save energy consumption of sensor nodes, and efficiently utilize network bandwidth. In addition, the system provides early warning score services which analyze e-health data and send information to caregivers about abnormalities.

The main contribution of this paper is presented as follows:

- Demonstrating the feasibility of practically applying Fog computing into the existing IoT-based health monitoring system
- Providing the proof of the concept of healthcare services at Fog assisted gateways and enhancing the existing healthcare services
- Presenting mobility and interoperability support in Fog-assisted smart gateways
- Presenting a case study of an early warning score in Fog-based systems for healthcare monitoring

Author's contribution: The author is the second author of this paper. The author implemented the sensor nodes and Fog-assisted smart gateways. In detail, the author constructed several e-health and contextual sensor nodes which use different communication protocols such as 6LoWPAN, Bluetooth, and Wi-Fi. In order to support these protocols, the author designed and implemented both the physical and operational structure of the Fog-assisted smart gateways. In addition, the author designed and implemented several Fog-based services such as interoperability, security, data filtering, data compression and local data storage. The author partly built the early warning score system. Furthermore, via practical experiments, the author demonstrated the possibility of saving the energy consumption of sensor nodes by shifting the burdens of sensor nodes to Fog-assisted gateways.

9.4 Paper IV: Fog Computing Approach for Mobility Support in Internet-of-Things Systems

In this paper, a Fog-based approach for mobility support in health monitoring IoT systems is presented. The proposed Fog computing approach has a handover mechanism which keeps the connection between sensor nodes and gateways with a low latency even though sensor nodes moves between several coverage areas of the system. Furthermore, the handover mechanism is able to handle oscillating nodes which move back and forth between the coverage areas of gateways over a short time period. In the proposed approach, heavy computation tasks are shifted to Fog-assisted smart gateway in order to save energy consumption in the sensor nodes and maintain the high-quality services. Results from experiments show that the latency of the Fog-based approach is 10% - 50% less than other state-of-the-art approaches for mobility support.

Author's contribution: The author is the main author of this paper. In this paper, the author discusses the mobility type in IoT systems and analyzes important metrics used in handover mechanism especially for Wi-Fi connection. The author designed and built the complete Fog-based health motoring system consisting of sensor nodes, Fog-assisted smart gateways and Cloud servers with terminals. In addition, the author designed, implemented, and evaluated the Fog-based approach for mobility support.

9.5 Paper V: IoT-based Continuous Glucose Monitoring System: A Feasibility Study

This paper presents an IoT-based system for continuous glucose monitoring. Via the system, caregivers can monitor glucose levels from many patients simultaneously and remotely in real-time while the patient's movements are not limited or intervened. The proposed system provides many advanced services (e.g., local distributed data storage) at smart mobile gateways to improve the quality of healthcare services. Historical and current motion-related data in particular can be accessed at any time remotely even though the connection between gateways and Cloud servers is occasionally interrupted. In addition, the system sends push notification messages to inform responsible persons such as caregivers when abnormalities such as high blood glucose level occur. In this paper, a wearable sensor device using the nRF protocol for glucose monitoring and its energy harvesting unit are proposed. The sensor device is energy-efficient with 1.4 mA current consumption.

The main contributions of this paper are:

- proposing a smart IoT-based system architecture using a smartphone-based gateway for remote continuous glucose monitoring in real-time
- designing an energy harvesting unit for a wearable glucose monitoring sensor device
- proposing an energy-efficient wearable sensor device for glucose and body temperature monitoring

Author's contribution: The author is the primary author of this paper. The author proposed and implemented the complete smart IoT-based system for remote continuous glucose monitoring. Specifically, the author designed and built energy-efficient wearable sensor devices for collecting glucose and body temperature in real-time. Furthermore, the author proposed and implemented a smart mobile gateway and its advanced services (e.g., local distributed data storage, push notification, data filtering and localhost with user interface). The author configured the Cloud servers and constructed a mobile application to monitor data and receive push notification messages in real-time.

9.6 Paper VI: Energy-efficient Wearable Sensor Node for IoT-based Fall Detection Systems

In this paper, an IoT-based fall detection system is proposed and implemented. The completed system consists of sensor nodes, smart gateways and Cloud servers with end-user terminals. The sensor node collects three-dimensional (3-D) acceleration, 3-D angular velocity and, 3-D magnetic forces and then sends to the smart gateway. The wearable sensor node is energy-efficient, small, light-weight and does not interfere with a user's daily activity. A smart gateway can detect a fall case in real-time with a light-weight fall detection algorithm. It then sends instant messages to inform caregivers. In order to achieve a high level of energy efficiency, we set up different configurations of sensor nodes. Each configuration had a specific sensor or a group of sensors which used several sampling rates to collect data. Several types of data communication protocols were applied such as SPI, I2C, and UART which are used for connecting sensors and a micro-controller. The experiments included several transmission ranges and types (e.g., line-of-sight and blocked path transmissions) between the sensor nodes and the gateway. We evaluated the energy consumption of the sensor nodes in these cases. In addition, acceleration and angular velocity in different cases such as *sit still*, *sit and fall*, *stand still*, *stand and fall*, *stand and fall forward*, *stand and fall backward*, and *walk and fall* were collected and analyzed. The experimental results showed that the proposed sensor node is energy-efficient and it can continuously work for up to 76 hours with a 1000 mAh lithium battery.

Author's contribution: The author is the primary author of this paper. The author proposed and implemented the complete fall detection IoT-based system. The author constructed energy-efficient and wearable sensor nodes. The author proposed a light-weight fall detection algorithm at a smart gateway. In addition, the author evaluated and analyzed the energy consumption of sensor nodes in different situations and configurations. Furthermore, the author provided a comprehensive discussion to improve the energy efficiency of a sensor node for the fall detection system.

9.7 Paper VII: Energy-efficient Fog-assisted IoT System for Monitoring Diabetic Patients with Cardiovascular Disease

In this paper, a Fog-assisted IoT system for monitoring diabetic patients with cardiovascular disease is presented. The system consists of sensor nodes, Fog-assisted smart gateways and a Cloud layer with terminals. Sensor nodes can be categorized into contextual and e-health nodes. The contextual sensor node is used to collect room temperature, humidity, air quality and other context-related data. The e-health sensor node is able to acquire Electrocardiography (ECG), blood glucose, body temperature, skin humidity and motion-related data (i.e., 3-D angular velocity and 3-D acceleration). In this paper, algorithms for ECG feature extraction, fall detection, and status activities categorization are presented and implemented at Fog-assisted smart gateways. When these algorithms detect abnormalities, the push notification service is triggered to inform caregivers. The connection between sensor nodes and smart gateways are secured via a light-weight cryptography algorithm which causes a light increase in the power consumption of sensor nodes. Furthermore, energy-efficient wearable sensor nodes are proposed in this paper. In most of cases, the sensor nodes can work for up to 157 hours in a daily working or living environment such as office or apartment. In addition, we analyzed and evaluated the power consumption, and latency of sensor nodes in different scenarios.

The main contributions of this paper are:

- Proposing and implementing a complete Fog-assisted healthcare IoT system
- Designing a light-weight algorithm for ECG feature extraction
- Proposing a light-weight algorithm for fall detection and activity status categorization
- Designing an energy-efficient wearable sensor node for collecting ECG, glucose and motion-related data
- Analyzing e-health data in different activity statuses
- Analyzing and evaluating energy consumption of sensor nodes in different scenarios

Author's contribution: In this paper, the author is the main author and contributor of all the contributions mentioned except for the ECG feature extraction algorithm.

Bibliography

[1] Older americans & cardiovascular diseases. *American Heart Association.* Available:https://www.heart.org/idc/groups/heart-public/wcm/sop/smd/documents/downloadable/ucm_319574.pdf.

[2] G. W. Albers *et al.* Heart rhythm monitoring strategies for cryptogenic stroke: 2015 diagnostics and monitoring stroke focus group report. *Journal of the American Heart Association*, 5(3):e002944, 2016.

[3] B. Riegel *et al.* Self-care for the prevention and management of cardiovascular disease and stroke: A scientific statement for healthcare professionals from the american heart association. *Journal of the American Heart Association*, 6(9):e006997, 2017.

[4] Diabetes. *WHO*, 2017. Available: http://www.who.int/mediacentre/factsheets/fs312/en/, Accessed: Jan 2018.

[5] K. D. Kochanek et al. Deaths: Final data for 2014. *National Vital Statistics Reports*, 65(4), 2016.

[6] C. D. Mathers and D. Loncar. Projections of global mortality and burden of disease from 2002 to 2030. *PLoS medicine*, 3(11):e442, 2006.

[7] Rates of new diagnosed cases of type 1 and type 2 diabetes on the rise among children, teens. Available: https://www.nih.gov/news-events/news-releases/rates-new-diagnosed-cases-type-1-type-2-diabetes-rise-among-children-teens, Accessed: May 2018.

[8] D. A. Sterling *et al.* Geriatric falls: injury severity is high and disproportionate to mechanism. *Journal of Trauma and Acute Care Surgery*, 50(1):116–119, 2001.

[9] J. A. Stevens *et al.* The costs of fatal and non-fatal falls among older adults. *Injury prevention*, 12(5):290–295, 2006.

[10] A. C. Scheffer *et al.* Fear of falling: measurement strategy, prevalence, risk factors and consequences among older persons. *Age and ageing*, 37(1):19–24, 2008.

[11] R. Igual *et al.* Challenges, issues and trends in fall detection systems. *Biomedical engineering online*, 12(1):1, 2013.

[12] S. Volpato *et al.* Risk factors for falls in older disabled women with diabetes: the women's health and aging study. *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 60(12):1539–1545, 2005.

[13] Cardiovascular disease & diabetes. Dec.2017. https://www.heart.org/HEARTORG/Conditions/More/Diabetes /WhyDiabetesMatters/CardiovascularDisease-Diabetes_UCM_313865_Article.jsp/, Accessed: Jan.2018.

[14] R. T. Robinson *et al.* Mechanisms of abnormal cardiac repolarization during insulin-induced hypoglycemia. *Diabetes*, 52(6):1469–1474, 2003.

[15] R. Vedanthan, V. Fuster, and A. Fischer. Sudden cardiac death in low-and middle-income countries. *Global heart*, 7(4):353–360, 2012.

[16] Cardiovascular disease costs will exceed $1 trillion by 2035, warns the american heart association. *American Heart Association*, 2017. Available: https://newsroom.heart.org/news/cardiovasculardisease-costswillexceed1trillionby2035warnstheamericanheartassociation, Accessed: May 2018.

[17] Kronos Inc. Guided analytics: Uncovering hidden labor costs. Updated: Jan. 2017, Accessed Feb 2018, https://www.hfma.org/DownloadAsset.aspx?id=38019.

[18] D. A. Leon and G. Walt. Poverty inequality and health in international perspective: a divided world? 2001.

[19] J. Zhang, V. L. Patel, and T. R. Johnson. Medical error: Is the solution medical or cognitive? *Journal of the American Medical Informatics Association*, 9(Supplement_6):S75–S77, 2002.

[20] H. Sundmaeker *et al.* Vision and challenges for realising the internet of things. *Cluster of European Research Projects on the Internet of Things, European Commision*, 2010.

[21] T. N. Gia *et al.* Customizing 6lowpan networks towards internet-of-things based ubiquitous healthcare systems. In *NORCHIP, 2014*, pages 1–6. IEEE, 2014.

[22] T. N. Gia *et al.* Fault tolerant and scalable IoT based architecture for healthcare monitoring. *Sensors Applications Symposium (SAS)*, pages 1–6, 2015.

[23] T. N. Gia *et al.* Iot-based fall detection system with energy efficient sensor nodes. In *Nordic Circuits and Systems Conference (NORCAS), 2016 IEEE*, pages 1–6. IEEE, 2016.

[24] I. Tcarenko *et al.* Energy-efficient iot-enabled fall detection system with messenger-based notification. In *International Conference on Wireless Mobile Communication and Healthcare*, pages 19–26. Springer, 2016.

[25] M. Jiang *et al.* Iot-based remote facial expression monitoring system with semg signal. In *Sensors Applications Symposium (SAS), 2016 IEEE*, pages 1–6. IEEE, 2016.

[26] D. C. Klonoff. Cybersecurity for connected diabetes devices. *Journal of diabetes science and technology*, 9(5):1143–1147, 2015.

[27] H. Shin. Ambient temperature effect on pulse rate variability as an alternative to heart rate variability in young adult. *Journal of clinical monitoring and computing*, 30(6):939–948, 2016.

[28] A. Rahmani *et al.* *Fog Computing in the Internet of Things*. Springer International Publishing, 2018.

[29] F. Bonomi *et al.* Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[30] V. K. Sarker *et al.* Portable multipurpose bio-signal acquisition and wireless streaming device for wearables. In *Sensors Applications Symposium (SAS), 2017 IEEE*, pages 1–6. IEEE, 2017.

[31] E. Casilari and M. A. Oviedo-Jiménez. Automatic fall detection system based on the combined use of a smartphone and a smartwatch. *PloS one*, 10(11):e0140929, 2015.

[32] D. Chen, W. Feng, Y. Zhang, X. Li, and T. Wang. A wearable wireless fall detection system with accelerators. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 2259–2263. IEEE, 2011.

[33] P. Pivato, S. Dalpez, D. Macii, and D. Petri. A wearable wireless sensor node for body fall detection. In *Measurements and Networking Proceedings (M&N), 2011 IEEE International Workshop on*, pages 116–121. IEEE, 2011.

[34] F. Wu, H. Zhao, Y. Zhao, and H. Zhong. Development of a wearable-sensor-based fall detection system. *International journal of telemedicine and applications*, 2015:2, 2015.

[35] D. Yacchirema *et al.* Fall detection system for elderly people using iot and big data. *Procedia computer science*, 130:603–610, 2018.

[36] J. He, C. Hu, and X. Wang. A smart device enabled system for autonomous fall detection and alert. *International Journal of Distributed Sensor Networks*, 12(2):2308183, 2016.

[37] O. Biros, J. Karchnak, D. Simsik, and A. Hosovsky. Implementation of wearable sensors for fall detection into smart household. In *Applied Machine Intelligence and Informatics (SAMI), 2014 IEEE 12th International Symposium on*, pages 19–22. IEEE, 2014.

[38] S. Z. Erdogan and T. T. Bilgin. A data mining approach for fall detection by using k-nearest neighbour algorithm on wireless sensor network data. *IET Communications*, 6(18):3281–3287, 2012.

[39] C. Shuo. Fall detection system using arduino fio. In *Proceedings of the IRC Conference on Science, Engineering and Technology, Singapore*, volume 13, 2015.

[40] A Murakami, MA Gutierrez, SHG Lage, MFS Rebelo, RHG Guiraldelli, and JAF Ramires. A continuous glucose monitoring system in critical cardiac patients in the intensive care unit. In *Computers in Cardiology, 2006*, pages 233–236. IEEE, 2006.

[41] M. Ali, L. Albasha, and H. Al-Nashash. A bluetooth low energy implantable glucose monitoring system. In *Radar Conference (EuRAD), 2011 European*, pages 377–380. IEEE, 2011.

[42] J. Y. Lucisano, T. L. Routh, J. T. Lin, and D. A. Gough. Glucose monitoring in individuals with diabetes using a long-term implanted sensor/telemetry system and model. *IEEE Transactions on Biomedical Engineering*, 64(9):1982–1993, 2017.

[43] M. U. H. Al Rasyid *et al.* Implementation of blood glucose levels monitoring system based on wireless body area network. In *Consumer Electronics-Taiwan (ICCE-TW), 2016 IEEE International Conference on*, pages 1–2. IEEE, 2016.

[44] N. Wang and G. Kang. A monitoring system for type 2 diabetes mellitus. In *e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on*, pages 62–67. IEEE, 2012.

[45] R. S. H. Istepanian *et al.* The potential of Internet of m-health Things ḿ-IoT́for non-invasive glucose level sensing. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 5264–5266. IEEE, 2011.

[46] A. J. Jara *et al.* An internet of things-based personal device for diabetes therapy management in ambient assisted living (aal). *Personal and Ubiquitous Computing*, 15(4):431–440, 2011.

[47] A. M. Rahmani *et al.* Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems. In *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*, pages 826–834. IEEE, 2015.

[48] P. Singh and A. Jasuja. Iot based low-cost distant patient ecg monitoring system. In *Computing, Communication and Automation (ICCA), 2017 International Conference on*, pages 1330–1334. IEEE, 2017.

[49] Z. Yang *et al.* An iot-cloud based wearable ecg monitoring system for smart healthcare. *Journal of medical systems*, 40(12):286, 2016.

[50] C. Li, X. Hu, and L. Zhang. The iot-based heart disease monitoring system for pervasive healthcare service. *Procedia Computer Science*, 112:2328–2334, 2017.

[51] M. Koshti, S. Ganorkar, and L. Chiari. Iot based health monitoring system by using raspberry pi and ecg signaly. *International Journal of Innovative Research in Science, Engineering and Technology*, 5(5), 2016.

[52] S. Valenzuela, M. Chen, and V. Leung. Mobility support for health monitoring at home using wearable sensors. *IEEE Trans. Inf. Technol. Biomed. 15 (4)*, pages 539–549, 2011.

[53] A. Jara, M. Zamora, and A. Skarmeta. An initial approach to support mobility in hospital wireless sensor networks based on 6LoWPAN (HWSN6). *J. Wirel. Mob. Netw., Ubiquitous Comput., Dependable Appl. 1 (2-3)*, pages 107–122, 2010.

[54] A. Jara *et al.* Hwsn6: Hospital wireless sensor networks based on 6lowpan technology: Mobility and fault tolerance management. *in: International Conference on Computational Science and Engineering*, 2:879–884, 2009.

[55] A. Jara, M. Zamora, and A. Skarmeta. Intra-mobility for hospital wireless sensor networks based on 6lowpan. *in: 6th International Conference on Wireless and Mobile Communications*, pages 389–394, 2010.

[56] H. C. Tung *et al.* A mobility enabled inpatient monitoring system using a zigbee medical sensor network. *Sensors*, 14(2):2397–2416, 2014.

[57] A. Boulemtafes, A. Rachedi, and N. Badache. A study of mobility support in wearable health monitoring systems: Design framework. In *Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of*, pages 1–8. IEEE, 2015.

[58] F. Touati and R. Tabish. U-healthcare system: State-of-the-art review and challenges. *Journal of medical systems*, 37(3):9949, 2013.

[59] J. J. Bailey *et al.* Recommendations for standardization and specifications in automated electrocardiography: bandwidth and digital signal processing. a report for health professionals by an ad hoc writing group of the committee on electrocardiography and cardiac electrophysiology of the council on clinical cardiology, american heart association. *Circulation*, 81(2):730, 1990.

[60] W. Einthoven. The string galvanometer and the measurement of the action currents of the heart. *Nobel Lecture, December*, 11, 1925.

[61] Clinical ECG interpretation. Ecg interpreation: Characteristics of the normal ecg (p-wave, qrs complex, st segment, t-wave). Updated: Jan. 2017, Accessed: May 2018, https://ecgwaves.com/ecg-normal-p-wave-qrs-complex-st-segment-t-wave-j-point/.

[62] W.G. James. eheart: An introduction to ecg/ekg. 2006. Updated: Jan. 2017, Accessed: May 2018, https://www.ndsu.edu/pubweb/ grier/eheart.html.

[63] ECG learning center. The standard 12 lead ecg. Updated: Jan. 2017, Accessed: May 2018, https://ecg.utah.edu/lesson/1.

[64] Bristish Heart Foundation. Your heart rate. Updated: Jan. 2017, Accessed Feb 2018, https://www.bhf.org.uk/-/media/files/publications/medical-information-sheets/your-heart-rate-is23.pdf.

[65] Å. Hjalmarson *et al.* Influence of heart rate on mortality after acute myocardial infarction. *American Journal of Cardiology*, 65(9):547–553, 1990.

[66] J .W. Mason *et al.* Electrocardiographic reference ranges derived from 79,743 ambulatory subjects. *Journal of electrocardiology*, 40(3):228–234, 2007.

[67] Tachycardia — Fast Heart Rate. Updated: Jan. 2017, Accessed Feb 2018,https://www.heart.org/HEARTORG/Conditions/Arrhythmia/AboutArrhythmia/Tachycardia_UCM_302018_Article.jsp.

[68] H. Tanaka, K. D. Monahan, and D. R. Seals. Age-predicted maximal heart rate revisited. *Journal of the American College of Cardiology*, 37(1):153–156, 2001.

[69] R. A. Robergs and R. Landwehr. The surprising history of the" hrmax= 220-age" equation. *Journal of Exercise Physiology Online*, 5(2):1–10, 2002.

[70] J. Karvonen and T. Vuorimaa. Heart rate and exercise intensity during sports activities. *Sports Medicine*, 5(5):303–311, 1988.

[71] Know your target heart rates for exercise, losing weight and health. 2018. Updated: Jan. 2018, Accessed: May 2018, https://www.heart.org/en/healthy-living/fitness/fitness-basics/target-heart-rates.

[72] D. H. Wasserman. Four grams of glucose. *American Journal of Physiology-Endocrinology and Metabolism*, 296(1):E11, 2009.

[73] R. Walker and J. R. Type. Diabetes–your questions answered, dorling kindersley, 2006. 2.

[74] J. S. Hutchison *et al.* Hypothermia therapy after traumatic brain injury in children. *New England Journal of Medicine*, 358(23):2447–2456, 2008.

[75] S. Kushimoto *et al.* The impact of body temperature abnormalities on the disease severity and outcome in patients with severe sepsis: an analysis from a multicenter, prospective survey of severe sepsis. *Critical care*, 17(6):R271, 2013.

[76] W. Tao, T. Liu, R. Zheng, and H. Feng. Gait analysis using wearable sensors. *Sensors*, 12(2):2255–2283, 2012.

[77] E. Sardini, M. Serpelloni, and M. Ometto. Smart vest for posture monitoring in rehabilitation exercises. In *Sensors Applications Symposium (SAS), 2012 IEEE*, pages 1–5. IEEE, 2012.

[78] P. S. Pandian *et al.* Smart vest: Wearable multi-parameter remote physiological monitoring system. *Medical engineering & physics*, 30(4):466–477, 2008.

[79] S. C. Chavan and A. Chavan. Smart wearable system for fall detection in elderly people using internet of things platform. In *Intelligent Computing and Control Systems (ICICCS), 2017 International Conference on*, pages 1135–1140. IEEE, 2017.

[80] A. Cadei *et al.* Kinetic and thermal energy harvesters for implantable medical devices and biomedical autonomous sensors. *Measurement Science and Technology*, 25(1):012003, 2013.

[81] K. Fotopoulou and B. W. Flynn. Wireless powering of implanted sensors using rf inductive coupling. In *5th IEEE Conference on Sensors*, pages 765–768, 2006.

[82] S. K. Park *et al.* Air pollution and heart rate variability: effect modification by chronic lead exposure. *Epidemiology (Cambridge, Mass.)*, 19(1):111, 2008.

[83] S. K. Park *et al.* Effects of air pollution on heart rate variability: the va normative aging study. *Environmental health perspectives*, 113(3):304, 2005.

[84] T. N. Gia *et al.* Fog computing in body sensor networks: An energy efficient approach. In *IEEE International Body Sensor Networks Conference (BSN15)*, 2015.

[85] A. M. Rahmani *et al.* Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach. *Future Generation Computer Systems*, 2017.

[86] S. M. Moosavi *et al.* Sea: a secure and efficient authentication and authorization architecture for iot-based healthcare using smart gateways. *Procedia Computer Science*, 52:452–459, 2015.

[87] S. M. Moosavi *et al.* Session resumption-based end-to-end security for healthcare internet-of-things. In *CIT 2015*, pages 581–588. IEEE, 2015.

[88] S. R. Moosavi *et al.* End-to-end security scheme for mobility enabled healthcare internet of things. *Future Generation Computer Systems*, 64:108–124, 2016.

[89] M. D. Francesco, S. K. Das, and G. Anastasi. Data collection in wireless sensor networks with mobile elements: A survey. *ACM Trans. Sensor Netw.*, 8:1–7, 2011.

[90] P. Kulkarni et al. Requirements and design spaces of mobile medical care. *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, 11:12–30, 2007.

[91] R. Mulligan and H.M. Ammari. Coverage in wireless sensor networks: A survey. *Netw. Protocols Algorithms*, 2(2):27–53, 2010.

[92] L. F. Bittencourt *et al.* Towards virtual machine migration in fog computing. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2015 10th International Conference on*, pages 1–8. IEEE, 2015.

[93] L. F. Bittencourt *et al.* Mobility-aware application scheduling in fog computing. *IEEE Cloud Computing*, 4(2):26–35, 2017.

[94] Y. Chen and Y. Tsai. A mobility management using follow-me cloud-cloudlet in fog-computing-based rans for smart cities. *Sensors*, 18(2):489, 2018.

[95] H. Fotouhi, M. Alves, M.Z. Zamalloa, and A. Koubaa. Reliable and fast hand-offs in low-power wireless networks. *IEEE Trans. Mob. Comput. 13 (11)*, pages 2620–2633, 2014.

[96] R. Silva, J. S. Silva, and F. Boavida. Mobility in wireless sensor networks–survey and proposal. *Computer Communications*, 52:1–20, 2014.

[97] I. Fredriksen and P. Kastnes. Choosing a mcu for your next design; 8 bit or 32 bit?, 2014. Accessed Jul 2017, http://www.atmel.com/images/45107a-choosing-a-mcu-fredriksen_article_103114.pdf.

[98] M. Sanchez-Alavez *et al.* Insulin causes hyperthermia by direct inhibition of warm-sensitive neurons. *Diabetes*, 59(1):43–50, 2010.

[99] M. Taghadosi et al. Miniaturised printed elliptical nested fractal multiband antenna for energy harvesting applications. *IET Microwaves, Antennas Propagation*, 2015.

[100] R. F. Yazicioglu, C. Van Hoof, and R. Puers. *Biopotential readout circuits for portable acquisition systems.* Springer Science & Business Media, 2008.

[101] Y. Li *et al.* Accelerometer-based fall detection sensor system for the elderly. In *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on*, volume 3, pages 1216–1220. IEEE, 2012.

[102] S-C. Lee and W-Y. Chung. A robust wearable u-healthcare platform in wireless sensor network. *Journal of Communications and Networks*, 16(4):465–474, 2014.

[103] S. Mahmud *et al.* An inexpensive and ultra-low power sensor node for wireless health monitoring system. In *E-health Networking, Application & Services, 2015 17th International Conference on*, pages 495–500. IEEE, 2015.

[104] R. Dilmaghani *et al.* Wireless sensor networks for monitoring physiological signals of multiple patients. *IEEE transactions on biomedical circuits and systems*, 5(4):347–356, 2011.

[105] Blood Sugar Level Ranges. Diabetes.co.uk, https://www.diabetes.co.uk/diabetes$_care/blood - sugar - level - ranges.html[accessed : Aug.$2018].

[106] Blood glucose monitoring. Diabetes Australia, https://www.diabetesaustralia.com.au/blood-glucose-monitoring [accessed: Aug. 2018].

[107] B. Vandenberk *et al.* Which qt correction formulae to use for qt monitoring? *Journal of the American Heart Association*, 5(6):e003264, 2016.

[108] H. C. Bazett. An analysis of the time-relations of electrocardiograms. *Annals of Noninvasive Electrocardiology*, 2(2):177–194, 1997.

[109] L. S. Fridericia. Die systolendauer im elektrokardiogramm bei normalen menschen und bei herzkranken. *Journal of Internal Medicine*, 53(1):469–486, 1920.

[110] A. Sagie *et al.* An improved method for adjusting the qt interval for heart rate (the framingham heart study). *American Journal of Cardiology*, 70(7):797–801, 1992.

[111] M Hodges. Bazett's qt correction reviewed: evidence that a linear qt correction for heart rate is better. *J Am Coll Cardiol*, 1:694, 1983.

[112] P. M. Rautaharju *et al.* Assessment of prolonged qt and jt intervals in ventricular conduction defects. *American Journal of Cardiology*, 93(8):1017–1021, 2004.

Part II

Original Publications

Paper I

Fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction

Tuan Nguyen Gia, Mingzhe Jiang, Amir-Mohammad Rahmani, Tomi Westerlund, Pasi Liljeberg, Hannu Tenhunen

Original published in Proceeding of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015, pages 356-363, Liverpool, United Kingdom.

2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing

Fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction

Tuan Nguyen Gia[1], Mingzhe Jiang[1] Amir-Mohammad Rahmani[1,2], Tomi Westerlund[1], Pasi Liljeberg[1], and Hannu Tenhunen[1,2]
[1]Department of Information Technology, University of Turku, Turku, Finland
[2] Department of Industrial and Medical Electronics, KTH Royal Institute of Technology, Stockholm, Sweden
Email: {tunggi, mizhji, amirah, tovewe, pakrli}@utu.fi, hannu@kth.se

***Abstract*—Internet of Things technology provides a competent and structured approach to improve health and wellbeing of mankind. One of the feasible ways to offer healthcare services based on IoT is to monitor humans health in real-time using ubiquitous health monitoring systems which have the ability to acquire bio-signals from sensor nodes and send the data to the gateway via a particular wireless communication protocol. The real-time data is then transmitted to a remote cloud server for real-time processing, visualization, and diagnosis. In this paper, we enhance such a health monitoring system by exploiting the concept of fog computing at smart gateways providing advanced techniques and services such as embedded data mining, distributed storage, and notification service at the edge of network. Particularly, we choose Electrocardiogram (ECG) feature extraction as the case study as it plays an important role in diagnosis of many cardiac diseases. ECG signals are analyzed in smart gateways with features extracted including heart rate, P wave and T wave via a flexible template based on a lightweight wavelet transform mechanism. Our experimental results reveal that fog computing helps achieving more than 90% bandwidth efficiency and offering low-latency real time response at the edge of the network.**

***Keywords*—Internet of Things, Healthcare, Smart Gateway, Sensor Network, Heart Rate, ECG feature extraction, Fog Computing.**

I. Introduction

Internet of Things (IoT) can be considered as a dynamic global network infrastructure where objects with unique identities are interconnected for enabling advanced services. Wireless Body Area Networks (WBAN), which is one of the fundamental technologies in healthcare IoT, is ubiquitous used for acquiring different vital signs such as Electrocardiogram (ECG), Electromyography (EMG), body temperature, blood pressure in a real-time, unobtrusive and efficient way. WBAN can be established from a multitude of implantable or wearable sensor nodes to sense and transmit the data over a wireless network via communication protocols such as Wi-Fi or IEEE.802.15.4 to end users for visualization and diagnosis. Low-cost and power efficient WBAN-based systems play important roles in various areas of healthcare environments from monitoring, clinical care to chronic disease management and disease prevention. For example, these systems are used for monitoring and treating many cardiovascular diseases.

In many health monitoring systems, remote cloud servers have been used for storing and processing big data collected from a large number of sensor nodes due to cloud computing benefits such as low-cost services, capability of large data storage volume and superfluous maintenance cost. However, there exists many challenges in these systems regarding latency-sensitive issues, location awareness and large data transmission. Undoubtedly, the more data is transmitted over a network, the higher possibility of error occurs because bit error, data transmission latency and packet dropping possibility are proportional with the volume of transmitted data. Especially, in case of emergency, a single error in analyzed data causes inaccurate treatment decisions and affect crucially to a human life. Therefore, there is a need of reducing the number of transmitted data over a network, on the other hand, quality of service (QoS) is still guaranteed.

A proper solution for fulfilling the requirement is the provision of an extra layer in between a conventional gateway and a remote cloud server. The extra layer denoted as fog layer helps diminishing the volume of transmitted data for guaranteeing QoS, and saving network bandwidth by preprocessing data. In addition, fog computing offers advanced services at the edge of the network and reduces the burden of cloud [1]. Fog computing not only brings the cloud computing paradigm to the edge of the network but also addresses unsupported or unfit fundamentals in the cloud computing paradigm. For instance, edge location, low latency, location awareness, geographical distribution, interoperability, and support for on-line analytics are some of fundamental characteristics of fog computing. Due to these characteristics, fog computing can be suitable for supporting human health monitoring WBAN-based systems which have features of low energy, low bandwidth, low processing power and include hardware constrained nodes. To this end, a combination of WBAN-based system, cloud computing and fog computing can be a sustainable solution for challenges in the current IoT healthcare systems.

In this paper, we present an efficient IoT-enabled healthcare system architecture which benefits from the concept of fog computing. Using this architecture, we demonstrate the effectiveness of fog computing in IoT-based healthcare systems in terms of bandwidth utilization, QoS assurance, and emergency notification. In addition, we utilize ECG feature extraction at the edge of the network in our implementation as a case study. In summary, the key contributions of this work are as follows:

- Low-latency data processing and low bandwidth utilization at smart gateways (i.e., edge of the network)

978-1-5090-0154-5/15 $31.00 © 2015 IEEE
DOI 10.1109/CIT/IUCC/DASC/PICOM.2015.51

- Support various sensor node types, communication protocols, operating systems at smart gateways (i.e., heterogeneity, and interoperability)
- Real-time rapid interaction at smart gateways in case of emergency (i.e., real-time push notification)
- Real-time and on-line analytic at the fog layer even in case of poor connection with the cloud (i.e., geographical distribution, location awareness, graphical user interface)

The rest of the paper is organized as follows: In Section II, the related work and the motivation of the paper are discussed. The concept of an e-health IoT-based system and a smart e-health gateway are presented in section III. Section IV presents an implementation of the smart IoT-based gateway. Demonstrations and experimental results are provided in section V. Finally, section VI concludes the paper and present discussion for further researching.

II. Related Work and Motivation

There have been many efforts in designing smart gateways for healthcare applications. For instance, Chen *et al*. [2] introduce a smart gateway for health care system using wireless sensor network. The proposed gateway acting as a bridge between wireless sensor network and public communication networks has a data decision system, a lightweight database and an ability to notify caregivers in case of emergency. In addition, the gateway provides a way of diminishing a remote server's burden by applying the request and response message method.

Mohapatra *et al*. discuss a hybrid framework for remote patient monitoring via a sensor cloud [3]. Advantages of using a sensor cloud for patient health status monitoring is demonstrated in their proposed system. In [4], authors present a cloud computing solution for patient's data collection in health care institutions. The proposed system uses sensors attached to medical equipments to collect patient data and sends the data to cloud for providing ubiquitous access. Yang *et al*. present a personal health monitoring gateway based on smart phone [5]. The proposed gateway uses a Bluetooth interface to upload gathered data to remote servers. In a work presented in [6], the sensor network system uses sensing servers as gateways in the system. However, the proposal is expensive, poor scalable, and inefficient for the number of IoT-based applications. In [7], authors discuss a prototype of a smart IPv6 Low power personal area network (6LoWPAN) border router based on Hidden Markov Model for making decisions of health states. In [8], authors propose a mobile gateway for ubiquitous health care system using ZigBee and Bluetooth. The gateway tenders various services such as alarms and analysis of medical data. However, the proposed gateway is inefficient in term of power consumption and it cannot be considered as a novel gateway for several IoT-based applications. Zhong *et al*. present a gateway based on a mobile phone for connecting sensor network nodes and devices supporting CDMA or Bluetooth [9]. In another work [10], the proposed architecture acquire data via multiple personal health devices via USB, ZigBee or Bluetooth.

The topic of body area sensor network systems has attracted a lot of research efforts in recent years. Especially, the number of works in the healthcare domain has dramatically increased to explore several undiscovered aspects or overcome existing drawbacks in health monitoring systems. With the purpose of novel health monitoring systems provision, some of works try to expand current systems with more services and functions while others attempt to propose new platforms or methods. However, as described in the aforementioned examples, a large number of systems focus on ZigBee whereas it is a challenge to assure the quality of service in ZigBee when monitoring streaming bio-signals such as ECG, EMG and Electroencephalography (EEG) as the maximum data rate of ZigBee is 250 kbps. Inversely, Bluetooth technology can overcome problems of low data rate in ZigBee and other short range communication protocols. Nevertheless, it might be more difficult to design a gateway based on Bluetooth technology for supporting mobility and acquiring data from multiple targets [8]. The discussed systems basically dispense conventional gateways for collecting data from nodes and sending these data to remote servers. More precisely, none of these works have considered to fully take advantage of the fog computing paradigm and bring intelligence to the gateways.

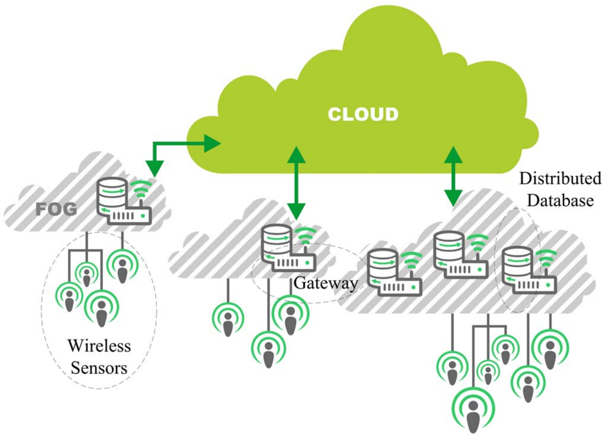


Fig. 1. Fog computing concept

The main motivation of this paper is to enhance a health monitoring IoT-based systems used in diversified environments such as home and hospital by providing a smart gateway [11] with the fog layer having quantities of advanced services. The role of fog computing is illustrated in Fig. 1. It is a middle computing layer between sensors and cloud computing which consists of gateways and distributed databases. Fog computing concept is the extension of the cloud computing paradigm in a view of pointing applications and domains which the paradigm of the cloud does not fully support. Some of them consist of i) real-time applications including video conference applications, and Voice over Internet Protocol (VoIP) require very low latency because substantial delays may cause diminishing QoS. ii) large data applications collecting a ton of data from numerous sensors and transmitting the data over networks stand in need of very high bandwidth [12]. iii) monitoring applications, which operate unremittingly, require uninterrupted data in case of losing connectivity between monitoring systems and the cloud [13]. Characteristics of these aforementioned applications are similar to characteristics of real-time health monitoring systems. In such systems, a large amount data are acquired from a multitude of bio and environmental sensors. Then, the large data is transmitted for networks for being ultimately remote monitoring by end users such as care-givers or medical doctors. Therefore, fog computing fits these systems marvellously. It is acknowledgeable that instead of replacing or lessening the role of the cloud in IoT applications, fog computing is completely cooperated and compatible with the cloud to enhance existing IoT applications from the aspects of location awareness, low latency, scalability, real-time interactions, heterogeneity and interoperability [14].

Electrooculography (EOG), EMG, EEG, ECG are important signals gathered by monitoring systems for diagnosing human health in both daily activities and medical abnormalities. However, in this paper, only a case study of ECG is addressed to present the concept of fog computing and its benefits in health monitoring IoT applications. In order to efficiently utilize network bandwidth between a health monitoring system's gateway and a cloud server, we present a flexible template with light-weight algorithms for extracting heart rate, P and T waves which can be presented at an end-user's browser. The extracted heart rate is presented at a secure graphical user interface and fed to our warning service for real-time notification in cases of emergency. In addition, a secure graphical user interface at the smart gateway is used for representing result of P, T waves after preprocessing and raw ECG data in graphical waveforms for real-time visualization and analytics. Furthermore, we introduce a method of providing gateway interoperability in the interest of supporting Ethernet, Wi-Fi, Bluetooth, ZigBee and 6LoWPAN.

III. System and Gateway Architecture

A health monitoring system often comprises of several devices to collect bio-data from a human body and transmit the data to a processing or visualization device via wires and cables for monitoring and diagnosis. Several drawbacks exist in such system such as unsupported mobility and remote monitoring which cause many inconveniences for both patients and doctors. For example, health of diabetes and cardiovaculus needs to be continuously monitored 24/7. When the conventional system is applied for this procedure, these patients must carry many devices and cables during long period of monitoring hours and incorrect data is acquired due to movement activities of these patients. A health monitoring system based on WBAN can handle these drawbacks effectively through its characteristics of mobility and wireless transmission. Although there are different types of health monitoring WBAN-based systems designed for particular bio-data and diseases, they have three main parts including sensor nodes, a gateway and a back-end part. The detailed description of these components is presented in the following.

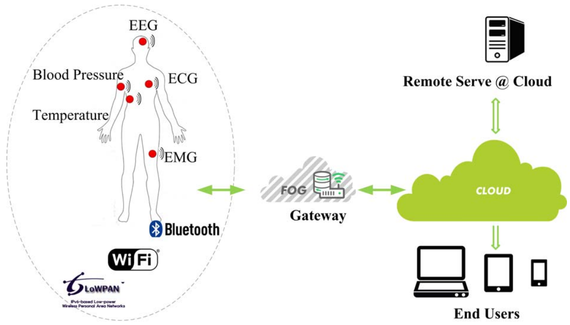


Fig. 2. The IoT-based health monitoring system architecture

1) Medical sensor node: is a composition of several physical devices including implantable or wearable sensors which are integrated to a tiny wireless module for gathering contextual and medical data such as temperature, location, humidity, SpO2, ECG, EMG, and EEG and then transmitting the data to a gateway via a specific communication protocol such as Wi-Fi, Bluetooth, ZigBee or 6LoWPAN. Depending on treatment methods for different diseases, particular bio-data and the contextual data are intentionally focused in various health monitoring WBAN-based systems.

2) Gateway: plays an important role in such a WBAN-based system because it connects a profusion of sensor nodes with a remote cloud server. When a gateway does not function properly or bottle-neck occurs at a gateway, the whole system will be affected. As a result, real-time bio-data cannot be appropriately accessed at a cloud server.

3) Back-end part: consists of a cloud server and back-services which can be different from systems to systems depending on technologies and services offered. A cloud server is used for storing, processing and broadcasting data while back-end services are responsible for representing real-time data for analysis and visualization.

The current human health monitoring WBAN-based systems containing several existing challenges can be augmented for shortening the gap towards the novelty level. In order to achieve this target, smart gateways with fog computing substituting for gateways in conventional IoT-based health monitoring systems are introduced. Our proposed IoT-based health monitoring system architecture is shown in Fig. 2.

A. Smart gateway architecture

In addition to perform functionalities of conventional gateways, the smart gateway should have abilities to offer a high level of advanced services in the fog computing platform. The smart gateway architecture including physical and operational structures is elaborately designed and described in the following.

The physical gateway structure shown in Fig. 3 includes several embedded devices which are an embedded router and one or several sink nodes. The embedded router supports some communication protocols such as Bluetooth Low Power (BLE), Wi-Fi and Ethernet but it cannot offer any possibility to connect with sensor nodes via low power wireless communication protocols. With the purpose of handling this challenge and achieving interoperability, sink nodes, which offers 6LoWPAN and other low power communication protocols are integrated into the gateway. The number of sink nodes does not limit within one or several ones but it can reach to dozens or even a hundred depending on specifications of the USB extension, SPI, I2C connections and the Internet service provider. For example, the maximum tier for USB extension is tier 7 and the gogo6 service [15] provider allows a maximum of 256 sink nodes per a network.

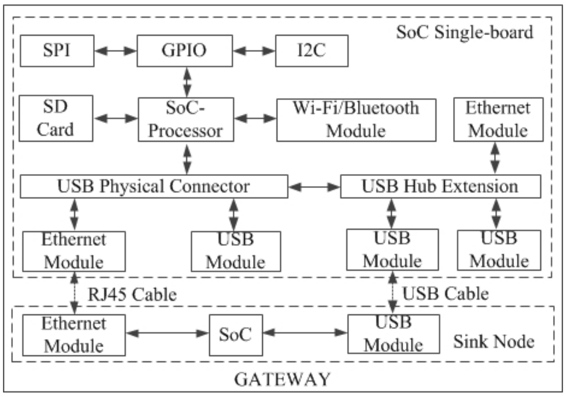


Fig. 3. Physical gateway structure

The embedded router contains different components such as processor, extendable SD card, USB connection, Ethernet, GPIO with SPI and I2C support, and a wireless connection module. Therefore, it can be able to run an operating system and process some heavy computing, which are clarified via the operational structure shown in Fig. 4. It presents the gateway operational structure comprising of a fog computing service layer, an embedded operating system and a hardware layer.

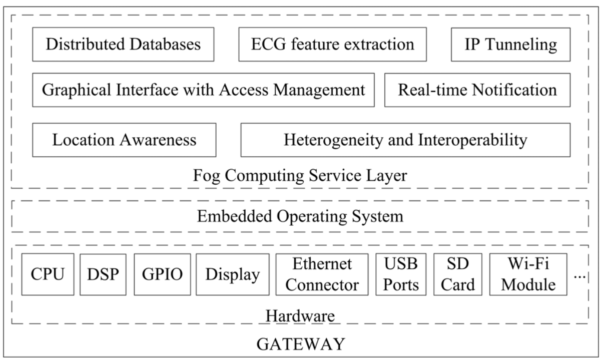


Fig. 4. Operational gateway structure

1) Hardware layer: A hardware layer operates as middleware between an embedded operating system and all physical components. Middleware receives the operating systems instructions for allocating physical hardware to gateway services. The hardware layer allows only one service to have a permission to access the component at a time. When two or more services need to access the same component, the rest must wait for its turn to acquire permission.

2) Embedded operating system: There are two types of embedded operating system used in a smart gateway including one type in the embedded router and another type in sink nodes. The embedded operating system in a sink node must be light-weight, compact, and does not require specification of powerful hardware. Inversely, the operating system in the embedded gateway does not need to be very compact because it has to provide various tools, and management mechanisms to ensure that real-time data is transmitted into a remote cloud server and all fog computing service function properly.

3) Fog computing service layer: The fog computing service layer plays the most essential role in a gateway because it contains a profusion of advanced services for not only embodying functionalities of a conventional gateway but also performing characteristics of fog computing. Details of these services are described in the following.

Distributed databases contain a static look-up storage, a general purpose storage, and a synchronized storage. The static look-up storage contains static and essential data required for several services and algorithms (e.g., security with username and password, references for data accessing and access management); therefore, the static database is kept intact for all cases except for the case of system administrators. The general purpose storage storing high data rate incoming data is used for both the fog computing service and graphical user interface. The general purpose database size can vary depending on specific applications. The synchronized storage is as an inventory of temporarily environmental data and bio-data which are sent from sensor nodes with a low data rate such as temperature, and humidity. Furthermore, it has responsibility for updating data at a remote server.

ECG feature extraction: ECG feature extraction has been processed and carried in many researches due to its important role in many applications, especially in healthcare domain. For example, in addition to help doctors monitoring and giving treatments to many diseases related to cardiovascular more efficiently, it helps to detect some abnormalities of the heart. Heart rate is one of the most concerned features extracted from ECG because it provides an overview of the heart which is necessary for emergency services and diagnosing many diseases. Furthermore, via heart rate information, some instant methods might be applied to keep the heart operating normally. For instance, a workout person can reduce the workout intensity level when heart rate is very high because heart rate is proportional with the workout intensity.

In order to extract the ECG feature at a smart gateway, we design a flexible template shown in Fig. 5. The template contains three main parts including ECG preprocessing, wavelet transformation, and ECG feature extraction. The ECG preprocessing part contains some filters such as notch filter or moving average filter in pursuance of movement artifact removal. The wavelet transformation part encompasses fast computation methods and wavelet algorithms. Finally, the ECG extraction part is used for applying various algorithms for extracting different data such as P-R interval, Q-T interval, S-T segment, QRS area and QRS energy. In this paper, we design a light-weight algorithm which is suitable with requirements of fast computation and low hardware resources consumption with the purpose of extracting P wave, T wave and heart rate from original ECG signals for keeping track of the heart's activities and verifying the smart gateway's warning service. The algorithm includes specifying a proper threshold value via scanning the whole ECG input signals, extracting the number of pulses via the help of the threshold value, and calculating heart rate based on the formula given as

$$\text{Heart rate} = \frac{60}{\text{R-R interval}} \quad [16]$$

The algorithm is designed in such a way that some parts of the algorithm can be reused for extracting other data besides R-R interval, P wave, T wave and heart rate. For instance, steps of scanning ECG signals and the method of calculating a threshold value are two reusable parts of the algorithm although parameters of these methods may vary depending on which parts of ECG signals, medical doctors want to diagnose.

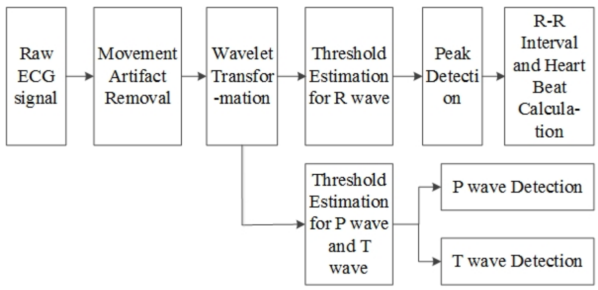


Fig. 5. The ECG feature extraction template

Graphical user interface with access management: The interface can be used by end users such as caregivers, medical doctors and system administrators for visualizing ECG and other bio-data at the smart gateway. In order to have an access to the gateway's graphical user interface, a user must login with his/her username and password which are created via the access management service. The service has a mechanism which categorizes end users into several groups depending on their responsibilities. Data available for each group will be different. For example, some sensitive data can be seen by medical doctors while the data is unavailable to patients or their relatives. The main reason for building this mechanism is to avoid some unexpected situations related to patients reaction during the disease treatment process. For example, when a patient knows his/her current status health, the patient may overreacts or has some strong emotions which may cause bad effects for disease treatment. With the intention of avoiding both the brute-force method and flooding the system's database, a user only has three chances during 10 minutes to login into the system. After three failed times, a user needs to wait for extra 10 minutes for the next login. Due to the limitation of the gateway's resources, all recorded data are stored in the database within 5 minutes. After that, the database will be purged for storing new incoming data.

Real-time notification: The notification service is a real-time notification to inform about abnormal situations. The gateway sends some real-time signals to a remote cloud which sends notification messages to an end users device when one of three following cases occurs. Firstly, the smart gateway does not receive any data from specific sensor nodes during a particular period of time. Secondly, an internal temperature of an embedded gateway or a sink node is higher than a predefined threshold. Finally, heart rate is not in the range of normal heart rate which is defined by American Heart Association. Notification messages are categorized into three groups having level 1, 2 and 3, respectively. Depending on particular situations, the push notification generates different messages with corresponding levels. For example, when the temperature of the embedded gateway is 50, 60, and 70 Celsius degrees, the push notification service sends warning messages level 1, 2, and 3 to system administrators, respectively. The higher warning level, the more urgent and critical the situation is. Similarly, this mechanism is applied for other cases.

Location awareness: Location awareness, which is used for providing information of geographical location of a device, is supported in our system via a method of tracking a physical MAC address of the systems gateway. Typically, a single gateway is setup for serving one or server rooms in the same corridor with the purpose of diminishing the complexity of the location awareness service and enhancing QoS. Through the location awareness service, the system administrators or caregivers can properly detect a patient location without any effort, which is necessary in case of emergency. In addition, the service is in conjunction with a congestion recognition mechanism which is used for monitoring an amount of incoming data of a gateway in an instant time in the direction of avoiding possibilities of exceeding a gateways bandwidth limit. When the amount of incoming data is too large, the service sends an instant signal to the real-time notification service which then notifies an administrator with real-time messages.

Heterogeneity and Interoperability: Cloud computing physical infrastructure can be described as a combination of homogeneous physical resources which are deployed and managed in a centralized manner. In contrast, fog computing physical infrastructure consist of heterogeneous resources managed in a distributed manner. Therefore, fog computing has possibilities of supporting interoperability which is an ability of serving different devices in terms of various manufactures, models, different operating systems and inconsistent communication protocols. The gateway architecture is designed in a way of supporting many operating systems and several versions of an operating system type. For instance, different versions of Linux can be installed as an operating system of a gateway. Moreover, sensor nodes produced from various producers including Texas Instrument, Zigduino, Arago Systems, and Zolertia operating under inconsistent communication protocol standards such as Wi-Fi, 6LoWPAN, and Bluetooth can connect with the gateway for establishing a complete heterogeneous wireless sensor network.

IP tunneling: IP tunneling constructed by a combination of router advertisement daemon and gogoc services is a gateway service for connecting 6LoWPAN with IPv4/IPv6. The router advertisement daemon service has two main functions including router advertisement provision and router solicitation listening while the gogoc service is responsible for querying a tunneling between a gateway and a tunnel server.

IV. Smart Gateway Implementation

Our smart gateway is built by combining various embedded hardware including Pandaboard [17] and a 6LoWPAN sink node, as shown in Fig. 6. Pandaboard is used in our implementation because it is constructed from the OMAP 4 platform which is a power efficiency and high performance system-on-chip. The processor of the OMAP 4 platform comprises of dual-core ARM Cortex-A9 MPCore in which the speed of each core is more than 1 GHz. The processor is suitable with our application due to its characteristics such as power efficiency, symmetric multiprocessing, hardware accelerator provision (i.e., a programmable digital signal processor). Furthermore, the platform supports non-volatile and volatile memories via high performance and comprehensive controllers. Therefore, external memory such as 64 GB SD card can be appended into Pandaboard for serving as a system storage. Additionally, Pandaboard is capable of dealing with different communication protocols such as Ethernet, Wi-Fi, and Bluetooth via various interfaces and pre-integrated hardware modules.

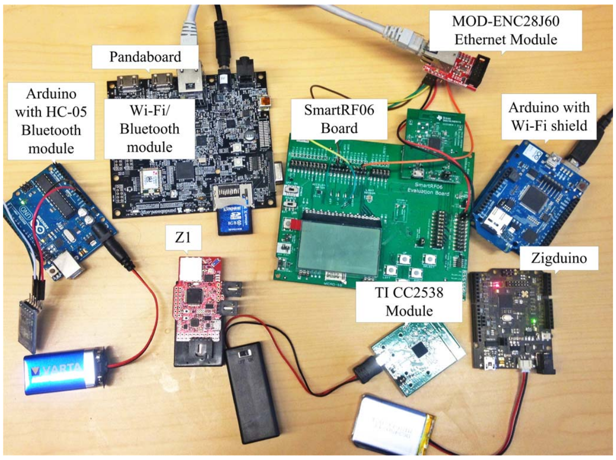


Fig. 6. Demonstration of our smart IoT-based healthcare gateway

Moreover, the OMAP4 platform supports different operating systems and applications such as Symbian, Linux (LiMo, Android, Ubuntu), and Windows Mobile. Convincingly, Ubuntu is selected as the gateway's operating system in our implementation due to its benefits including open-source operating system, various software support and high performance. In addition, Linaro operating system, which is another embedded Linux operating system, is installed in another gateway for proving the gateway's interoperability in terms of operating system support. Furthermore, we experiment heterogeneity and interoperability in our gateway by applying various hardware including Arduino with Wi-Fi shield, TI CC2538, Zigduino, Z1, and Arduino with HC-05 Bluetooth module as wireless sensor network nodes for collecting and sending data to the gateway via different communication protocols such as Wi-Fi, 6LoWPAN, ZigBee and Bluetooth.

MySQL comprising of sub-databases and tables is used as a main inventory for storing configurations of services and mechanisms, e-health and environmental data. The MySQL ensures that the data in a remote server is always updated regularly via the support of the third party tool (i.e., xSQL Lite database synchronization tool).

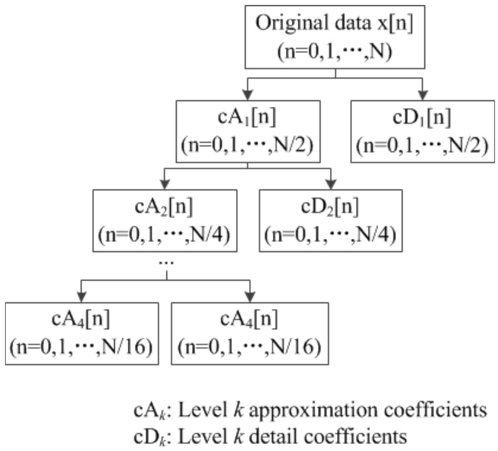


Fig. 7. Four-level Discrete Wavelet Decomposition

In a direction of implementing fog computing in the smart e-health gateway, many services and tools including gogoc, tunslip, router advertisement, socket receiver, push notification, graphical interface with access management are applied or constructed.

With a view of gateway workload reduction, a push notification service is completely implemented in a cloud server by using WebSocket. In the gateway, only a push notification mechanism is created. When the push notification service in cloud receives a desired signal from the gateway, it sends a notification to end users or system administrators.

With the interest of ECG heart rate, P wave, T wave extraction, we applied our proposed template. As mentioned above, the template has several phases including movement artifact removal, wavelet transform and ECG feature extraction. Initially, the baseline drift elimination step is implemented with a 20-points moving average filter. Then Discrete Wavelet Decomposition (DWT) with db 4 wavelet, which is in the family of Daubechies, is used as the mother wavelet. With the purpose of achieving proper results, a 4-level DWT shown in

Fig. 7, is applied. Finally, R-R interval value, P wave, T wave and heart rate can be extracted.

Graphical user interface is built by MySQL database, PHP used as server-side scripting and JavaScript (JQuery) for HTML content generation such as plotting charts. All mechanisms for access management, verifying username, password and checking the number of login times are built in PHP with the assistance of MySQL database.

A 6LoWPAN sink node consists of three components such as the Olimex Ethernet module [18], the TI Smart RF06 board, and the CC2538 module [19], shown in Fig. 6. The Ethernet module is used for data communicating between the 6LoWPAN sink node and Pandaboard. The CC2538 module takes responsibility for receiving data from sensor nodes through 6LoWPAN while the TI Smart RF06 board is used for interface provision and debugging.

V. Experimental Results and Demonstration

According to MIT-BIT Arrhythmia database [20], ECG signals were recorded from 47 subjects by two-channel ambulatory ECG system and the recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10 mV range. In addition, the database recorded a large number of statistics and records related to ECG data such as a total number of normal heart beats and other types heat beats during 30 minutes recording time of each person. Due to these statistics, ECG data in the MIT-BIT Arrhythmia database is a suitable candidate for our experiments. Initially, ECG data was stored in the gateway storage and then processed with the fog computing service. Finally, ECG features (e.g., heart rate, P wave and T wave) were extracted. In order to have a closer view to our proposed template functionality, we applied the file "101" from MIT-BIT Arrhythmia database into the template. The result is shown in Fig. 8.

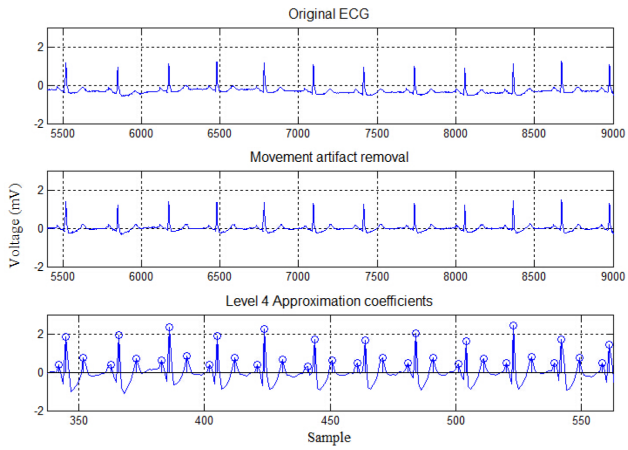


Fig. 8. ECG processing implementation

With the intention of presenting benefits of fog computing towards a healthcare IoT system, latency of transmitting various raw ECG data from the gateway to a remote cloud is compared with a total latency of processing the fog computing service and transmitting preprocessed ECG data. The comparison results shown in Table I are achieved via Wi-Fi whose information such as network condition, data rate, frequency and link quality are provided in Table II. In case of fog computing, a volume of data transmitted over Wi-Fi is small on the grounds that processed data including heart rate, P wave, and T wave is transmitted.

TABLE I
Data rate and latency comparison

Data rate (Mb/s)	Raw data		Fog computing			Improvement	
	Data size(B)	Latency (ms)	Data size(B)	Processing(ms)	Transmitting(ms)	Data size reduction(%)	Latency reduction(%)
18		106.6			⩽6.6		⩾3.5
12	240000	152.2	⩽15840	≈ 96.3	⩽9.5	⩾93	⩾30.5
9		213.3			⩽13.5		⩾48.5

TABLE II
Information of Wi-Fi network

Network condition	Data rate	Frequency	Link quality
Not busy	18Mbit/s	2.437GHz	53/70
Busy	12Mbit/s	2.412GHz	51/70
Busiest	9Mbit/s	2.402GHz	50/70

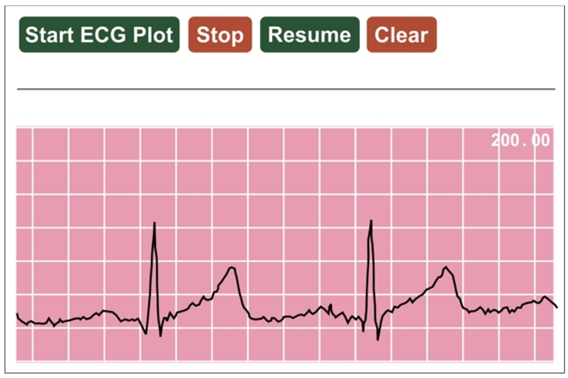


Fig. 9. ECG waveforms

Throughout these reported results, the fog computing service is efficient in terms of transmission latency minimization. In addition, the fog computing service helps to reduce the number of data transmitted over a network. As a consequence, a volume of transmitted data reduce dramatically, which leads to efficient utilization of network bandwidth.

As mentioned above, users can log in to our gateway graphical user interface by their usernames and passwords for visualizing and diagnosing real-time ECG data shown in Fig. 9, P wave, T wave and heart rate. In this experiment, instead of using ECG data from the MIT-BIT Arrhythmia database, ECG data is collected from nodes and sent to the smart gateway via Wi-Fi. In addition, the gateway user interface is designed for achieving user friendliness with several functional buttons for manipulating ECG waveforms.

VI. Conclusion and Discussion

In this paper, we present fog computing at a gateway for augmenting health monitoring systems. We have implemented fog computing services including interoperability, distributed database, real-time notification mechanism, location awareness and graphical user interface with access management. In addition, we introduce a flexible, light-weight template for ECG feature (e.g. heart rate, P wave, and T wave) extraction. The demonstration and results show the achievements provided by the smart gateway. The template based on wavelet transform can be used for extracting various ECG features by detecting different parts of ECG waveforms (e.g. PR interval, and QT interval), or extracting EMG, EEG features.

References

[1] J. Burt, "Fog computing aims to reduce processing burden of cloud systems," http://www.eweek.com/networking/fog-computing-aims-to-reduce-processing-burden-of-cloud-systems.html, 2010.

[2] Y. Chen, W. Shen, H. Huo, and Y. Xu, "A smart gateway for health care system using wireless sensor network," in *2010 Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM)*, July 2010, pp. 545–550.

[3] S. Mohapatra and K. S. Rekha, "Sensor-cloud: a hybrid framework for remote patient monitoring," *Int. J. Comput. Appl*, vol. 55, no. 2, pp. 1–11, 2012.

[4] C. O. Rolim, F. L. Koch, C. B. Westphall, J. Werner, A. Fracalossi, and G. S. Salvador, "A cloud computing solution for patient's data collection in health care institutions," in *ETELEMED'10. Second International Conference on eHealth, Telemedicine, and Social Medicine, 2010.* IEEE, 2010, pp. 95–99.

[5] S. Yang and M. Gerla, "Personal gateway in mobile health monitoring," in *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, March 2011, pp. 636–641.

[6] J. W. Yoon, Y. K. Ku, C. S. Nam, and D. R. Shin, "Sensor network middleware for distributed and heterogeneous environments," in *NISS'09. International Conference on New Trends in Information and Service Science, 2009.* IEEE, 2009, pp. 979–982.

[7] W. Shen, Y. Xu, D. Xie, T. Zhang, and A. Johansson, "Smart border routers for ehealthcare wireless sensor networks," in *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, Sept 2011, pp. 1–4.

[8] T. H. Laine, C. Lee, and H. Suk, "Mobile gateway for ubiquitous health care system using zigbee and bluetooth," in *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. IEEE, 2014, pp. 139–145.

[9] *A Phone-centered body sensor network platform: Cost, energy efficiency & user interface*. Institute of Electrical and Electronics Engineers, Inc., April 2006. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=103003

[10] K. Park and J. Pak, "An integrated gateway for various phds in u-healthcare environments," *BioMed Research International*, vol. 2012, 2012.

[11] A. M. Rahmani, N. K. Thanigaivelan, T. N. Gia, J. Granados, B. Negash, P. Liljeberg, and H. Tenhunen, "Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems," in *Annual IEEE Consumer Communications and Networking Conference*. IEEE, 2015, pp. 826–834.

[12] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 169–186.

[13] N. Bessis and C. Dobre, *Big data and internet of things: a roadmap for smart environments*. Springer, 2014.

[14] 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*. ACM, 2012, pp. 13–16.

[15] J. Whitby, "Systemd for gogoc," http://www.gogo6.com/forum/topics/systemd-for-gogoc, jun 2012.

[16] K. Goh, J. Lavanya, Y. Kim, E. Tan, and C. Soh, "A pda-based ecg beat detector for home cardiac care," in *IEEE-EMBS 2005. 27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005.* IEEE, 2006, pp. 375–378.

[17] "Resources, pandaboard.org." http://pandaboard.org/content/resources/references.

[18] "Olimex. mod-enc28j60 development board, users manual." https://www.olimex.com/Products/Modules/Ethernet-/MOD-ENC28J60/resources/MOD-ENC28J60.pdf.

[19] "Smartrf06 evaluation board users guide (rev.a)." http://www.ti.com/lit/pdf/swru321.

[20] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 20, no. 3, pp. 45–50, 2001.

Paper II

Low-cost Fog-assisted Health-care IoT System with Energy-Efficient Sensor Nodes

Tuan Nguyen Gia, Mingzhe Jiang, Victor Kathan Sarker, Amir M Rahmani, Tomi Westerlund, Pasi Liljeberg, Hannu Tenhunen

Original published in IEEE Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International Conference, pages 1765-1770, Valencia, Spain

Low-cost Fog-assisted Health-care IoT System with Energy-efficient Sensor Nodes

Tuan Nguyen Gia[1], Mingzhe Jiang[1], Victor Kathan Sarker[1], Amir M. Rahmani[2,3], Tomi Westerlund[1], Pasi Liljeberg[1], and Hannu Tenhunen[1]
[1]Department of Future Technologies, University of Turku, Turku, Finland
[2]Department of Computer Science, University of California Irvine, USA
[3] Institute of Computer Technology, TU Wien, Austria
Email: {tunggi, mizhji, vikasar, tovewe, pakrli, hatenhu}@utu.fi, amirr1@uci.edu

***Abstract*—A better lifestyle starts with a healthy heart. Unfortunately, millions of people around the world are either directly affected by heart diseases such as coronary artery disease and heart muscle disease (Cardiomyopathy), or are indirectly having heart-related problems like heart attack and/or heart rate irregularity. Monitoring and analyzing these heart conditions in some cases could save a life if proper actions are taken accordingly. A widely used method to monitor these heart conditions is to use ECG or electrocardiography. However, devices used for ECG are costly, energy inefficient, bulky, and mostly limited to the ambulatory environment. With the advancement and higher affordability of Internet of Things (IoT), it is possible to establish better health-care by providing real-time monitoring and analysis of ECG. In this paper, we present a low-cost health monitoring system that provides continuous remote monitoring of ECG together with automatic analysis and notification. The system consists of energy-efficient sensor nodes and a fog layer altogether taking advantage of IoT. The sensor nodes collect and wirelessly transmit ECG, respiration rate, and body temperature to a smart gateway which can be accessed by appropriate care-givers. In addition, the system can represent the collected data in useful ways, perform automatic decision making and provide many advanced services such as real-time notifications for immediate attention.**

***Keywords*—** Internet-of-Things, low-cost, wearable, fog computing, health monitoring, energy-efficient, low energy.

I. INTRODUCTION

More than 422 million people in the world are related to diabetes and cardiovascular diseases which can directly or indirectly cause serious consequences such as congestive heart failure, other irregular heart rhythms, heart attack, stroke, and kidney failure [1, 2]. Delay or incorrectness in the treatment of these abnormalities can endanger a patient. Therefore, it is necessary to monitor these patients' health and notify abnormal situations to doctors in real-time. Continuous health monitoring systems can be considered as a solution for this issue. However, they are often high-cost with several limitations such as non-support advanced services including remote monitoring, or real-time notification.

Internet of Things (IoT) can be described as a convergent network infrastructure where physical and virtual objects are interconnected together [3]. With the involvement of many advanced technologies such as wireless sensor network, wireless body area network, wearable and implanted sensor, IoT shows its capabilities of solving existing problems or difficulties in health-care monitoring systems. It can help to improve the quality of service i.e. offering remote monitoring, push notification whilst reducing health-care costs.

In order to monitor patients' health, wireless sensor devices (i.e. implanted or wearable sensors) acquiring bio-signals from a human body and wirelessly sending the signals to a gateway are popularly applied. These sensor devices are small and resource constraint (i.e. limited power supply capacity). Accordingly, it is important to achieve some levels of energy efficiency in sensor devices. However, it is a challenge to reduce power consumption of sensor devices dramatically while maintaining the high quality of signals. Also, when high resolution signals are needed, it costs high power consumption for data acquisition and wireless transmission.

In conventional IoT-based systems, primary tasks of gateways are data receiving and transmitting. To improve the quality of health-care service, gateways can be upgraded by the assistance of Fog computing which can be described as a virtual platform extending the Cloud computing paradigm to the edge of the network and reducing burdens of Cloud [4–6]. For example, diversified advanced services such as edge location, low latency, geographical distribution, and mobility support can be provided with the assistance of Fog.

In order to reduce health-care costs and improve the quality of health-care service, we propose a novel low-cost remote health monitoring IoT-based system with Fog computing and energy-efficient wearable sensor devices. The wearable device, which is small and low-cost, is able to collect and wirelessly transmit the large number of high resolution signals (i.e. ECG and respiration rate) to a smart gateway. Furthermore, the wearable device's power consumption is dramatically reduced by a combination of hardware design and software-based techniques. In the system, smart gateways are integrated with the Fog layer providing a large number of advanced services such as data analysis and data processing at gateways, decision making, notifications and local data storage. Real-time decision making is regularly carried out for checking abnormal situations. When abnormality such as too low or high heart rate is detected, it sends real-time notifications to the patient and his/her doctor. Accordingly, the early stage of deterioration can be timely detected. Last but not least, doctors can remotely monitor patients' health represented in

978-1-5090-4372-9/17/$31.00

text and graphical forms in real-time via an end-terminal i.e. a smart-phone or a computer's browser. To summarize, the key contributions of this work are as follows:

- A design of energy-efficient, low-cost and wearable sensor device for collecting and wirelessly transmitting ECG, respiration rate, humidity, body temperature and room temperature
- A complete remote health monitoring system based on IoT and customized ultra low power 2.4GHz radio frequency protocol (nRF)
- Fog services for representing bio-signals in graphical waveforms, performing decision making, categorization, real-time notifications, and channel managing

This paper is organized as follows. Section II discusses related works and motivations. Section III describes the system architecture in details. Section IV presents a design of a low-cost and energy-efficient wearable sensor device. Section V presents the system's gateway architecture and a back-end system. Section VI shows an implementation of the entire health monitoring system. Section VII shows the experimental results. Section VIII concludes the work.

II. Related Work and Motivation

There have been a lot of efforts in developing remote health monitoring IoT-based systems. Gia et al. [7] propose a continuous health monitoring system based on customized 6LoWPAN. The system enables remote and real-time ECG monitoring via a reliable network.

Jiang et al. [8] propose an IoT-based system for remote facial expression monitoring. The system's sensor nodes acquire Electromyography (EMG) and transmit the data to a gateway via Wi-Fi. The bio-signals are processed and classified in Cloud with the assistance of LabVIEW.

Gomez et al. [9] introduce a patient monitoring system based on IoT for monitoring health status and recommending workouts to patients with chronic diseases. The system acquires not only bio-signals (i.e. ECG) but also contextual data (i.e. time and location). Doctor and patient can access collected data via an Android app.

Nemati et al. [10] propose a wireless wearable ECG sensor for long-term applications. The small-size sensor node can be conveniently deployed in T-shirt or undergarment for collecting and transmitting ECG data wirelessly via an ANT protocol. A patient carrying the sensor can perform his or her daily activities without any disturbance.

Fanucci et al. [11] present an integrated information and communication technology system for monitoring patients at home. The system collects ECG, SpO2, blood pressure, and a patient's weight via biomedical sensors. The collected data is transmitted to the hospital information system for remote monitoring. The system helps to reduce the number of subsequence hospitalization via its capability of supporting early detection of the alterations in vital signs.

In [12], authors present a smart health-care system using Internet of Things. The system is able to monitor different signals such as glucose level, ECG, blood pressure, body temperature, SpO2 and transmit the collected data wirelessly to Raspberry Pie via Zigbee. End-users such as doctors and care-givers can monitor the data via a mobile application.

Other systems based on Bluetooth Low Energy and IoT [13, 14] are capable of acquiring and transmitting ECG wirelessly with low power consumption. By applying these systems at home and hospital, doctors can monitor ECG and heart rate of patients in real-time.

Although these systems can improve the quality of health-care service via their advancement (i.e. remote and real-time monitoring), they still have limitations such as high power consumption of sensor nodes or lack of necessary services (i.e. push notification, and local storage). Some systems based on Wi-Fi and Bluetooth are not energy-efficient because these protocols consume high power. Although other systems pay attention to reducing transmission power consumption of sensor nodes by using low power wireless transmission protocols (i.e. ANT, 6LoWPAN, and BLE), sensor nodes are still not energy-efficient due to high power consumption of other components (i.e. memory, micro-controller, and voltage regulator). By applying a comprehensive combination of software and hardware design methods altogether with a customized low-power wireless transmission protocol, power consumption of sensor nodes can be considerably reduced. In most of the systems, the quality of health-care service cannot be considered as comprehensive since the number of advanced services is limited.

This paper aims to provide an enhanced real-time and remote health monitoring IoT system. The major difference of this system from previous works is the adoption of energy-efficient sensor nodes based on the customized nRF protocol. The sensor nodes are carefully designed in terms of both software and hardware for reducing power consumption as much as possible. In addition, the system overcomes previously mentioned limitations in other systems. With the assistance of Fog computing in smart gateways, the quality of health-care service is dramatically improved.

III. Health monitoring IoT-based system architecture

The proposed health monitoring IoT-based system, whose architecture is shown in Fig. 1, is comprised of sensor nodes, gateways, and a back-end system. The sensor node acquires bio-signals (i.e. ECG, respiration, human temperature) and contextual data (i.e. room humidity and temperature). Then, it transmits the collected data to a gateway via an nRF module which is low-power, low-cost (about 1 Euro) and has fully customizable parameters. A selection of low transmission data rate is preferred for reducing energy consumption of the sensor node. Depending on particular signals and usages, signals can be kept intact or preprocessed before transmitting. The gateway in the system receives incoming data from the sensor nodes and transmits the data to Cloud. Similarly, data can be raw or processed data. In addition, the gateway with Fog can provide the large number of advanced services shown in Fig. 3 for improving the quality of health-care service.

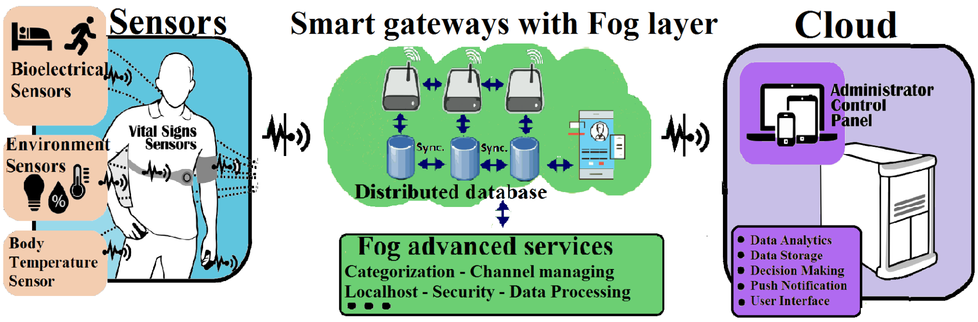


Fig. 1. A IoT-based e-health monitoring system architecture

A back-end system consisting of Cloud and an end-user application performs several tasks such as storage, data analysis and graphing, and push notification. End-users (e.g. doctors) can monitor their patients remotely by accessing real-time and history data in Cloud via an Internet browser or a mobile app.

IV. Sensor Node Design

A sensor node in the system primarily consists of a micro-controller, an nRF block, and specified sensors which are connected via SPI or I2C. The SPI protocol is preferred in the design due to its benefits of high data rate support and low energy consumption [15]. However, the more components are connected via SPI, the more difficult the issues get in terms of data categorization and verification. If the issues cannot be appropriately handled, the quality of data reduces.

Obviously, surrounding temperature, humidity, and body temperature do not change rapidly in seconds. Therefore, these can be acquired with a low data rate (i.e. 1 sample/s or 1 sample/30s). According to [16], a slight difference of 15 μJ energy consumption is observed when communicating with low data rate via SPI and I2C. Hence, I2C is used for connecting temperature and humidity sensors to the micro-controller while SPI is applied for connecting other components. The architecture of the sensor node is shown in Fig. 2.

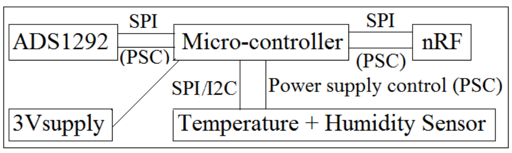


Fig. 2. Architecture of Sensor Node

Sensors including bio-potential and contextual measurement sensors must fulfill the requirement of low energy consumption. In addition, sensors must be capable of fast response and data sampling with low-noise, high precision and accurateness.

A micro-controller plays the most important role in the sensor node. It acquires data from sensors and transmits the data to the nRF block via SPI. In addition, it controls power supply of sensors and the nRF block. Therefore, a high performance and ultra-low-power micro-controller with power management modes such as several sleep modes is suitable for the sensor node.

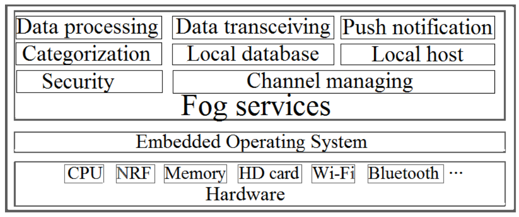


Fig. 3. Gateway structure

an nRF module consisting of an nRF integrated circuit (IC) and an on-PCB printed antenna is chosen for the design because it consumes low energy while supporting high data rates and communication bandwidth. In addition, it can be fully customized for the system. For example, it supports on-air data rate up to 2Mbps but 250kbps can be selected for reducing power consumption.

V. Gateway Structure

A gateway is supplied by a wall power outlet and fixed at a single room (i.e. a hospital room). The gateway integrated with Fog services shown in Fig. 3 is designed for serving several sensor nodes (i.e. 5-10 nodes). Detailed information of these services are explained as follows:

A. *Data transceiving*

To receive data via an nRF, the gateway is equipped with an nRF transceiver component. The component includes a micro-controller and an nRF module which are similar to the ones used in the sensor node. All collected data from the nRF transceiver is transmitted via UART to the gateway's primary MCU for further processing because the gateway's MCU is more powerful. The processed data is sent to Cloud via Ethernet or Wi-Fi. Accordingly, real-time data can be stored and monitored at Cloud servers.

B. *Data processing*

Processing of bio-signals includes pre-processing to eliminate noise from signals and extract useful features for further interpretation. Data processing service in the system is similar to ECG feature extraction presented in [17]. Data processing helps to improve the quality of health-care service and save transmission bandwidth between gateways and Cloud.

C. Local database

Local database includes two distinct databases which are internal and external usage databases. Internal usage database stores intact information which is only edited or updated by system administrators. For example, reference data (i.e. reference tables) used in algorithms or services (i.e. data processing and push notification) is stored in this database. The internal database is not synchronized with Cloud in most of the cases except a case of back-up data. In this case, data is encoded before being sending to Cloud. These specifications of the internal usage database help to avoid some unexpected security attacks from outsiders. Oppositely, the external usage database stores bio-medical signals and contextual data which are real-time synchronized with Cloud servers. Due to a limited storage capacity, the data is stored in the database for a period of time (i.e. several hours or a day), then replaced by new-coming data. Accordingly, the real-time data can be accessed at the Fog layer or at Cloud. For monitoring data in history, Cloud must be accessed.

D. Security

In order to protect information and resources of the system from unauthorized accesses, security and cryptography methods are applied in the Fog layer. It is a challenging task because the methods must not cause a dramatic increase in the system's latency.

E. Localhost with user interface

In order to provide real-time health monitoring at the gateway, localhost with user interface is integrated into the Fog layer. Concisely, a web server is run at the gateway for hosting a web-page which is user-friendly and able to represent both raw and processed data in text and graphical forms. The web-page provides functions such as a log-in form with a username and a password, or a searching tool.

F. Categorization service

In the Fog layer, a mechanism for classifying connected devices is integrated. The system regularly performs the mechanism for categorizing local and external connected devices. For local connected devices, real-time monitored data is directly retrieved from the Fog layer's local database instead of Cloud. This mechanism helps to reduce latency of the monitored data because the data transmission's path is shorter. When devices do not connect to the local network, data monitored at an end-user' s terminal (i.e. an Internet browser, or a mobile application) is retrieved from Cloud.

G. Push notification

Push notification is used for notifying end-users in case of abnormality. For example, when body temperature of a patient is too high over a threshold value, the push notification is triggered to send real-time messages to the patient and a doctor responsible for the patient.

H. Channel managing

It is important to provide a channel managing service in the Fog layer because it helps to avoid channel conflicting which causes incorrect data at a gateway's receiver. The service manages 126 channels of an nRF protocol to guarantee that each specific channel is reserved for a sensor node or a group of sensor nodes in which a channel with a higher frequency (channel with a higher number) will be given first. There is a table for recording assigned and unassigned channels. Unassigned channels will be verified for availability before being assigned for a device. The main purpose is to avoid channel conflicting between nRF channels. In the future work, channel conflicting between nRF and other technologies based on 2.4 GHz such as Wi-Fi will be investigated. Some channels (i.e. channel 116-126) are reserved for emergency notification and future usage. The channel managing service verifies incoming data regularly. When it detects some abnormality at a specific channel, it sends a request message to the channel and waits for an acknowledgement message(s) from a sensor node or a group of sensor nodes. By analyzing and investigating the acknowledgement message(s), it can detect channel conflicting. In case of a conflict, a push notification is triggered to notify the problem to system administrators.

VI. Implementation

The implementation of the system is divided into two parts described in detailed as follows:

A. Node implementation

ADS1292 is a low-cost (about 11 Euros), low-noise, and low-power analog front-end device for acquiring multichannel ECG with high data rates (i.e. 1000 samples/s). In the implementation, 2 ECG channels with a data rate of 250 samples/s are used. According to [18], the high quality of ECG signals can be obtained when sampling at 250 samples/s and higher data rates.

For acquiring humidity, environmental and body temperature, two BME280 sensors are used. These sensors are low-cost (5 Euros for each) and low-power while they provide high precision data and accurate measurements.

A low cost (1 Euro) and ultra-low power AVR ATMEGA328P micro-controller is used in the sensor node. It can support up to 20 MHz but power consumption is high. To reduce power consumption, 8 MHz is applied. As mentioned, the micro-controller controls voltage supply of sensors and an nRF block. Therefore, voltage supply must be appropriately chosen. In our implementation, 3V is the best voltage supply as suiting to all components. When the voltage supply is slightly less than 3V (i.e. 2.7V) due to a voltage drop characteristic of a battery, the sensor node is able to operate appropriately.

an nRF24L01 transceiver is used in the sensor node because of its low power consumption and low cost. As mentioned, it is customized for sending and receiving data with a data rate of 250kbps for reducing power consumption.

In order to protect data transmitted over an nRF network, AES-256 [19], which is a block cipher utilizing a 256-bit symmetric key for encryption and decryption, is implemented in the sensor node. The AES-256 is used because the algorithm is strong and the sensor node can perform the encryption algorithm fast. However, applying the algorithm increases power consumption of the sensor node and latency of the system. Results are shown in Section VII.

B. Gateway and back-end system implementation

In our implementation, Orange Pi One (about 14 Euros) is used as a core of the gateway for providing all mentioned services. Orange Pi One consists of Quad-core Cortex-A7 at 600 MHz each, 512MB low power memory, high speed and high capacity SD card (64GB), and several types of connectivity including Ethernet. As mentioned, in order to provide a capability of nRF wireless communication, the nRF component is added to Orange Pi via a UART port. The nRF component in the gateway including nRFL2401 [20] and AT-Mega328P [21] is similar to the nRF component in the sensor node. In order to receive data from the nRF component, a Python application is constructed in Orange Pi One. The application reads available data from the UART port then stores the data into the synchronized database. Simultaneously, the application transmits the collected data to Cloud.

Data processing is implemented in the Fog layer via several filtering and advanced processing algorithms. For example, moving average filter is first applied on raw ECG signal to remove baseline wander. Then 50 Hz notch Butterworth filter is applied to eliminate powerline interference. Finally, R peaks in ECG waveform are detected with conditional peak detection and R to R intervals are calculated for further application specific feature extraction. All of the data processing algorithms are implemented in Python.

Local database including both reference and synchronized database is implemented with the assistance of MySQL and a local SD card. For example, bio-signals and contextual data altogether with the recorded time are stored in the MySQL database. In addition, the MySQL database stores usernames and passwords of all users.

IPtables [22] and AES-256 used at Fog are implemented in C. IPtables is tables containing chains of rules for the treatment of incoming and outgoing packets at a gateway.

For implementing the web-page and server in the Fog layer, several up-to-date technologies such as HTML5, CSS, JavaScript, JSON, Python, and XML are used. The web-page is user-friendly and it is able to represent real-time data in text and graphical forms.

A channel managing service is implemented at Fog in C and Python. C is mainly used at an nRF receiving part while Python is used at Orange Pi One.

Categorization is implemented by a combination of a scanning service and database. By customizing an ”iw” package provided in Linux kernel, information of all devices connecting to a specific gateway via Wi-Fi can be acquired without any effort. The ”iw” package based on CLI configuration utility supports all new drivers of wireless devices. Although the ”iw” package has been in a further development process, it is suitable for the categorization service. The acquired information of connected wireless devices is recorded in tables in the database. The scanning service triggers an ”iw” command to update the information of connected devices regularly. The latency of running the ”iw” package regularly is not high because each gateway only serves several connected devices in a single room.

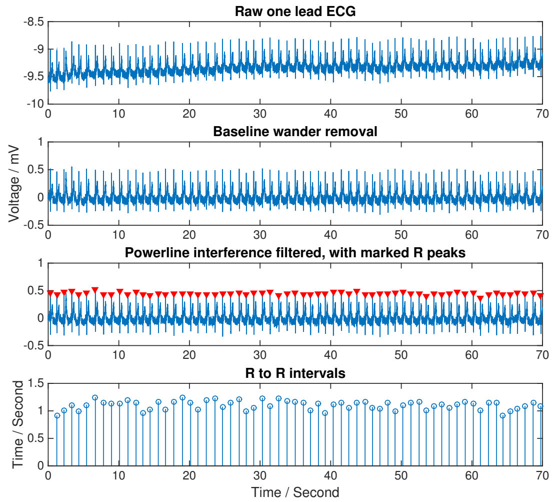


Fig. 4. Signal processing with one lead ECG

Push notification is implemented in both the fog layer and Cloud. When an end-user such as a doctor currently connects a local network (i.e. the same network with gateway’s network), the push notification service at the Fog layer is triggered for sending notification messages to the end-user directly via a TCP server installed in the gateway. When an end-user does not connect to a local network, the push notification service in Cloud implemented by a Google push service API is triggered for sending real-time notification messages.

VII. Experimental results

Results of ECG signal processing at the system’s gateway is presented in Fig. 4. It depicts the ECG data presented at Fog’s interface in graphical waveforms. The data includes raw data and processed data i.e. R-R intervals for calculating the heart rate.The ECG is acquired with one channel from a healthy person at a sample rate of 1000 samples/s, where the two electrodes are placed on the left wrist and the right wrist, respectively. Although the sensor node is designed for acquiring ECG with a data rate of 250 samples/s, a data rate of 1000 samples/s is applied in the experiment for testing the sensor node’s capability of sampling and transmitting with higher data rates. Results show that the quality of signals is still high when acquiring and transmitting at 1000 samples/s.

In order to measure power consumption of a sensor node, the developed prototype is tested while in operation. Results shown in Table I indicate that average current of the sensor node is very low about 6.5 mA for gathering and transmitting all data including ECG, body temperature, environment temperature, and humidity. In case of applying AES-256, average current of the node increases up to 7.01 mA.

The developed prototype of the sensor node shown in Fig. 5 approves its small physical size beside a two Euro coin for comparison. The actual size of the sensor node can be reduced dramatically since the prototype has extra many components for debugging purposes. The device and its battery are lightweight. With a 1000 mAh lithium button cell, the sensor node

TABLE I

AVERAGE POWER CONSUMPTION OF THE HEALTH MONITORING DEVICE AT A DATA RATE OF 18 KBPS

Mode	Voltage (V)	Average power (mW)
Idle	3	1.2
Active without AES-256	3	19.5
Active with AES-256	3	21.03

TABLE II

LATENCY OF RUNNING AES-256 FOR ENCRYPTING AND DECRYPTING 16 BYTES DATA AT SENSOR NODES AND GATEWAYS

Device	Algorithm (V)	Latency (μs)
sensor node	AES-256 encryption	170
Gateway	AES-256 decryption	38
Gateway	AES-256 encryption	42
Cloud server	AES-256 decryption	8

can operate up to 155 hours. Furthermore, the cost of the sensor node and the gateway is low, around 26 Euros and 20 Euros, respectively. Hence, the sensor node can be as a wearable device.

For testing the quality of data during transmission, two types of data including fixed data and actual data collected from sensors are used during experiments. Results show that data loss does not occur during communicating and longer range transmission requires higher power.

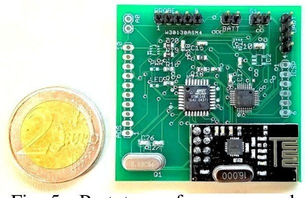


Fig. 5. Prototype of a sensor node

Table II shows that, when using AES-256 in a sensor node, average latency of the sensor node and the system increases about 170 μs and 260 μs, respectively. However, these small increases do not cause dramatically negative impacts on the system's performance and latency.

VIII. CONCLUSIONS

In this paper, a low-cost remote health monitoring IoT-based system with the Fog layer has been proposed. The designed system is able to acquire data including bio-signals (i.e. ECG and respiration) and contextual data (i.e. environment temperature and humidity) and transmit the data wirelessly for real-time and remote monitoring. In addition, with the assistance of the Fog layer, the system provides advanced services such as data processing, categorization, push notification and channel management for improving the quality of health-care service. Furthermore, a design of a low-cost and portable sensor node has been presented. The sensor node is able to operate for a long period of time reaching up to 155 hours due to its high energy efficiency. By applying this system at hospitals and homes, emergencies (i.e. related to cardiovascular diseases) can be notified in real-time to medical doctors for in time action to avoid serious consequences.

REFERENCES

[1] WHO, "Diabetes." http://www.who.int/mediacentre/factsheets/fs312/en/ [accessed 2017-01-22].

[2] WHO, "Cardiovascular disease." http://www.who.int/cardiovascular_diseases/en/ [accessed 2017-01-22].

[3] D. Uckelmann *et al.*, "An architectural approach towards the future internet of things," in *Architecting the internet of things*, pp. 1–24, Springer, 2011.

[4] F. Bonomi *et al.*, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, ACM, 2012.

[5] A. M. Rahmani *et al.*, "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach," *Future Generation Computer Systems*, 2017.

[6] B. Shiferaw Negash *et al.*, "Leveraging fog computing for healthcare iot," *Fog Computing in the Internet of Things - Intelligence at the Edge*, 2017.

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

[8] M. Jiang *et al.*, "IoT-based remote facial expression monitoring system with sEMG signal," in *IEEE Sensors Applications Symposium*, pp. 1–6, 2016.

[9] J. Gómez *et al.*, "Patient monitoring system based on internet of things," *Procedia Computer Science*, vol. 83, pp. 90–97, 2016.

[10] E. Nemati *et al.*, "A wireless wearable ecg sensor for long-term applications," *IEEE Communications Magazine*, vol. 50, no. 1, 2012.

[11] L. Fanucci *et al.*, "Sensing devices and sensor signal processing for remote monitoring of vital signs in chf patients," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 3, pp. 553–569, 2013.

[12] K. Natarajan *et al.*, "Smart health care system using internet of things," *Journal of Network Communications and Emerging Technologies*, vol. 6, no. 3, 2016.

[13] F. Touati, R. Tabish, and A. B. Mnaouer, "A real-time ble enabled ecg system for remote monitoring," *APCBEE Procedia*, vol. 7, pp. 124–131, 2013.

[14] B. Yu *et al.*, "Bluetooth low energy (ble) based mobile electrocardiogram monitoring system," in *(ICIA), 2012 International Conference on*, pp. 763–767, IEEE, 2012.

[15] T. N. Gia *et al.*, "IoT-based fall detection system with energy efficient sensor nodes," in *Nordic Circuits and Systems Conference (NORCAS), 2016 IEEE*, pp. 1–6, IEEE, 2016.

[16] K. Mikhaylov *et al.*, "Evaluation of power efficiency for digital serial interfaces of microcontrollers," in *5th NTMS'2012*, pp. 1–5, 2012.

[17] T. N. Gia *et al.*, "Fog computing in healthcare internet of things: A case study on ecg feature extraction," in *(CIT), 2015 IEEE International Conference on*, pp. 356–363, IEEE, 2015.

[18] R. F. Yazicioglu *et al.*, *Biopotential readout circuits for portable acquisition systems*. Springer Science & Business Media, 2008.

[19] NIST, "Announcing the advanced encryption standard (aes)," 2001.

[20] Nordic Semiconductor, "nRF24L01+Single Chip 2.4GHz Transceiver." http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01 [accessed 2017-01-30].

[21] ATmel, "ATmega328P." http://www.atmel.com/devices/ATMEGA328P.aspx [accessed 2017-01-30].

[22] Rusty Russell, "iptables(8) - Linux man page." https://linux.die.net/man/8/iptables [accessed 2017-01-30].

Paper III

Exploiting Smart E-Health Gateways at the Edge of Healthcare Internet-of-Things: A Fog Computing Approach

Amir M Rahmani, Tuan Nguyen Gia, Behailu Negash, Arman Anzanpour, Iman Azimi, Mingzhe Jiang, Pasi Liljeberg

Original published in Future Generation Computer Systems, The International Journal of eScience, Elsevier, 78, pages 641-658

Future Generation Computer Systems 78 (2018) 641–658

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach

Amir M. Rahmani [a,b,*], Tuan Nguyen Gia [c], Behailu Negash [c], Arman Anzanpour [c], Iman Azimi [c], Mingzhe Jiang [c], Pasi Liljeberg [c]

[a] Department of Computer Science, University of California Irvine, USA
[b] Institute of Computer Technology, TU Wien, Vienna, Austria
[c] Department of Information Technology, University of Turku, Turku, Finland

ARTICLE INFO

Article history:
Received 26 May 2016
Received in revised form 8 November 2016
Accepted 8 February 2017
Available online 10 February 2017

Keywords:
Internet of Things
Healthcare
Edge/Fog computing
Mobility
Smart hospital
Home care
Smart gateway
Sensor network

ABSTRACT

Current developments in ICTs such as in Internet-of-Things (IoT) and Cyber–Physical Systems (CPS) allow us to develop healthcare solutions with more intelligent and prediction capabilities both for daily life (home/office) and in-hospitals. In most of IoT-based healthcare systems, especially at smart homes or hospitals, a bridging point (i.e., gateway) is needed between sensor infrastructure network and the Internet. The gateway at the edge of the network often just performs basic functions such as translating between the protocols used in the Internet and sensor networks. These gateways have beneficial knowledge and constructive control over both the sensor network and the data to be transmitted through the Internet. In this paper, we exploit the strategic position of such gateways at the edge of the network to offer several higher-level services such as local storage, real-time local data processing, embedded data mining, etc., presenting thus a Smart e-Health Gateway. We then propose to exploit the concept of Fog Computing in Healthcare IoT systems by forming a Geo-distributed intermediary layer of intelligence between sensor nodes and Cloud. By taking responsibility for handling some burdens of the sensor network and a remote healthcare center, our Fog-assisted system architecture can cope with many challenges in ubiquitous healthcare systems such as mobility, energy efficiency, scalability, and reliability issues. A successful implementation of Smart e-Health Gateways can enable massive deployment of ubiquitous health monitoring systems especially in clinical environments. We also present a prototype of a Smart e-Health Gateway called UT-GATE where some of the discussed higher-level features have been implemented. We also implement an IoT-based Early Warning Score (EWS) health monitoring to practically show the efficiency and relevance of our system on addressing a medical case study. Our proof-of-concept design demonstrates an IoT-based health monitoring system with enhanced overall system intelligence, energy efficiency, mobility, performance, interoperability, security, and reliability.

1. Introduction

Internet of Things (IoT) is getting a wide acceptance and a growing adoption in many aspects of our daily life [1,2]. IoT technology provides a competent and structured approach to improve health and wellbeing of mankind. It is predicted that IoT-based systems will remodel the healthcare sector in terms of social benefits and penetration as well as cost-efficiency [3,4]. Due to the ubiquitous computing nature of IoT, all the healthcare system entities (individuals, appliances, medicine) can be monitored and managed continuously. By applying IoT technologies to healthcare, the quality and cost of medical care can be improved by automating tasks previously performed by humans [5–7]. In that sense, IoT enables Electronic Health (eHealth), Mobile Health (mHealth) and Ambient Assisted Living (AAL) that allow remote monitoring and tracking of patients living alone at home or treated in hospitals, and creates a continuum among these through cloud access [4,8].

It is no longer sufficient enough to design just standalone wearable devices, instead it becomes vital to create a complete ecosystem in which sensors in a body area network seamlessly synchronize data to cloud services through the IoT infrastructure [9–11]. The architectural elements generally needed in healthcare IoT systems (Health-IoT) are illustrated in Fig. 1. The architecture includes three main components: (i) body area sensor network, (ii) Internet-connected gateways, and (iii) cloud and big data support. Various applications provide services to different stakeholders in the system through this platform. Data generated from

* Corresponding author at: Department of Computer Science, University of California Irvine, USA.
E-mail address: amirr1@uci.edu (A.M. Rahmani).

http://dx.doi.org/10.1016/j.future.2017.02.014

Fig. 1. General IoT-based health monitoring system.

sensors attached to users is made available to caregivers, family members and authorized parties giving them the ability to check the subject's vital signs from anywhere at any time.

According to predictions, the current hospital-centered healthcare systems will evolve first to hospital–home-balanced in 2020, and then ultimately to home-centered in 2030 [12]. In order to realize such evolution, new system architectures, technologies, and computing paradigms are required, particularly in the smart spaces and e-Health domains. It should be noted that the paradigm shift towards smart ubiquitous healthcare systems results in new challenges to manifest themselves in fulfilling different system requirements such as reliability, interoperability, energy-efficiency, low-latency response, mobility, security, etc.

Gateways generally act as a hub between a sensor layer and cloud services. With an in-depth observation of a gateway's role in a smart home/hospital, where the mobility and location of users and things are confined to the hospital premises or the building, it can be noticed that the stationary nature of gateways empowers them with the luxury of being non-resource constrained in terms of processing power, power consumption and communication bandwidth. Such a valuable characteristics can be exploited by reinforcing the gateways with sufficient processing power, intelligence, and orchestrated networking capabilities, thus becoming a smart e-Health gateway.

However, the advantageous services that can be potentially offered by a smart gateway will be limited if the gateway is deployed in a standalone and independent fashion. Scalability and mobility issues can easily arise and the efficacy of the solution will be significantly limited. This reveals the demand for an intermediary layer of computation where a geo-distributed network of smart gateways provides intelligence at the edge of the network and facilitates the interplay between sensors layer and cloud layer. This paradigm, which is also called fog or edge computing [13–15], enables the system to support seamless mobility, load balancing, efficient scalability, low-latency response, and developing applications utilizing services offered by multiple sensors and gateways, just to mention a few.

In this paper, which is a major extension of our recent works published in [16], we present a fog computing-based solution to enhance different characteristics of IoT architectures used for healthcare applications in terms of energy-efficiency, performance, reliability, interoperability, to name a few. The main contributions of this article are as follows:

- Presenting a practical solution to take advantage of fog computing in IoT-health systems.
- Elaborating the features of a fog computing based health-IoT system and its services from different perspectives.
- Proposing fog-based mobility support to enable seamless connectivity for mobile sensors.
- Providing a proof of concept full-system implementation from development of cloud services to hardware–software demonstration of our prototype of Smart e-Health Gateways.
- Demonstrating the system with a medical case study called Early Warning Scores (EWS) with hierarchical fog-assisted cloud computing.

The rest of the paper is organized as follows: In Section 2, related work and motivation of this paper are presented. Section 3 describes the architecture of a fog-assisted IoT based e-Health platform using Smart e-Health Gateways. The properties and features of the networked smart e-Health gateways are presented in more detail in Section 4. Demonstration of our Smart e-Health Gateways on a medical case study along with experimental results are provided and discussed in Section 5. Finally, Section 6 concludes the paper.

2. Related work and motivation

In the healthcare context, designing an efficient IoT-based system is a challenging task due to the following main issues. First, the chosen sensor networking technology must be resource-efficient and customized for e-Health applications. Medical sensor nodes, especially implanted ones, have much lower processing power, memory, transmission speed, and energy supply than sensors in other sensor networks domain. Second, unlike common sensor networks where interval-based data transmission is used (e.g., temperature and humidity monitoring), e-Health applications often need to manage streaming-based transmissions where realtime requirements need to be considered. Consequently a considerable energy is dissipated during the transmission process. For instance, Electrocardiogram (ECG) signal transmission requires 4 kbps bandwidth per channel. Third, in multi-patient applications such as in smart hospitals, hardware platforms with a high processing power and parallel processing features (e.g., multi-core processors) are needed in the gateway due to concurrent nature of the workloads. However, as we discuss in this section, the existing general-purpose gateways are not designed for such scenarios. Fourth, reliability in e-Health application is of utmost importance and even short system unavailability often cannot be tolerated. Thus, as we discuss in the following, the limited resources of medical sensor nodes render the use of general purpose gateways inefficient in most circumstances with respect to delay, energy, and reliability.

Using a three tier architecture, with varying computational capacity, for IoT applications is common in both industry and research. The focal point of most of the related works is the gateway used in the middle tier of the IoT architecture. One of many such efforts is presented in [17,18], which proposes gateways to transparently connect sensor networks with different protocols such as ZigBee, Bluetooth, and Ethernet to the Internet. However, these gateways have limited flexibility as they cannot be customized for different applications. In a different category of related work, Mueller et al. [19] present a gateway called SwissGate which handles and optimize the operation of a sensor network. They specifically apply SwissGate on home automation applications such as measuring heating, ventilation, and air conditioning control (HVAC) parameters. Bimschas et al. [20] aim at providing some levels of intelligence to gateways by enabling them to execute application code. They propose a middleware for the gateway to offer four possible services: protocol conversion, request caching, intelligent caching, and discovery.

Another work by Jong-Wan et al. [21] present a sensor network system comprising of a main server and several sensing-servers acting as gateways and connecting with different sensor networks. Using network-dependent sensing-servers as gateways results in high implementation and hardware cost as well as poor scalability, making such a design inefficient for many IoT applications. In a related work presented in [22], a plug-configurable-play service-oriented generic gateway is proposed in order to provide simple and rapid deployment of various external sensor network applications. The gateway offers a proper level of interoperability by facilitating heterogeneous sensor networks to work together. However, the middleware presented in their work lacks intelligence and runs on PC, limiting its applicability for many IoT applications. In a similar attempt, Guoqiang et al. [23] propose a general purpose smart gateway. It provides pluggable architecture which enables communication among different protocols, unified external interfaces fitting for flexible software development, and flexible protocol to translate different sensor data.

In order to save energy and reduce the cost of smart home, Bian et al. [24] present a new type of intelligent home control system, using an Android Phone as a temporary home gateway instead of the default home gateway. The aim of the work is to automatically shut down the unused devices by predicting user behavior. In a different application domain, the work presented in [25] proposes a prototype of a smart 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) border router which makes local decisions of health states using a Hidden Markov Model.

In another work, Satyanarayanan et al. [26] propose that mobile devices can use complex algorithms such as facial recognition and language translation to augment human cognition. However, limitations in processing power and long WAN latency are unacceptable. The authors then propose cloudlets which consist of a computer with wireless connectivity in the vicinity of the mobile device. A base virtual machine (VM) is installed in the cloudlets. VM instances are launched from the cloudlets and configured using an overlay script from the mobile device with desired application. The VMs are designed to migrate from one cloudlet to another to allow ubiquity. The authors claim that applications run faster with this configuration due to lower latencies and higher processing power. However, the authors face challenges regarding long launching times. In [27], Stantchev et al. present the benefits of three-level (i.e., fog based) architecture for a smart healthcare infrastructure from servitization and business point of view. However, they only focus on high level architectural modeling aspects and do not discuss real world implementation and experimental evaluation of the services.

Although efforts of using a gateway in IoT have been greatly expanded in recent years, there are only small improvements towards realizing smart gateways streamlined specifically for the healthcare domain. Most of the presented efforts focus on general purpose gateway designs which affects the provided level of intelligence due to lack of information about the application domain. Some of these efforts limit their level of intelligence only for the sake of plug-and-play ability, or reconfigurability to various domains. Some others just focus on specific domains such as smart home.

Existing contributions using a gateway as an intermediary between sensors and cloud storage, consider a minimal role for the gateway, for example applying simple set of rules. In few cases, a gateway is leveraged for domain specific purposes. However, such platforms fail to satisfy the requirements of other domains. In the healthcare sector, particularly for remote health monitoring, a high level of reliability, availability and robustness is demanded. Moreover, security and privacy issues are of critical importance. The purpose of our smart e-Health gateway is to satisfy these domain-specific requirements by customizing gateways for the healthcare domain and providing intelligence closer to patients.

Our proposal is motivated by the fact that in a smart hospital or in-home healthcare, the gateway is in the unique position between both the BAN/PAN/LAN and the wide area network (WAN). This promising opportunity can be exploited by different means such as collecting health and context information from these networks and providing different services accordingly. By geographically distributing and networking smart e-Health gateways, a smart intermediary layer can be formed to provide smooth and efficient healthcare services without limiting the mobility of patients. In general, the motivations of utilizing a Geo-distributed network of smart e-Health gateways for Health-IoT are manifold. The major ones are:

1. To provide local data processing for real-time notification for medical professionals as described in Sections 4.1, 4.3 and 4.4.
2. To secure the sensitive medical data gathered by sensors and keep the privacy of patients, discussed in Section 4.5.
3. To give interoperability for heterogeneous platforms and communication protocols used in medical sensor networks, elaborated in Section 4.6.
4. To provide mobility of patients across the area of coverage of the Fog layer for hospital and home based care, discussed in Section 4.7.
5. To enable the underlying sensor network to be more efficient in terms of energy and communication bandwidth presented in Section 4.8.

The following section provides the system architecture of the overall Health-IoT system.

3. System architecture and the role of fog computing

The large scale implementation of IoT is expected to introduce billions of additional resource-constrained devices connected to the Internet. The majority of these devices, for example wearable and implantable medical sensors, are not capable of storing data they generate. A straightforward design approach is to transfer this data to a cloud for processing. Given the large number of connected devices, the latency of the connection with the cloud could be significant. Moreover, these devices are power and bandwidth constrained, that make them unfit directly to the cloud architecture. Fog Computing [13] is an essential paradigm shift towards a hierarchical system architecture and a more responsive design. As shown in Fig. 2, Fog is an intermediate computing layer between the cloud and end devices that complements the advantages of cloud computing by providing additional services for the emerging requirements in the field of IoT. This intermediate layer is discussed in different terms in various articles, such as Mobile Edge computing [28], Micro-clouds [29], or simply just Edge Computing [30]. The concept behind our smart e-Health gateway is to provide different services at the edge of the network between the smart objects and the cloud.

Fig. 3 shows a detailed view on how the components of a Health-IoT system can be organized in a distributed manner across the three layers to be used in smart hospitals or home. In such

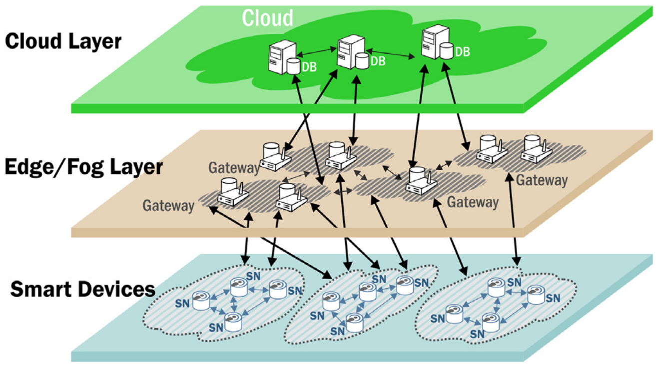


Fig. 2. Generic fog-based IoT architecture.

systems, patient health related information is recorded by body-worn or implanted sensors, with which the patient is equipped for personal monitoring of multiple parameters. This health data can be also supplemented with context information (e.g., date, time, location, temperature). Context-awareness enables to identify unusual patterns and make more precise inferences about the situation. Other sensors and actuators (e.g., medical equipment) can be also connected to the systems to transmit data to medical staff such as high-resolution images (e.g., CAT scan, magnetic resonance imaging). The system architecture includes the following main components:

1. **Medical Sensors and Actuators Network:** Enabled by the ubiquitous identification, sensing, and communication capability, biomedical and context signals are captured from the body and room. The data is then transmitted to the gateway via wireless or wired communication protocols such as Bluetooth, Wi-Fi, ZigBee or 6LoWPAN.
2. **Network of Smart e-Health Gateways:** This layer is built from multiple geographically distributed smart e-Health gateways, i.e., forming the fog. Each gateway, which supports different communication protocols, acts as a dynamic touching point between a sensor network and the local switch/Internet. It receives data from different sub-networks, performs protocol conversion, and provides other higher level services such as data aggregation, filtering and dimensionality reduction.
3. **Back-End System:** Back-end system consists of a cloud computing platform that implements broadcasting, data warehouse and data analytics. Finally, it provides demonstrations for web client as a graphical user interface for final visualization and feedback. The collected health and context data represents a source of big data [31,32] for statistical and epidemiological medical research (e.g., detecting approaching epidemic diseases).

As can be observed from Fig. 3, the intermediate layer is composed of a network of smart e-Health gateways at a strategic location to offer many higher level services to enhance the system characteristics in different aspects. The following section gives an overview of the advantages of this layer in IoT systems.

4. Properties and features of smart e-Health gateways at the fog layer

As mentioned before, the main role of a gateway is to support various wireless protocols and take care of inter-device communication. In this section, we extend its role to become fog enabler by (i) forming an orchestrated network of gateways and (ii) implementing several features such as acting as repository (i.e., local database) to temporarily store sensors' and users' data, and incorporating it with data fusion, aggregation, and interpretation techniques. These are essential to provide local pre-processing of sensors' data, becoming thus a Smart e-Health Gateway.

4.1. Local data processing

As a key feature of fog computing, local data processing is implemented to provide intelligence at the gateway by which the

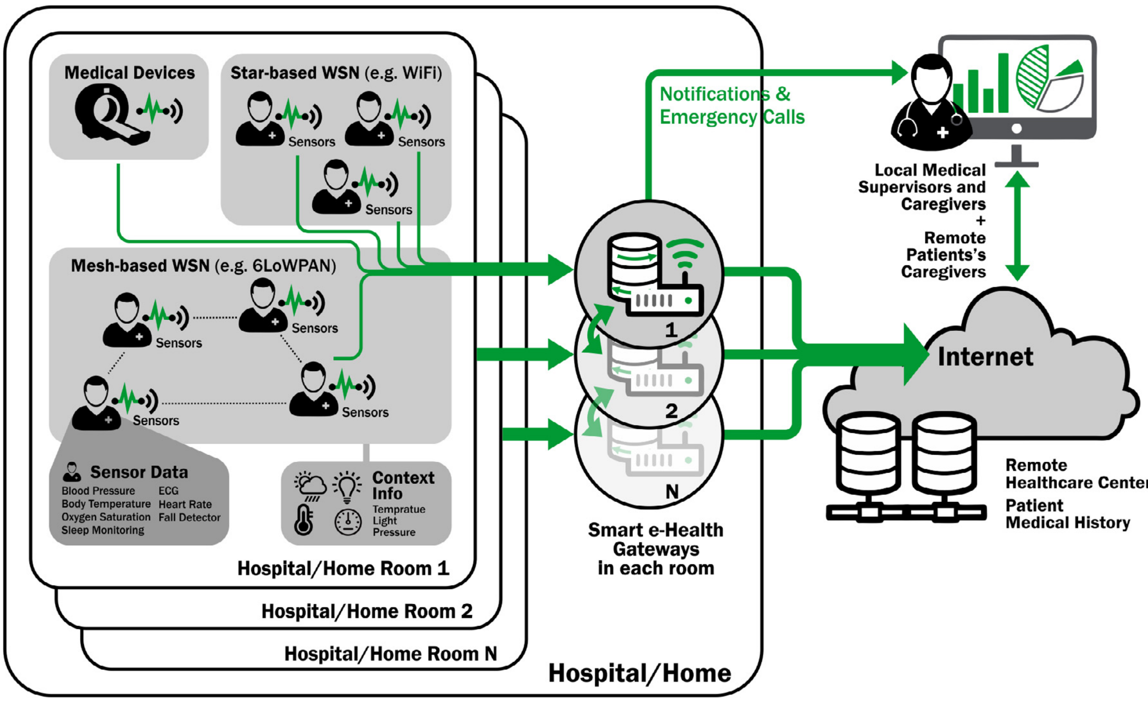


Fig. 3. Components of IoT-based health monitoring system.

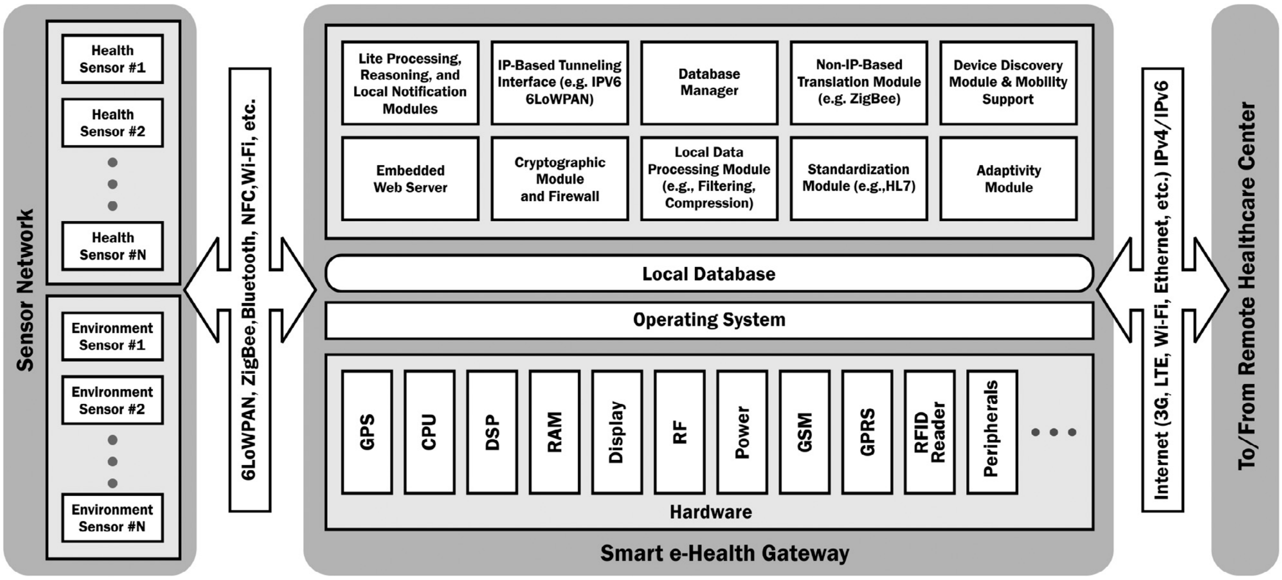


Fig. 4. Smart e-Health gateway architecture.

streaming data is analyzed locally. According to the system architecture, fog layer requires to continuously handle a large amount of sensory data in a short time and response appropriately with respect to various conditions. This task becomes more important in medical cases by enabling the system to react as fast as possible in medical emergencies [33]. Fig. 4 illustrates a conceptual architecture of a smart e-Health gateway utilizing a local processing unit for data filtering, data compression, data fusion, and data analysis.

4.1.1. Data filtering

Receiving data from various sensors makes it essential to implement appropriate pre-processing at the edge before any more advanced processing such as data analysis is performed. Bio-signals (e.g., ECG, EEG and EMG) collected from users' body are the primary sources of information for assessing a patient health status. They usually contain complicated shapes with small amplitude (i.e., in the range of millivolts) and various frequencies. During the sensing of human body, noise is also accumulated to the bio-signals and distort the signal quality. Such noises are produced by different sources such as oscillations of alternating current in the electric power grid, electromagnetic interference from other electrical devices, and improper attachment of sensors to users' body.

The smart e-Health gateway at the fog layer can address this issue as it interfaces sensors directly. The fog layer receives digitalized signals from sensors via various communication protocols. Although sensors may implement light-weight filtering to remove some noises at the data collection phase, more robust and complex data filtering is still required at the fog layer.

4.1.2. Data compression

In the context of data communication, data compression is used for reducing communication latency and energy consumed during transaction. Both lossy and lossless data compression are widely used in the IoT domain depending on the application. Lossless data compression has the shortcoming of requiring rather heavy computation for performing complex algorithms. Therefore, there are processing power requirements in terms of processor speed and memory size when using a lossless data compression method.

In device layer of Health-IoT systems, both lossy and lossless compression methods are useful. However, in many cases lossy data compression is more suitable for resource-constrained sensors due to limitations such as battery life time and available processing power. For instance, many popular lossless ECG compression methods [34–36] do not fit to many types of sensors while lossy compression methods, for example a method introduced by Yu et al. [37], are feasible in terms of hardware requirements. However, for applications such as real-time ECG monitoring, it is desirable to have lossless compression to ensure that all features of the signals all observable with a high precision. Fog computing provides the required computational power for efficiently running complex lossless data compression algorithms by offloading the burden from device layer. Furthermore, it enables real-time operation while using lossless data compression.

4.1.3. Data fusion

Data fusion enables the system to effectively decrease the volume of data, and consequently reduce the energy needed for data transmission. Data fusion is categorized into three classes: complementary, competitive, and cooperative [38]. Complementary data fusion can be performed at the fog layer to achieve better global knowledge. Obtaining temperature difference between body and the environment is an instance provided from two sensory data. Competitive data fusion can be also utilized at the edge in a way that data from a single parameter is collected from different sources to improve the accuracy and consistency of results in case of sensors' failure. Finally, cooperative data fusion can also provide benefits at the edge in a way that new information is extracted in smart gateways from the heterogeneous data collected from diverse sources. For instance, cooperative data fusion can provide comprehensive information about the medical state of a patient from his/her vital signs.

4.1.4. Data analysis

The sensitivity of the system is improved by applying local data analysis at the edge. It can assist the system to detect and predict emergency situations. For instance, in case of fall detection for elderly people, fog layer can locally offer fall-detection related processing rather than sending parameters to a cloud and waiting for the responses. Consequently, the system reacts to the emergency situation faster and more reliable and implements real-time responses. In addition to the sensitivity of the system, utilizing data analysis in the fog layer enables the system to minimize the processing latencies of critical parameters.

Furthermore, local data analysis and local feedback from the sensory data improve the system reliability and consistency in case of unavailability of Internet connection. For long-term remote monitoring of individuals suffering from chronic diseases, Internet disconnection may occur frequently. In this case, fog computing enables to keep the functionality of the system operational locally. Moreover, it is possible to save the sensory data and processing results in a local storage at the fog layer and synchronize them with the Cloud later.

4.2. Adaptivity

Considering the application of fog layer in various cases, some predefined parameters (e.g., transmission rate from sensors to the cloud) are set depending on the use case. However, it is also essential for the fog layer to be reconfigurable and adaptive over time, particularly when critical events take place. The reconfiguration can be dynamically applied for various services utilizing incremental machine learning algorithms such as incremental support vector machine and incremental neural networks.

Data transmission at the fog layer needs to be adaptively tuned. This includes not only data requests from sensors but also data transmission rate from the fog layer to the cloud. For instance, in long-term monitoring of a patient suffering from a cardiovascular disease, the system should learn to increase the request rates (priority) for heart-related parameters when detecting an abnormal heart-related sign. Furthermore, transmission rate to the cloud is performed by assigning priorities to different services and parameters. The priority of data transmission rate to the cloud for patients with acute diseases needs to be higher, while patients suffering from chronic diseases require a lower transmission rate. As a result, adaptive fog computing improves the system performance by increasing sensitivity and specificity of critical parameters.

4.3. Local storage

To ensure that the system can smoothly recover the data, gateways should store the incoming data in a local storage. The operating system on a gateway handles the local repository and stores the data in a non-volatile memory. Based on the type and significance, data can be stored in local storage in a compressed or encrypted way. Data in the repository can be exported to medical standard formats such as Health Level-7 (HL7) [39] if required.

Data storage is also necessary for other functionalities of the gateway. As discussed earlier, gateway is responsible for data analysis, compression, filtering, and encryption, all these functions need a local temporary storage. Because the speed to transfer data from the gateway to the cloud is limited by network bandwidth, and computations are limited by processing power of the gateway, in case of inequality of data processing and data transfer, local storage will act as a cache to implement a continuous data flow. Data storage on gateways makes the system reliable and robust even when network is unavailable. The local repository is handled by the database manager unit shown in Fig. 4.

4.4. Local actuation

In IoT-based healthcare system, actuation can be classified into different forms. It can be used in the form of information streaming, controlling medical actuators, and sensor network reconfiguration. In most of these cases, a predictable and fast response time is demanded. Examples for fast response actuations are adjusting the frequency of electrical nerve stimulation based on the heart rate, or adjusting insulin release rate in automatic pumps based on the patient blood Glucose and other vital signs.

Streaming patient medical signals in real-time to a control panel for medical experts is also a sensitive case to transmission delay where a minimum samples per second rate needs to be met. Using the local processing power and networking facilities, the gateway is able to stream real-time signals such as ECG and PPG (photoplethysmogram) to a client device (i.e., tablet), without relying on the Internet connectivity.

Notifications are also necessary features for smart e-Health gateways at the edge of the network. Health monitoring systems often need to inform and warn medical teams, caregivers, and the patient about an emergency situations. Any failure in the notification service may cause serious problems for both patients and medical treatments. Compared to a cloud server which is able to send notifications via several methods, a gateway has limited resources and can only notify via some specific media. However, the advantage is that gateway-based notifications act independently (e.g., via the local network or GSM) even during unavailability of cloud server, to maximize the reliability of the system and to ensure that users can receive critical notifications in time.

4.5. Security

Security can be considered as one of the most essential requirements in Health-IoT applications on the ground that an unsecured systems can have serious vulnerabilities. In order to provide a high level of security, operating system level techniques can be utilized at gateways such as IPtable offered by Linux. More precisely, IPtables and IPFW provided by Linux kernel firewall can be used for configuring IP packet table which is basically a set of rules for network packets. Typically, IP tables are configured to grant permissions to some ports for communication while other ports are blocked for preventing unnecessary traffic [40]. As the gateway can also act as an embedded web server during network unavailability or whenever needed, it can communicate over secure HTTPS and authenticate sensor nodes to maintain the confidentiality, integrity, and authenticity of the system. Although IPtables provide some advantages, they cannot be considered as a robust tool for security. In order to have a higher level of security, IPtables must be in cooperation with other advanced security methods. To address these issues, different approaches have been recently proposed in the literature [41–43]. However, cryptographic operations in these approaches are heavy in terms of required processing power and energy making them unfeasible for resource-constrained devices. In a recent proposal focused on security, Rahimi et al. [44,45] introduced a secure and efficient authentication and authorization approach for Health-IoT systems which requires some processing power at the edge. More precisely, on the contrary of running secure methods at resource-constrained sensor nodes, the approach exploits properties of a smart gateway in fog computing for heavy and security-related jobs.

4.6. Interoperability and reconfigurability

Besides the standardization efforts, it is evident that interoperability plays a key role to the success of Health-IoT systems. With such heterogeneous mix of networking technologies, protocols and platform choices to implement IoT-based system, integrating these application silos is an evident challenge. Our smart e-Health gateway plays a key role in providing interoperability for the various sensors connected via distinct network interfaces. As shown in Fig. 4, the health sensors and the context sensors are connected to the Smart e-Health Gateway using either wireless network or wired connections while using different standards (e.g., ZigBee, 6LoWPAN, Bluetooth, Wi-Fi) to communicate with the gateway. The smartness of the gateway comes here in the form of easy integration of these heterogeneous networking technologies, protocols

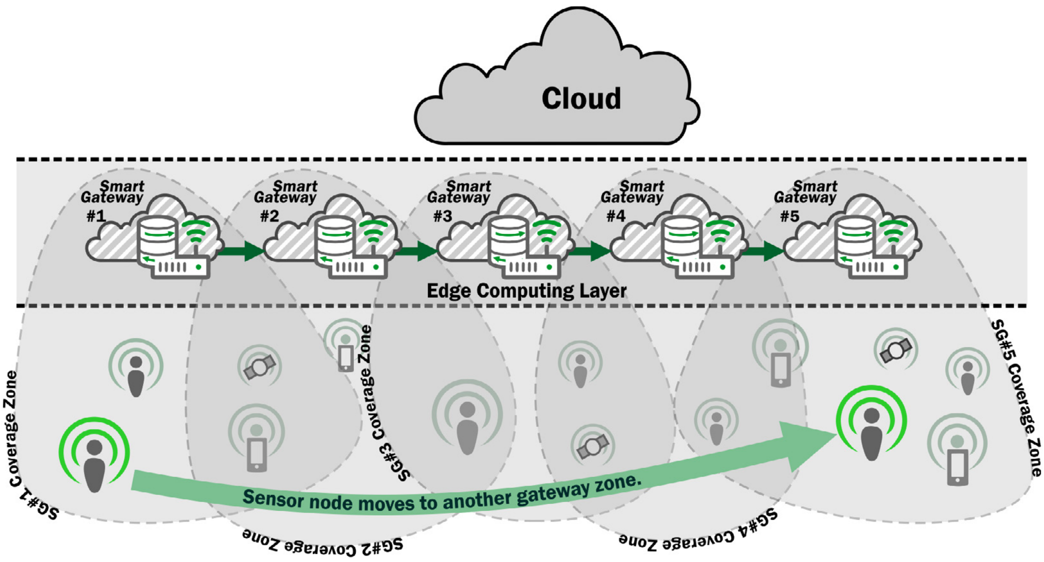


Fig. 5. Node mobility in fog computing.

and standards thereby enabling them to exchange information and work seamlessly.

Technical Interoperability: The various system components in IoT-based system are built by different vendors, and hence they use different network interfaces and standards. In our smart e-Health gateway implementation, technical interoperability is achieved by directing exchanged information to the smart e-Health gateway which has multiple interfaces. Adaptation layers in the gateway facilitate inter protocol exchange of messages and format conversion, which is part of the syntactic interoperability discussed later. One should not have all the possible interfaces in the gateway unless they are used by sensor nodes. This dynamic inclusion or removal of such interfaces is managed by the reconfigurability feature of the smart gateway.

Syntactic Interoperability: Syntactic interoperability layer relies on the previous, technical interoperability layer. It deals with the format of messages exchanged between systems. Once a message is delivered, the receiver has to identify the content of the message and hence the need for the protocol support modules in the central gateway. Variations in protocols result in differences in the format of the message. One common example of tunneling is the case when a 6LoWPAN edge router needs to tunnel between 6LoWPAN and IPv4/IPv6 protocols [46]. Protocol translators can also be used to buffer an incoming message and forward in another format. These functions are realized using the two modules, IP based tunneling interface and non-IP based translation module, shown in Fig. 4.

In addition to network level protocols, medical data is formated in a specific format. Data in any of the Electronic Health Record (EHR) standards, such as HL7 [39], is re-formatted when necessary by the smart gateway, in the standardization module shown in Fig. 4. Sensor nodes can be free from processing overhead that results in formatting the data into standards. In addition, the overhead on the communication channel due to the standards related information that could be sent with the data is removed. This makes the sensor nodes to be energy and bandwidth efficient by sending unformatted data to the gateway.

Semantic Interoperability: To have this common understanding of the meaning of a certain data, a vocabulary (i.e., ontology [47]) of the terms used in that specific context has to be shared first. In addition to the definition of the terms exchanged, the relationship among these terms has to be drawn. Our smart e-Health gateway is designed to provide semantic interoperability in two ways. The first interoperability is for other devices interested in the data collected from sensor nodes. On the other side, the gateway is connected to the Internet and human readable data should be presented.

4.7. Device discovery and mobility support

Mobility in general involves two main processes, handover and roaming, that are needed in order to avoid data loss and service interruptions, and to maintain quality of service (QoS). In a mobile host, handover arises in case of switching from a connection channel to another channel while roaming takes place when moving from one network to another. Mobility can also be categorized into macro and micro types which are defined as mobility between different network domains and within a network domain, respectively [48].

There exists a few methods supporting mobility at edge routers [49]. However, there is no comprehensive method available at the moment to fully address the challenges in the IoT realm. For example, incompatibility with multihop routing, and requirements of NS (Neighbor Solicitation)/NA (Neighbor Advertisement) exchanges are two open issues in the proxy mobile IPv6. NEMO becomes more complicated when different types of mobility (micro and macro) happen simultaneously. There are also some mobility-related mechanisms in the literature handling the mobility support either at the cloud or via additional remote assistant servers. This results in increased handover latency due to often a long distance from a mobile node to the cloud. Particularly in healthcare environments, latency of the network and services need to follow some standards, for example IEEE 1073 [50]. This clearly shows the demand for additional layer between the nodes and the cloud to enhance mobility.

A simplified view of how smart gateways are used in the fog layer to assist nodes during mobility from one geographic location to another domain is shown in Fig. 5. Device discovery helps in identifying a new node entering to the domain under primary control of the associated gateway. Taking a single node that intends to move from gateway #1 to gateway #6, in the path shown by the arrow, each gateway utilizes device discovery and mobility support module to provide uninterrupted service for the node. The initial configuration shows that each gateway manages a set of nodes. In [51], we show in detail how to provide device discovery and mobility at the edge, in particular for interoperability purposes. As a node moves, it receives a broadcast message from the gateways regarding its identity. When the node receives the broadcast message it replies with a discovery request to the respective gateway, which is processed by the device discovery and mobility support module in the gateway. At the fog layer, the two gateways (the source and destination) exchange information regarding the profile of the node and handle the handover process.

4.8. Energy efficiency for sensor nodes

Data processing at sensor nodes has several drawbacks due to sensor nodes' resource constraints. In some cases, complex algorithms might be successfully executed at sensor nodes, however at a high cost of energy. Therefore, shifting selective signal processing tasks from sensor nodes to smart gateways at the fog layer can be an effective solution to address the aforementioned issues, particularly when gateways are not battery-powered.

Recently, several approaches [52,53] have focused on providing energy efficiency for Health-IoT applications. In [52], Otto et al. perform real-time signal processing on sensor nodes while Gia et al. [53] utilize a low power transmission protocol to save transmission energy for sensor nodes. Although such techniques enhance energy efficiency of sensor nodes, a considerable amount of energy can be still saved via fog computing by outsourcing some loads from sensor nodes to smart gateways.

4.9. Latency

For a continuous 24/7 remote health monitoring system, rapid decision making and agile responses are essential for several acute diseases and emergencies, where data processing and transmission time should be minimized. In cloud computing where raw data is transferred from sensor nodes to cloud, if network condition is unpredictable, it may cause uncertainty to response latencies. The situation is more critical when streaming-based data processing is needed (e.g., signal processing on ECG or EEG signals). Comparatively, implementing high priority data analytics in distributed smart gateways and making critical and time-sensitive decisions within the local network make the system more robust and predictable. The processed data can be then transmitted to the cloud for storage and further analysis. Moreover, in a large scale sensor network, local signal processing at the fog layer can minimize the traffic between gateways and the cloud.

5. Demonstration and evaluation

To demonstrate our hypothesis, *an enhanced healthcare IoT system realized through the use of a network of smart e-Health gateways at the fog layer*, a set of demonstrations and evaluations are presented in this section. It starts with demonstrators and evaluations that show the characteristics and performance of the smart gateway and the benefits they provide. These demonstrations show the behavior of a single gateway in a standalone condition as well as the collaborative benefits which can be provided via a layer of networked smart gateways. A medical case study follows the features of the gateways to demonstrate a medical early warning scenario, where a network of gateways forming a fog layer.

To begin with the demonstrations, we present the architecture of our demonstration system and evaluate the system characteristics including higher-level functions developed on the gateway. In our architecture, the system implementation is divided into three major phases: node implementation, networked smart gateways implementation, and back-end and user interface implementation. Our Smart e-Health Gateway, called UT-GATE, collaborates with sensor nodes, other smart gateways, remote server, and clients. The implemented system architecture is shown in Fig. 6. To demonstrate the device and protocol level interoperability of UT-GATE, we have implemented different network topologies, such as mesh and star topologies, using several wireless sensor technologies, like 6LoWPAN, Wi-Fi and Bluetooth, so that each sensor in each subnetwork utilizes different platform but works in harmony with UT-GATE. The tunneling interface module in UT-GATE is used by the Mesh-based 6LoWPAN network to interoperate with the rest of the system i.e., to tunnel between 6LoWPAN and IPv4/IPv6 protocols. The non-IP based translation module supports the star-based Bluetooth network to interoperate with the IP-based system, i.e., to translate between Bluetooth and IPv4/IPv6 protocols.

As shown in Fig. 7, UT-GATE is constructed from combination of Pandaboard [54] and Texas Instruments (TI) SmartRF06 board integrated with CC2538 module [55] and MOD-ENC28J60 Ethernet Module [56]. The Pandaboard is low-power, low-cost single-board computer development platform based on TI OMAP4430 system-on-chip following OMAP architecture and fabricated using 45 nm technology for providing high-performance [54]. OMAP4430 processor is composed of microprocessor unit (MPU) subsystem including dual-core ARM Cortex-A9 cores with symmetric multiprocessing at up to 1.2 GHz each. Pandaboard can support different operating systems such as Windows CE, WinMobile, Symbian, Linux, and Palm, and integrate on-chip memory and external memory interfaces, and support memory management and connecting peripherals. In our implementation, 8 GB external memory added to the Pandaboard and powered by Ubuntu operating system which allows to control devices and services such as local storage and notification. Furthermore, it supports different network interfaces such as 802.11 b/g/n (based on WiLink 6.0), Bluetooth v2.1 + Enhanced Data Rate (BDR) (based on WiLink 6.0), and Onboard 10/100 Ethernet.

Bluetooth sink node is created by configuring Bluetooth module (WiLink 6.0) while Bluetooth sensor node is constructed by the combination of Bluetooth module, Arduino Due and analog front-end (AFE) devices. E-health data collected from the AFE devices is sent to the Arduino Due through SPI connection after digitization and then the data is transferred to the sink node through the Bluetooth module. All operations in the Arduino Due are executed on FreeRTOS [57], an open source real-time operating system.

A Wi-Fi network is built by the combination of a sink node constructed by configuring the Wi-Fi Module (WiLink 6.0) in the Pandaboard and a Wi-Fi sensor node formed by using an RTX4140 module [58] which includes micro controller unit (EMF32), Wi-Fi module (Atheros) and the AFE devices. Similar to the Bluetooth module, AFE devices are connected to the RTX module through SPI connection. Unlike Bluetooth, a UDP client running on RTXOS [59], an operating system for the RTX module, sends the data to the Wi-Fi sink node.

The SmartRF06 along with the CC2538 module form the sink node for the 6LoWPAN network collect data from other 6LoWPAN nodes and forward to the Pandaboard through the Ethernet port. Wi-Fi is used for data exchange between the Pandaboard and the remote server.

The SmartRF06 board is the motherboard for low-power RF ARM Cortex M3 based SoCs from Texas Instruments [55]. As shown in Fig. 7, the board is plugged to the CC2538 module to collect the data from the 6LoWPAN subnetwork and then send the data to the Pandaboard via the plugged MOD-ENC28J60 Ethernet Module. To enable communication between the 6LoWPAN nodes and UT-GATE, we used RPL (IPv6 Routing Protocol for Low-power and Lossy Networks) implementation in Contiki OS [60] which is an open source operating system focusing on low-power IoT devices.

The sensor nodes receive Electrocardiogram (ECG), Electroencephalogram (EEG), and Electromyogram (EMG) digital signals through 2 SPI connectors from AFE devices (i.e., TI ADS1292 [61]

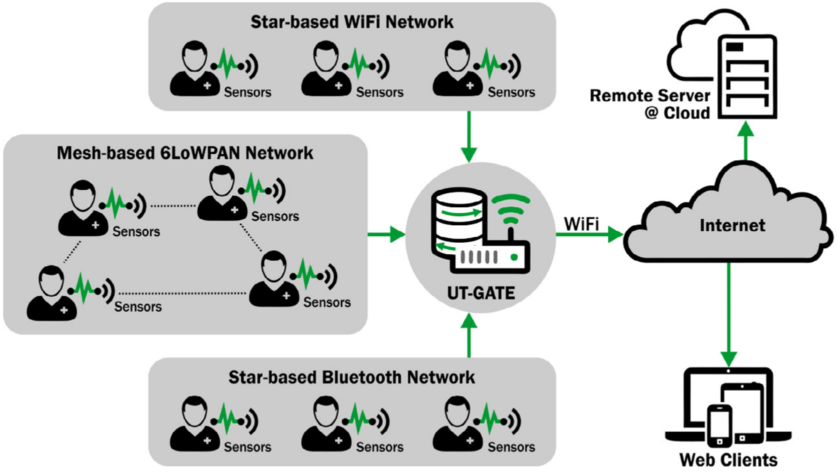


Fig. 6. Architecture overview.

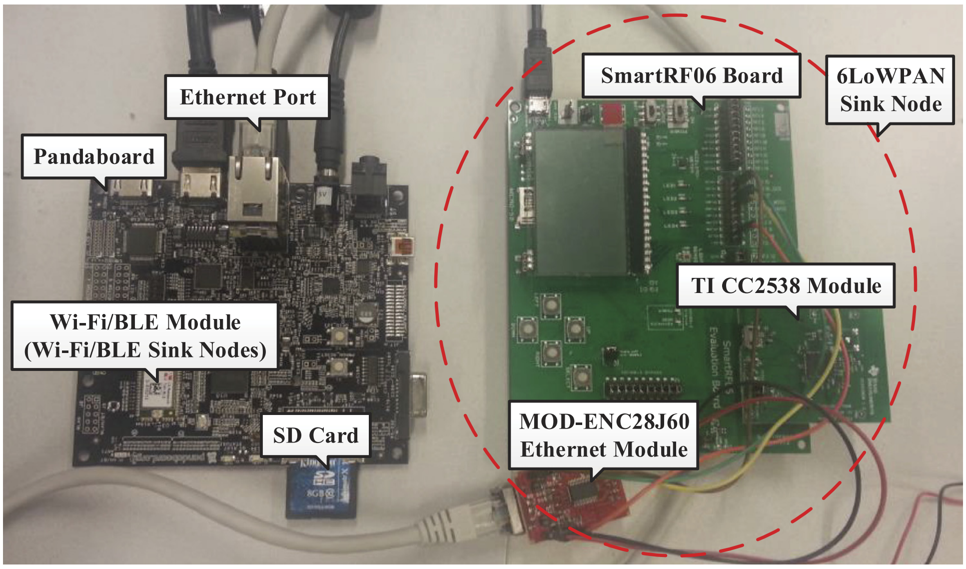


Fig. 7. UT-GATE: Our demonstrator of a smart e-Health gateway.

Table 1
Hardware specification of sensor nodes and UT-GATE.

Device	Micro-controller	Flash (KB)	RAM (KB)	EEPROM (KB)	Clock (MHz)	Voltage (V)
Zigduino R2	ATMega 128	128	16	4	16	3.3
Arduino Uno R3	ATMega 328P	32	2	1	16	5
Arduino Mega	ATMega 1280	128	8	4	16	5
Arduino Due	ARM CortexM3	512	86	–	84	3.3
Zolertia Z1	MSP430	92	8	–	16	3.3
TI-CC2538	ARM Cortex M3	Up to 512	32	–	32	3.3
Pandaboard	Dual-core ARM Cortex-A9	Up to 32 000	1000	–	1200	5

Table 2
Power consumption of sensor nodes when transmitting at 8.7 kbps.

Communication type	Current (mA)	Voltage (V)	Power consumption (mW)
6LoWPAN node	24.6	3.3	81.2
Wi-Fi node	114	3.3	376.2
Bluetooth 2.0 node	56.9	3.3	187.7
BLE node	31.6	3.3	104.4

and TI ADS1298 [62]). The analog front-end devices get analog values from electrodes and perform analog to digital conversion. In SPI connection between a node and ADS1292, a node (i.e., CC2538 module) acts as the master while the ADS129x module act as the slave. The specification and power consumption of the sensor nodes are respectively shown in Tables 1 and 2.

In our IoT-based health monitoring system, the remote server is responsible for handling client requests by providing the requested

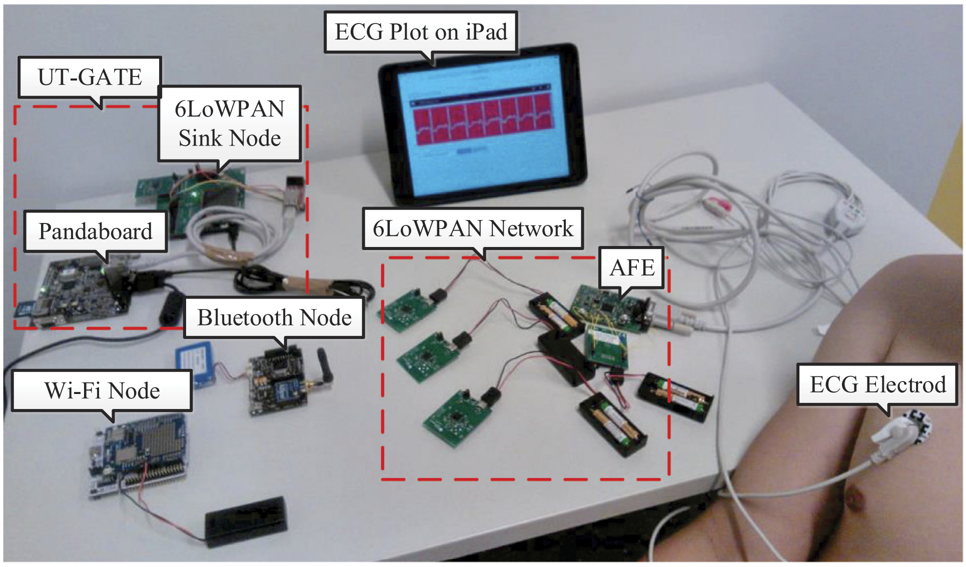


Fig. 8. IoT-based health monitoring system demonstration.

Table 3
Sensing to actuation latency comparison for local fog based vs. remote cloud based scenarios.

Latency of the sensing-to-actuation loop	using Wi-Fi (ms)	using BLE (ms)
Fog-based (locally via UT-GATE)	21	33
Cloud-based (remotely via the Cloud)	161	176

data along with graphical user interface. For the remote server over the cloud implementation, we used the free service provided by "heliohost.org" including MySQL server with remote access facility. The database along with tables are created in the server and the server side scripts are developed in PHP.

As shown in Fig. 8, we have implemented an IoT-based health monitoring system including 6LoWPAN, Wi-Fi and Bluetooth sub-networks to practically demonstrate the features of UT-GATE. The currently implemented functionalities of UT-GATE such as data compression, data fusion, WebSocket server and local storage are discussed in detail in the following subsections.

Table 3 shows benefits of the local fog-based decision making compared to that of the centralized cloud computing scenario in terms of latency. As can be seen from the table, the latency for the sense–decide–act loop for the local scenario is considerably lower then its counterparts making it a viable option for real-time streaming or actuation purposes.

5.1. Compression at UT-GATE

Applying data compression into remote e-Health monitoring systems helps reducing the size of data transmitted over a network. The amount of compressed data varies depending on a particular data compression method. Some methods can achieve a high compression ratio, for example 8:1, 9:1 or even higher, while others cannot reach to these ratios. The key requirement when applying data compression in real-time monitoring systems is computation time of the compression and decompression process because maximum latency for ECG, EMG, EEG signals have to be less than 500ms, according to IEEE 1073 [63]. With the purpose of fulfilling this latency requirement, a LZW [64] algorithm is implemented. LZW is a lossless data compression method which provides rapid compression and decompression. In addition, balanced execution time for compression and decompression in this method helps keeping harmony between input and output. In order to provide a general view of data compression in a real-time e-Health monitoring system, data compression is applied at both sensor nodes and gateways in distinct processes. Even though these compression algorithms are among the least expensive methods available in the literature in terms of minimum hardware requirements, it is still not feasible nor efficient to implement them at sensor nodes. Therefore, data compression is operated at the fog layer as a unit in the smart e-Health gateway.

The LZW algorithm reacts differently to various data sizes. It is more efficient with a higher input size in terms of computation time. For example, when the size of input data increases 10 times, computation time increases about 8 times. However, the computation cost in terms of latency rapidly grows by increasing the data size. Therefore, the data size needs to be carefully chosen to meet the real-time requirements of e-Health data.

Table 4 shows the time required for compression and decompression and compressed, and data size with the number of connected sensor nodes to UT-GATE varying from 1 to 50. Data is collected from sensor nodes having 8 channels for EMG with sampling rate of 1000 samples/second. In addition, other signals including environmental data is obtained resulting to a sample size of approximately 70B. When 120 samples are used as an input, the compression takes around 3.1ms. Transmission time for sending 70B data using 6LoWPAN technology from a sensor node to a gateway, is around 40 μs when transmission rate is 250 kbps. This shows that the transmission time is negligible compared to the compression/decompression time, and total required time for compression, transmission, and decompression is tolerable for real-time transmission of e-Health data. It can be also observed that the smart gateway at the fog has the ability to serve a large number of sensor nodes with still reasonable compression and decompression time.

5.2. Benefits of data processing at UT-GATE

In an IoT-based health monitoring system, as presented in Fig. 3, there are options to apply signal processing at each layer of the

Table 4
Compression results at UT-GATE and latency reduction.

Number of senors nodes connected to UT-GATE	1	2	5	10	50
Number of analog channels	8	8	8	8	8
Data size (120 samples) (B)	8400	16 800	42 000	84 000	420 000
Compressed data size (B)	808	1597	3893	7696	38333
Compression time (ms)	3.1	4.4	9.2	16.6	73.0
Decompression time (ms)	3.3	4.6	11.3	23.0	83.9
Total time of comp., tran., and decomp. (ms)	12.86	21.77	51.64	101.16	463.5
Transmission time of non-processed data (ms)	67.2	134.4	336	672	3360
Total latency reduction (%)	80.8	83.8	84.6	84.8	86.1

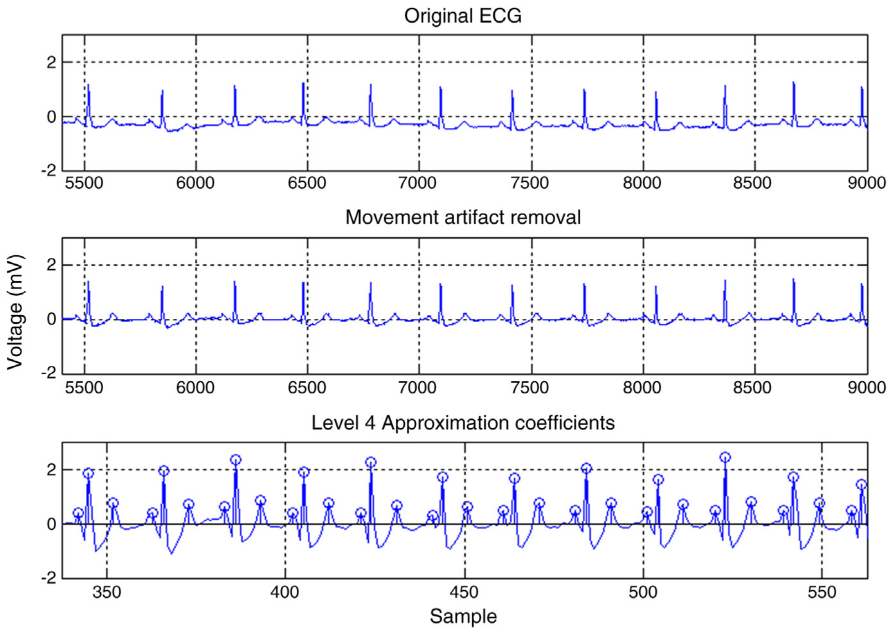


Fig. 9. ECG processing implementation.

system architecture. As introduced in Sections 4.8 and 4.9, fog layer has adequate computational resource and can bring benefits to both sensor node energy efficiency and data transmission time in the system.

As a widely used function in health monitoring applications, we chose to implement ECG signal processing for movement artifact removal and feature extraction (i.e., heart rate calculation from *R* to *R* interval, *P* wave, and *T* wave). We implement this function under three different scenarios: (i) on a sensor node, where the processed data and extracted features are sent to the cloud through UT-GATE, (ii) on a UT-GATE where the raw data is received from sensor nodes and the results are sent to a cloud (fog-assisted cloud computing), and (iii) on the cloud, where raw data is passed from two prior layers. For all these scenarios, we measured energy consumption of sensor node, the size of samples transmitting from UT-GATE to cloud, and the latency of data delivery between them.

We use MIT-BIT Arrhythmia database [65] which includes abundant ECG data sources sampled at 360 samples per second in 11-bit over 10 mV. Data is pre-stored in sensor node or UT-GATE before being processed. The ECG threshold-based feature extraction algorithm is applied after moving average filter and four level discrete wavelet transformation with Daubechies 4 wavelet. Discrete wavelet transformation is a time-frequency method which is suitable for non-stationary signal processing. With each level of transformation, the signal is divided into approximation coefficients and detail coefficients in even data sizes which represent low and high part of the signal respectively. ECG features lie in the low frequency part, so approximation coefficients are kept in each level for the next level transformation. ECG signal processing and peak detection are shown in Fig. 9.

To access the overheads of local processing on sensor nodes in terms of energy consumption, application execution time and energy consumption are measured from Arduino Due acting as a sensor node. Sensor node in real-time application needs to accumulate data for a few seconds before processing. In our case, the collection time for 1000 samples is 2.78 s. A low-cost Wi-Fi module ESP8266 is utilized to transmit data from the sensor node to the UT-GATE. The energy consumption calculation and comparison for the above mentioned three scenarios are presented in Table 5. As can be seen from the table, by outsourcing the processing burden from the sensor node to the smart gateway, about 55.7% of energy is saved thanks to fog computing.

As shown in Table 6, fog computing considerably reduces the data transmission to the cloud due to local data processing. For every 1000 raw samples, it provides 74.1% reduction in sample size, due to sample downsizing through wavelet decomposition in gateway. Meanwhile, essential information about ECG, including waveform and extracted features, are kept and sent to cloud, instead of transmitting raw samples. Consequently, data transmission time from gateway to cloud is reduced, especially for large data sets. The latency reduction when transmitting 240 KB raw samples of data between gateways and the cloud under different Wi-Fi network conditions are presented in Table 7. It can be observed from the table that fog computing can reduce the communication traffic in particular in high-traffic network conditions.

5.3. Local storage, notification, and security at UT-GATE

The UDP server running at the gateway on port 5700 receives data modules for the 6LoWPAN network and under RTXOS on

Table 5
Providing energy efficiency for sensor nodes.

	Time	Current	Energy
Processing at sensor node (collection + execution)	2.78 s + 101 ms	10.1 mA + 106.8 mA	127.34 mJ
Process at UT-GATE/cloud (transmitting raw data)	95 ms	180 mA	56.34 mJ

Table 6
Number of transmitted samples from fog to cloud.

	Processing at sensor node or UT-GATE	Processing at cloud	Improvement (%)
No. of samples	259	1000	74.1

Table 7
Latency reduction between gateways and cloud server for 240 KB of raw samples.

Network condition	Data rate (Mbits/s)	Raw samples trans. time (ms)	Raw samples proc. time + trans. time of processed samples (ms)	Latency reduction (%)
Light load	18	106.6	96.3 + 6.6	3.5
Medium load	12	152.2	96.3 + 9.5	30.5
Heavy load	9	213.3	96.3 + 13.5	48.5

Table 8
XML status code and description.

Code	Description
0	Invalid request or Error
1	Notification — {total in number}
2	No new notification

RTX Wi-Fi modules for the Wi-Fi network. Similarly, the Bluetooth nodes send data to the Bluetooth module in UT-GATE. Received data is processed and stored in the local repository apart from forwarding the same data to the remote server. We have implemented a local repository on UT-GATE using MySQL database which offers several engines. Federated engine is one such kind of engine used to create references to the tables in the remote server without the requirement of database mirroring or replication. The tables created with federated engine on local repository will hold of the same structure and record as in the remote server. Whenever new rule is added or existing rule is updated in remote server, the same information is available in the local repository which helps to give least priority to synchronization process on rules. During processing, if the received data does not conform to the rules, a notification will be logged in the repository. The notification table will be populated directly from local repository and the notification mechanism configured in the remote server will act accordingly.

The gateway purges the locally stored information in repository, which is received 30 min earlier ensuring that data synchronization with the remote server has been successfully completed. If the connection with the remote server is not available, it will store the data as long as possible and begin to delete old data to accommodate new data, if it runs out of memory. During network unavailability, UT-GATE can also act as a local web server by handling the client application's request along with notifications taking its operation to a higher level. While acting as local web server, it will send responses either in XML or JSON format as requested, leaving the user interface rendering at the client-end by utilizing the client resources efficiently to minimize its resource usage.

Notification mechanism configured in the gateway can be used on permanent basis parallel to remote server or whenever the connection to the remote server is unavailable. For notification implementation, we have developed an Android application which communicates with UT-GATE over Wi-Fi on demand. If new notification exists, the gateway communicates with the respective node with a message along with node ID and timestamp in XML format.

The XML format has two sections: header and content. The header section contains the status of the current request in the form of code and description. Currently, gateway responds with 3 status codes; the codes along with its description are given in Table 8. When mobile application receives response, at first it will check the code in status header and if status returns '1', it will proceed with content section, otherwise it will deliver the status header description as message.

As mentioned before, UT-GATE has been powered by the Ubuntu operating system which comes with a firewall called Uncomplicated Firewall (UFW) which is used to restrict the accessibility of protocols and ports. With proper configuration, the gateway can be tuned to achieve certain degree of security level. For our implementation, we blocked all ports and protocols except TCP and UDP over ports 80, 443, 3306 and 5700.

5.4. *WebSocket server on UT-GATE*

An embedded WebSocket server was implemented on UT-GATE using the Tornado non-blocking Web server framework for Python. The server receives data as a UDP server directly from the sensor nodes functioning as a UDP client. Another configuration involves receiving the signal from the MySQL database configured to serve as a streaming database. The benefit of this approach is multi-user support for the WebSocket server since the signal is always stored and can be retrieved many times. On the client side, a WebSocket enabled browser accesses an HTML page hosted at the gateway that offers the JavaScript interface and necessary parameters to establish the two-way asynchronous WebSocket link between the browser and the gateway. The ECG signal is buffered to 400 samples and sent as WebSocket messages of 800 bytes each averaging a data rate of 1.1 KB/s. In our LAN setup, it takes 32 ms for the Web client to receive the packet and render the continuous chart. The buffer size can be decreased to lower the latency at the expense of higher processing overhead for the Web Client. A JavaScript client plots the near real-time chart and a set of commands is implemented to control transmission start-stop. Future work includes the expansion of the command set into a complete API for gateway management and a generic library capable of listening to different transport layer protocol sockets for easy interoperability of variety of nodes with different protocols.

5.5. *Medical early warning scores: A case study*

Early Warning System is a guide tool in hospitals for estimating the degree of illness and predicting the risk of deterioration to reduce the complications and prevent intensive care unit admission. It is based on recording patient's medical parameters periodically to find abnormal signs. This guide system works based on the fact

Table 9
A typical early warning score model [70].

Physiological parameters	3	2	1	0	1	2	3
Respiration rate (breaths/min)	≤8		9-10	12-20		21-24	≥25
Oxygen saturation (%)	≤ 91	92-93	94-95	≥96			
Temperature (°C)	≤35.0		35.1-36.0	36.1-38.0	38.1-39.0	≥39.1	
Systolic BP (mmHg)	≤90	91-100	101-110	111-219			≥220
Heart rate (beats/min)	≤40		41-50	51-90	91-110	111-130	≥131
Level of consciousness				A*			V,P or U*

that clinical deterioration is a visible pattern in patient's vital sign up to 24 h before deterioration happens [66,67,68]. The system is designed based on a method called Early Warning Score (EWS) which calculates a total score using a set of values related to some medical parameter. This score reflects the health status of a patient with respect to her/his vital signs. To find the score, a nurse should measure and record the vital signs of a patient in an observation chart. The nurse marks a vital sign with a score, based on its value in its range. A higher score means more abnormalities of a specific vital sign and the sum of all scores indicates the overall health status of a patient [69]. Table 9 shows a typical EWS model.

Considering a patient's final calculated score, medical staff can modify the therapy orders and recording intervals. Such a scoring method for medical early warning was presented for first time in 1997 [71]. Initial versions of the EWS were using five vital signs: heart pulse, respiration rate, blood pressure, body temperature, and blood Oxygen level (SpO2). Several years after first implementations of this method in hospitals, some enhancements were proposed and applied to the original EWS algorithm. Modified early warning system (MEWS) [72], standardized early warning system (SEWS) [73], and a national early warning system in UK (NEWS) [74] are some examples. The main difference of these enhanced methods is the number of parameters they use to calculate the final score. Also they customize the threshold values of vital signs based on the average property value in the target country.

Despite many benefits offered by EWS in hospitals in terms of reduced mortality rate and healthcare costs, some researchers have reported false diagnosis and errors mainly caused by inaccurate records. The necessity of implementing an accurate early warning system drives hospitals to move towards automatic electronic solutions [75].

The objective of this case study is to demonstrate an in-home early warning system using the discussed system architecture and the concepts of fog computing. Therefore, we implemented an in-home EWS system where patients are supposed to be provided with medically acceptable services and environments. To this end, it is essential to monitor environmental properties as well as patient's activities to consider their effect on patient's vital signs. The components in our proposed system comprises three layers shown in Fig. 10.

At the first layer, we implemented a network of sensors which are divided into three groups. The first and main group includes medical sensors to monitor vital signs such as heart rate, respiration rate, body temperature, blood pressure, blood oxygen level, and ECG. The second group of sensors contains environmental sensors for recording light, temperature, and humidity of the room. Activity sensors are in the last group for recording the patterns of movements, posture, and total daily steps of the patient.

At the second layer, data received from sensor network is handled by UT-GATE via wireless communication. The gateway receives data from several type of medical sensors via a UDP server implemented by Node.js. This UDP server stores data for each sensor in separate files considering the information of patient whom the data is coming from. Another service running on UT-GATE with Apache server reads data from locally created files for further processings using a Python script. As properties of the collected data are not similar, an adaptation is needed to unify the data structure. The frequency of data collection is not the same for all the sensors, ranging from 250 samples per second (e.g., ECG) to about 5 samples per day (e.g., step counts). Communication protocols also differs for different sensors (e.g., Bluetooth and Wi-Fi with UDP and TCP transmission protocols).

Node.js UDP server is responsible for handling different data transmission protocols while continuously receiving data from sensor network. In the gateway, using Python, preliminary data analysis is implemented to detect critical conditions before performing detailed analysis at the cloud. UT-GATE first filters the data for noise reduction. A bandpass filter (0.5–100 Hz) with Finite Impulse Response (FIR) is implemented in the gateway to reduce noises from the incoming ECG signal (250 samples per second). Then, heart rate data is extracted from the signal considering RR intervals. Fig. 11 demonstrates a window of raw and filtered ECG data.

Moreover, the gateway applies data fusion on the sensory data. In this regard, two heart rate signals from the monitored person are collected using two different devices. Due to the presence of inevitable noises (e.g., movement's noises) on the acquired signals, two obtained data may not be identical during the monitoring. To reduce the noise impact, first, outliers and meaningless data are removed using an anomaly detection method. Considering a threshold for the minimum value of heart rate, zero values are removed. In our case, outliers in heart rate data sources are mainly because of improper probe connections, often due to loose connection or low conductivity between the probe and the patient's skin (low moisture). Second, a weighted average is implemented for two sources values to achieve more reliable heart rate (see Fig. 12). The weight of each data set is determined by the sensor accuracy mentioned in the sensor's datasheet.

After the aforementioned steps, real-time EWS calculation, based on the constraints and rules shown in Table 9, are performed at the fog layer. We believe EWS is a proper case study for fog computing in healthcare, as it demands reliability, rapid response, and real-time processing, and deals with heterogeneous sensory data which calls for pre-processing. To calculate the EWS score, a window of collected vital signs are chosen. Using a rule-based system defined based on Table 9, the score is calculated at UT-GATE. In our use case, a 35 years old male subject (BMI = 28.3) is monitored for 8 h. Fig. 13 shows one minute of monitoring in which the subject encountered a sudden variation in his vital signs. Fig. 14 also demonstrates the calculated score. The increasing score indicates the higher probability of the patient's health deterioration for this specific window. Therefore, in this case, the medical experts are notified about the critical state of the patient for further considerations.

In parallel with the real-time analysis and notification, the gateway applies compression and encryption to the filtered and processed data, sends feedback and notifications to the sensor network and cloud respectively, and stores the data in UT-GATE's local storage (see Fig. 10).

We compress processed data in the gateway to have a backup for the situation when Internet connectivity drops off. The compression method we used here is different with the method described earlier for the real-time data compression during transmissions. In this case study, we use *tar* method to create a file

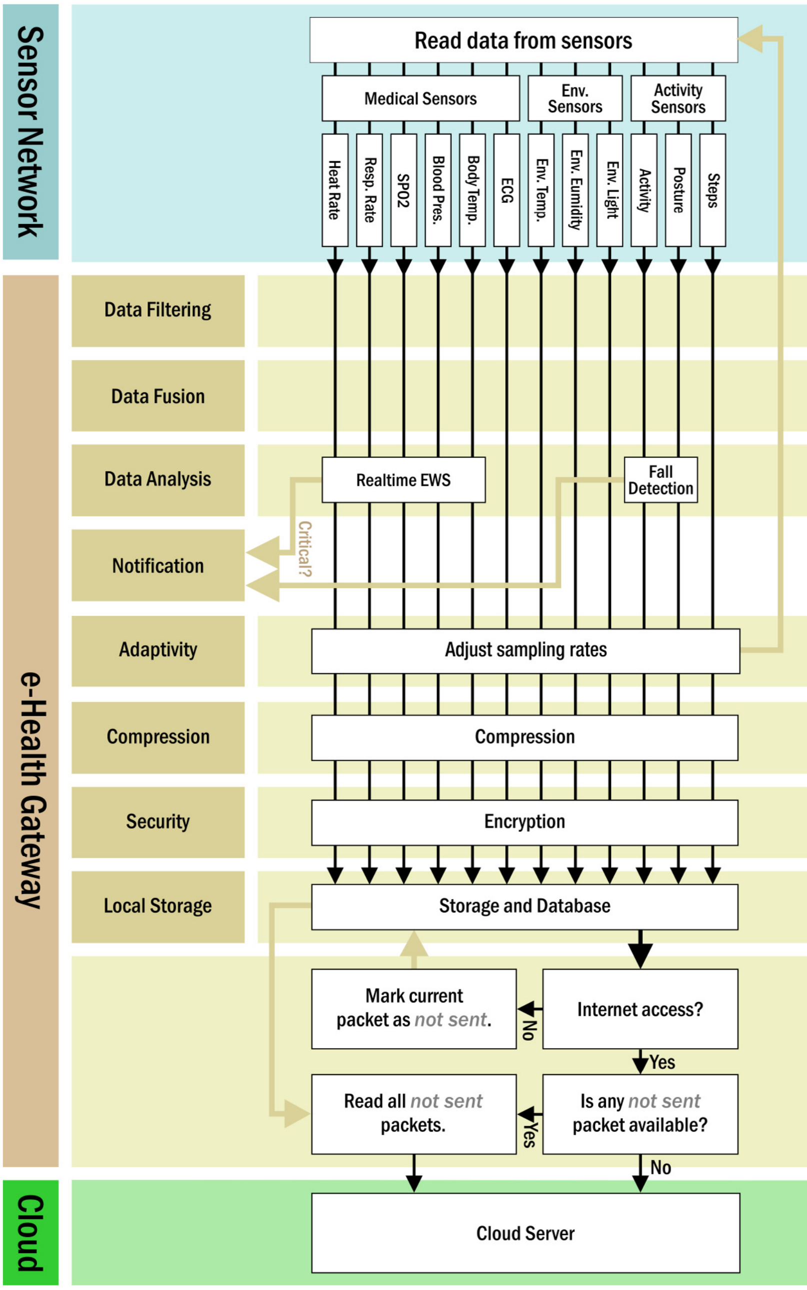


Fig. 10. Fog-based EWS System services, components, and data flow.

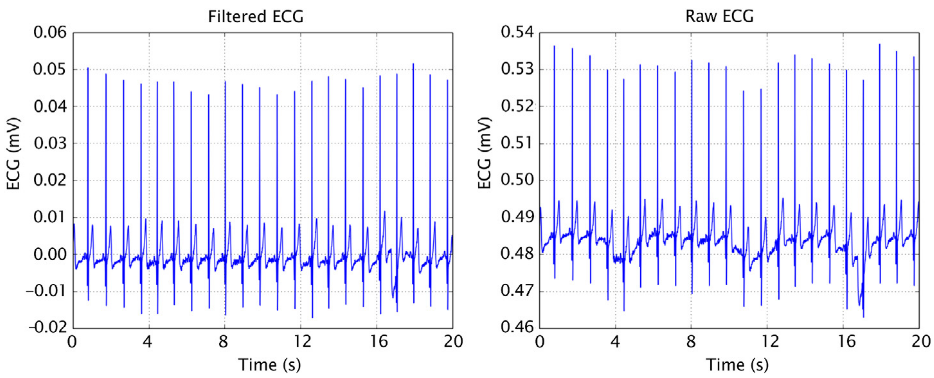


Fig. 11. Raw and filtered ECG data.

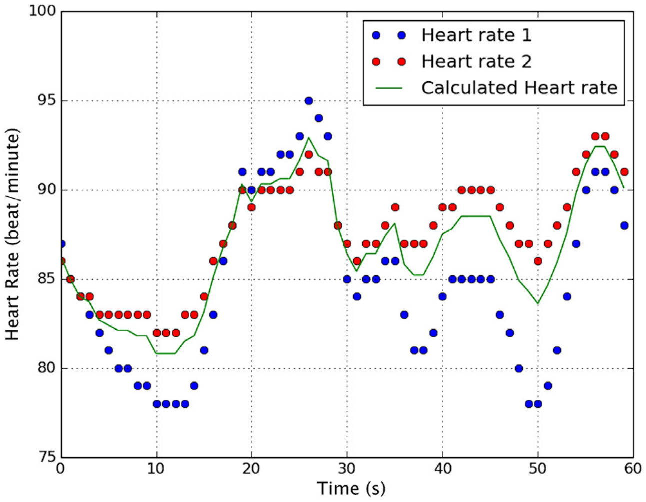


Fig. 12. Heart rate data fusion.

as temporary location for collected data and use *tar.gz* method to compress temporary file when the size of file reaches to a certain value (e.g., 500 KBytes). The effect of temporary file size on the compression ratio is shown in Fig. 15. The larger the size of temporary file, the higher the compression ratio would be. However, the compression ratio shows an insignificant improvement for temporary files larger than 500 KByte.

To keep stored data secure, we use an asymmetric encryption method using *Crypto* library [76] in Python. All the compressed files on the gateway are encrypted with a public key and only the data collector service in the cloud has the private key to decrypt the files.

The gateway adjusts the sampling rate according to the calculated EWS score. When the score increases, the gateway considers priority for patient's health situation by increasing the sensor's sampling rate to track the momentary changes more accurately. On the other hand, when the score is zero or within the normal range, the sampling rate is reduced by sending feedback to sensor network in order to improve the energy efficiency.

Local storage consists of a file storage and database to keep the properties and indexes of the files. In the last phase, the gateway checks the availability of the Internet connection and sends the data to the could server. In case of connectivity issues, it tags the unsent data to be sent in future. A process in local storage service checks and synchronizes the stored data with the cloud server and removes old and duplicated files from the storage, and deletes their indexes from the database.

The third layer of our in-home early warning system consists of a cloud server and user interface for patients, caregivers, and

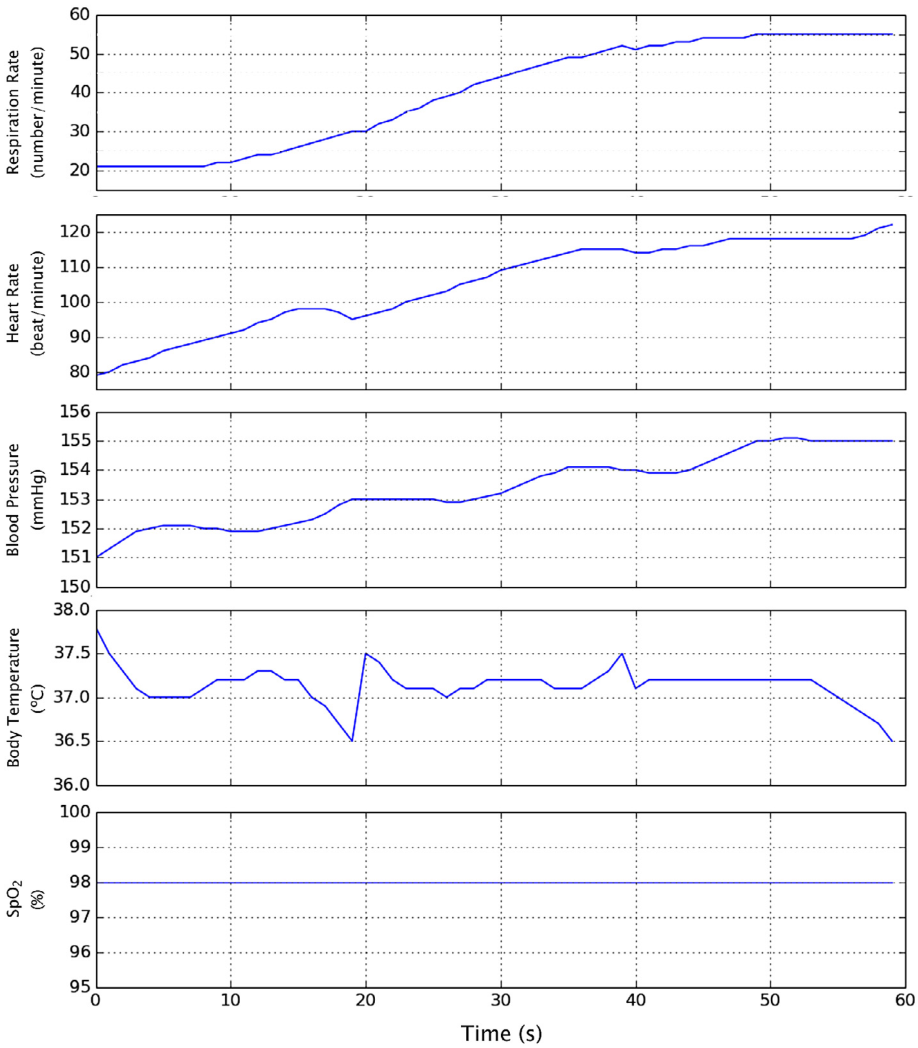


Fig. 13. Vital signs.

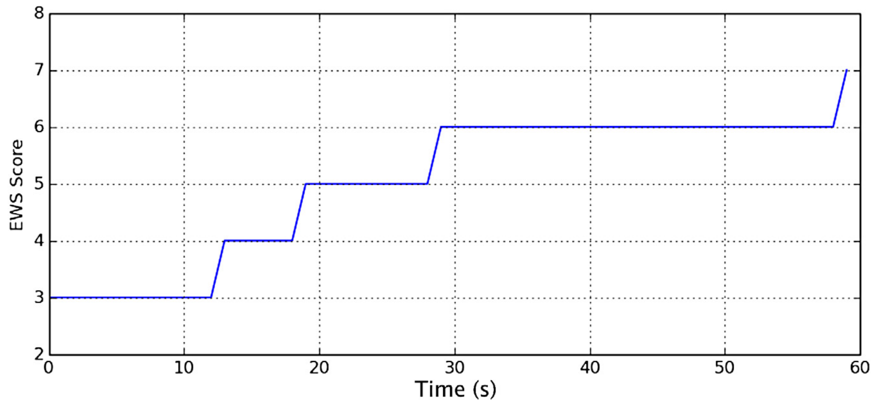


Fig. 14. Calculated EWS score.

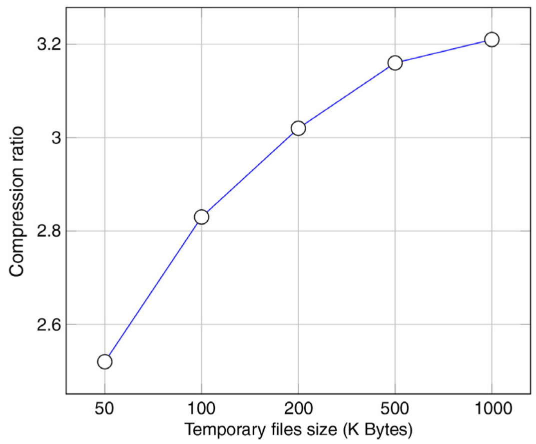


Fig. 15. The compression ratio for different temporary file size.

medical experts. At this layer, the cloud server receives and records gateway's data (e.g., EWS score, vital signs and notifications) for further processing. Using the incoming data from the gateway and the patient medical history data, the server provides reports, suggestions and possible alerts for medical experts and caregivers. Cloud server also has a user interface to provide data access for health professionals. Fig. 16 shows a snapshot of the developed HTML5 web-based user interface which is used as a control panel. In addition, patients and caregivers have also access to an Android-based smartphone application to receive notes and notifications.

6. Conclusions

In this paper, the concept of fog computing and Smart e-Health Gateways in the context of Internet-of-Things based healthcare systems was presented. Smart gateways at the close proximity of sensor nodes in smart home or hospital premises can exploit their unique strategic position to tackle many challenges in IoT-based health systems such as mobility, energy efficiency, scalability, interoperability, and reliability issues. We investigated in detail a range of high level services which can be offered by smart gateways to sensors and end-users in a Geo-distributed fashion at the edge of the network (e.g., local processing, storage, notification, standardization, firewall, web services, compression, etc.). We presented a proof of concept implementation of an IoT-based remote health monitoring system which includes our demonstration of a Smart e-Health Gateway called UT-GATE. By exploiting a number of UT-GATEs, we formed an intermediary processing layer to demonstrate the fog computing concept for IoT-based healthcare systems. Our fog-assisted system was applied to a medical case study called Early Warning Scores, targeted to monitoring patients with acute illnesses. Our full system demonstration includes all the data flow processes from data acquisition at sensor nodes to the cloud and end-users.

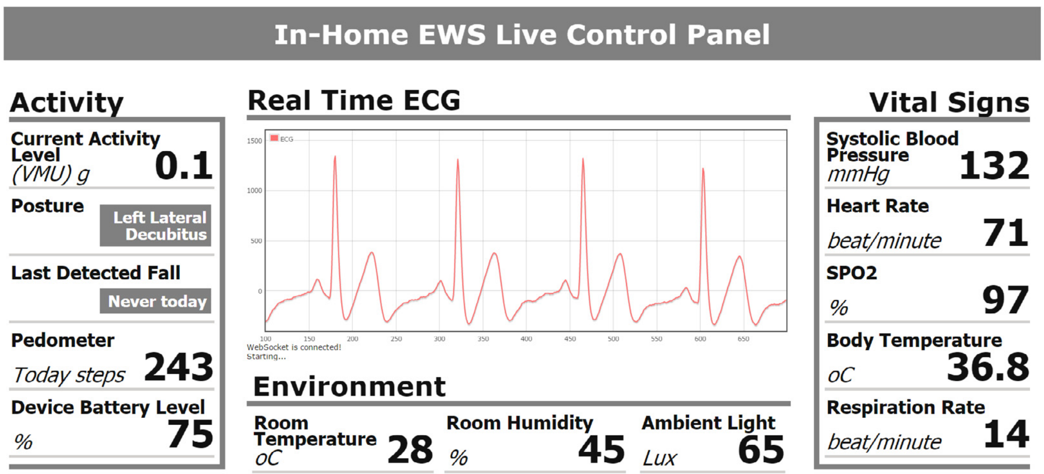


Fig. 16. Live control panel web interface.

References

[1] European Commission Information Society, Internet of things strategic research roadmap, 2009. http://www.internet-of-things-research.eu/. (Accessed 14 July 2015).

[2] European Commission Information Society, Internet of things in 2020: a roadmap for the future, 2008. http://www.iot-visitthefuture.eu. (Accessed 14 July 2015).

[3] A. Dohr, R. Modre-Opsrian, M. Drobics, D. Hayn, G. Schreier, The internet of things for ambient assisted living, in: Proceedings of the International Conference on Information Technology: New Generations, 2010, pp. 804–809.

[4] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac, Internet of things: Vision, applications and research challenges, Ad Hoc Netw. 10 (7) (2012) 1497–1516.

[5] M. Carmen Domingo, An overview of the internet of things for people with disabilities, J. Netw. Comput. Appl. 35 (2) (2012) 584–596.

[6] Hairong Yan, Li Da Xu, Zhuming Bi, Zhibo Pang, Jie Zhang, Yong Chen, An emerging technology â wearable wireless sensor networks with applications in human health condition monitoring, J. Manag. Anal. 2 (2) (2015) 121–137.

[7] Y.J. Fan, Y.H. Yin, L.D. Xu, Y. Zeng, F. Wu, IoT-based smart rehabilitation system, IEEE Trans. Ind. Inf. 10 (2) (2014) 1568–1577.

[8] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, K. Mankodiya, Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare, Elsevier Future Generation Computer Systems (2017).

[9] European Research Cluster on the Internet of Things, IoT semantic interoperability: research challenges, best practices, solutions and next steps, IERC AC4 Manifesto - ''Present and Future'', 2014.

[10] B. Xu, L.D. Xu, H. Cai, C. Xie, J. Hu, F. Bu, Ubiquitous data accessing method in IoT-based information system for emergency medical services, IEEE Trans. Ind. Inf. 10 (2) (2014) 1578–1586.

[11] L. Jiang, L.D. Xu, H. Cai, Z. Jiang, F. Bu, B. Xu, An IoT-oriented data storage framework in cloud computing platform, IEEE Trans. Ind. Inf. 10 (2) (2014) 1443–1451.

[12] C.E. Koop, R. Mosher, L. Kun, J. Geiling, E. Grigg, S. Long, C. Macedonia, R. Merrell, R. Satava, J. Rosen, Future delivery of health care: Cybercare, IEEE Eng. Med. Biol. Mag. 27 (6) (2008) 29–38.

[13] F. Bonomi, R. Milito, J. Zhu, 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.

[14] M. Aazam, E.N. Huh, Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT, in: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, 2015, pp. 687–694.

[15] M. Aazam, E.N. Huh, Fog computing and smart gateway based communication for cloud of things, in: Future Internet of Things and Cloud (FiCloud), 2014 International Conference on, 2014, pp. 464–470.

[16] A.-M. Rahmani, N.K. Thanigaivelan, Tuan Nguyen Gia, J. Granados, B. Negash, P. Liljeberg, H. Tenhunen, Smart e-Health gateway: Bringing intelligence to IoT-based ubiquitous healthcare systems, in: Proceeding of 12th Annual IEEE Consumer Communications and Networking Conference, 2015, pp. 826–834.

[17] K.A Emara, M. Abdeen, M. Hashem, A gateway-based framework for transparent interconnection between WSN and IP network, in: Proceedings of the EUROCON, 2009, pp. 1775–1780.

[18] Q. Zhu, R. Wang, Q. Chen, Y. Liu, W. Qin, IoT gateway: Bridging wireless sensor networks into internet of things, in: Proceedings of the International Conference on Embedded and Ubiquitous Computing, 2010, pp. 347–352.

[19] R. Mueller, J.S. Rellermeyer, M. Duller, G. Alonso, Demo: A generic platform for sensor network applications, in: Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems, 2007, pp. 1–3.

[20] D. Bimschas, H. Hellbrück, R. Mietz, D. Pfisterer, K. Römer, T. Teubler, Middleware for smart gateways connecting sensornets to the internet, in: Proceedings of the International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks, 2010, pp. 8–14.

[21] J.-W. Yoon, Y.-K. Ku, C.-S. Nam, D.-R. Shin, Sensor network middleware for distributed and heterogeneous environments, in: Proceedings of the International Conference on New Trends in Information and Service Science, 2009, pp. 979–982.

[22] L. Wu, Y. Xu, C. Xu, F. Wang, Plug-configure-play service-oriented gateway - for fast and easy sensor network application development, in: Proceedings of the International Conference on Sensor Networks, 2013, pp. 53–58.

[23] S. Guoqiang, C. Yanming, Z. Chao, Z. Yanxu, Design and implementation of a smart IoT gateway, in: Proceedings of the International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, 2013, pp. 720–723.

[24] J. Bian, D. Fan, J. Zhang, The new intelligent home control system based on the dynamic and intelligent gateway, in: Proceedings of the International Conference on Broadband Network and Multimedia Technology, 2011, pp. 526–530.

[25] W. Shen, Y. Xu, D. Xie, T. Zhang, A. Johansson, Smart border routers for eHealthCare wireless sensor networks, in: Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, 2011, pp. 1–4.

[26] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for VM-based cloudlets in mobile computing, IEEE Perv. Comput. 8 (4) (2009) 14–23.

[27] Vladimir Stantchev, Ahmed Barnawi, Sarfaraz Ghulam, Johannes Schubert, Gerrit Tamm, Smart items, fog and cloud computing as enablers of servitization in healthcare, Sensors Transducers 185 (2) (2015) 121.

[28] S. Nunna, A. Kousaridas, M. Ibrahim, M. Dillinger, C. Thuemmler, H. Feussner, A. Schneider, Enabling real-time context-aware collaboration through 5G and mobile edge computing, in: Proceedings of 12th International Conference on Information Technology - New Generations, 2015, pp. 601–605.

[29] Shiqiang Wang, R. Urgaonkar, Ting He, M. Zafer, K. Chan, K.K. Leung, Mobility-Induced service migration in mobile micro-clouds, in: Proceedings of Military Communications Conference, 2014, pp. 835–840.

[30] Yi Lin, B. Kemme, M. Patino-Martinez, R. Jimenez-Peris, Enhancing edge computing with database replication, in: Proceedings of 26th IEEE International Symposium on Reliable Distributed Systems, 2007, pp. 45–54.

[31] D. Laney, 3D Data Management: Controlling Data Volume, Velocity, and Variety, Technical Report, Application Delivery Strategies by META Group Inc., 2001.

[32] M. Beyer, Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data, 2016. http://www.gartner.com/newsroom/id/1731916.

[33] J. Granados, A.M. Rahmani, P. Nikander, P. Liljeberg, H. Tenhunen, Towards energy-efficient HealthCare: An internet-of-things architecture using intelligent gateways, in: Proc. of International Conference on Wireless Mobile Communication and Healthcare, 2014, pp. 279–282.

[34] M.L. Hilton, Wavelet and wavelet packet compression of electrocardiograms, IEEE Trans. Biomed. 44 (5) (1997) 394–402.

[35] Z. Lu, D. Youn Kim, W.A. Pearlman, Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm, IEEE Trans. Biomed. 47 (7) (2000) 849–856.

[36] R. Benzid, A. Messaoudi, A. Boussaad, Constrained ECG compression algorithm using the block-based discrete cosine transform, Digit. Signal Process. 18 (1) (2008) 56–64.

[37] F. Touati, R. Tabish, U-healthcare system: State-of-the-art review and challenges, J. Med. Syst. 37 (3) (2013).

[38] H.F. Durrant-Whyte, Sensor models and multisensor integration, Int. J. Robot. Res. 7 (6) (1988) 97–113.

[39] Health Level Seven Int'l. Introduction to HL7 Standards, 2012. www.hl7.org/implement/standards. (Accessed 30 July 2015).

[40] netfilter/iptables - nftables project, http://netfilter.org/projects/nftables/. (Accessed 24 July 2015).

[41] G. Kambourakiset, E. Klaoudatou, S. Gritzalis, Securing medical sensor environments: The codeblue framework case, in: Proceeding of the Second International Conference on Availability, Reliability and Security, 2007, pp. 637–643.

[42] R. Chakravorty, A programmable service architecture for mobile medical care, in: Proceeding of Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2006, pp. 1–5.

[43] J. Ko, J.H. Lim, Y. Chen, R. Musvaloiu-E, A. Terzis, G.M. Masson, T. Gao, W. Destler, L. Selavo, R.P. Dutton, MEDiSN: medical emergency detection in sensor networks, ACM Trans. Embed. Comput. Syst. 10 (1) (2010) 11:1–11:29.

[44] S.R. Moosavi, T.N. Gia, A. Rahmani, E. Nigussie, S. Virtanen, H. Tenhunen, J. Isoaho, SEA: A secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways, in: Proceeding of 6th International Conference on Ambient Systems, Networks and Technologies, 2015, pp. 452–459.

[45] S.R. Moosavi, T.N. Gia, E. Nigussie, A. Rahmani, S. Virtanen, H. Tenhunen, J. Isoaho, Session resumption-based end-to-end security for healthcare internet-of-things, in: Proceeding of IEEE International Conference on Computer and Information Technology, 2015, pp. 581–588.

[46] Z. Shelby, C. Bormann, 6LoWPAN: The Wireless Embedded Internet, Wiley Publishing, 2010.

[47] T. Bittner, M. Donnelly, S. Winter, Ontology and semantic interoperability, in: Proceeding of Large-Scale 3D Data Integration Challenges and Opportunities, 2005.

[48] Z. Shelby, C. Bormann, 6LoWPAN: The Wireless Embedded Internet, Wiley, UK, 2009.

[49] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, B. Patil, Proxy Mobile 2 IPv6, Internet Engineering Task Force, 2008.

[50] IEEE Standard for Medical Device Communication, Overview and Framework, 1996.

[51] B. Negash, A.M. Rahmani, T. Westerlund, P. Liljeberg, H. Tenhunen, LISA: Lightweight internet of things service bus architecture, Proc. Comput. Sci. 52 (2015) 436–443. The 6th International Conference on Ambient Systems,

Networks and Technologies, ANT-2015, the 5th International Conference on Sustainable Energy Information Technology, SEIT-2015.
[52] C. Otto, A. Milenković, C. Sanders, E. Jovanov, System architecture of a wireless body area sensor network for ubiquitous health monitoring, J. Mob. Multimedia 1 (4) (2006) 307–326.
[53] T. Nguyen Gia, N.K Thanigaivelan, A.M. Rahmani, T. Westerlund, P. Liljeberg, H. Tenhunen, Customizing 6LoWPAN networks towards Internet-of-Things based ubiquitous healthcare systems, in: Proceeding of NORCHIP, 2014, pp. 1–6.
[54] OMAP® 4 PandaBoard System Reference Manual, 2010. http://pandaboard.org. (Accessed 04 August 2015).
[55] SmartRF06 Evaluation Board User's Guide, 2013. http://www.ti.com/lit/ug/swru321a/swru321a.pdf. (Accessed 04 August 2015).
[56] Olimex, MOD-ENC28J60 Development Board, Users Manual, 2008. https://www.olimex.com/Products/Modules/Ethernet/MOD-ENC28J60/resources/MOD-ENC28J60.pdf. (Accessed 04 August 2015).
[57] R. Barry, Using the FreeRTOS Real Time Kernel, Microchip PIC32 Edition, FreeRTOS Tutorial Books, 2010.
[58] RTX41xx Low Power Modules, http://www.rtx.dk/RTX41xx_Modules-4024.aspx. (Accessed 04 August 2015).
[59] RTX Real-Time Operating System, http://www.keil.com/rl-arm/kernel.asp. (Accessed 04 August 2015).
[60] A. Dunkels, B. Gronvall, T. Voigt, Contiki - a lightweight and flexible operating system for tiny networked sensors, in: Proceeding of International Conference on Local Computer Networks, 2004, pp. 455–462.
[61] Texas Instruments, Low-Power, 2-Channel, 24-Bit Analog Front-End for Biopotential Measurements, 2012.
[62] Texas Instruments, ECG Implementation on the TMS320C5515 DSP Medical Development Kit (MDK) with the ADS1298 ECG-FE, 2011.
[63] IEEE standard for medical device communication, overview and framework, in: ISO/IEEE 11073 Committee, 1996.
[64] V.S. Miller, Data compression method, US4814746 A, Filing date Aug 11, 1986, Publication date Mar 21, 1989.
[65] G.B. Moody, R.G. Mark, The impact of the MIT-BIH arrhythmia database, IEEE Eng. Med. Biol. Mag. 20 (3) (2001) 45–50.
[66] J. McGaughey, F. Alderdice, R. Fowler, A. Kapila, A. Mayhew, M. Moutray, Outreach and Early Warning Systems (EWS) for the prevention of Intensive Care admission and death of critically ill adult patients on general hospital wards, Cochrane Database Syst. Rev. (3) (2007).
[67] C. Franklin, J. Mathew, Developing strategies to prevent inhospital cardiac arrest: analyzing responses of physicians and nurses in the hours before the event, Crit. Care Med. 22 (2) (1994) 244–247.
[68] R.M. Schein, N. Hazday, M. Pena, B.H. Ruben, C.L. Sprung, Clinical antecedents to in-hospital cardiopulmonary arrest, Chest 98 (6) (1990) 1388–1392.
[69] M. Odell, C. Victor, D. Oliver, Nurses' role in detecting deterioration in ward patients: systematic literature review, J. Adv. Nurs. 65 (10) (2009).
[70] D. Georgaka, M. Mparmparousi, M. Vitos, Early warning systems, Hosp. Chronicle 7 (1) (2012) 37–43.
[71] R.J.M. Morgan, F. Williams, M. Wright, An early warning scoring system for detecting developing critical illness, Clin. Intensive Care 8 (2) (1997) 100.
[72] R. Paterson, et al., Prediction of in-hospital mortality and length of stay using an early warning scoring system: clinical audit, Clin. Med. 6 (3) (2006) 281–284.
[73] G.D. Barlow, Standardised early warning scoring system, Clin. Med. 6 (4) (2006).
[74] Royal College of Physicians, National Early Warning Score (NEWS): standardising the assessment of acute-illness severity in the NHS, 2012.
[75] A. Zarabzadeh, M. O'Connell, J. O'Donoghue, T. O'Kane, S. Woodworth, J. Gallagher, S. O'Connor, F. Adam, Features of electronic early warning systems which impact clinical decision making, in: Proceeding of 25th International Symposium on Computer-Based Medical Systems, 2012, pp. 1–4.
[76] PyCrypto API Documentation, https://pythonhosted.org/pycrypto/. (Accessed 26 May 2016).

Amir M. Rahmani received his Master's degree from Department of Electrical and Computer Engineering, University of Tehran, Iran, in 2009 and Ph.D. degree from Department of Information Technology, University of Turku, Finland, in 2012. He also received his MBA jointly from Turku School of Economics and European Institute of Innovation & Technology (EIT) ICT Labs, in 2014. He is currently an EU Marie Curie Global Fellow at University of California Irvine, USA, and TU Wien, Austria. He is also adjunct professor in embedded parallel and distributed computing at University of Turku, Finland. He is the author of more than 120 peer-reviewed publications. His research interests span Self-aware Computing, Energy-efficient Many-core Systems, Runtime Resource Management, Healthcare Internet of Things, and Fog Computing.

Tuan Nguyen Gia: received his B.Sc. (Tech.) degree in Information technology from Department of Information Technology, Helsinki Metropolia University of Applied Sciences, Helsinki, Finland in 2012, and M.Sc. (Tech.) degree in Information Technology, Embedded Computing from the Department of Information Technology and Communication Systems, University of Turku, Finland in 2014. He is currently working towards his Ph.D. degree at the University of Turku, Finland. His research interests include Internet of Things (IoT), Smart Healthcare, and Medical cyber–physical system, FPGA and Wireless Body Sensor Networks.

Behailu Negash received B.Sc. degree in Electrical Engineering from Mekelle University (Ethiopia) and M.Sc. (Tech.) degree in Embedded Computing from University of Turku (Finland) in 2006 and 2015, respectively. He is currently a Ph.D. student in Embedded Electronics laboratory, IoT4Health research group, at University of Turku. His research focuses on architecture and interoperability of Internet of Things, network architecture and embedded software.

Arman Anzanpour is Ph.D. student in "IoT for Health" group at University of Turku since September, 2014. He received his Master in Biomedical Engineering from Amirkabir University of Technology, Iran and his Bachelor in Material Engineering from Ferdwosi University of Mashhad. His current research focuses on Internet of Things and smart health monitoring frameworks.

Iman Azimi received his bachelor degree in Biomedical Engineering at University of Isfahan (Iran) in 2010, and his master degree in Artificial Intelligence and Robotics at Sapienza, University of Rome (Italy) in 2014. He started his Ph.D. research in IoT4health group, department of Information Technology at University of Turku in August 2015. His current research area is intelligent health big data analytics based on Internet-of-Things.

Mingzhe Jiang is currently a Ph.D. student in IoT4Health research group in University of Turku. She received her M.Sc and B.Sc degree in Instrument Science and Technology from Harbin Institute of Technology in the year 2014 and 2012. Her current research is related with bio-signal processing and pattern recognition in Healthcare Internet of Things.

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 is a Senior University Lecturer in Embedded Electronics Laboratory and an adjunct professor in embedded computing architectures at the University of Turku, Embedded Computer Systems laboratory. During the period 2007–2009, he held an Academy of Finland researcher position. He is the author of more than 200 peer-reviewed publications, has supervised nine Ph.D. theses. Liljeberg is the leader of the Internet-of-Things for Healthcare (IoT4Health) research group.

Paper IV

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

Tuan Nguyen Gia, Amir M Rahmani, Tomi Westerlund, Pasi Liljeberg, Hannu Tenhunen

Original published in IEEE Access Journal, pages 36064-36082

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2848119, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

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, Finland, (e-mail: tunggi, tovewe, pasi.liljeberg, hannu.tenhunen@utu.fi)
[2]Department of Computer Science, University of California Irvine, USA (e-mail: amirr1@uci.edu)
[3]Institute of Computer Technology, TU Wien, Austria

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

This work is financially supported by Academy of Finland, Nokia Foundation, University of Turku Graduate School (UTUGS), Finnish Foundation for Technology Promotion (TES)

ABSTRACT Handover mechanism for mobility support in a remote real-time streaming 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, location awareness, and real-time interaction. As shown in [19], [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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2848119, IEEE Access

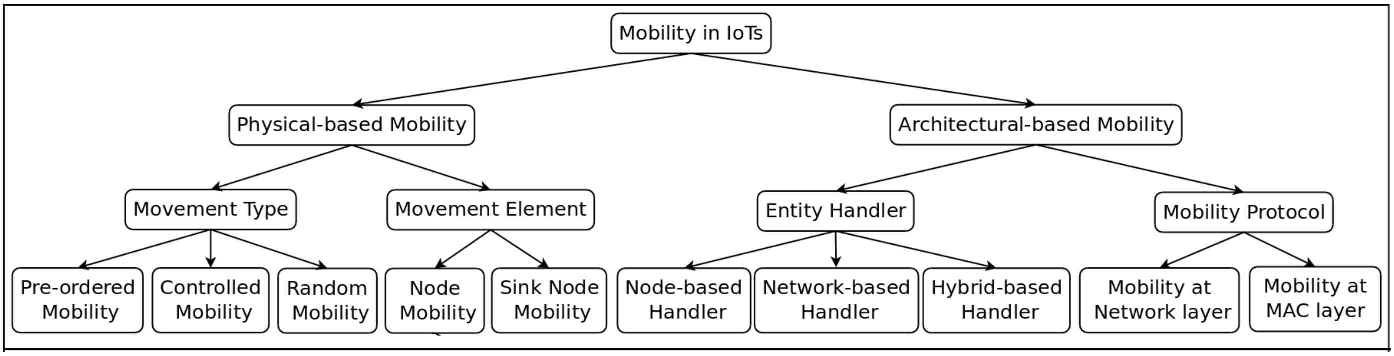


FIGURE 1. Mobility in IoT

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 *et al.* [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, 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], authors 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, Chen *et al.* [43] propose a mobility management method using follow-me Cloud-Cloudlet [44] 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 [45]. 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 [46]. In practice [47], 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 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

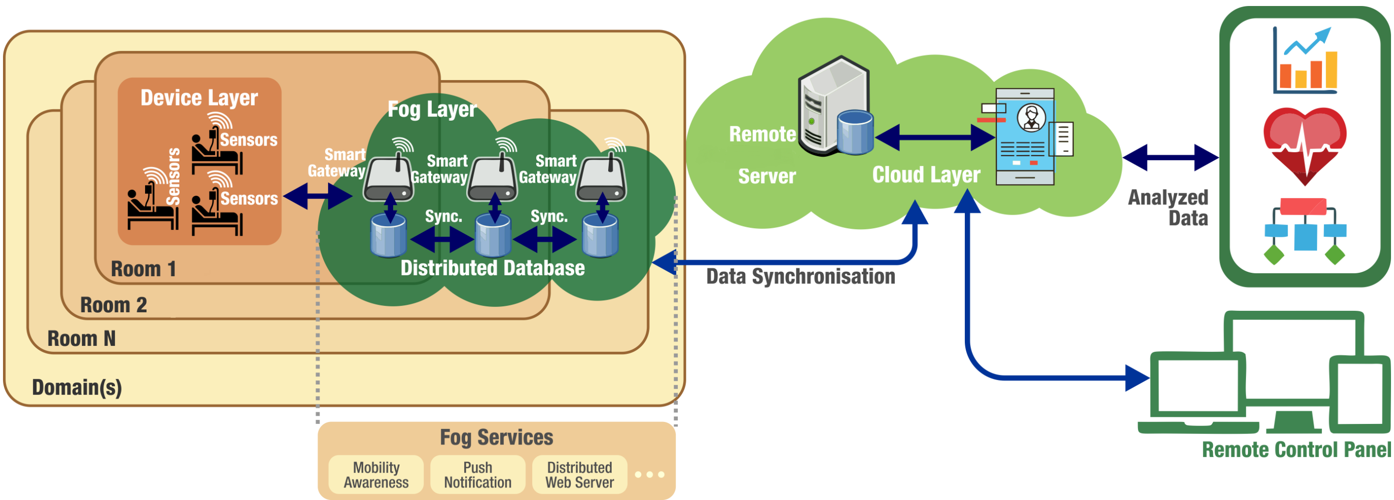


FIGURE 2. Remote real-time monitoring IoT system

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 [48], [49]. 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 [50]. 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 [50]. 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 [51].

Link Quality Indicator (LQI): In addition to RSSI, LQI can be used for handover mechanisms [52] 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 [51], [53]. 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2848119, IEEE Access

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 [54]. 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 [51].

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 [55].

Velocity is used alongside with RSSI or LQI in handover mechanisms [50]. When the velocity of a sensor node increases, a handover latency proportionally rises [56]. 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 [57] 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 [50].

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 [58], [59], angle of arrival [60], and the time of arrival [61].

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 [62]. 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.

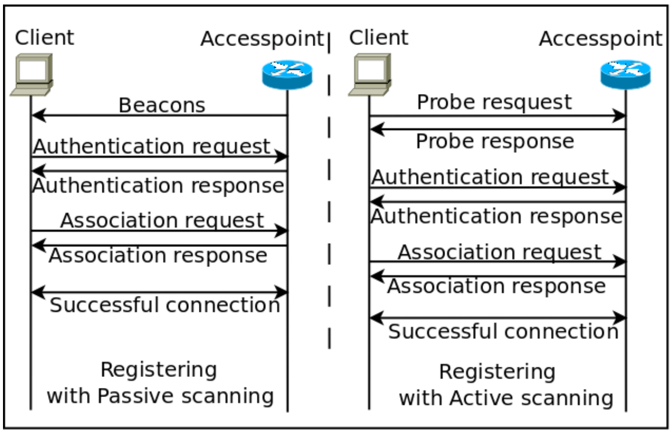


FIGURE 3. Wi-Fi connection

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 indoor or outdoor locations, (ii) node mobility between indoor 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.* [63], 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 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 [64].

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.

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.

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 "neighbouring" 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.

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 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2848119, IEEE Access

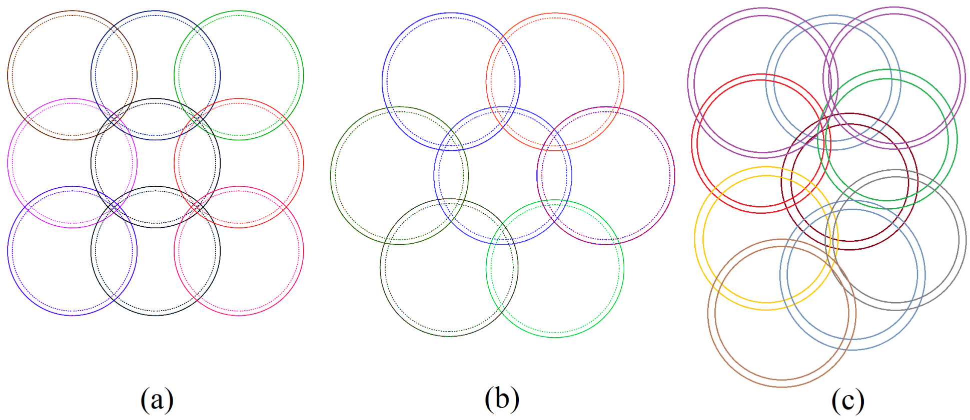


FIGURE 4. Gateway topology (a) Square topology (b) Hexagon topology (c) Random topology

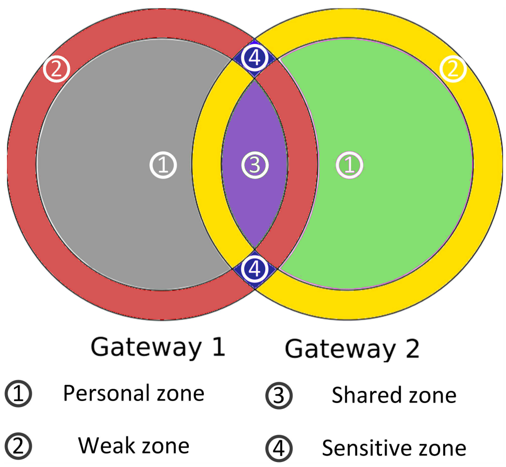


FIGURE 5. Setting up two adjacent gateways

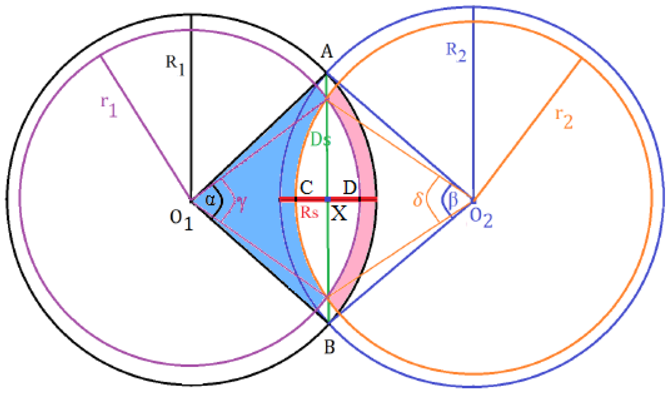


FIGURE 6. Areas of two adjacent gateways

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:

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

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

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
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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2848119, IEEE Access

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:

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

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:

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2848119, IEEE Access

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.

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.

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.

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.

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2848119, IEEE Access

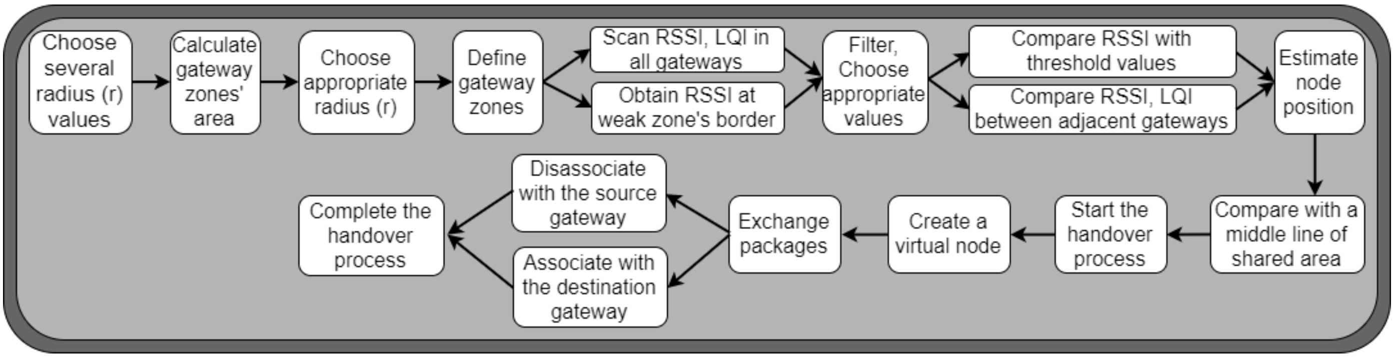


FIGURE 7. Mobility handover mechanism

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.

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.

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.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2848119, IEEE Access

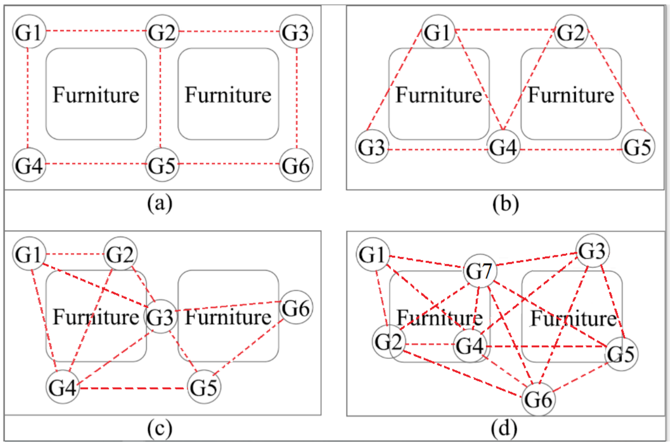


FIGURE 8. Gateway placement in a room
(a) Square topology
(b) Hexagon topology
(c) Random topology (setup 1)
(d) Random topology (setup 2)
(- -) indicates adjacent gateways having overlapping areas

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 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 [65], ADS1299 [45], ESP8266 [66] 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 [67] and Wi-Fi shield [68] 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 [64] and a 300Mbps wireless USB adapter [69]. 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) [70] 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 [71].

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 [72] 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).

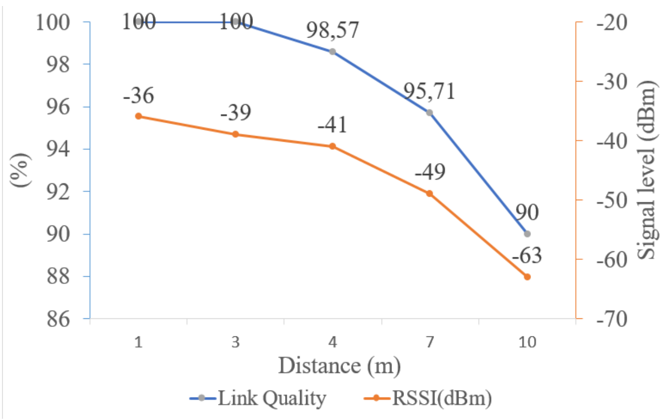


FIGURE 9. Gateway's signal level and link quality

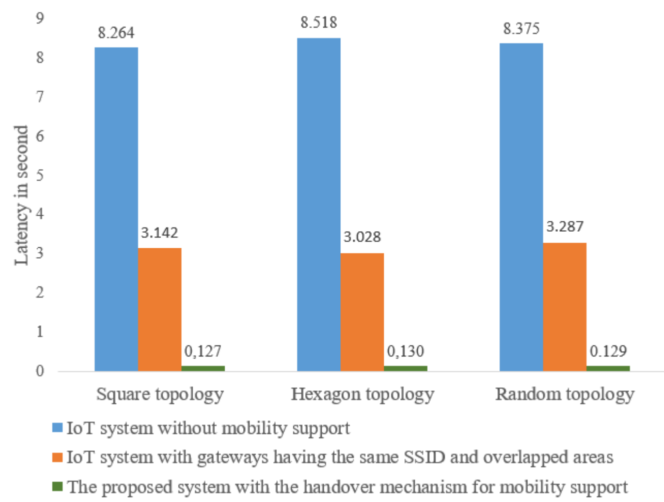


FIGURE 10. Latency of IoT systems with different configurations

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.

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 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'.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2848119, IEEE Access

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

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 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2848119, IEEE Access

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

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.

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

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.

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.

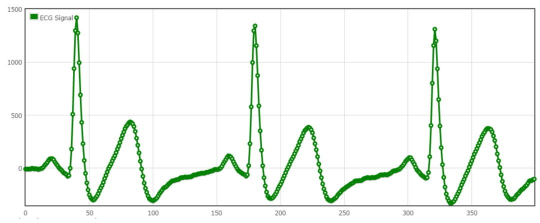


FIGURE 11. Graphical ECG waveforms at a remote browser

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 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 qual-

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]

ity 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.

ACKNOWLEDGMENT

The authors would like to thank Academy of Finland, Nokia Foundation, University of Turku Graduate School (UTUGS), Finnish Foundation for Technology Promotion (TES) for the financial support this project.

REFERENCES

[1] "European commission information society, internet of things strategic research roadmap," pp. 118–173, 2009.
[2] L. DaXu, W. He, and S. Li, "The internet of things: A survey," IEEE Trans. Ind. Inf. 10 (4), pp. 2233–2243, 2014.
[3] S. Li, L. DaXu, and S. Zhao, "Internet of things in industries: A survey," Inf.Syst.Front.17(2), pp. 243–259, 2015.
[4] T. N. Gia et al., "Iot-based fall detection system with energy efficient sensor nodes," in Nordic Circuits and Systems Conference (NORCAS), 2016 IEEE. IEEE, 2016, pp. 1–6.
[5] S.R. Moosavi et al., "Session resumption-based end-to-end security for healthcare internet-of-things," in (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on. IEEE, 2015, pp. 581–588.
[6] T. N. Gia et al., "Fault tolerant and scalable iot-based architecture for health monitoring," in Sensors Applications Symposium (SAS), 2015 IEEE. IEEE, 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 Computer Science, vol. 52, pp. 452–459, 2015.
[8] T. N. Gia et al. , "Energy efficient wearable sensor node for iot-based fall detection systems," Microprocessors and Microsystems, vol. 56, pp. 34–46, 2018.
[9] T. N. Gia et al., "Iot-based continuous glucose monitoring system: A feasibility study," Procedia Computer Science, vol. 109, pp. 327–334, 2017.
[10] A.M. Rahmani et al., "Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems," IEEE Consumer Communications and Networking Conference (CCNC'15), pp. 826–834, 2015, USA.
[11] V.K. Sarker et al., "Portable multipurpose bio-signal acquisition and wireless streaming device for wearables," in Sensors Applications Symposium (SAS), 2017 IEEE. IEEE, 2017, pp. 1–6.
[12] M. Ali et al., "Autonomous patient/home health monitoring powered by energy harvesting," pp. 1–7, 2017.
[13] M. Jiang et al., "Iot-based remote facial expression monitoring system with semg signal," IEEE Sensors Applications Symposium (SAS'16), 2016.
[14] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "Iot gateway: Bridging wireless sensor networks into internet of things," in Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on. Ieee, 2010, pp. 347–352.
[15] I. Grønbæk, "Architecture for the internet of things (iot): Api and interconnect," in Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on. IEEE, 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 Generation Computer Systems, 2017.
[17] A. M. Rahmani, P. Liljeberg, J.-S. Preden, and A. Jantsch, "Fog computing in the internet of things: Intelligence at the edge," 2017.
[18] B. Negash et al., "Leveraging fog computing for healthcare iot," in Fog Computing in the Internet of Things. Springer, 2018, pp. 145–169.
[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," IEEE International Conference on Computer and Information Technology (CIT'15), 2015.

[20] T. N. Gia et al., "Low-cost fog-assisted health-care iot system with energy-efficient sensor nodes," in Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International. IEEE, 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," The Journal of Supercomputing, pp. 1–19, 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, 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), 2015, pp. 1–7.

[24] I. Tcarenko et al., "Energy-efficient iot-enabled fall detection system with messenger-based notification," in International Conference on Wireless Mobile Communication and Healthcare. Springer, 2016, pp. 19–26.

[25] P. Kulkarni et al., "Requirements and design spaces of mobile medical care," ACM SIGMOBILE Mobile Comput. Commun. Rev., vol. 11, 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.

[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 Computing, vol. 4, no. 2, pp. 26–35, 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 2018, vol. 18, no. 2, 2018.

[30] L. F. Bittencourt, M. M. Lopes, I. Petri, and O. F. Rana, "Towards virtual machine migration in fog computing," in P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2015 10th International Conference on. IEEE, 2015, pp. 1–8.

[31] Z. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, HICSS'05, pp. 287a–297a, 2005, doi:http://dx.doi.org/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: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, ACM, New York, NY, USA,, pp. pp. 193–204, 2003, doi:http://doi.acm.org//10.1145/958491.958515.

[33] S. Yang and M. Gerla, "Personal gateway in mobile health monitoring," in Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on. IEEE, 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 Internet Multimedia Systems Architecture and Application (IMSAA), 2011 IEEE 5th International Conference on. IEEE, 2011, pp. 1–6.

[35] A. Raja and X. Su, "Mobility handling in mac for wireless ad hoc networks," J. Wireless Commun. Mobile Comput., Distrib. Syst. Sensors Actuators, vol. 9, no. 3, pp. 303–311, 2009.

[36] R.S. Koodli, and C.E. Perkins, "Mobile inter-networking with ipv6," John Wiley Sons, Inc,, vol. Print ISBN 9780471681656, pp. 27–53, 2007, dOI:10.1002/9780470126486.

[37] R. Silva, J. S. Silva, and F. Boavida, "Mobility in wireless sensor networks–survey and proposal," Computer Communications, vol. 52, pp. 1–20, 2014.

[38] S. Valenzuela, M. Chen, and V. Leung, "Mobility support for health monitoring at home using wearable sensors," IEEE Trans. Inf. Technol. Biomed. 15 (4), pp. 539–549, 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. Wirel. Mob. Netw., Ubiquitous Comput., Dependable Appl. 1 (2-3), pp. 107–122, 2010.

[40] A. Jara et al., "Hwsn6: Hospital wireless sensor networks based on 6lowpan technology: Mobility and fault tolerance management," in: International Conference on Computational Science and Engineering, vol. 2, pp. 879–884, 2009.

[41] A. Jara, M. Zamora, and A. Skarmeta, "Intra-mobility for hospital wireless sensor networks based on 6lowpan," in: 6th International Conference on Wireless and Mobile Communications, pp. 389–394, 2010.

[42] H. Fotouhi, M. Alves, M.Z. Zamalloa, and A. Koubaa, "Reliable and fast hand-offs in low-power wireless networks," IEEE Trans. Mob. Comput. 13 (11), pp. 2620–2633, 2014.

[43] Y.-S. Chen and Y.-T. Tsai, "A mobility management using follow-me cloud-cloudlet in fog-computing-based rans for smart cities," Sensors, vol. 18, no. 2, p. 489, 2018.

[44] M. Haseeb, A. Ahsan, and A. W. Malik, "Cloud to cloudlet-an intelligent recommendation system for efficient resources management: Mobile cache," in Frontiers of Information Technology (FIT), 2016 International Conference on. IEEE, 2016, pp. 40–45.

[45] "Ads1299," accessed: May 2016. [Online]. Available: http://www.ti.com/product/ADS1299

[46] F. Touati and R. Tabish, "U-health system: State-of-the-art review and challenges," Journal of Medical Systems 37(3):9949,, 2013.

[47] 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 NORCHIP, 2014. IEEE, 2014, pp. 1–6.

[48] T. Rappaport et al., "Wireless communications: principles and practice," vol. 2, 1996.

[49] K. Atalah, E. Macias, and A. Suarez, "A proactive horizontal handover algorithm based on rssi supported by a new gradient predictor," Ubiquitous Computing and Communication Journal, vol. 3, 2008, doi:http://dx.doi.org/10.1016/j.compind.2014.09.002.

[50] J. M. L. P. Caldeira et al., "Intra-mobility support solutions for healthcare wireless sensor networks-handover issues," IEEE Sensors Journal, vol. 13, no. 11, Nov. 2013.

[51] A. Vlavianos, L.K. Law, I. Broustis, S.V. Krishnamurthy, and K. Faloutsos, "Assesing link quality in IEEE 802.11 wireless networks: which is the right metric?" In: The 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–6, 2008.

[52] K. Srinivasan and P. Levis, "Rssi is under appreciated," in Workshop on Embedded Networked Sensors, 2006.

[53] C. Reis et al., "Measurement - based models of delivery and interference in static wireless networks," In ACM SIGCOMM, 2006.

[54] C. Guo et al., "Improving packet delivery ratio estimation for indoor ad hoc and wireless sensor networks," Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE, pp. 1–5, 2009.

[55] G. Breed, "Bit error rate: Fundamental concepts and measurement issues," High Frequency Electronics tutorial, 2003.

[56] 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," Computers in Industry 69 (2015), pp. 92–104, 2015, doi:http://dx.doi.org/10.1016/j.compind.2014.09.002.

[57] C. Pelczar, K. Sung, J. Kim, and B. Jang, "Vehicle speed measurement using wireless sensor nodes," Proceedings of the 2008 IEEE International Conference on Vehicular Electronics and Safety, 2008.

[58] S. Capkun et al., "Gps-free positioning in mobile ad hoc networks," Cluster Computing, vol. 5, no. 2, pp. 157–167, 2002.

[59] L. Wang and Q. Xu, "Gps-free localization algorithm for wireless sensor networks," Sensors, vol. 10, no. 6, pp. 5899–5926, 2010.

[60] K. Ashton, "That internet of things thing," RFiD Journal, vol. 22, no. 7, pp. 97–114, 2009.

[61] J. Caffery et al., "Overview of radiolocation in cdma cellular systems," Communications Magazine, IEEE, vol. 36, no. 4, pp. 38–45, 1998.

[62] M. Lin and W.J. Hsu, "Mining gps data for mobility patterns: A survey," Pervasive and Mobile Computing, vol. 12, pp. 1–16, 2014.

[63] L. Xu et al., "Variation of received signal strength in wireless sensor network," 3rd International Conference on Advanced Computer Control (ICACC 2011), vol. 3, 2011.

[64] "Pandaboard," Available: accessed: April 2018. [Online]. Available https://elinux.org/PandaBoard.

[65] "Arduino mega," accessed: May 2016. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardMega

[66] "Esp8266," accessed: May 2016. [Online]. Available: http://www.esp8266.com/

[67] "Arduino uno," accessed: May 2016. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno

[68] "Wifi shield for arduino uno," accessed: May 2016. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoWiFiShield

[69] "Esp8266," accessed: May 2016. [Online]. Available: http://www.netis-systems.com/en/Products/Wireless%20USB%20Adapters/861.html

[70] T. A. Beardsley and J. Qian, "The tcp split handshake: Practical effects on modern network equipment," Network Protocols and Algorithms, vol. 2, no. 1, pp. 197–217, 2010.

[71] S.R. Moosavi et al., "End-to-end security scheme for mobility enabled healthcare internet of things," Future Generation Computer Systems, 2016.

[72] M. Fontanini, "Libtins - packet crafting and sniffing library," 2016, accessed: May 2016. [Online]. Available: http://libtins.github.io/

TUAN NGUYEN GIA received his Master (Tech) from the University of Turku (Turku, Finland) in 2014. He received a 4-year funded position in University of Turku and a grant scholarship from Nokia Foundation for an efficient and excellent researcher in 2015. In addition, he receive a grant scholarship for excellent research from the Finnish Foundation for Technology Promotion in 2016. Currently, he works at the Internet-of-Things for Healthcare (IoT4Health) research group in Future Technologies department, University of Turku. His research field is Fog computing, Internet-of-Things, healthcare, FPGA, and autonomous system.

AMIR M. RAHMANI received his MSc from Department of ECE, University of Tehran, Iran, in 2009 and Ph.D. degree from Department of IT, University of Turku, Finland, in 2012. He also received his MBA jointly from Turku School of Economics and European Institute of Innovation & Technology (EIT) Digital, in 2014. He is currently Marie Curie Global Fellow at University of California Irvine (USA) and TU Wien (Austria). He is also an adjunct professor (Docent) in embedded parallel and distributed computing at 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 is the author of more than 150 peer-reviewed publications. He is a senior member of the IEEE.

TOMI WESTERLUND received his PhD (Tech) 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 acted yearly as a visiting scholar at Fudan University, Shanghai, China. His current research interest is Internet of Things (IoT); how we can utilise 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 MSc and PhD degrees in electronics and information technology from the University of Turku, Turku, Finland, in 1999 and 2005, respectively. He received Adjunct professorship in embedded computing architectures in 2010. Currently he is working as a professor in University of Turku in the field of Embedded Systems and Internet of Things. At the moment his research is focused on Internet of Things, Fog computing, biomedical engineering, self-aware systems, approximation computing and health technology. In that context he has established and leading the Internet-of-Things for Healthcare (IoT4Health) research group. Liljeberg is the author of more than 270 peer-reviewed publications.

HANNU TENHUNEN received the diplomas from the Helsinki University of Technology, Finland, 1982, and the PhD degree from Cornell University, Ithaca, NY, 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 department director. Since 1992, he has been a professor at the Royal Institute of Technology (KTH), Sweden, where he also served as a dean. He has more than 600 reviewed publications and 16 patents internationality. He is a member of the IEEE.

Paper V

IoT-based Continuous Glucose Monitoring System: A Feasibility Study

Tuan Nguyen Gia, Mai Ali, Imed Ben Dhaou, Amir M Rahmani, Tomi Westerlund, Pasi Liljeberg, Hannu Tenhunen

Original published in Procedia Computer Science, Elsevier Journal, 109, pages 327-334

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 327–334

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Conference on Ambient Systems, Networks and Technologies (ANT-2017)

IoT-based continuous glucose monitoring system: A feasibility study

Tuan Nguyen Gia[1]*, Mai Ali[2], Imed Ben Dhaou[3], Amir M. Rahmani[4,5], Tomi Westerlund[1], Pasi Liljeberg[1], Hannu Tenhunen[1]

[1]*Department of Future Technologies, University of Turku, Turku, Finland*
[2]*Department of Electrical Engineering, Alfaisal University, Riyadh, Saudi Arabia*
[3]*Department of Electrical Engineering, Qassim University, Buraydah, Saudi Arabia,*
[4]*Department of Computer Science, University of California Irvine, USA*
[5]*Institute of Computer Technology, TU Wien, Austria*

Abstract

Health monitoring systems based on Internet-of-things (IoT) have been recently introduced to improve the quality of health care services. However, the number of advanced IoT-based continuous glucose monitoring systems is small and the existing systems have several limitations. In this paper we study feasibility of invasive and continuous glucose monitoring (CGM) system utilizing IoT based approach. We designed an IoT-based system architecture from a sensor device to a back-end system for presenting real-time glucose, body temperature and contextual data (i.e. environmental temperature) in graphical and human-readable forms to end-users such as patients and doctors. In addition, nRF communication protocol is customized for suiting to the glucose monitoring system and achieving a high level of energy efficiency. Furthermore, we investigate energy consumption of the sensor device and design energy harvesting units for the device. Finally, the work provides many advanced services at a gateway level such as a push notification service for notifying patient and doctors in case of abnormal situations (i.e. too low or too high glucose level). The results show that our system is able to achieve continuous glucose monitoring remotely in real-time. In addition, the results reveal that a high level of energy efficiency can be achieved by applying the customized nRF component, the power management unit and the energy harvesting unit altogether in the sensor device.

Keywords: Internet-of-Things, wearable, health monitoring, energy harvesting, energy efficient, power management, glucose monitoring

1. Introduction

Internet of Things (IoT) can be viewed as a dynamic network where physical and virtual objects are interconnected together[1]. IoT encompassing advanced technologies such as wireless sensor networks (WSN), artificial intelligence, and cloud computing plays an important role in many domains comprising of robotics, logistics, transportation, and health-care. For instance, IoT-based systems for health-care consisting of sensing, WSN, smart gateways, and Cloud provide a way to remote and real-time e-health monitoring.

Advances in WSNs have created an innovative ground for e-health and wellness application development. Ambient assisted living, ambient intelligence, and smart homes are becoming increasingly popular[2]. These can be combined to other health solutions such as fitness and wellness, chronic disease management and diet or nutrition monitoring applications. The new initiatives tend to be integrated into the patient information ecosystem instead of being sepa-

* Corresponding author. Tel.: +3-582-333-8444.
E-mail address: tunggi@utu.fi

10.1016/j.procs.2017.05.359

rated into monitoring and decision processes. There are potential benefits to ageing population, where elderly people could be monitored and treated at the comfort of their own homes.

Fully autonomous health monitoring wireless systems can have many useful applications. Among those applications is glucose level measurement for diabetics. Diabetes is a major health concern. According to a WHO report, the number of people with diabetes has exceeded 422 millions and in 2012, over 1.5 million people died because of diabetes. The WHO classified diabetes as a top ten causes of mortality. Diabetes has serious effects on the well-being of a person and the society. Unfortunately, there is still no known permanent cure for diabetes [3]. However, one solution to this problem is to continuously measure blood glucose levels and close the loop with appropriate insulin delivery. Statistics published by the UK Prospective Diabetes Group demonstrate that CGM can reduce the long term complications between 40 % and 75 %. [4]. Hence, CGM equipped with alarm systems can help patients to take corrective action(s) such as decisions on their diet, physical exercise and when to take medication.

Energy harvesters incorporated into wearable devices allow powering wireless sensor operated applications , thereby making them autonomously operated. This regime has many useful implications on patients and heath-care providers, specially for implanted sensors where battery changing could cause pain and discomfort. Cautious design of both low-power electronic circuitry and efficient energy harvesting scheme is pivotal to fully autonomous wearable systems.

In this paper, the presented work aims to study the feasibility of invasive and secure CGMS using IoT. The work is to design an IoT-based system architecture from a sensor device to a back-end system for presenting real-time glucose, body temperature and contextual data (i.e. environmental temperature) in graphical and text forms to end-users such as patient and doctor. Moreover, the work customizes the nRF communication protocol for suiting to the glucose monitoring system and achieving a high level of energy efficiency. Furthermore, we investigate energy consumption of a sensor device and design energy harvesting units for the device. Finally, we present a push notification service for notifying patient and doctors in case of abnormal situations such too low or too high glucose level. In summary, our main contributions in this paper are as follows:

- proposing continuous glucose monitoring IoT-based system
- designing an energy efficient sensor device using nRF protocol
- designing an energy harvesting unit for the sensor device to extend the sensor device's battery life

The remainder of the paper is organized as follows: In section 2 related works are presented. Section 3 presents the continuous glucose monitoring IoT-based system architecture. In section 4, an implementation of the glucose monitoring system is shown. In section 5, experimental results are discussed. Section 6 concludes the work.

2. Related works

Many research applications in glucose monitoring are not based on IoT-based architectures. Correspondingly, doctors or caregivers cannot monitor glucose levels of a patient remotely in real-time. Murakami et al. [5] present a CGM system in critical cardiac patients in the intensive care unit. The system is built by a disposable subcutaneous glucose sensor, a glucose client, and a server. The system collects glucose data four times per day and stores in a hospital information system. Doctors can use the bedside monitor to monitor the glucose data.

Ali et al. [6] propose a Bluetooth low energy (BLE) implantable glucose monitoring system. Glucose data collected from the system is transmitted via BLE to a PDA (smart-phone, or Ipad) which represents the received data in text forms for visualization. The system shows some achievements in reducing power consumption of an external power unit and an implantable unit.

Lucisano et al. [7] present a glucose monitoring in individuals with diabetes using long-term implanted sensor system and model. Glucose data is sent every two minutes to external receivers. The system shows its capability of continuous long-term glucose monitoring. In addition, the system proves that implanted sensors can be placed inside a human body for a long period time (i.e. 180 days) for managing diabetes and other diseases.

Menon et al. [8] propose a non-invasive blood glucose monitoring system using near-infrared (NIR). Glucose in blood is predicted based on the analysis of the variation in the received signal intensity obtained from a NIR sensor. The predicted glucose data is sent wirelessly to a remote computer for visualization.

Recently, some IoT-based applications for glucose monitoring have been built. However, those systems do not attentively consider energy efficiency of sensor nodes and the communication between sensor devices and a gateway. Rasyid et al. [9] propose a blood glucose level monitoring system based on wireless body area network for detecting diabetes. The system is built by using a glucometer sensor, Arduino Uno, and a Zigbee module. Doctor and caregiver can access to a web-page to monitor glucose levels of a patient remotely. However, the system is not energy efficient due to high power consumption of the Arduino Uno board and the Zigbee module.

Wang et al. [10] introduce a monitoring system for types 2 diabetes mellitus. The system is able to make decision on the statues of diabetes control and predict future glucose of an individual. Obtained glucose data can be monitored remotely by medical staffs via wide area networks.

Although these systems show their advantages in continuous glucose monitoring, there are still many limitations. For example, some systems do not consider real-time and remote monitoring while other systems do not pay attention on energy efficiency sensor devices/nodes. In addition, they are not able to inform to medical doctors in real-time in cases of emergency.

The main motivation of the paper is to provide an advanced IoT-based system for real-time and remote continuous monitoring glucose, contextual data, and body temperature. The differences between our work and others are energy efficient sensor devices integrated with an energy harvesting unit, a power management unit and an ultra low energy nRF wireless communication, together with dedicated gateways equipped with advanced services such as push notification for real-time notifying both doctor and patient in case of abnormality.

3. System architecture

In the furtherance of providing continuous glucose monitoring in real-time locally and remotely, the CGMS architecture shown in Fig .1 is based on an IoT architecture. The system includes three main components such as a portable sensor device, a gateway and a back-end system.

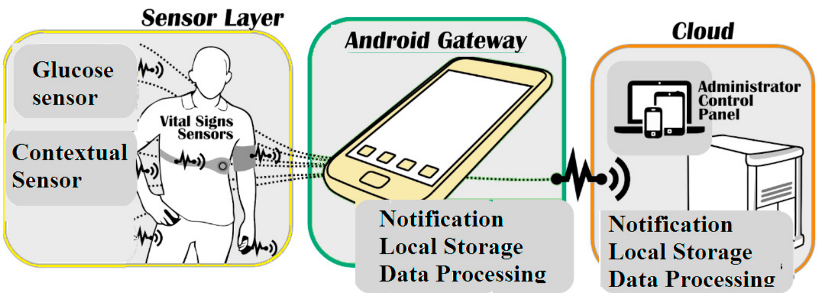


Fig. 1: Continuous Glucose Monitoring using IoT

3.1. Sensor device structure

The sensor device whose structure is shown in Fig .2 consists of primary component blocks such as sensors, a micro-controller, a wireless communication block, energy harvesting and management components. The micro-controller performs primary tasks of the device such as data acquisition and transmission. Therefore, it consumes a large part of the device's total power consumption. Reducing power consumption micro-controller can save a lot of power consumption of the device. The ultra low power micro-controller capable of operating with sleep modes is a suitable candidate for the target. In the device, the micro-controller receives glucose data from an implantable glucose sensor via a wireless inductive link receiver while it collects environmental and body temperature via data link wires such as UART, SPI or I2C. In the system, SPI is more preferable due to its lowest power consumption between these interfaces [11].

The nRF wireless communication block is responsible for transmitting data from the micro-controller to the gateway equipped with an nRF transceiver. The block includes a RF transceiver IC for the 2.4GHz ISM band and an embedded antenna. Due to 2Mbps supporting, nRF completely fulfills the requirements of transmission data rates in a CGM system. Transmission data rates of nRF can be configured for achieving some levels of energy efficiency. For example, instead of using 2Mbps, a data rate of 256kbps can be used for saving power when sending glucose, temperature, and contextual data. In addition, nRF is capable of both short and long range transmission from a few centimeters to a hundred of meters. Depending particular applications, the transmission range and transmission power can be configured. With a short range communication, nRF consumes lower energy.

In the sensor node, the energy harvesting unit and the power management unit described in the followings are two of the most important components because they directly impact on energy consumption and an operating duration of the sensor node.

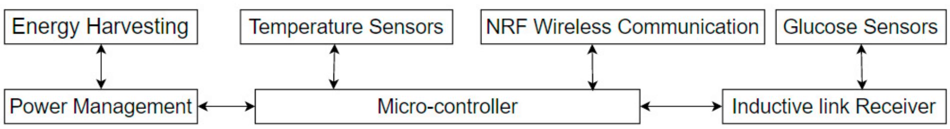


Fig. 2: Portable sensor device structure

3.1.1. Energy harvesting unit

The exponential advancements in WSNs, WBANs and the emerging field of IoT have opened the doors wide for numerous intelligent applications. Unfortunately, this development is not reflected at the battery capacity side. A major limitation of untethered nodes is a limited battery capacity which limits the operation time of the nodes. The finite lifetime of a node implies the nite lifetime of the applications or additional costs and complexity to regularly

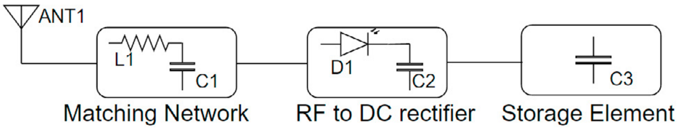


Fig. 3: RF energy harvesting system

change batteries. Nodes could possibly use large batteries for longer lifetimes, but will have to deal with increased size, weight and cost. Nodes may also opt to use low-power hardware like a low-power processor and radio, at the cost of lesser computation ability and lower transmission ranges. Several solution techniques have been proposed to maximize the lifetime of battery-powered sensor nodes. Some of these include energy-aware MAC protocols, power aware storage, routing and data dissemination protocols, duty-cycling strategies, adaptive sensing rate, tiered system architectures and redundant placement of nodes. While all the above techniques optimize and adapt energy usage to maximize the lifetime of a sensor node, the lifetime remains bounded and nite. The above techniques help prolong the application lifetime and/or the time interval between battery replacements but do not preclude energy related inhibitions [12].

Energy harvesting could be a solution to the above mentioned dilemma. Energy harvesting refers to harnessing energy from the environment or other energy sources (body heat, foot strike, nger strokes) and converting it to electrical energy. If the harvested energy source is large and periodically/continuously available, a sensor node can be powered perpetually. Energy sources can be broadly classified into the following two categories, (i) Ambient Energy Sources: Sources of energy from the surrounding environment, e.g., solar energy, wind energy and RF energy, and (ii) Human Power: Energy harvested from body movements of humans. Passive human power sources are those which are not user controllable. Some examples are blood pressure, body heat and breath. Active human power sources are those that are under user control, and the user exerts a specic force to generate the energy for harvesting, e.g., nger motion, paddling and walking. No single energy source is ideal for all applications. The choice depends on every applications requirements and constraints.

To power the glucose sensor node a combination of ambient and human powered sources is selected. Due to its ubiquitous availability RF energy is an adequate source for this application. Also, since the sensor is mounted on human body it makes sense to exploit this medium a source of energy. Through the use of a Thermoelectric Generator (TEG), thermal energy can be converted into electrical energy. The conversion process is based on the See-beck effect where electricity can be generated from the temperature gradient across two conductors connected together. In this paper the RF energy harvesting system is presented, thermal energy harvesting will be integrated into harvesting in future work.

The RF energy harvesting system illustrated in Fig. 3 is designed . A first step in designing the RF energy harvesting system is deciding on the frequencies at which power will be harvested. The wireless spectrum is full of signals with different frequencies and power levels, ranging from cellular standards, WLANs and TV signals. The criteria that control the selection of certain frequencies for the purpose of energy harvesting are wide deployment and power level. GSM 900/1800 and DTV signals cover most of the world. Thus it is almost guaranteed that GSM signals would be available wherever the harvesting system is placed. On the other hand, the power level of GSM signals can be very low reaching -102 dBm at receiver sensitivity, and transmitting power of DTV stations can be as high as 70 dBm [13]. Wi-Fi and Bluetooth signals abounds in urban environments, hence they can be exploited as well. A miniaturized printed elliptical nested fractal multiband antenna (PENF) was designed for this purpose in [13]. The proposed antenna covers GSM 900, 2.4 GHz Bluetooth/WLAN, 3.2 GHz (Radiolocation, 3G), 3.8 GHz (for LTE, 4G) and 5 GHz Wi-Fi bands. The antenna was designed and fabricated using FR4 substrate. The measured radiation patterns of PENF antenna in different planes verify the omni-directional feature of the proposed antenna. This feature is of interest in RF energy harvesting applications to receive the ambient signals from all directions.

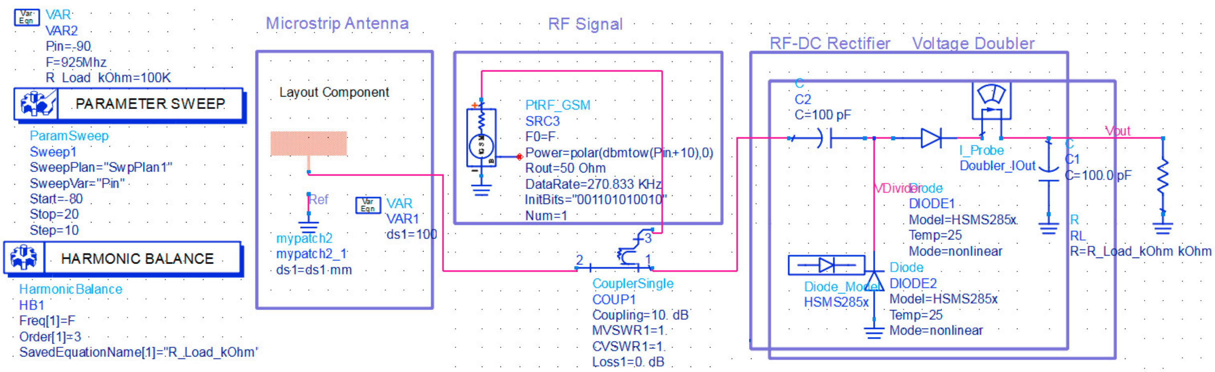


Fig. 4: RF-DC rectifier schematic and test bench

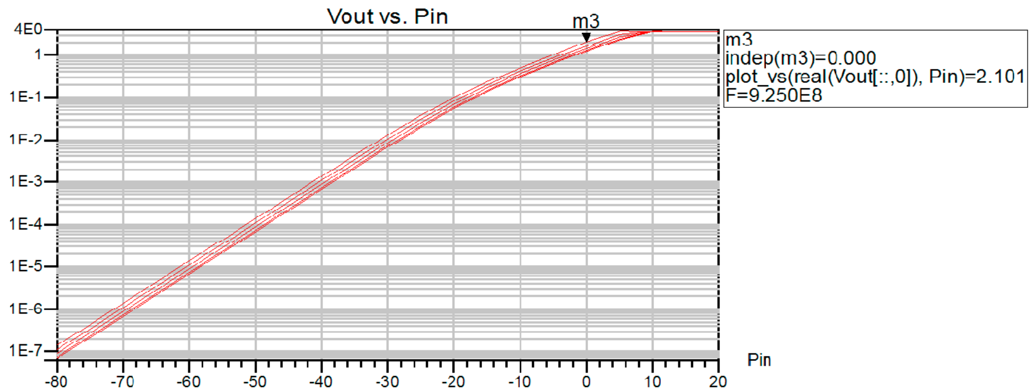


Fig. 5: Rectifier output voltage Vs. input power level

Due to their low forward voltage drop Schottky diodes are used in designing the rectifier and voltage doubler circuits. A Schottky diode is a rectifying metal semiconductor junction which is fabricated by depositing an n-type or p-type semiconductor material on a variety of metals. While the threshold voltage of a P-N diode is around 0.6 V to 0.7 V, Schottky diodes can achieve similar performance at lower threshold levels (0.2 V to 0.3 V). Fig. 4 illustrates the RF-DC rectifier schematic and test bench used in evaluating its performance. The rectifier unit itself shown at the right corner of Fig. 4 consists of a single unit voltage doubler constructed using a non-linear model for HSMS285 Schottky diode. Using a single patch antenna integrated into the schematic as a layout component is tested. At 925 MHz RF input signal, the corresponding output voltage for a wide range of input power levels is shown in Fig. 5. The sensor node constructed at this work requires 2 volts input for proper operation, the RF energy harvesting system designed was able to achieve this voltage at 0 dBm input power which corresponds to 1 mW.

3.1.2. power management unit

In battery powered wireless sensor network, the sensor goes to the idle status to prevent power draining. However, in case the sensor receives an incoming message (packet), the sensor should go from idle to active status. Duty-cycling is an efficient approach that reduces idle-mode power consumption [14]. There are three-types of duty cycling: (I) synchronous, (II) pseudo-synchronous and (III) asynchronous. The latter scheme has the lowest power dissipation.

A number of circuit techniques have been proposed to implement the wake-up unit. In [15], the author laid the foundation for radio-triggered wake-up circuit. The circuit consists of passive components (antenna, resistance, diode, capacitance and inductance). The circuit has low-sensitivity and may cause false wake-up.

The authors of [16] describe a dual source energy scavenging circuits for wireless sensor network. The power management system is composed of both a radio-triggered wake-up circuit and a voltage sensor. The latter pushes the sensor into a sleep-mode should the voltage level across the storage capacitor drops below a predefined voltage level. The wake-up circuit is implemented using a low-power Schmitt trigger circuit. The voltage sensor is implemented using a differential amplifier with positive feedback.

3.1.3. Proposed power management unit

The communication between the sensor and the gateway is quasi-unidirectional in which the information is relayed from the sensor towards the rest of the system. This unique characteristics makes the need for a radio-triggered wake-up circuit. However, the power management unit for the sensor needs to efficiently use the collected energy. To prevent an abrupt shutdown and to enable a better resource utilization, in case the power management unit descries insufficient amount of energy in the capacitor, drives the sensor into a deep-sleep mode and grants the energy scavenging unit enough time to charge the capacitor.

The core of the power management unit is the voltage sensor, which can be realized using the low-power Schmitt trigger circuit reported in [17]. Fig. 6 details the transistor level circuit for the PMU.

3.2. Gateway and back-end structure

Similar to conventional gateways in IoT systems, the proposed gateway collects data from wireless sensor devices and transmits the data to Cloud servers. The gateway performs its tasks by using a nRF transceiver and a wireless IP-based transceiver (i.e. Wifi, GPRS or 3G). The nRF transceiver, which is a plug-able component, is compatible with all types of smart devices (i.e. Android, Iphone, tablet). It is possible to use smart phones and fixed gateways as the system gateways. However, in this work, gateways based on smart-phones are more focused. The nRF transceiver consists of a micro-controller and a low power RF transceiver IC, and a FTDI component. The micro-controller and the nRF components are the same as the ones used in the sensor device. The FTDI chip is used for converting from an UART connection to an USB connection. All physical connections of an nRF component and a smart phone (Android) are shown in Fig.7.

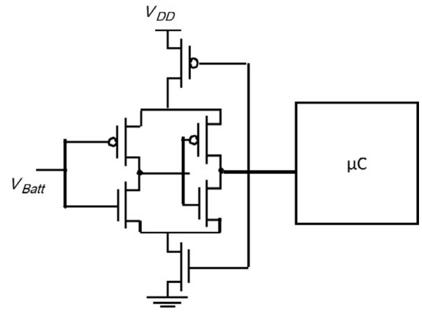


Fig. 6: Power Management unit using Schmitt triger circuit proposed in [17]

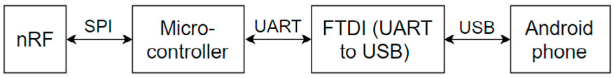


Fig. 7: Android gateway structure

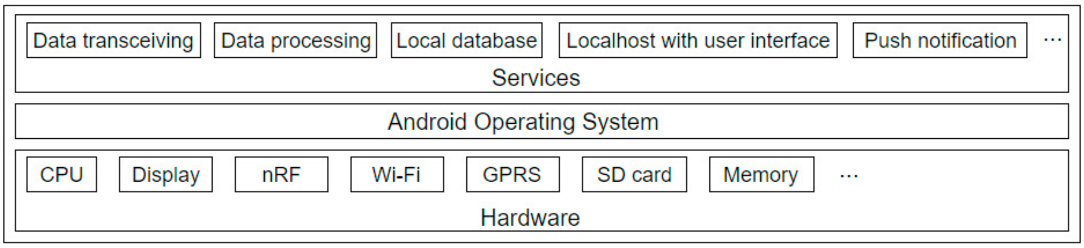


Fig. 8: Gateway operating structure

In addition to mentioned tasks, the gateway provides advanced services such as data processing, local database, local host with user interface and push notification shown in Fig.8. For example, the collected data might consist of noise and corrupted data. In order to provide a high quality of data, the noise and corrupted data must be filtered. In the gateway, the data processing unit not only performs filtering tasks but also run algorithms to process data such as decision making and categorization of diabetes statuses.

Local database in the gateway consists of an intact database and a real-time database. The intact database stores algorithms' information and configuration data while the real-time database is used for storing e-health and contextual data. Therefore, the intact database is only used for internal usage and managed by system administrators while the real-time database is regularly updated and synchronized with Cloud's database. Due to a small amount of collected data (i.e. 4-8 samples per 10 minutes), local database can store the data during a long period of time (i.e. several days) before getting full.

By supplying a local host with user interface, real-time data can be monitored directly from the gateway without requiring Cloud servers. In this case, this helps to eliminate an unnecessary latency of transmitting and receiving data to and from Cloud, respectively.

In the gateway, decision making and push notification services work together to provide real-time notifications to doctors or caregivers. For example, when a monitored glucose level is higher and lower than an acceptable level, the decision making service triggers the push notification to send messages for notifying a doctor in real-time. The back-end part comprises of Cloud and an user accessible terminal. Doctor can access real-time data in Cloud remotely via a web browser or a mobile application.

4. Implementation

With the purpose of evaluating feasibility of the CGM system using IoT, the entire system shown in Fig. 1 is implemented. First, the interaction of the biological tissue under investigation is studied. Since the glucose sensor will be subcutaneous, the electrical characteristics of the biological tissue i.e. skin will be evaluated from which the amount of power loss and absorption due to propagation through the biological tissue will be estimated.It is imperative to make sure that subjecting the human body to this continuous signals is within the safe specified measures. The guidelines for Electromagnetic Field exposure (EMF) are in terms of Specific Absorption Rate (SAR) and the equivalent plane wave power density (SW/m2). SAR is a measure of the rate of energy absorption per unit mass due to exposure to an RF source. SAR is normalized to mass and is defined as [18]:

$$SAR = ({}_{e}ffE_{r}ms^{2})/(W/Kg) \quad (1)$$

Where ${}_{e}ff$ is the effective conductivity of the biological material such as skin and is proportional to the frequency of the applied field, is the mass density which is approximately 1000 kg/m^{3} for most biological tissues,and $E_{r}ms$ is the root mean square value of the electric field E at the measurement point. As specified in [19] at an operating frequency

Table 1: Glucose levels

Type	Before meals	2 hours after meals	Wake up	Risk of hypoglycaemia (low blood glucose)
Healthy Person	4 to 5.9 mmol/L	under 7.8 mmol/L		under 4 mmol/L
Type I diabetes	4 to 7 mmol/L	5 to 9 mmol/L	5 to 7 mmol/L	under 4 mmol/L
Type 2 diabetes	4 to 7 mmol/L	under 8.5 mmol/L		under 4 mmol/L
Children with type 1 diabetes	4 to 7 mmol/L	5 to 9 mmol/L	4 to 7 mmol/L	under 4 mmol/L

Table 2: Blood glucose levels in diagnosing diabetes

Plasma glucose test	Normal	Prediabetes	Diabetes
Random	Below 11.1 mmol/l Below 200 mg/dl		11.1 mmol/l or more 200 mg/dl or more
Fasting	Below 6.1 mmol/l Below 108 mg/dl	6.1 to 6.9 mmol/l 128 to 125 mg/dl	7.0 mmol/l or more 126 mg/dl or more
2 hour post-prandial	Below 7.8 mmol/l Below 140 mg/dl	7.8 to 11.0 mmol/l 140 to 199 md/dl	11.1 mmol/l or more 200 mg/dl or more

of 2.4 GHz, the maximum E and S are 61 V/m and 10 W/m^2 respectively which are well below the targeted operation power of the wireless sensor node.

In the sensor device, an ATMega328P micro-controller is used because it can achieve a high level of energy efficiency. The micro-controller can run at 16Mhz. However, it needs to use an external oscillator and requires 5V power supply for running at this clock. In contrast, it merely uses 1Mhz internal oscillator and requires 2V power supply for operating at 1Mhz. On the grounds that the sensor device does not perform any heavy computation, 1MHz clock speed and 2V power supply are suitable. In the implementation, the sensor device is in a deep sleep mode in most of the time. It is waken up regularly i.e. for receiving incoming data from a glucose sensor and temperature sensors. Then, it wakes up a nRF component for transmitting the data to a gateway. After all, it goes back to the deep sleep mode.

The nRF24L01 IC is an ultra low power 2Mbps RF transceiver IC for the 2.4GHz ISM band. It is built-in with an advanced power management method. The nRF24L01 IC communicates with the micro-controller via a SPI interface. For the reason that it is no required a high transmission data rate to collect glucose, body temperature and environment temperature every 10 minutes, SPI with a data rate of 250kbps is applied in the implementation.

The gateway includes a nRF transceiver and a smart phone in which the nRF transceiver is connected to the phone via a USB port. In terms of the hardware implementation, the nRF transceiver is implemented by an ATMega328P micro-controller and a nRF24L01 IC. Similar to the micro-controller in the sensor device, the micro-controller of the nRF transceiver also run at 1Mhz. The micro-controller is supplied with 3.3V from the FTDI component which converts 5V from the phone's USB port to 3.3V. For saving power consumption, it is in a deep sleep mode in most of the time. It is only waken up by an interrupt for receiving incoming data from a sensor device and immediately forwarding to the smart phone. After completing these tasks, it goes back to a deep sleep node. When it is in a sleep mode, the nRF component is also in a sleep mode.

An Android app is built in the gateway for receiving data from the nRF component and performing other services. When data is available at one-end of the USB port, the app automatically reads the data and performs the data processing service. In addition, the app is capable of representing the processed data in text and graphical forms and triggering a push notification service.

The push notification service is implemented by a Google push notification API. When the mobile app detects abnormal situations (i.e. too low or too high glucose level), the push notification service in the gateway is triggered for sending notification messages to Cloud which then notifies doctors and an end-user wearing the sensor device.

Local database in gateways is implemented by MySQL database, and local storage (HD card). For example, the table of glucose levels based on the Australian and UK national diabetes service scheme and the global diabetes community[20]-[21] shown in TABLE .1-2 is stored MySQL tables.

The server is implemented by HTML5, Web-Socket and Node.js because they support real-time and streaming data. In addition, MySQL database for storing synchronized data and Javascipt for plotting graphical charts are utilized.

5. Experiments and Results

In order to verify the quality of data transmitted via nRF from a sensor node to a gateway, two sets of data including random and predefined data are used. The data collected at the sensor node and the received data at the gateway is compared in several cases such as the sensor node and the gateway in pockets, or the gateway in the environment less than 0 degree Celsius. In the experiments, the distance between the sensor node and the gateway is in a range of a few meters although in most of the cases, the distance is less than 1 meter. The results show that the sent and received data (i.e. environmental temperature and body temperature) is the same, and there is no lost during the transmission for all cases. In some cases when radio signals are blocked, the sensor node tries to send the data with a higher power. Fortunately, average power consumption of the sensor node does not vary dramatically due to a long interval (i.e. every 10 minutes) between transmission times. Power consumption of components and devices used in the implementation is shown in Table .3.

Table 3: Power consumption of nRF transceiver, sensor node and gateway

Device	Voltage supply (V)	Average Current (mA)
nRF transmitter (nRF + ATMEGA328P)	2	0.5
nRF receiver (nRF + ATMEGA328P + FTDI board)	5	5
Sensor node	2	1.4
gateway(Android phone without nRF receiver)	5	70
Android phone with nRF receiver	5	75

To the best of our knowledge, our sensor node consumes the least power than other sensor nodes existing in the market and proposed by other authors. Most of existing energy efficient sensor nodes for CGM consume more than 5mA while the proposed sensor node consume merely around 1.4mA. Table 3 shows that power consumption of the Android phone increases about 6.5% when attaching an nRF receiver prototype including nRF transceiver, ATMEGA328P and FTDI components. The power consumption can be reduced when the prototype is replaced by an entire circuit device. In this case, surplus components such as LEDs, and IO ports can be removed for saving power consumption. For testing functionality of the application in the gateway, several glucose values including low, medium and high glucose levels altogether with temperature values are sent from the sensor node to the gateway. The result shows that data is categorized and represented in text forms accurately. In addition, the push notification service operates accurately in real-time when abnormality (e.g. too high or too low glucose levels) is detected.

6. Conclusion

In this paper, we presented a real-time remote IoT-based continuous glucose monitoring system. The implemented IoT-based architecture is complete system starting from sensor node to a back-end server. Through the system, doctors and caregivers can easily monitor their patient anytime, anywhere via a browser or a smart-phone application. Sensor nodes of the system are able to obtain several types of data (i.e. glucose, body temperature, and environmental data) and transmit the data wirelessly to the gateway efficiently in term of energy consumption. In addition, the sensor node is integrated with the power management unit and the energy harvesting unit for extending operating duration of the sensor device. With the assistance of the customized nRF receiver, a patient's smart-phone becomes a gateway for receiving data from sensor nodes. In addition, the gateway with its application provides advanced services to users, such as a notification service. The result showed that it is feasible to remote monitor glucose continuously in real-time and the system can be made energy efficient.

Acknowledgement

This research was partially supported by IRG16202 - Alfaisal University.

References

1. S.A. Haque et al. Review of cyber-physical system in healthcare. *International Journal of Distributed Sensor Networks*, 2014, 2014.
2. A. Aragues et al. Trends and challenges of the emerging technologies toward interoperability and standardization in e-health communications. *IEEE Communications Magazine*, 2011.
3. WHO. Global report on diabetes. http://apps.who.int/iris/bitstream/10665/204871/1/9789241565257$_eng.pdf[accessed 2016-12-22]$.
4. P. King et al. The uk prospective diabetes study (ukpds): clinical and therapeutic implications for type 2 diabetes. *British Journal of Clinical Pharmacology*, 1999.
5. A. Murakami et al. A continuous glucose monitoring system in critical cardiac patients in the intensive care unit. In *2006 Computers in Cardiology*, pages 233–236. IEEE, 2006.
6. M. Ali et al. A bluetooth low energy implantable glucose monitoring system. In *EuMC 2011*, pages 1265–1268. IEEE, 2011.
7. J. Lucisano et al. Glucose monitoring in individuals with diabetes using a long-term implanted sensor/telemetry system and model. *IEEE Transactions on Biomedical Engineering*, 2016.
8. KAU. Menon et al. A survey on non-invasive blood glucose monitoring using nir. In *ICCSP 2013*, pages 1069–1072. IEEE, 2013.
9. MUH. Al Rasyid et al. Implementation of blood glucose levels monitoring system based on wireless body area network. In *Consumer Electronics-Taiwan (ICCE-TW), 2016 IEEE International Conference on*, pages 1–2. IEEE, 2016.
10. N. Wang and G. Kang. A monitoring system for type 2 diabetes mellitus. In *Healthcom 2012*, pages 62–67. IEEE, 2012.
11. TN. Gia et al. Iot-based fall detection system with energy efficient sensor nodes. In *NORCAS 2016*, pages 1–6. IEEE, 2016.
12. S. Sudevalayam and P. Kulkarni. Energy harvesting sensor nodes: Survey and implications. *IEEE Communications Surveys Tutorials*, 2011.
13. M. Taghadosi et al., L. Albasha, N. Qaddoumi, and M. Ali. Miniaturised printed elliptical nested fractal multiband antenna for energy harvesting applications. *IET Microwaves, Antennas Propagation*, 2015.
14. V. Jelicic et al. Analytic comparison of wake-up receivers for wsns and benefits over the wake-on radio scheme. In *PM2HW2N '12*, pages 99–106. ACM, 2012.
15. L. Gu et al. Radio-triggered wake-up capability for sensor networks. In *RTAS 2004*, pages 27–36, 2004.
16. Kuan-Yu Lin, T. K. K. Tsang, M. Sawan, and M. N. El-Gamal. Radio-triggered solar and rf power scavenging and management for ultra low power wireless medical applications. In *2006 IEEE International Symposium on Circuits and Systems*, pages 4 pp.–5731, May 2006.
17. S. F. Al-Sarawi. Low power schmitt trigger circuit. *Electronics Letters*, 38(18):1009–1010, Aug 2002.
18. M. Ali. *Low Power Wireless Subcutaneous Transmitter*. PhD thesis, 2010.
19. International Commission on Non Ionizing Radiation Protection. Icnirp guidelines for limiting exposure to time varying electric, magnetic and electromagnetic fields (up to 300 ghz). *Health Physics*, 1998.
20. Blood glucose monitoring. Diabetes Australia, https://www.diabetesaustralia.com.au/blood-glucose-monitoring [accessed 2016-12-22].
21. Blood Sugar Level Ranges. Diabetes.co.uk, https://www.diabetes.co.uk/diabetes$_care/blood - sugar - level - ranges.html[accessed 2016 - 12 - 22]$.

Paper VI

Energy Efficient Wearable Sensor Node for IoT-based Fall Detection Systems

Tuan Nguyen Gia, Victor Kathan Sarker, Igor Tcarenko, Amir M. Rahmani, Tomi Westerlund,Pasi Liljeberg, Hannu Tenhunen

Original published in Microprocessors and Microsystems, Elsevier Journal, 56, pages 34-46

Microprocessors and Microsystems 56 (2018) 34–46

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

Energy efficient wearable sensor node for IoT-based fall detection systems

Tuan Nguyen Gia*,[a], Victor Kathan Sarker[a], Igor Tcarenko[a], Amir M. Rahmani[b,c], Tomi Westerlund[a], Pasi Liljeberg[a], Hannu Tenhunen[a]

[a] Department of Future Technologies, University of Turku, Turku, Finland
[b] Department of Computer Science, University of California, Irvine, USA
[c] Institute of Computer Technology, TU Wien, Vienna, Austria

ARTICLE INFO

Keywords:
Internet-of-Things
IoT
Fall detection
Energy efficiency
Wearable devices
Accelerometer
Gyroscope
Magnetometer
nRF

ABSTRACT

Falls can cause serious traumas such as brain injuries and bone fractures, especially among elderly people. Fear of falling might reduce physical activities resulting in declining social interactions and eventually causing depression. To lessen the effects of a fall, timely delivery of medical treatment can play a vital role. In a similar scenario, an IoT-based wearable system can pave the most promising way to mitigate serious consequences of a fall while providing the convenience of usage. However, to deliver sufficient degree of monitoring and reliability, wearable devices working at the core of fall detection systems are required to work for a prolonged period of time. In this work, we focus on energy efficiency of a wearable sensor node in an Internet-of-Things (IoT) based fall detection system. We propose the design of a tiny, lightweight, flexible and energy efficient wearable device. We investigate different parameters (e.g. sampling rate, communication bus interface, transmission protocol, and transmission rate) impacting on energy consumption of the wearable device. In addition, we provide a comprehensive analysis of energy consumption of the wearable in different configurations and operating conditions. Furthermore, we provide hints (hardware and software) for system designers implementing the optimal wearable device for IoT-based fall detection systems in terms of energy efficiency and high quality of service. The results clearly indicate that the proposed sensor node is novel and energy efficient. In a critical condition, the wearable device can be used continuously for 76 h with a 1000 mAh li-ion battery.

1. Introduction

Fall is one of the most trivial reasons causing traumas and serious injuries (e.g. bone fractures or traumatic brain damages caused by head traumas) [1,2]. Elderly people are likely to fall and they often have more serious consequences after falling than people of other ages. According to statistics, 30% of those over 65 and 50% of those over 80 years old fall every year with hazardous results [1]. Because of high morbidity (almost 20% of fall lead to serious traumas), about 40% of all nursing home admissions are related to fall [3].

Treatment of injuries from a fall often lasts over a long period of time and is very costly (e.g. 30,000 US dollars for a serious case in hospital) [4,5]. The proportion is as follows: 63% of fall-related costs accounts for hospitalizations, 21% is for emergency department visits and 16% is for outpatient visits. However, despite the high significance of the problem, timely aid is only delivered in half of the cases. Unreported cases lead to the deterioration of injury which might complicate treatments later.

Fear of falling amplifies the negative post-fall consequences and might decrease patient's confidence [6]. As a result, it limits the patient's activities, reduces social interactions and eventually causes depression [7,8]. Thus, there is an urgent need of fall detection systems. A quick response to the incident might decrease the risk of serious consequences after a fall. Correspondingly, it helps to reduce treatment costs and to increase chance of recovery. In [9], authors have separated fall detection systems into three groups based on wearable devices, ambiance sensors, and cameras. Systems based on wearable devices seem to be more popular because they can detect a fall more accurately regardless of the patient's location (i.e. indoor and outdoor) and do not interfere the patient's privacy and daily activities. Wearable devices often acquire parameters related to motion such as acceleration, rotation and the direction of motion [10].

It is a challenge for wearable sensor nodes to differentiate between fall events and casual daily activities, or to notify doctors in real-time. Due to their resource constraints (e.g. limited power and storage capacity), it is required to have an advanced system which helps to reduce

* Corresponding author.
E-mail addresses: tunggi@utu.fi (T. Nguyen Gia), vikasar@utu.fi (V.K. Sarker), igotsa@utu.fi (I. Tcarenko), tovewe@utu.fi (A.M. Rahmani), pakrli@utu.fi (T. Westerlund), amirr1@uci.edu (P. Liljeberg), hannu@kth.se (H. Tenhunen).

http://dx.doi.org/10.1016/j.micpro.2017.10.014
Received 5 March 2017; Received in revised form 9 October 2017; Accepted 30 October 2017
Available online 02 November 2017

computationally heavy loads on wearable sensor nodes, while maintaining or improving quality of service. Internet-of-Things (IoT) is one of the most suitable candidates for such systems as it consists of a wide range of advanced technologies such as sensing, wireless sensor network and cloud computing for interconnecting virtual objects with physical objects. IoT-based systems can help to reduce wearable devices' burdens by shifting high-computational tasks from wearable devices to their smart gateways. For example, the gateways can perform complex fall detection algorithms (i.e. algorithms based on discrete wavelet transform or data mining). In addition, smart gateways help to improve quality of service by providing advanced services i.e. local storage for storing temporary data or push notification for informing abnormality in real-time.

It is inevitable that IoT can comprehensively help to reduce power consumption of wearable devices by sharing the work load. However, IoT cannot always guarantee a high level of energy efficiency in wearable devices. Other primary issues (i.e. data acquisition and data transmission) causing high energy consumption in wearable sensor nodes must be attentively considered. When a wearable sensor node is energy inefficient, it possibly causes unreliability and reduces quality of service.

In the previous work [11], we have proposed an IoT-based fall detection system. The system comprises of energy efficient sensor nodes, a smart gateway, and a back-end system. The gateway with a Fog layer [12,13] helps to achieve energy efficiency at sensor nodes. In that paper, a sensor node attached to human chest acquires data from a three-dimensional (3-d) accelerometer and transmits the data to the smart gateway via BLE (Bluetooth Low Energy). The main computation (i.e. a customized fall detection algorithm) is performed at the smart gateway since the gateways are powerful in terms of hardware specification and it is supplied by a wall power outlet. The work shows several analysis of primary communication interface buses' power consumption. The results show that SPI (Serial Peripheral interface) consumes less power than I^2C (Inter-Integrated Circuit) and UART (universal asynchronous receiver/transmitter) while SPI's data rates are higher than others (e.g., SPI can support a high data rate of 4 Mbps and more).

The work presented in this paper is a major extension of our recent work published in [11]. In the paper, we aim to study and minimize energy consumption of the wearable sensor node in an IoT-based fall detection system. Furthermore, we analyze undisclosed issues in the previous work. For example, the analysis of advantages and disadvantages of software-based SPI and its impact on energy consumption of a sensor node are presented. Moreover, we analyze energy consumption of the sensor node in various transmission distances and different transmission conditions (e.g. line-of-sight transmission, and transmission via objects). We also investigate and discuss impacts of different sensors (e.g. accelerometer, gyroscope and magnetometer) on both total energy consumption of the sensor node and an accuracy of the fall detection mechanism. We analyze the accuracy of the fall detection system in exceptional cases such as users having abnormal postures. In addition, we discuss and provide comprehensive methods for overcoming limitations (e.g. P2P communication) in the previous work. In this paper, we present the design and implementation of an energy efficient wearable sensor node based on a customized nRF module. The design helps to solve the limitation of P2P communication by offering many-to-many communication between sensor nodes and gateways. Unlike BLE used in the previous work [11] which is connected to the micro-controller via UART, the nRF module in the proposed design uses SPI as its communication bus. Therefore, it incurs a new issue of using several SPI buses simultaneously by a single micro-controller (i.e., SPI communication buses for collecting data from sensors and for transmitting the data via nRF). Therefore, these issues are discussed to find out the most appropriate solution in terms of energy efficiency, feasibility, and complexity. The proposed wearable sensor node is low-cost, lightweight, tiny, energy efficient and flexible. It can be configured to suit to different fall detection algorithms based on motion (e.g. acceleration or angle). The wearable sensor node can provide a viable solution for everyday use without interfering user's daily life. Furthermore, we customize the fall detection algorithm presented in our previous paper for suiting to the proposed sensor node and improving QoS (e.g. the accuracy of the fall detection system).

The rest of the paper is organized as follows: Section 2 includes related work and motivation for this work. Section 3 provides an overview of the IoT-based fall detection system's architecture. Section 4 emphasizes on design principles and reasons behind component and technology selection. Section 5 illustrates the implementation details of the proposed sensor node. Section 6 provides insights about experimental setup and results. Section 7 discusses various issues and findings, and proposes possible solutions. Finally, Section 8 concludes the work.

2. Related work and motivation

Several efforts have been devoted in proposing wearable sensor nodes for fall detection systems. For instance, Casilari et al. use an accelerometer in a smart watch to detect a fall. Accelerometer data is transmitted via BLE from the smart watch to a smart phone which processes data and detects a fall. Then, the smart phone, which acts as a gateway, sends a notification to Cloud via 3G/4G [14]. In another work [15], authors use a depth camera (Kinect) with an accelerometer-based wearable to improve the accuracy of fall detection. Collected data is processed at PandaBoard for detecting a fall in real-time.

Pivato et al. [16] present a wearable wireless sensor node for fall detection. The wearable node whose size is about three times larger than a 2 Euro coin, requires low average current about 15 mA and 25 mA at 50% and 100% duty cycle, respectively. The node is equipped with a 3-d accelerometer ADXL345 and a wireless chip (i.e. CC2420) for gathering and sending acceleration data to a gateway, respectively.

Chen et al. [17] present wearable sensors for a reliable fall detection system. The sensors collect data from low-cost and low-power MEMS accelerometers and send the data via RF. By deploying the sensors at home, the position of the fallen person can be detected.

Biros et al. [18] propose a wearable sensor for a smart household environment. The wearable sensor collects 3-d acceleration and angles from an accelerometer and a gyroscope, respectively. The sensor sends the collected data via ZigBee to Arduino Uno connected to a computer for further processing and detecting a fall.

Erdogan et al. [19] discuss a data mining approach by using k-nearest neighbors for a fall detection system. A wearable device in the system is based on a general purpose board equipped with motion sensors.

In another work [20], the authors present a sensor node based on GSM communication and 3-d accelerometer for a fall detection system. A fall location can be easily detected by the system.

In other works [21,22], authors utilize general purpose boards (e.g. Arduino Uno, Arduino Fio) as the core of fall detection sensor nodes. Although the sensor nodes are low-cost and provide some useful services, they still have several drawbacks such as high power consumption and large physical size. It is known that general purpose boards are often equipped with extra components such as a voltage regulator and a FTDI USB to UART chip ultimately resulting in energy inefficiency.

In several works [14,16–20], fall detection sensor nodes based on motion data often utilizes one or several types of sensors such as accelerometer, gyroscope or magnetometer. The selection of a sensor type or a combination of several sensor types in a single sensor node is mainly focused on functions and features of the sensor(s) while energy consumption of the sensor(s) is not attentively considered. For example, the accelerometer and the gyroscope are often used together in the fall detection applications so as to improve the accuracy of fall detection.

It is known that energy consumption of a sensor node dramatically impacts quality of service. When energy consumption is high, it may cause or lead to negative consequences such as a short operating duration, discontinuation of services or unreliability. However, to the best of our knowledge, the actual issues limiting energy efficiency of a sensor node in an IoT-based fall detection system have not been elaborately investigated. For example, energy consumption of communication buses between a micro-controller and its slave devices (i.e. sensors or a wireless communication module) is not considered in many sensor node designs. Therefore, we investigate energy consumption of communication bus interfaces such as SPI, I2C, and UART. The results showing the impact of primary communication buses on energy consumption of a sensor node can be used as a premise to design the high energy efficient sensor node suiting for different fall detection applications.

The relationship between an IoT-based fall detection sensor node's sampling rate and energy consumption has not been examined in other works. Therefore, in this paper, we analyze the relationship with different configurations and discuss optimal solutions for achieving both high levels of energy efficiency and fall detection accuracy.

A low-power sensor node in fall detection applications often uses BLE as a primary wireless communication protocol [11,23–25]. Although BLE provides many advantages (e.g. low power, fast cyclic redundancy check, and connection improvements), it still has several limitations (e.g. p2p communication, a complex stack with several profiles) which may increase service costs and may not guarantee the highest level of energy efficiency. Therefore, we analyze another low power wireless communication protocol which helps to avoid the limitations of BLE while maintaining high quality of service.

In the paper, we also investigate factors impacting on energy consumption of a wearable sensor device. These factors are such as a micro-controller, motion sensors (accelerometer, gyroscope, and magnetometer), sampling rate, wireless transmission data rate, transmission distance, and software. By applying an optimal combination of hardware design and software techniques, it is possible to provide a novel tiny, and light-weight wearable sensor node with a high level of energy efficiency.

3. Overview of an IoT-based fall detection system's architecture

An overview of an IoT-based fall detection system is presented with the purpose of showing the role and the hierarchical position of wearable devices in the system. The system architecture shown in Fig. 1 consists of three main parts including wearable sensor nodes, a gateway and a back-end system.

A sensor node of an IoT-based fall detection system is responsible for acquiring motion data (i.e. acceleration or rotation angle) and transmitting the data via a wireless communication protocol to a smart gateway. Depending on particular fall detection systems, the collected data can be pre-processed or kept intact before being transmitted. In most of the cases, collected data (raw data) is transmitted without preprocessing by complex algorithms or methods (i.e. wavelet transformation or neural filtering) [26] because pre-processing with complex mechanisms like fall detection based on k-nearest neighbor algorithm requires significant computational power. Correspondingly, energy efficiency must be sacrificed and latency dramatically increases for running such complex algorithms at a sensor node. In order to avoid these issues, complex algorithms are implemented and run at smart gateways [27,28].

In addition to primary tasks of receiving data from sensors and transmitting the data to Cloud servers, a smart gateway with a fog layer provides advanced services such as push notifications, local storage, web-host, and fall detection. Complex algorithms can be run effectively with low latency at the fog layer of a smart gateway due to their advantages of the constant power supply, embedded operating system, and powerful hardware (e.g. a gateway is often about 50–100 times more powerful than a sensor node). Correspondingly, fall can be detected and notified to doctors or caregivers in real-time.

A back-end system consists of Cloud servers and terminals (i.e. end-user's Internet browsers or mobile phone applications). Via the back-end system, doctors, and caregivers can monitor a patient in real-time or history of patient records remotely. In addition, the back-end system may help doctors in disease treatment by providing analyzed data and history of records.

4. Sensor node design

A sensor node for an IoT-based fall detection system primarily comprises of a micro-controller, a motion sensor or sensors, and an nRF block whose connections are shown in Fig. 2. The micro-controller performs main tasks of gathering data from sensors, formatting and transmitting the collected data to the nRF block, and controlling sensors and I/O interfaces (i.e. SPI, I^2C or UART). It consumes a large portion of total power consumption of the sensor node. Therefore, it is important to apply an optimal micro-controller for performing mentioned tasks efficiently in terms of latency and energy consumption.

In our application, a 8-bit micro-controller is more suitable than a 32-bit micro-controller. Based on experiments run by Atmel [29], an Atmel 8-bit AVR device is more efficient than an Atmel ARM Cortex®M0 + based 32-bit MCU in terms of hardware near-functions. For example, an Atmel 8-bit AVR device requires 12 cycles to receive one byte from SPI using interrupt while an Atmel ARM Cortex®M0 + based 32-bit MCU requires 33 cycles for performing the same task. When running a recursive 15-stage Fibonacci algorithm, a 8-bit AVR micro-controller needs 70 bytes of stack while the 32-bit ARM-based device needs 192 bytes [29]. In simple applications such as receiving data from SPI using interrupt, assuming a SPI data bandwidth of 80 kbps, the 8-bit AVR micro-controller consumes 36.1 uA while the 32-bit ARM-based micro-controller consumes 48.1 uA. During sleep mode, a 8-bit AVR micro-controller consumes 100 nA while a 32-bit ARM-based micro-controller consumes 200 nA [29].

In [11], we have shown that a 8-bit AVR ATMega micro-controller is capable of successfully performing several tasks (e.g. data gathering and data transceiving) without infringing latency requirements of real-time monitoring systems. The 8-bit AVR micro-controller supports several clock frequencies such as 4, 8, 16 and 20 MHz which completely

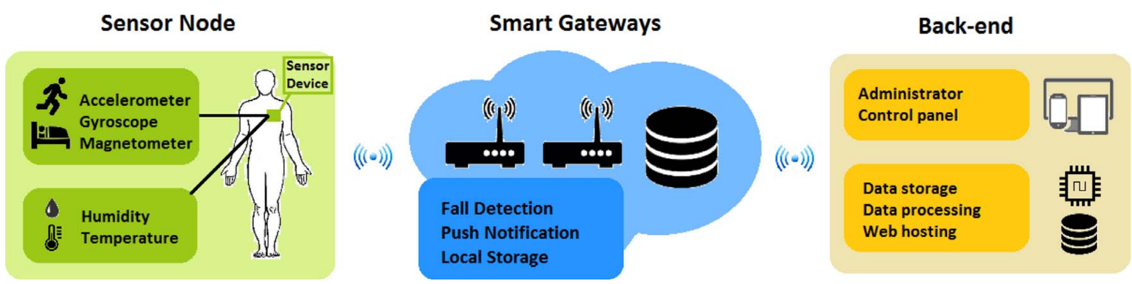


Fig. 1. The three layers of system architecture: edge, fog and cloud. Measurements collected by wearable devices in the edge layer are processed in the fog layer while cloud layer provide information to caregivers.

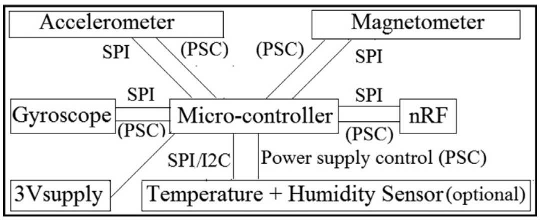


Fig. 2. Connection of sensor node's primary components.

fits to our fall detection application using a few hundred samples per second. In addition, it consumes low power in an active mode, and supports several sleep modes and advanced features for saving power consumption. The micro-controller supports all popular communication bus interfaces (e.g. UART, SPI, and I^2C). Furthermore, it is small and low-cost (around 1–2 dollars). Therefore, it is completely suitable for our fall detection application.

Depending on particular fall detection algorithms running on smart gateways, one or several types of motion sensors (such as accelerometer and gyroscope) can be integrated in a sensor node [30]. A combination of several types of motion sensors (e.g. accelerometer, gyroscope, and magnetometer) altogether may help to improve the accuracy of a fall detection system but it causes higher energy consumption. Fortunately, these sensors can be controlled by software (e.g. entering sleep mode) for saving energy. For example, replacing a 3-d accelerometer (e.g. ADXL345 accelerometer) in a sensor node by a combination of a 3-d accelerometer, a 3-d gyroscope and a temperature sensor (e.g. ADXL345, Kionix KXG07, and STML20), energy consumption during the idle mode in a second only increases about a few µW (e.g. less than 10 µW). In this paper, in order to provide a flexible and low-energy wearable sensor node suiting to different IoT-based fall detection systems, three motion sensors including 3-d accelerometer, 3-d gyroscope, and 3-d magnetometer are integrated in our sensor node. When sensors are not in use, they are forced to sleep. In addition, in order to provide the optimal sensor node in terms of both energy efficiency and the fall detection accuracy, a comprehensive analysis and a discussion of sensor node in different configurations are presented in Section 6 and Section 7. It is known that sampling rates and communication protocols (UART, I^2C and SPI) dramatically impact on energy consumption of sensors [11]. Often, these sensors support several sampling rates of which low sampling rates (50–100 Hz) can be run in a low-power mode and high sampling rates are run in a normal mode. However, when the sampling is too low, it negatively impacts on the accuracy of fall detection. A relationship between sensors sampling rate and energy consumption is investigated in Section 6 for finding an appropriate sampling rate which provides a high level of fall detection accuracy while consuming low energy.

An nRF module consisting of an nRF integrated circuit (IC) and an on-PCB printed antenna is chosen for the design because it consumes less energy while supporting high data rates. Also, comparing to Wi-Fi, XBee and Bluetooth, nRF is more suitable for the sensor node because it consumes the least power (i.e. about 5–10%, 5–10%, and 80% less power than BLE, XBee, and Wi-Fi [31]) and it supports software customization enhancing a sensor node's flexibility. Depending on particular application requirements, a transmission data rate can be customized. In some cases, it can transmit data with a data rate of 2 Mbps.

According to our previous work [11], SPI consumes less power than I^2C and UART communication interfaces at the same data rate. Therefore, SPI is utilized for connecting the micro-controller with motion sensors as well as the nRF module. However, applying multiple SPI communication bus causes some difficulties in data management and data verification.

5. System implementation

5.1. Sensor node implementation

A sensor node must be able to operate reliably for a long period of time. To achieve this, each component of the sensor node must be energy-efficient in both hardware and software. Based on our previous work's investigation [11] and the specification of an AVR ATMega328P micro-controller, a 8-bit AVR ATMega328P micro-controller is suitable for the sensor node. In the implementation, all unneeded interfaces (e.g. UART, I^2C) and internal modules (blocks) of the micro-controller are intentionally disabled for reducing energy. For example, unneeded internal modules (i.e. Serial, ADC, or brownout detection) are turned off. Similarly, necessary interfaces and modules are forced to be disabled in most of the time. They are merely enabled or waken up only for performing their tasks. Energy consumption of the sensor node with and without disabling unneeded modules is shown in Table 3.

The micro-controller supports up to 20 MHz. However, the higher clock frequency is applied, the higher power the sensor node has to be provided because the micro-controller requires higher voltage supply and draws more current when running at high frequencies. For instance, at 16 MHz an ATMega328P micro-controller needs 5 V and consumes about 57.6 mJ for running a test function while at 8 MHz an ATMega328P micro-controller only needs 3 V and consumes approximately 46.8 mJ for running the same test function. In both cases, an array of 100 values is retrieved and a sum of two adjacent values is written back to the array. When supplying the micro-controller 2.2–2.5 V for running at 4 MHz, the micro-controller consumes less power than at 8 MHz. However, if applying 2.2 V power supply, it would be incompatible for other primary components of the sensor node. For example, sensors (e.g. MPU-9250) and nRF require 3 V power supply for a stable operation. In order to solve the incompatibility issue, the micro-controller must be supplied with 3 V or the sensor node must be equipped with a voltage regulator converting a higher voltage down to 3 V. Correspondingly, in both cases, it may waste 15–30% of total power consumption while it may not operate stably. Therefore, running the micro-controller at 8 MHz is suitable for our sensor node because extra components like voltage regulator(s) can be removed while a high clock frequency can be utilized. Another reason of choosing 8 MHz and 3 V power supply is that when a 3 V battery drains, the voltage supply from the battery may drop until around 2.7 V which is still suitable for the sensor node. In addition, the choice of 8 MHz and 3 V is suitable for extending the sensor node for the future use such as collecting e-health data (e.g. ECG, EMG and EEG). Analog front-end ICs for these signals often require 3 or 3.3 V power supply.

MPU-9250, which is 9-axis MotionTracking sensor combining a 3-d accelerometer, a 3-d MEMS gyroscope, a 3-d MEMS magnetometer and a Digital Motion Processor hardware accelerator engine, is used in the implementation for sensing motion data. The MPU-9250 sensor is fully programmable and able to support low-power and sleep modes. For example, the gyroscope sensor consumes 8 µA in sleep mode. One of advantages is that each internal module such as accelerometer, gyroscope, and magnetometer can be controlled separately. Correspondingly, the sensor node can be customized for particular fall detection applications without sacrificing energy efficiency of the sensor node intensively. Depending on the applications or scenarios, some internal modules can be activated while others can be in sleep modes. In the implementation, several scenarios described in Section 6 are applied for investigating energy consumption of our sensor nodes and the accuracy of fall detection. The sensor requires a supply voltage from 2.4 V to 3.6 V. The MPU-9250 sensor supports both SPI and I^2C.

An nRFL2401 module, which is a low-power transceiver operating in ISM frequency band from 2.4 GHz to 2.4835 GHz, is used. The module integrated with an embedded base-band protocol engine supports several operating modes. For example, the module can operate at 250 kbps, 1 Mbps, and 2 Mbps. In the implementation, 250 kbps is

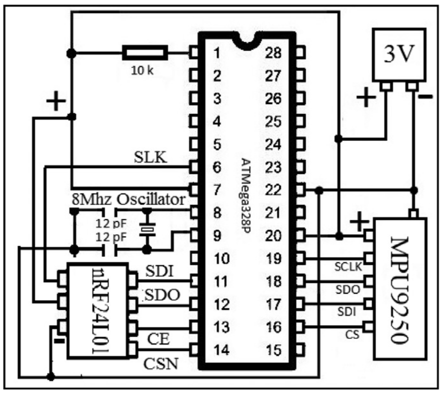


Fig. 3. Minimal setup of a sensor node using both hardware and software based SPI.

preferred because it fulfills the data rate requirement of the system and consumes the lowest energy than other data rates. The module is connected to the micro-controller via SPI.

In addition, for evaluating energy consumption of the sensor node when using SPI and software SPI, two different sensor nodes based on SPI and software SPI are implemented. The first node uses a combination of SPI and software SPI while the second node merely uses SPI. Minimal setup of these nodes is shown in Figs. 3 and 4.

Finally, the proposed wearable sensor node is built, as shown in Fig. 5. The wearable sensor is tiny, light-weight and low-cost. The total cost of the wearable sensor node is less than 11 Euros in which a motion sensor MPU9250 and an nRF24L01 module cost about 5 Euros and 2 Euros, respectively.

5.2. Gateway and back end implementation

A gateway is implemented by a combination of an nRF transceiver and Raspberry Pi [32]. An nRFL2401 module described above is used as an nRF transceiver of the gateway. The module is connected to the Raspberry Pi via SPI.

Several algorithms, presented in [11,33], are applied in a smart gateway for testing functionality of sensor nodes. These algorithms are chosen because they can be replicated easily in the gateway for the verification purposes and they provide a high level of accuracy in detecting fall. These algorithms operate based on acceleration and angular motion which vary in time during a fall, as shown in Fig. 7. In these algorithms, different types of filters are used for removing noise from the collected data. Then, the fall-related parameters such as Sum Vector Magnitude (SVM) and differential SVM (DSVM) are calculated by the formula shown in Equation (1,2 and 3). It is noted that the Eq. (2) is not applied for gyroscope.

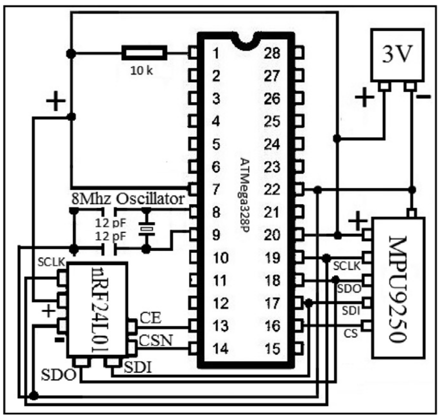


Fig. 4. Minimal setup of a sensor node using hardware SPI only.

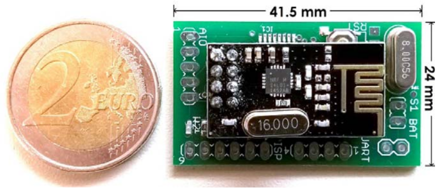


Fig. 5. Prototype of proposed sensor node beside a 2 Euro coin for size comparison.

$$SVM_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \tag{1}$$

$$\Phi = \arctan\left(\frac{\sqrt{y_i^2 + z_i^2}}{x_i}\right) * \frac{180}{\Pi} \tag{2}$$

$$DSVM_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \tag{3}$$

SVM: Sum vector magnitude

i: sample number

x,y,z : accelerometer value or gyroscope value of x, y, z axis

Φ : the angle between y-axis and vertical direction

DSVM: Differential sum vector magnitude

These fall-related parameters will be further processed or compared with several pre-defined thresholds. If the processed data or fall-related parameters are larger than the predefined thresholds, a fall is detected. In details, each sensor has a specific threshold. For example, SVM of 3-d acceleration is around 1 g in most of the cases (e.g. standing, sitting or walking). When a patient falls, the SVM value increases instantly more than 1.9 g at the fall moment. Therefore, the threshold value can be defined as 1.6 or 1.7 g. Similarly, the threshold values of 3-d gyroscope and 3-d magnetometer can be defined. In this paper, we do not focus on fall detection algorithms in a smart gateway. Therefore, we customize the threshold-based fall detection algorithm presented in our previous paper [11]. The customized algorithm includes several stages such as filtering, calculating fall feature parameters, combining fall feature parameters from several sensors, and comparing with two-level thresholds. The detailed flow of the customized fall detection algorithm is shown in Fig. 6. Based on our experiments and results shown in Section 6, relying on data collected from a single sensor type does not provide a high level of accuracy in some cases. Therefore, in the paper, we add two extra stages to the fall detection algorithm. The first stage combines and analyzes several fall feature parameters from several sensors such as 3-d accelerometer, 3-d gyroscope, and 3-d magnetometer. In case that only two sensor types are used (e.g. accelerometer and gyroscope), the parameters from the absent sensor will be ignored. After the first stage, there will be two cases: (i) if all fall feature parameter values from collected sensors are larger than their own

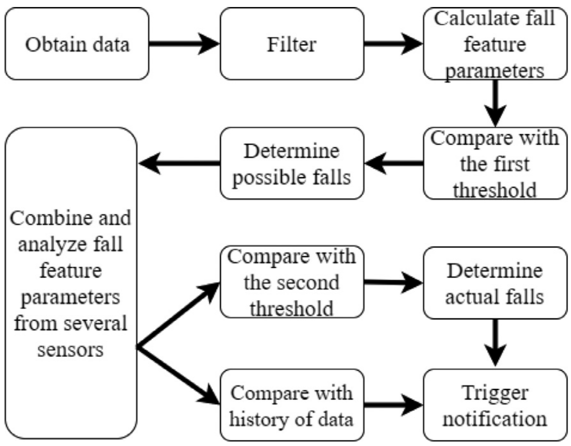


Fig. 6. Fall detection algorithm flow.

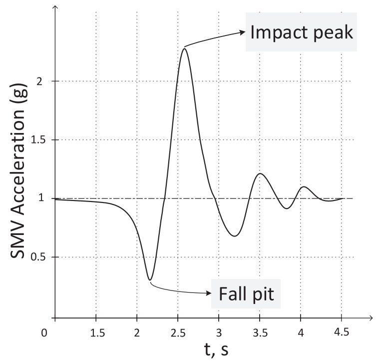


Fig. 7. Acceleration changes in time during a fall.

thresholds, they are compared with their second thresholds. If one of the comparison results shows that the fall feature parameter value is larger than the second threshold, it triggers notification for informing a fall; (ii) if one of the fall feature parameter values is less than the first threshold while other parameters are larger, they are compared with their own values in the past 1 and 2 s for finding dysfunctional or unstable sensor(s).

In our implementation, 1.6 and 1.9 g are used as the first and the second threshold for SVM of 3-d acceleration while 130 and 160 deg/s are used as the first and the second threshold for SVM of 3-d gyroscope. Depending on particular requirements of a fall detection system, it is possible to run one of complex threshold-based algorithms presented in [34–37] or complex machine-learning based algorithms presented in [38–40] at our smart gateway. In such cases, our sensor nodes are still compatible and able to operate efficiently.

The push notification is implemented at Cloud and an Android application via Google's Push API. When the gateway detects a fall, it sends a message (a patient id and time when the patient falls) to the Push service at Cloud servers which then remotely notifies responsible doctors and caregivers in real-time. In addition to mentioned services, smart gateways are implemented with a Fog layer for providing advanced services such as local storage, local host with user interface, data processing, data compression, security, channel managing, categorization. However, in this paper, these Fog services provided at smart gateways are not our main focuses. Therefore, only an overview of Fog and Fog services are presented in the paper while details of the Fog layer and services including description, structure, design and implementation are presented in our other papers and book [26–28,41–44].

In our system, all collected data from sensor nodes are temporarily stored in local database of smart gateways. The local database helps to avoid losing data when the connection between smart gateways and Cloud servers is interrupted. When the connection is re-established, gateways send all recorded data in the database to Cloud. The database is implemented with MongoDB. Categorization service is used to distinguish Intranet users and Internet users with the purpose of reducing latency of services. For example, when the system detects a fall, it checks the status of a doctor or a caregiver responsible for the person falling. If he or she is currently connected to the local network, the system sends the push notification message directly from smart gateways to him or her. This helps to avoid a long latency of transmission via Cloud. The service is implemented by a combination of scanning service and database. The "iw" package helps to check the status of connected users in the local network. Then, the results are stored in the database. The categorization service and local host with user interface allow doctors or caregivers access real-time data directly at the gateways. In order to implement the local host, HTML5, CSS, JavaScript, JSON, Python, and XML are used. Channel management helps to avoid channel conflict by assigning free channels for newly connected sensor nodes. The channel management service triggers the push notification service for informing to system administrators in case of channel conflict.

6. Experimental setup and results

Energy consumption of a sensor node for a fall detection IoT system is calculated with Eq. (4) [45]. Total energy consumption of the node is equal to a sum of energy consumed during operating and waiting.

$$E = V \times I(w) \times t(w) + V \times I(o) \times t(o) \tag{4}$$

E : Total energy consumption (mJ)
V : Voltage supply
$I(w)$: Average current draw during waiting time (mA)
$I(o)$: Average current draw during operating (mA)
$t(w)$: Waiting time (s)
$t(o)$: Operating time (s)

In order to provide an overview of sensor nodes used in measurements and comparisons, their hardware specifications are shown in Table 2. In the experiments, each measurement is carried out during 5–10 min and a professional power monitoring tool from Monsoon Solution is used [46]. This tool is able to accurately monitor minimum, maximum and average voltage, current draw, power consumption of a sensor node. In addition, Monsoon provides an advanced utility for plotting monitored values in time series, which helps us to detect abnormality during measurements. Although average power consumption in one second and energy consumption per second is identical, to maintain consistency throughout the paper, energy consumption per second is reported instead of power consumption retrieved from the monitor.

In order to determine the suitable method for waking up the micro-controller from deep sleep or normal sleep modes, several general-purpose timers and a watchdog timer are used. In the experiment, the nRF module is not active and the sensor node acquires data from different sensors (i.e. accelerometer, gyroscope, and magnetometer) via 1 Mbps SPI with a data rate of 50 samples/s. Energy consumption of the sensor node in one second is captured and shown in Fig. 8. Results from the Fig. 8 show that energy consumption of the sensor node significantly decreases when using a watchdog timer instead of general-purpose timers. The main reason is that a watchdog timer can wake up the micro-controller from the deepest sleep mode(s) whilst other timers cannot. Results show that an 8-bit timer is more energy efficient than a 16-bit timer. Since the maximum data rate supported by the watchdog timer of ATMega328P is 62 samples/s, a data rate of 50 samples/s is most suitable for the wearable sensor node.

For evaluating energy consumption of the wearable sensor node when using different protocols, 50 samples/s data is acquired from several sensors such as accelerometer, gyroscope and magnetometer via SPI and I^2C. In the experiment, the nRF module is not active and energy

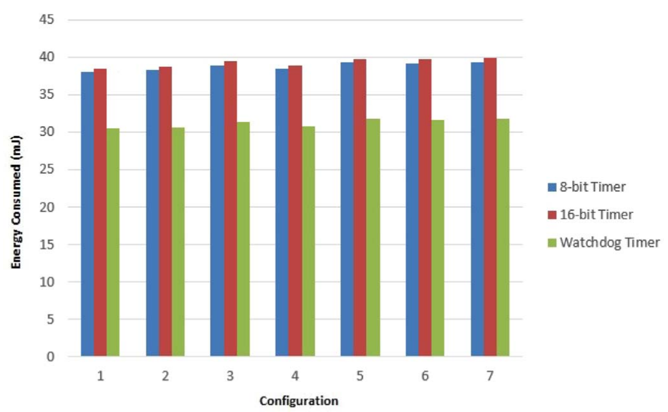


Fig. 8. Energy consumed per second when collecting data from several sensors at 50 samples/s via different techniques.

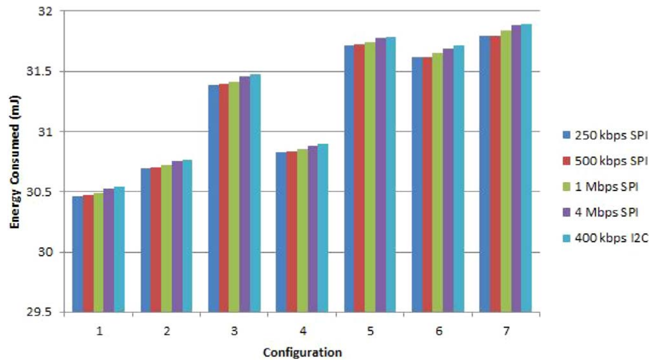


Fig. 9. Energy consumed per second of the sensor node when collecting data from multiple sensors at 50 samples/s using SPI and I^2C.

consumption of the sensor node shown in Fig. 9 is measured for one second. It can be seen that SPI consumes less energy than I^2C in most of the cases.

We compare energy consumption of several sensor nodes based on general purpose platforms and our sensor node. Energy consumption is measured when collecting data from a 3-d accelerometer for one second with a data rate of 50, 100, 200 and 500 samples/s via SPI. In the experiment, software-based techniques for energy efficiency are not applied and all modules for wireless communication (e.g. nRF and BLE) are neither active nor used. Results shown in Fig. 10 indicate that the proposed sensor node consumes the least energy for collecting 3-d acceleration via SPI in all applied data rates. One of the reasons for a high level of energy efficiency in the proposed sensor node is that the sensor node is designed with a minimum number of required components (unnecessary components e.g. FTDI or voltage regulators are removed from the design).

In order to analyze energy consumption of the sensor node, several configurations shown in Table 1 are used. An accelerometer module MPU9250 [47] can support a data rate up to 4000 Hz. However, the low-power mode of the accelerometer cannot be applied at this high data rate. Obviously, the normal operating mode requires more energy than the low-power mode. Therefore, the low-power mode of the accelerometer, which supports a maximum data rate of 500 Hz is applied. For investigating energy consumption of the accelerometer module, several data rates lower than 500 Hz are used in the experiments. Similarly, the low data rate and the low-power mode are applied for gyroscope and magnetometer. In the experiment, several data rates (i.e. 50, 100, 200, and 500 samples/s) are applied to collect data via 1 Mbps SPI in different configurations and the nRF module is not active. Results shown in Fig. 11 indicate that for data rates in a low-power mode, a 3-d accelerometer consumes the least amount of energy among three sensors while a 3-d magnetometer consumes the most energy. In addition, the results reveal that utilizing both accelerometer and gyroscope

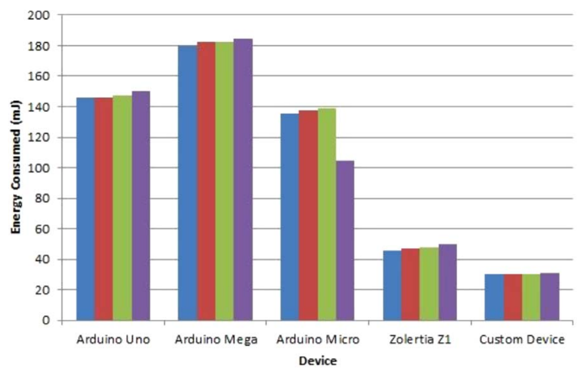


Fig. 10. Energy consumption per second of different devices when collecting 3-d accelerometer data at different sampling rates via SPI.

Table 1
Scenarios setup.

	Accelerometer	Gyroscope	Magnetometer
Configuration 1 (Conf 1)	X		
Configuration 2 (Conf 2)		X	
Configuration 3 (Conf 3)			X
Configuration 4 (Conf 4)	X	X	
Configuration 5 (Conf 5)		X	X
Configuration 6 (Conf 6)	X		X
Configuration 7 (Conf 7)	X	X	X

Table 2
Devices specifications.

Device	Micro-controller (MHz)	Flash (KB)	SRAM (KB)	Voltage (V)
Arduino Uno	ATmega328P-PU (16)	32	2	5
Arduino Mega	ATMega1280 (16)	128	8	5
Our sensor node	ATmega328P-PU (8)	32	2	3
Arduino Micro	ATmega32U4 (16)	32	2.5	5
Sensor node in [18]	ATMega32L (8)	256	8	5
Sensor node in [19]	ATMega128L (8)	128	4	3
Sensor node in [16]	MSP430F2617 (8)	92	8	3.7
Sensor node in [21]	MSP430 (8)	48	10	3
Sensor node in [20]	MSP430F1611 (8)	48	10	3.7
Z1	MSP430 (8)	92	8	3

Table 3
Energy consumption of the sensor node when collecting 50 samples/s acceleration data via SPI in Mode 1 and Mode 2 during a second.

	Mode 1(mJ)	Mode 2(mJ)
Energy consumption of a sensor node	30.47	27.98

Mode 1: when unneeded modules are turned on or enabled
Mode 2: when unneeded modules are turned off or disabled

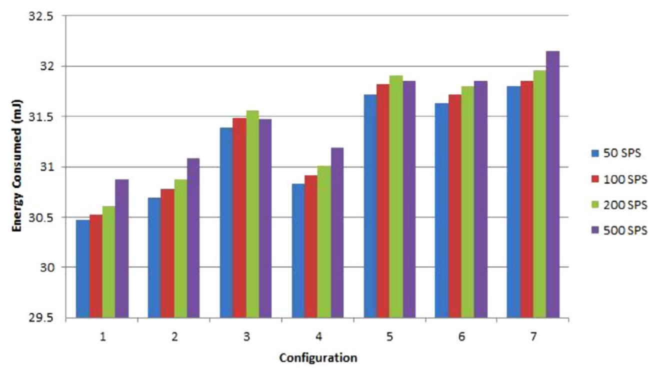


Fig. 11. Energy consumed per second when collecting multiple sensor data at different sampling rates via SPI.

modules at the same time causes a slight increase in energy consumption when compared with applying a single sensor (i.e. 3-d accelerometer or 3-d gyroscope).

In order to investigate the impact of software-based techniques on energy consumption of the sensor node, we measure energy consumption of the sensor node for one second with two cases: (i) when unneeded modules (e.g. UART, ADC, I^2C, and brownout detection) of a micro-controller are turned on or enabled; (ii) when unneeded modules are turned off or disabled. In the experiment, the sensor node collects 3-d acceleration data at a rate of 50 samples/s in a second via 1 Mbps SPI and the nRF module is not active. Results from Table 3 show that energy required by the sensor node can be reduced by about 8–10% when unneeded internal modules of a micro-controller are disabled.

The nRF module can be configured for two modes of communication- one-way and two-way. In one way communication, a sensor node only sends data to a receiver (i.e. a gateway) regardless of the success of a transmitted package. On the other hand, in two-way communication, a sensor node sends data to a gateway and waits for an acknowledgement message from the gateway. If it receives an acknowledgement message from the gateway, it continuously sends new data to the gateway. In contrast, when it does not receive any acknowledgement message from the gateway, it automatically increases transmission power and re-sends the package which was not successfully received by the gateway in the previous attempt. There is a trade-off between QoS and energy consumption when applying these communication types. Two-way communication guarantees that a data package is received at a gateway after being sent by a sensor node. However, it causes higher energy consumption because a sensor node's down-link must be active to wait for the response package. In contrast, one-way communication consumes less energy because its down-link is disabled, however, the best possible QoS cannot be guaranteed. Correspondingly, these communication types must be attentively investigated for exposing the optimal configuration providing a high level of QoS and energy efficiency.

At first, energy consumption of a sensor node in a two-way communication is measured. Several sensor nodes are used during the experiments in which each sensor node is attached to a patient's clothing at the middle of the chest area. Sensor nodes collected 3-d accelerometer data with a data rate of 50 samples/s via 1 Mbps SPI and transmit the data via nRF to a gateway which is fixed in a single room. Several distances between sensor nodes and a smart gateway such as 5, 10 and 20 m are applied for evaluating variations of sensor nodes' energy consumption. In each measurement, both cases of the line of sight transmission and transmission via blocked objects (i.e. door and wall) are applied. Results of two-way communication are shown in Table 4. Energy consumption of a sensor node in this case includes both energy consumption of transmitting and receiving. The results indicate that energy consumption of the sensor node increases when the distance between the sensor node and the gateway increases.

Obviously, two-way communication often provides a high level of QoS because a loss package is always re-transmitted. In case of higher distances or transmission way blocked, transmission power in two-way communication is increased for ensuring a successful data transfer. In the experiments, the transmission power is retrieved effortlessly via the monitor utility. For achieving such a high level of QoS, the transmission power used in two-way communication can be re-applied into a case of one-way communication. Similarly, the same test-bed with similar distances (5, 10, and 20 m) is applied for one-way communication. Results shown in Table 5 indicate that energy consumption of a sensor node in case of one way communication is much less than in case of two-way communication in both situations (e.g. line-of-sight transmission and transmission through blocked objects) even though transmission power of a sensor node in case of one-way communication is forced to be increased for assuring a high level of QoS.

Table 4
Energy consumption of the sensor node when collecting 50 samples/s acceleration via SPI and transmitting the data via nRF in two-way communication to a gateway during a second.

Distance	5 m(mJ)	10 m(mJ)	20 m(mJ)
Line-of-sight transmission	41.96	42.28	43.99
Transmission through blocked objects	43.21	43.94	45.80

Table 5
Energy consumption of the sensor node when collecting 50 samples/s acceleration via SPI and transmitting the data via nRF to a gateway (one-way communication) during a second.

Distance	5 m(mJ)	10 m(mJ)	20 m(mJ)
Line-of-sight transmission	28.71	29.01	30.72
Transmission through blocked objects	29.81	30.48	32.61

Although many slave devices (i.e. sensors) can be connected to a master (i.e. micro-controller) via a single SPI interface, it is challenging to perform such a connection in some cases. For example, hangout wires may occur in the layout design when connecting several devices to a single SPI port or SPI libraries of slave devices may conflict. In order to avoid these issues, software SPI which written in C utilizes Pulse Width Modulation (PWM) pins for replicating an SPI transmission, can be used. In the experiments, energy consumption of two different sensor nodes is measured in which the first node uses only SPI and the second node uses a combination of SPI and software SPI. Both nodes use their watchdog timer for waking up the micro-controller from the deep sleep mode(s). Several distances (5, 10, 20 m) are applied and the data is transmitted via nRF with a line of sight transmission during these experiments. Results shown in Table 6 show that hardware SPI is more energy efficient than software SPI in all experimental cases. For avoiding missing package when applying one-way communication, we apply the same method of reusing transmission power in case of two-way communication into one-way communication for both sensor nodes (nodes using hardware SPI and nodes using both hardware and software SPI) in each measurement. Depending on particular distances, the transmission power is different.

With the purpose of providing a complete view of energy consumption of the sensor node, we investigate energy consumption of the sensor node in different configurations shown in Table 1. In the experiments, the sensor node acquires different data from one or several sensor types with a sampling rate of 50 samples/s and transmits the data via nRF with a line-of-sight transmission condition. Energy consumption is measured for one second. Results of the experiments shown in Table 7 show that 3-d accelerometer consumes the least energy while 3-d gyroscope and 3-d magnetometer consume higher and the most energy, respectively. In addition, the results reveal that applying two types of sensors in the sensor node consumes about 12–16% more energy and energy consumption of the sensor node equipped with two or three types of sensors is slightly different, approximately 3–5%.

When supplying with a 1000 mAh 3 V lithium battery having a size of 32 mm * 43 mm * 5 mm and a weight of 30 g, the wearable sensor node can operate for about 76–90 h depending on particular conditions. In the paper, we simply categorize into three conditions including the worst, normal and the best condition. In the worst condition, transmission path between a sensor node and a gateway is blocked with different indoor objects and doors. For example, the sensor node is placed is a room while a gateway is located in another room and these rooms are separated by walls and doors. In normal situation, there are very few objects in transmission path. In the experiment, some high wardrobes are placed in a room. In the best situation, the transmission path is clear (i.e. line-of-sight) and the experimentation room is almost

Table 6
Energy consumption of the sensor node when collecting 50 samples/s acceleration and transmitting the data to an nRF block via software and hardware SPI during a second.

	Distance		
Method	5 m (mJ)	10 m (mJ)	20 m (mJ)
Software SPI	30.98	31.93	33.7
Hardware SPI	28.68	29.01	30.72

Table 7
Energy consumption of the sensor node when collecting 50 samples/s motion data via SPI and transmitting the data via nRF with a distance of 20 m in different configurations.

\Distance	20 m (mJ)
Conf 1	30.72
Conf 2	30.95
Conf 3	32.1
Conf 4	34.53
Conf 5	35.56
Conf 6	35.35
Conf 7	36.68

empty (e.g. only short tables and chairs). In these experiments, the sensor node is placed on the top the battery for forming a compact device and it takes about 2–5 h to charge the battery depending on the current supplied. The sensor node collects data from 3-d accelerometer, 3-d gyroscope, and 3-d magnetometer with a data rate of 50 samples/s via 1 Mbps SPI and transmits the data via nRF. When the sensor node is not active (e.g. all modules such as nRF and sensors are disabled), it consumes about 3.6 mW. Results of these experiments are shown in Fig. 14. It can be seen that the sensor node can operate up to 76 h in the worst condition while its operating time can reach up to 90 h in the best condition. It is recommended that, the transmission power of the sensor node should be configured for suiting the worst condition because it can provide a high level of QoS for all cases. In the worst condition, some levels of energy efficiency (about 5–8%) must be sacrificed.

For providing a comprehensive view of the wearable sensor node, the sensor node is compared with other nodes proposed by other research in terms of energy consumption, size, weight and flexibility. In our context, high flexibility indicates that a sensor node can be customized easily and flexibly for suiting to different fall detection IoT-based systems and vice versa. Results shown in Table 8 summarize that our wearable sensor node is tiny, light-weight and energy efficient. In addition, the sensor node, which is highly flexible, suits to different IoT-based fall detection systems using motion data whilst other nodes are not completely suitable for other fall detection systems. Correspondingly, a user can wear the sensor node 24/7 without interfering daily activities.

In addition to the previous experiments, to evaluate quality of acquired signals at the gateway, 6 more measurements have been carried out. In details, each measurement uses 5 separate sensor nodes placed on the body of five volunteers for acquiring both 3-d accelerometer data and 3-d gyroscope data. Then the data is transmitted via nRF to the gateway with a line-of-sight transmission path condition. Each measurement is carried out for 30 min. Data received at the gateway is applied for the fall detection algorithms mentioned in Section 5.2. Results from the experiments shown in Fig. 12 reveal that the sensor node operates reliably in different scenarios (i.e. different daily activities) in most of the cases. In order to provide an incisive view of the data received at the gateway, the data is graphed in MatLab. In addition, the exceptional case is shown in Fig. 13.

Table 8
Devices specifications.

Device	Energy consumption	Size	Weight	Flexibility
Arduino Uno	High	Large	Medium	Partly
Arduino Mega	High	Large	Medium	Partly
Arduino Micro	Medium	Small	Light	Partly
Sensor node in [18]	Low	Medium	Medium	Partly
Sensor node in [19]	Medium	Medium	Medium	Partly
Sensor node in [16]	Low	Medium	Light	Partly
Sensor node in [21]	Low	Medium	Medium	Partly
Sensor node in [20]	High	Large	Medium	Partly
Z1	Low	Medium	Medium	Partly
Our sensor node	Low	Small	Light	Completely

Fig. 12 shows 3-d accelerometer data and 3-d gyroscope data whose sampling rate is 50 samples/s during a user's daily activities such as standing, sitting and walking. As seen in Fig. 12, quality of data collected from 3-d accelerometer and 3-d gyroscope is high in different cases such as "stand still", "sit still", "stand with body movements", "sit with body movements" and "walking". Due to some movements of the upper part of the body when walking, SVM of 3-d accelerometer values and 3-d gyroscope values fluctuate. However, the fluctuation of SVM of 3-d accelerometer values and 3-d gyroscope values is not large enough to dramatically impact on the result of fall detection since a variation of the fluctuation is much smaller than the peak magnitude of SVM of 3-d accelerometer values and 3-d gyroscope values when a user falls shown in Fig. 13. In other cases, the fluctuation of SVM from 3-d accelerometer and 3-d gyroscope is small, around 1 g and 0°s, respectively, which are similar to expected values discussed in Section 5.

Fig. 13 shows SVM of 3-d accelerometer values and 3-d gyroscope values with a sampling rate is 50 samples/s when a volunteer falls. It can be seen that magnitude of SVM increases dramatically in all cases. In case of 3-d accelerometer, the peak of SVM goes over 1.6 g for all of considered cases. In some of cases, the peak even reaches up to 2.5 g or 3 g. In case of "walk and fall", when a person falls, SVM of 3-d accelerometer reaches up to 3.5 g. However, when a person stands still after falling, the SVM value does not go back to 1 g (the expected value) but it remains at 2 g. However, in reality, the 2 g value is not the correct value. The reason might be incorrect calibration in the 3-d accelerometer or the large movement of the body when standing up. In this case, if the fall detection algorithm sets the threshold for detecting a fall at 1.8 g, the fall detection results are completely incorrect. In contrast, SVM from 3-d gyroscope, which is around 0°s, is correct as expected.

It is known that all artifacts with large angles negatively impact on the fall detection decision of the gyroscope-based system because the noise amplitude caused by movement artifacts are sometime larger that pre-defined thresholds used for determining a fall case. Therefore, to validate the system in such a case, the system is applied to a person who has an unbalance stance (e.g. moving his shoulder, hands and an upper part of his body with large angles) while walking. The results of the experiment are shown in Figs. 15 and 16. It can be seen that, SVM values in all experiment cases except the case "walking", which are around 1 g from SVM of 3-d accelerometer values and 0 deg/s from SVM of 3-d gyroscope values, are as expected. In case of walking, SVM of 3-d accelerometer values is 1 g as expected while SVM of 3-d gyroscope values varies dramatically. At some instances, the SVM values are even larger than 100 deg/s. In those cases, it is obvious that if the system applies a low threshold value close to 100 deg/s, an incorrect alarm will be triggered. Defining threshold values for a fall detection system based on accelerometer and gyroscope is not an easy task. Low threshold values help to reduce missing cases when SVM of 3-d accelerometer values or 3-d gyroscope values are not large. However, they may cause incorrect alarms or notifications. In contrast, high threshold values may cause missing fall detection cases, but they help to reduce incorrect alarms of falling cases. In addition, the results in a walking case in Fig. 15 unveil that replying on merely 3-d gyroscope may lead to incorrect alarms of falling cases. Comparing between results from "walking" in Fig. 15 and "walk and fall forward" in Fig. 16, it can be seen that SVM of 3-d gyroscope values are slightly different (i.e. around 105 versus 130 deg/s) while SVM of 3-d accelerometer values are largely different. In this case, values from a 3-d accelerometer are better in terms of the fall detection accuracy. To sum up, in order to avoid an incorrect fall detection alarm, many types of sensors (i.e. 3-d accelerometer, 3-d gyroscope and 3-d magnetometer) and two-level thresholds should be considered to be applied in a sensor node.

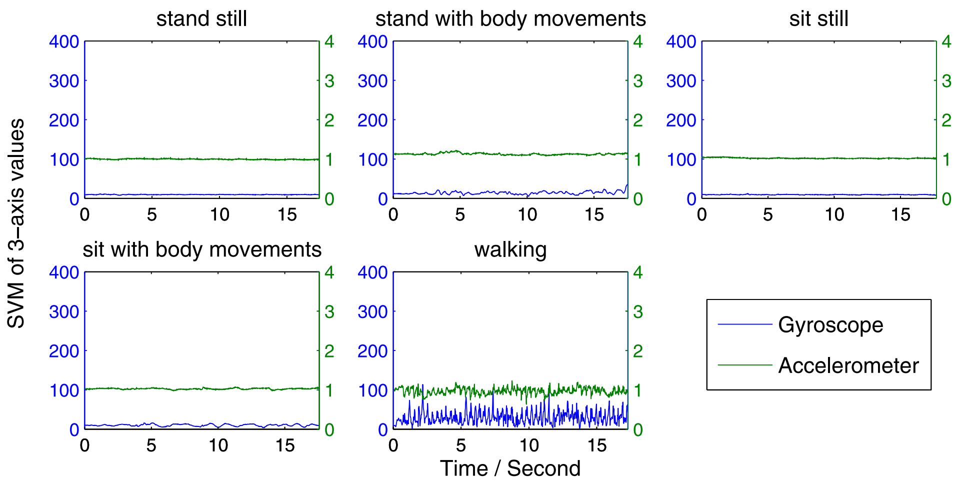


Fig. 12. Accelerometer's and Gyroscope's data at the gateway's nRF receiver during daily activities.

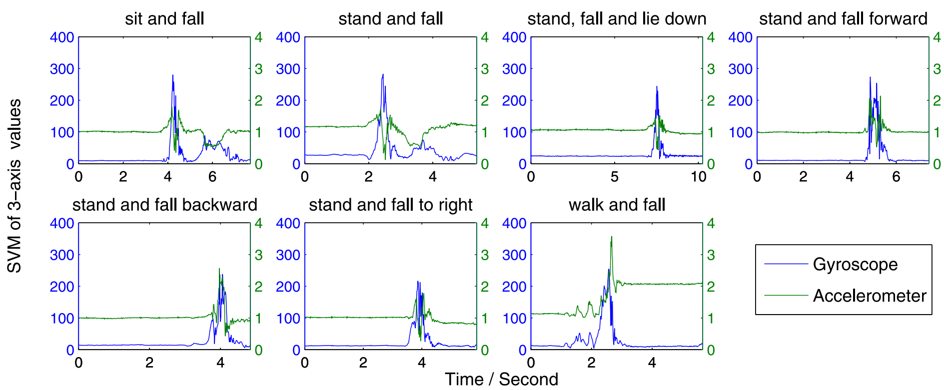


Fig. 13. Accelerometer's and Gyroscope's data at the gateway's nRF receiver during daily activities and fall.

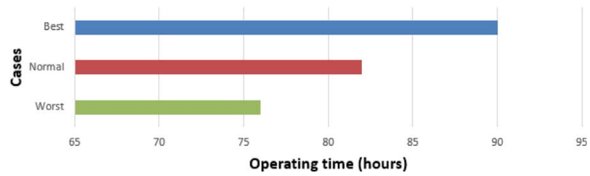


Fig. 14. Operating duration of the wearable sensor node supplied with a 1000 mAh battery when collecting data from accelerometer, gyroscope and magnetometer with a data rate of 50 samples/s via SPI and sending the data via nRF during a second under different conditions.

7. Discussions

Designing an energy efficient sensor node for fall detection and other health-care systems is not a simple task because the sensor node must fulfill strict requirements of healthcare IoT systems (e.g. latency and high quality of signal) while consuming low energy.

In order to achieve a high level of energy efficiency, wireless communication protocols are often attentively considered first. Before applying nRF for our sensor nodes, some of ESP8266 chips are integrated in sensor nodes for the experiments. Sensor nodes communicate via Wi-

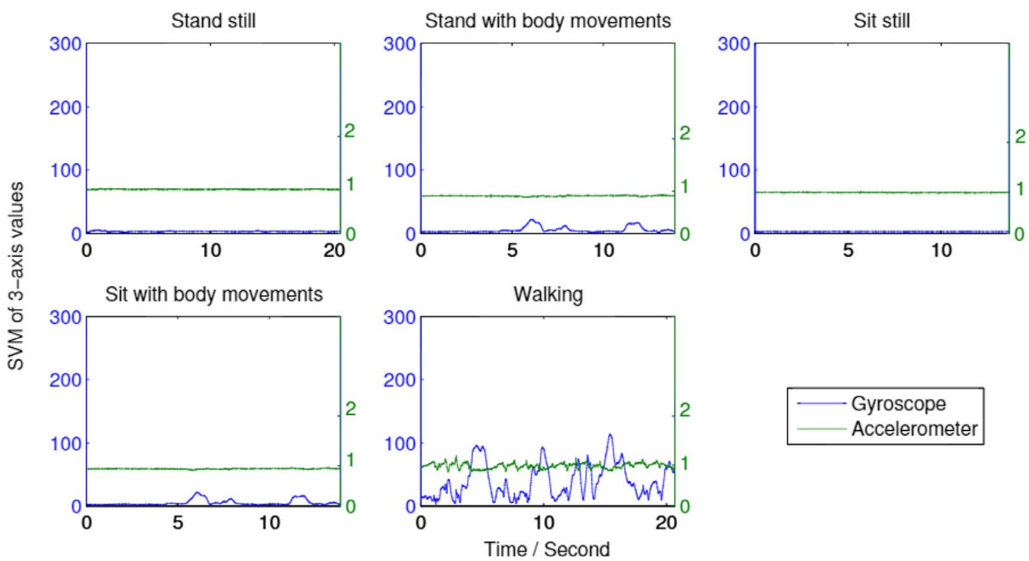


Fig. 15. Accelerometer and Gyroscope data at the gateway's nRF receiver during daily activities and fall.

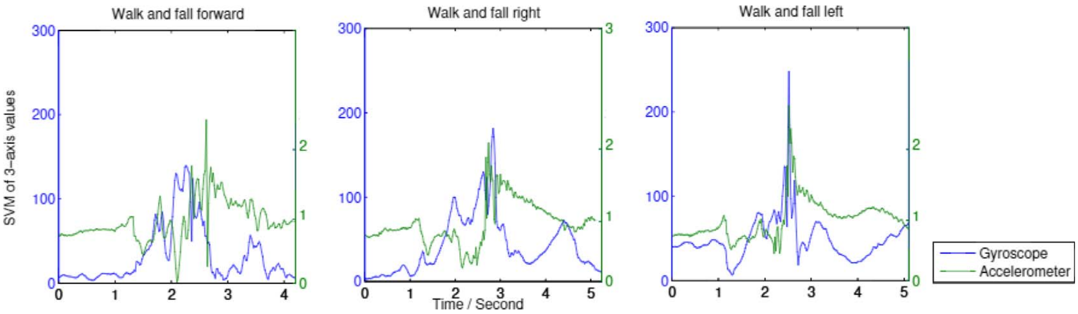


Fig. 16. Accelerometer and Gyroscope data at the gateway's nRF receiver during daily activities and fall.

Fi with a gateway fixed at the roof of the room and the transmission is line-of-sight with a distance of 7 m. When applying those ESP8266 chips, energy consumption during a second of a sensor node equipped with ESP8266 is about 270.6 mJ and the sensor node requires 460–480 mW for sending a packet. In the experiments, the sensor nodes collect data from a 3-d accelerometer, a 3-d gyroscope with a data rate of 50 samples/s. They accumulate the collected data and send one packet per second assuming that the maximum latency is 1 s. In case of other sampling rates (i.e. 50, 100 and 200 samples/s) of 3-d accelerometer and 3-d gyroscope, the maximum volume of data which the sensor node can accumulate for a single packet is 1800 Bytes, 3600 Bytes, or 7200 Bytes, respectively. The sensor node cannot accumulate more data for a single packet because the maximum latency (i.e. 1 s) requirement will be infringed. In case of the sensor node equipped with nRF, it is required 75, 150 and 300 packets to transmit the same amount of data in one second because the sensor node with nRF can send the maximum of 24 Bytes per packet. Energy consumption of the sensor node for transmitting 75, 150 and 300 packets in a second with nRF is around 31–50 mJ. Hence, in an IoT-based fall detection application, nRF is more suitable than Wi-Fi. In other applications (e.g. Electroencephalography (EEG) real-time monitoring), Wi-Fi might be more suitable due to a large amount of data (e.g. 10,000 Bytes/s per channel) is collected in a short period of time.

In terms of energy efficiency between low-power wireless communication protocols (i.e. BLE, ANT, 6LoWPAN and nRF) nRF and BLE are more energy efficient [48,49]. Precisely, nRF is more energy efficient than BLE [50,51]. For example, power consumption of a BLE chip and an nRF chip is around 46.2 mW and 37.29 mW at 0 dBm output power, respectively [50,51]. One of the main reasons of low energy in nRF is that nRF does not use any software stack. In our experiments, by replacing BLE in the sensor node presented in our previous work [11] by nRF, approximately 10% energy can be saved. In terms of connectivity, nRF is more suitable for IoT-based fall detection systems than BLE because BLE supports peer-to-peer communication while nRF supports many-to-many communication. Although nRF is energy efficient, it has several limitations (e.g. difficulty for a gateway to handle data sent simultaneously by many sensor nodes). When applying nRF, a transmission payload must be attentively considered for achieving energy efficiency and accuracy. In default, nRF uses a maximum payload of 32 Bytes as a static payload in each packet. When data is less than 32 Bytes (e.g. 4 Bytes), the nRF protocol automatically adds extra Bytes for filling up 32 Bytes payload (e.g. adding 28 Bytes). However, sending data with a size of 32 Bytes is not an optimal choice because some bytes of data may be collapsed at the receiver(s). Therefore, it is recommended to send the data with a size of 20–24 Bytes per packet.

8. Conclusions

We presented the design and implementation of an energy efficient wearable device for IoT-based fall detection systems. The device is tiny, light-weight and flexible hence suits to different IoT-based fall detection systems and can be used regularly without interfering user's daily activities. In this paper, we evaluated energy consumption of wearable sensor nodes in different configurations and scenarios to find optimal solutions for improving energy efficiency. We investigated configuration parameters (i.e. communication bus interface, and sampling rate) affecting energy consumption of the wearable device. In addition, important hardware and software factors or techniques impacting on the life-time of the sensor node are investigated. Besides, we evaluated energy consumption of the device in different transmission conditions for providing hints to system administrators for avoiding missed data while maintaining a high level of energy efficiency in the wearable device. Furthermore, we compared the wearable device with different devices proposed by others. The result shows that our wearable sensor node is the best among compared nodes. The results from conducted experiments conclude that our sensor node can operate around 76 h with a 1000 mAh battery in a tough transmission condition. Moreover, we implemented a complete IoT-based fall detection system consisting of smart gateways with Fog computing and a back-end system. When a fall occurs, the system can detect and remotely inform responsible personnel such as a doctor or caregiver in real-time. It can be concluded that the proposed wearable device is a solution to drawbacks of typical sensor nodes in IoT-based fall detection systems.

Acknowledgements

We sincerely thank MATTI program from University of Turku Graduate School, Academy of Finland, Nokia Foundation and Finnish Foundation of Technology Promotion (TES) for supporting our project.

References

[1] D.A. Sterling, J.A. O'Connor, J. Bonadies, Geriatric falls: injury severity is high and disproportionate to mechanism, J. Trauma Acute Care Surg. 50 (1) (2001) 116–119.
[2] J.A. Stevens, et al., The costs of fatal and non-fatal falls among older adults, Injury Prev. 12 (5) (2006) 290–295.
[3] M.E. Tinetti, M. Speechley, S.F. Ginter, Risk factors for falls among elderly persons living in the community, N. Engl. J. Med. 319 (26) (1988) 1701–1707.
[4] B.H. Alexander, F.P. Rivara, M.E. Wolf, The cost and frequency of hospitalization for fall-related injuries in older adults. Am. J. Public Health 82 (7) (1992) 1020–1023.
[5] Important facts about falls. Updated: Jan. 2017, Accessed Feb 2017, https://www.cdc.gov/homeandrecreationalsafety/falls/adultfalls.html.
[6] S.M. Friedman, et al., Falls and fear of falling: which comes first? A longitudinal prediction model suggests strategies for primary and secondary prevention, J. Am. Geriatr. Soc. 50 (8) (2002) 1329–1335.
[7] A. C. Scheffer, et al., Fear of falling: measurement strategy, prevalence, risk factors and consequences among older persons, Age Ageing 37 (1) (2008) 19–24.
[8] R. Igual, et al., Challenges, issues and trends in fall detection systems, Biomed. Eng. Online 12 (1) (2013) 1.
[9] M. Mubashir, et al., A survey on fall detection: principles and approaches, Neurocomputing 100 (2013) 144–152.
[10] J. T. Perry, et al., Survey and evaluation of real-time fall detection approaches, High-Capacity Optical Networks and Enabling Technologies (HONET), 2009 6th International Symposium on, IEEE, 2009, pp. 158–164.
[11] T. N. Gia, et al., Iot-based fall detection system with energy efficient sensor nodes, Nordic Circuits and Systems Conference (NORCAS), 2016 IEEE, IEEE, 2016, pp. 1–6.
[12] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ACM, 2012, pp. 13–16.

[13] A. Rahmani, et al., Fog Computing in the Internet of Things, Springer International Publishing, 2018.
[14] E. Casilari, M.A. Oviedo-Jiménez, Automatic fall detection system based on the combined use of a smartphone and a smartwatch, PLoS ONE 10 (11) (2015) e0140929.
[15] M. Kepski, B. Kwolek, Embedded system for fall detection using body-worn accelerometer and depth sensor, (IDAACS), 2015 IEEE 8th International Conference on, 2 IEEE, 2015, pp. 755–759.
[16] P. Pivato, et al., A wearable wireless sensor node for body fall detection, Measurements and Networking Proceedings (M&N), 2011 IEEE International Workshop on, IEEE, 2011, pp. 116–121.
[17] D. Chen, W. Feng, Y. Zhang, X. Li, T. Wang, A wearable wireless fall detection system with accelerators, Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on, IEEE, 2011, pp. 2259–2263.
[18] O. Biros, et al., Implementation of wearable sensors for fall detection into smart household, Applied Machine Intelligence and Informatics (SAMI), 2014 IEEE 12th International Symposium on, IEEE, 2014, pp. 19–22.
[19] S.Z. Erdogan, T.T. Bilgin, A data mining approach for fall detection by using k-nearest neighbour algorithm on wireless sensor network data, IET Commun. 6 (18) (2012) 3281–3287.
[20] F. Wu, H. Zhao, Y. Zhao, H. Zhong, Development of a wearable-sensor-based fall detection system, Int. J. Telemed. Appl. 2015 (2015) 2.
[21] Y. Li, et al., Accelerometer-based fall detection sensor system for the elderly, 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, 3 IEEE, 2012, pp. 1216–1220.
[22] C. Shuo, Fall detection system using Arduino Fio, Proceedings of the IRC Conference on Science, Engineering and Technology, Singapore, 13 (2015).
[23] Y. Cheng, C. Jiang, J. Shi, A fall detection system based on sensortag and Windows 10 iot core (2015).
[24] R. Freitas, M. Terroso, M. Marques, J. Gabriel, A.T. Marques, R. Simoes, Wearable sensor networks supported by mobile devices for fall detection, SENSORS, 2014 IEEE, IEEE, 2014, pp. 2246–2249.
[25] I. Tcarenko, et al., Energy-efficient iot-enabled fall detection system with messenger-based notification, International Conference on Wireless Mobile Communication and Healthcare, Springer, 2016, pp. 19–26.
[26] T. N. Gia, et al., Fog computing in healthcare internet of things: a case study on ecg feature extraction, (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on, IEEE, 2015, pp. 356–363.
[27] A. M. Rahmani, et al., Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach, Future Gener. Comput. Syst. (2017).
[28] B. Negash, et al., Leveraging fog computing for healthcare iot, Fog Computing in the Internet of Things, Springer, 2018, pp. 145–169.
[29] I. Fredriksen, P. Kastnes, Choosing a mcu for your next design; 8 bit or 32 bit?, 2014, Accessed Jul 2017, http://www.atmel.com/images/45107a-choosing-a-mcu-fredriksen_article_103114.pdf.
[30] Y.S. Delahoz, M.A. Labrador, Survey on fall detection and fall prevention using wearable and external sensors, Sensors 14 (10) (2014) 19806–19842.
[31] F. Touati, R. Tabish, U-healthcare system: state-of-the-art review and challenges, J. Med. Syst. 37 (3) (2013) 9949.
[32] RaspberryOrg, Raspberry-pi-3-model-b. Updated: Jan. 2016, Accessed Jul 2016, https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.
[33] A. Z. Rakhman, et al., Fall detection system using accelerometer and gyroscope based on smartphone, Information Technology, Computer and Electrical Engineering (ICITACEE), 2014 1st International Conference on, IEEE, 2014, pp. 99–104.
[34] P-K. Chao, et al., A comparison of automatic fall detection by the cross-product and magnitude of tri-axial acceleration, Physiol. Meas. 30 (10) (2009) 1027.
[35] A. Bourke, et al., Evaluation of waist-mounted tri-axial accelerometer based fall-detection algorithms during scripted and continuous unscripted activities, J. Biomech. 43 (15) (2010) 3051–3057.
[36] Q. T. Huynh, et al., Optimization of an accelerometer and gyroscope-based fall detection algorithm, J. Sens. 2015 (2015).
[37] L. Palmerini, et al., A wavelet-based approach to fall detection, Sensors 15 (5) (2015) 11575–11586.
[38] G. Rescio, et al., Support vector machine for tri-axial accelerometer-based fall detector, Advances in Sensors and Interfaces (IWASI), 2013 5th IEEE International Workshop on, IEEE, 2013, pp. 25–30.
[39] J. He, C. Hu, X. Wang, A smart device enabled system for autonomous fall detection and alert, Int. J. Distrib. Sens. Netw. 12 (2) (2016) 2308183.
[40] R. M. Gibson, et al., Multiple comparator classifier framework for accelerometer-based fall detection and diagnostic, Appl. Soft Comput. 39 (2016) 94–103.
[41] T. N. Gia, et al., Low-cost fog-assisted health-care iot system with energy-efficient sensor nodes, The 13th International Wireless Communications and Mobile Computing Conference (IWCMC 2017), IEEE, 2017.
[42] S. R. Moosavi, et al., End-to-end security scheme for mobility enabled healthcare internet of things, Future Gener. Comput. Syst. 64 (2016) 108–124.
[43] A. M. Rahmani, et al., Smart e-health gateway: bringing intelligence to internet-of-things based ubiquitous healthcare systems, (CCNC), 2015 12th Annual IEEE, IEEE, 2015, pp. 826–834.
[44] T. N. Gia, et al., Iot-based continuous glucose monitoring system: a feasibility study, Procedia Comput. Sci. 109 (2017) 327–334.
[45] T. N. Gia, et al., Fog computing in body sensor networks: an energy efficient approach, IEEE International Body Sensor Networks Conference (BSN15), (2015).
[46] M. Solutions, Power monitor. Updated: Jan. 2016, Accessed Jul 2016, https://www.msoon.com/LabEquipment/PowerMonitor/.
[47] Mpu-9250. Updated: Jan. 2017, Accessed Feb 2017, https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/.
[48] T. N. Gia, et al., Customizing 6LoWPAN networks towards Internet-of-Things based ubiquitous healthcare systems, NORCHIP, 2014, IEEE, 2014, pp. 1–6.
[49] A. Dementyev, et al., Power consumption analysis of bluetooth low energy, zigbee and ant sensor nodes in a cyclic sleep scenario, Wireless Symposium (IWS), 2013 IEEE International, IEEE, 2013, pp. 1–4.
[50] S. Kamath, J. Lindh, Measuring bluetooth low energy power consumption, Texas instruments application note AN092, Dallas(2010).
[51] nrf24l01 +. Updated: Jan. 2017, Accessed Feb 2017, https://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01P.

Tuan Nguyen Gia received his Master's degree (Tech) from the University of Turku (Finland) in 2014. He received a 4-year funded position in University of Turku, a grant scholarship from Nokia Foundation for an efficient and excellent researcher in 2015, and a grant scholarship from Finnish Foundation of Technology Promotion in 2017. He joined Internet-of-Things for Healthcare (IoT4Health) research group, University of Turku in 2014. Currently, he works as a researcher and a doctoral candidate in the same group.

Victor Kathan Sarker received his Bachelor's degree on Electronic and Telecommunication Engineering from North South University, Dhaka, Bangladesh in 2012. Currently he is pursuing Master's degree on Embedded Computing in University of Turku. Besides, he works in the Internet-of-Things for Healthcare (IoT4Health) research group in Department of Future Technologies, University of Turku.

Igor Tcarenko received his Master (Tech) from the University of Turku (Turku, Finland) in 2014. Currently, he works at the Internet-of-Things for Healthcare (IoT4Health) research group in IT department, University of Turku.

Amir M. Rahmani received his Master's degree from Department of Electrical and Computer Engineering, University of Tehran, Iran, in 2009 and Ph.D. degree from Department of Information Technology, University of Turku, Finland, in 2012. He also received his MBA jointly from Turku School of Economics and European Institute of Innovation & Technology (EIT) ICT Labs, in 2014. He is currently Marie Curie Global Fellow at University of California Irvine (USA) and TU Wien (Austria). He is also an adjunct professor (Docent) in embedded parallel and distributed computing at the University of Turku, Finland. He is the author of more than 140 peer-reviewed publications. He is a senior member of the IEEE.

Tomi Westerlund received his Ph.D. (Tech) 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 acted yearly as a visiting scholar at Fudan University, Shanghai, China. His current research interest is Internet of Things (IoT); how we can utilise 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 is a Senior University Lecturer in Embedded Electronics Laboratory and an adjunct professor in embedded computing architectures at the University of Turku, Embedded Computer Systems laboratory. During the period 2007–2009, he held an Academy of Finland researcher position. He is the author of more than 200 peer-reviewed publications, has supervised nine Ph.D. theses. Liljeberg is the leader of the Internet-of-Things for Healthcare (IoT4Health) research group.

Hannu Tenhunen received the diplomas from the Helsinki University of Technology, Finland, 1982, and the Ph.D. degree from Cornell University, Ithaca, NY, 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 department director. Since 1992, ha has been a professor at the Royal Institute of Technology (KTH), Sweden, where he also served as a dean. He has more than 600 reviewed publications and 16 patents internationality. He is a member of the IEEE.

Paper VII

Energy Efficient Fog-assisted IoT System for Monitoring Diabetic Patients with Cardiovascular Disease

Tuan Nguyen Gia, Imed Ben Dhaou, Mai Ali, Amir M Rahmani, Tomi Westerlund, Pasi Liljeberg, Hannu Tenhunen

Original published in Future Generation Computer Systems, The International Journal of eScience, Elsevier, 93, Pages 198-211

Future Generation Computer Systems 93 (2019) 198–211

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Energy efficient fog-assisted IoT system for monitoring diabetic patients with cardiovascular disease

Tuan Nguyen Gia [a,*], Imed Ben Dhaou [b], Mai Ali [c], Amir M. Rahmani [d,e], Tomi Westerlund [a], Pasi Liljeberg [a], Hannu Tenhunen [a]

[a] Department of Information Technology, University of Turku, Turku, Finland
[b] Department of Electrical Engineering, Qassim University, Saudi Arabia
[c] Department of Electrical Engineering, Alfaisal University, Saudi Arabia
[d] Department of Computer Science, University of California Irvine, USA
[e] Institute of Computer Technology, TU Wien, Vienna, Austria

ARTICLE INFO

Article history:
Received 14 June 2018
Received in revised form 21 August 2018
Accepted 18 October 2018
Available online 2 November 2018

Keywords:
Glucose
Diabetes
ECG
Hypoglycemia
Fall detection
Health monitoring
IoT
Fog/edge computing
Energy efficiency

ABSTRACT

Blood glucose plays an important role in maintaining body's activities. For example, brain only uses glucose as its energy source. However, when blood glucose level is abnormal, it causes some serious consequences. For instance, low-blood glucose phenomenon referred to as hypoglycemia can cause heart repolarization and induce cardiac arrhythmia causing sudden cardiac deaths. Diabetes, which can be viewed as a high-blood glucose level for a long period of time, is a dangerous disease as it can directly or indirectly cause heart attack, stroke, heart failure, and other vicious diseases. A solution for reducing the serious consequences caused by diabetes and hypoglycemia is to continuously monitor blood glucose level for real-time responses such as adjusting insulin levels from the insulin pump. Nonetheless, it is a misstep when merely monitoring blood glucose without considering other signals or data such as Electrocardiography (ECG) and activity status since they have close relationships. When hypoglycemia occurs, a fall can easily occur especially in case of people over 65 years old. Fall's consequences are more hazardous when a fall is not detected. Therefore, we present a Fog-based system for remote health monitoring and fall detection. Through the system, both e-health signals such as glucose, ECG, body temperature and contextual data such as room temperature, humidity, and air quality can be monitored remotely in real-time. By leveraging Fog computing at the edge of the network, the system offers many advanced services such as ECG feature extraction, security, and local distributed storage. Results show that the system works accurately and the wearable sensor node is energy efficient. Even though the node is equipped with many types of sensors, it can operate in a secure way for up to 157 h per a single charge when applying a 1000 mAh Lithium battery.

1. Introduction

Hypoglycemia describes an abnormal phenomenon when the blood glucose level goes below 60 mg/dl. Hypoglycemia causes heart repolarization and may induce cardiac arrhythmia which is one of the primary causes of sudden cardiac deaths [1,2]. According to Centers for Disease Control and Prevention [3], millions of people are affected by arrhythmia. Especially, people with the age of 60 or more are at high risk of arrhythmia or atrial fibrillation (AF). QT-interval lengthening of ECG is a sign of arrhythmia. Recently, research works have proposed the prediction of hypoglycemia by analyzing QT-period and T-Wave [1,4]. Hyperglycemia describes an abnormal high-blood glucose level (e.g. 100–125 mg/dl as pre-diabetes and higher than 126 mg/dl as diabetes). Heart repolarization time is altered by hyperglycemia [5,6]. Similarly, hyperglycemia can be predicted by measuring QT-interval length, QT-interval variability (QTV) and corrected QT interval variability (QTcV). According to WHO, this number increased dramatically from 108 to 422 million during 1990–2014 [7] and is projected to raise up to at two or three times by 2030 [8]. People from any gender and any age can have diabetes. For example, approximately 200,000 people who live in the USA and are under 20 years old have diabetes [9]. Diabetes not only occurs in developed countries but also in developing countries. According to National Vital Statistics Reports (NVSR), diabetes has a rank of 7 among the 15 leading causes of deaths in 2014 [7]. The number of deaths directly caused by diabetes is approximately 1.5 million in 2014 [8]. In addition, diabetes can directly or indirectly cause heart attack, stroke, heart

* Corresponding author.
E-mail address: tunggi@utu.fi (T. Nguyen Gia).

https://doi.org/10.1016/j.future.2018.10.029
0167-739X/

failure, kidney failure, blindness and other vicious diseases which are primary causes of deaths [10]. Unfortunately, diabetes cannot be cured with the existing knowledge. One of the methods for reducing the serious consequences caused by diabetes is to continuously monitor the blood glucose level and adjusting the insulin level in real-time.

Fall and its consequences cannot be neglected or underestimated because they might be the cause of serious injuries and traumas. For instance, bone fracture, broken knee, neck fracture, head bruises and head traumas can be caused by falls [11,12]. The injuries require a long period of time to be healed and fully recovered. Correspondingly, they cause significant costs and reduce the quality of life [13]. However, merely 50% of the falling cases are reported and in-time aided. Unreported cases might cause difficulty and complication in treatments later.

Diabetes, cardiovascular diseases, fall and old people often have some relationships. Diabetes has been identified as a risk factor for falls [14,15]. For example, people who are over 65 years old are like to have diabetes, cardiovascular diseases and fall more often. According to statistics, more than 25% of people who are over 65 years old have diabetes and more than 30% of these people fall every year with hazardous consequences [11,16]. In addition, more than 68% of these people having diabetes die from cardiovascular diseases [15]. Therefore, it is required to have a system which can both monitor diabetes, ECG and inform abnormalities (e.g., a fall, very low or high glucose level, and abnormal heart rate) in real-time without interfering the patient's daily activities.

Internet of Things (IoT) can be considered as one the most suitable candidates for addressing the target. IoT can be expressed as a platform where physical and virtual objects are interconnected and communicate together. IoT consisting of many advanced technologies such as sensing, sensor network, Internet and Cloud computing is capable of providing remote health monitoring in real-time while the quality of life can be maintained. Via IoT systems, collected data is stored in Cloud servers. Therefore, real-time data and the historical data can be accessed remotely in anytime. In addition, IoT systems are able to perform real-time responses or actions. For example, an insulin pump of IoT systems can automatically or be remotely controlled for injecting insulin into the patient's blood when the blood glucose is high.

Although glucose monitoring IoT systems have advantages such as remote real-time monitoring and global data storage, they have limitations. For example, most of them are not secured because the data transmitted over the network is not protected and encrypted. The transmitted data can be listened and altered by unauthorized parties [17]. Furthermore, many health monitoring IoT systems do not provide advanced services such as distributed local storage, push notification, and data analysis. Correspondingly, incorrect diagnosis and treatments of diseases may occur when both contextual data and patient's activity status are not analyzed altogether with e-health data. As a result, caregivers such as doctors might not be able to save a patient life. Most of the existing IoT health monitoring systems do not consider the close relationship of diabetes, cardiovascular disease, and fall cases.

A proper approach to solve the challenges in IoT systems is to enhance sensor nodes and apply an extra layer named as Fog between gateways and Cloud servers. Fog layer is run on the top of smart gateways to provide advanced services for enhancing the quality of services. For example, Fog helps to save network bandwidth between gateways and Cloud servers by processing and compressing data [18,19]. Furthermore, Fog helps to reduce the burdens of Cloud servers by pre-processing data at smart gateways. Fog provides distributed local storage for temporarily storing data. In addition to mentioned services, Fog helps to facilitate many other advanced services such as system fault detection, database synchronization, interoperability, mobility-awareness [20]. Moreover, Fog creates a convergent network of interconnected and intercommunicated gateways, that helps to overcome service interruption. For instance, when a connection between a smart gateway and Cloud servers is interrupted, real-time data streaming is still maintained. In this case, data is sent to Cloud servers via adjacent smart gateways which are directly connected and geographically close to the "interrupted" gateway. With the benefits of advanced services, Fog not only solves many challenges of IoT systems but also enhances the quality of services dramatically.

In this paper, a smart IoT system based on Fog for remote healthcare monitoring is introduced. For improving the accuracy of diseases analysis and diagnosis, the system monitors not only e-health data such as blood glucose, ECG, patient's movement and body temperature but also contextual data such as room temperature, humidity, and air quality. The system is secured with cryptography algorithms for protecting the collected data. Particularly, data is encrypted at sensor nodes before being transmitted and decrypted at smart gateways. The system with the Fog layer offers advanced services such as interoperability, distributed local storage, data processing (i.e., QT intervals extracted from an ECG waveform, activity status categorization, and fall detection via lightweight algorithms). Last but not least, the enhanced energy-efficient sensor node for monitoring vital signals is presented. The main contributions of this work are summarized as follows:

- A complete implementation of a Fog-assisted IoT system for monitoring diabetes patients with cardiovascular diseases
- Light-weight algorithm at Fog-assisted gateways for ECG feature extraction
- Light-weight algorithm at Fog-assisted gateways for activity status categorization and fall detection
- Designing and implementing energy efficient wearable sensor nodes for collecting ECG, glucose, body temperature and motion-related data
- Analysis of e-health data in different scenarios and patient's activities

The rest of the paper is organized as follows: Section 2 includes related work and motivation for this work. Section 3 provides an overview architecture of the health monitoring IoT system having Fog assistance. Section 4 emphasizes on Fog services such as the algorithm for extracting QT intervals from an ECG waveform, fall detection, and activity status categorization. Section 5 presents test-bed and the system implementation. Section 6 provides insights about experimental results. Finally, Section 7 concludes the work.

2. Related work and motivations

Many efforts have been made for proposing real-time and remote health monitoring IoT-based systems. In [21,22], an ECG monitoring IoT-based system using 6LoWPAN is proposed. The system consists of smart gateways and energy efficiency sensor nodes. In [23,24], authors present IoT systems for fall detection. The systems use wearable sensor nodes for collecting 3-dimensional (3-D) acceleration and 3-D angular velocity. The systems with smart gateways offer push notification services for informing a fall to caregivers. In [25], authors present an IoT system with a smart gateway for e-health monitoring. The gateway supports interoperability with Bluetooth Low Energy (BLE), Wi-Fi, and IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN). In addition, the gateway provides many advanced services such as data compression, data storage, and security. In [26], authors present a glucose monitoring IoT-based system which shows some levels of energy efficiency by applying 6LoWPAN and RFID. The system can distinguish non-fasting and fasting cases for an accurate diagnosis. In [27], authors propose an IoT system for non-invasive glucose level sensing. The system uses a laptop as a gateway for receiving data from 6LoWPAN nodes and sending the data to Cloud servers. In most of the discussed work, security is not attentively considered. Especially, the connection between sensor nodes and the smart gateway is not fully protected. Some of the works provide smart gateways. However, services provided at the smart gateways are limited. Some of the works do not support interoperability which limits the flexibility and ubiquity of the health monitoring systems.

Other works [28,29] consider high levels of security for health monitoring IoT systems. The connection between sensor nodes and smart gateways are secured by light-weight cryptography algorithms. However, the energy efficiency of sensor nodes is not attentively considered.

Recently, researchers have proposed health monitoring IoT systems with Fog computing. The Fog-based systems have advantages such as bandwidth saving, energy efficiency, and a high level of security. In [30], authors apply a smart gateway and Fog computing into an ECG monitoring IoT system. The system provides many services such as categorization, push notification and distributed local storage. In [31], authors propose an IoT system with Fog computing for continuous glucose monitoring system. The system uses a mobile-based gateway for processing and analyzing data. When the smart gateway detects abnormalities such as too low or too high blood glucose level, it sends push messages for informing medical doctors in real-time. In [18,19], authors propose Fog approaches for ECG monitoring systems. The systems can extract ECG features at Fog and achieve some levels of energy efficiency at sensor nodes. In [32,33], health monitoring IoT systems with Fog computing are proposed. These systems provide many advanced Fog services such as data analysis, data fusion, distributed local data storage, and data compression. By using a web-browser, real-time ECG data can be remotely and real-time monitored. In [34], authors propose a Fog approach for enhancing telehealth big data. The system analyzes ECG data for finding similar patterns. In [35], authors present a Fog approach for a fall detection IoT system. Fog computing and Cloud servers run the U-fall algorithms for detecting fall automatically in real-time. In [36], a Fog approach for a medical warning system is proposed. ECG is analyzed at Fog for early-detecting patient deterioration. In [37] Fog computing for reliable e-health applications (i.e., human fall detection) is applied. The system has both e-health and contextual sensor nodes which are built from general purpose devices (e.g., Arduino Uno and Lilypad). These sensor nodes transmit the collected data to a computer for processing. In [38], authors present a Fog-based fall detection system. In the system, a set of fall detection algorithms is developed and proposed. In the system, tasks are split between edge devices and Cloud servers to achieve real-time analysis. In [39], authors present a Fog-based system for monitoring mild dementia and chronic obstructive pulmonary disease (COPD) patients. The system consists of e-health, contextual nodes and Fog nodes. An environment node of the system based on Arduino Uno collects temperature, humidity, gas, CO2, and oxygen information while e-health node acquires 3-D acceleration. The data is sent via Zigbee. The Fog node is responsible for real-time processing and notification. In [40], authors claimed that cloud-based healthcare services with intermediate Fog nodes can help to improve quality of healthcare. For instance, health insights can be acquired accurately while information privacy is protected. The Fog node running a privacy middle-ware helps to reduce the burden of IoT sensor nodes. Accurate results provided by the system can benefit both caregivers and end-users. In [41], authors propose a system for monitoring and adjusting the oxygen level in real-time for obstructive pulmonary disease patients. The system is able to acquire different types of data including both contextual and e-health data. In addition, the system is equipped with a Fog-to-Cloud (F2C) service to process the data and adjust an oxygen level in real-time.

Although the mentioned Fog-based approaches provide advanced services for enhancing health monitoring systems, none of them consider all aspects of sensor node's energy efficiency, security, and the relationship of diabetes, cardiovascular disease and patient fall together. Energy efficiency is a vital characteristic of healthcare IoT systems. When a sensor node is not energy efficient, it can cause service interruption which is one of the reasons for reducing the accuracy of disease analysis. When the system is not secured, patient's information can be stolen or the system can be instructed for doing unacceptable actions. Disease analysis and diagnosis might be inaccurate when standalone e-health data is used without considering contextual data or activity status. For instance, heart rate during sitting and training is different. In this paper, we propose an IoT system with Fog computing for health monitoring. The system not only monitors e-health data (i.e., blood glucose, ECG, and body temperature), daily activity, and contextual data (i.e., room temperature, humidity, air quality) but also provides advanced services for improving the accuracy of disease analysis and informing abnormalities (i.e., hypoglycemia, hyperglycemia, and cardiac disease). In addition, many techniques and Fog approaches are applied to achieve a high level of energy efficiency while the connection between sensor nodes and smart gateways is protected by an AES algorithm.

3. Architecture

The architecture of the proposed system shown in Fig. 1 has 3 layers including sensor node layer, Fog computing layer consisting of Fog-assisted smart gateways, and Cloud servers with end-user terminals. Each layer is discussed as follows:

3.1. Sensor layer

Sensor layer includes different types of sensor nodes such as contextual sensor nodes, e-health sensor nodes, and actuator nodes. Contextual sensor nodes can be fixed at a single room for gathering contextual data from surrounding environments such as a room temperature, humidity, time, location and air quality. The contextual data plays an important role in achieving accurate analysis. For example, human pulse rate is likely to increase when the temperature rises. In [42], from the experiments with more than 30 thousand children, authors showed pulse rate increases more than 20 pulses when the temperature rises from 35 to 40 Celsius degrees.

E-health sensor nodes can be categorized into different types depending on the given health monitoring application. In this paper, three types of e-health sensor nodes such as low data rate sensor nodes, high data rate sensor nodes and hybrid sensor nodes equipped with both low and high data rate sensors. Low data rate sensor nodes can be used for acquiring blood glucose, body temperature, and humidity. High data rate sensor nodes can be used for collecting ECG and body motion (e.g., acceleration and angular velocity). Hybrid sensor nodes equipped with both low and high data rate sensors can be used for collecting all types of data such as ECG, body temperature, glucose, humidity, and body motion.

Actuator nodes are used for controlling actions related to health or the surrounding environment. Actuator nodes often receive instructions from a smart gateway. For example, an atmosphere controlling actuator can adjust a room's temperature and humidity.

The collected data from sensor nodes is sent to smart gateways via one of several wireless protocols. A choice of a specific

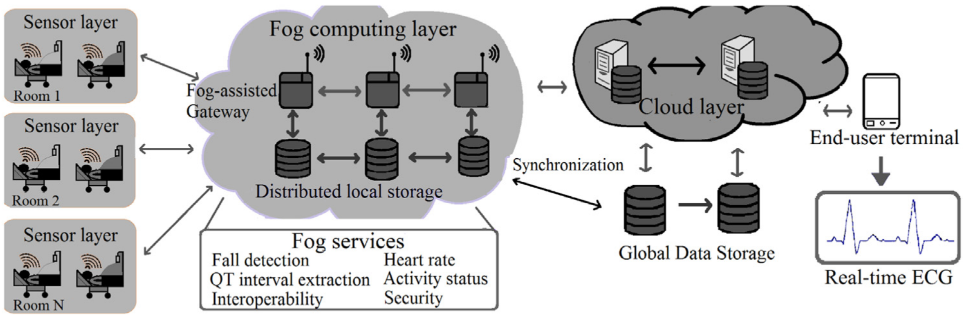


Fig. 1. Architecture of Fog-assisted IoT system for monitoring diabetes patients with cardiovascular disease.

wireless protocol depends on the application's requirements. For example, Wi-Fi is used for high data rate monitoring applications (e.g., 8-channel EMG monitoring in which each channel collects 1000 24-bit samples/s [43,44]). Bluetooth Low Energy (BLE) or IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) is used for low data rate health monitoring applications (1-channel ECG monitoring in which each channel collects 125 16-bit samples/s [21,22]). In the proposed system, an nRF protocol, which is ultra low power 2.4 GHz ISM band wireless protocol, is utilized because of its flexibility of data rate support and energy efficiency [45]. The nRF protocol supports data rates of 250 kbps, 1 Mbps, and 2 Mbps. The collected data can be kept intact or pre-processed before being transmitted. Particularly, some of the data (i.e., body temperature, and blood glucose) can be filtered with light-weight filters based on threshold. For instance, a human body temperature cannot be higher 47 Celsius degrees. This value can be used as a threshold value. When a measured sample value is higher than this threshold and the sample value is different than the historical values which were measured in a minute before. The sample is considered as dummy data and it is eliminated. When the sample value is similar to the historical values. The push notification message, which inform the abnormal case such as extremely high body temperature or malfunctioned body temperature sensors, will be sent to administrators via gateways and Cloud. Similarly, 400 mg/dl is used as a threshold value for blood glucose. In other cases, data (i.e., ECG, 3-D acceleration, and 3-D angular velocity) is kept in tact and sent to Fog-assisted smart gateways which perform heavy computational tasks such as wavelet transform and ECG feature extraction [18,19]. For providing some levels of security, data can be encrypted at sensor nodes before being sent to Fog-assisted smart gateways which decrypt the data and perform heavy processing.

3.2. Fog computing layer consisting of fog-assisted gateways

Smart gateways can be fixed or movable depending on the given application. In many health monitoring applications in hospitals, fixed gateways are more preferred because they can use power and Ethernet from wall sockets. Accordingly, fixed gateways can serve many sensor nodes simultaneously and offer advanced services running heavy computational algorithms whilst movable gateways are not capable due to limited battery capacity. In addition, fixed gateways are more secured because they are kept intact and can perform advanced security algorithms.

In addition to conventional tasks of receiving and transmitting data, Fog-assisted smart gateways proffer many advanced services for enhancing the quality of healthcare services. Some of the Fog services are distributed local storage, data compression, localhost with user interface, categorization service, and push notification.

Distributed local storage often consists of synchronized and intact databases. The synchronized database stores real-time contextual and e-health data while the intact database stores data used for algorithms, services and the system's configurations such as username and password and algorithm's parameters. The synchronized database is synchronized with Cloud servers' database while data in the intact database is only updated by system administrators. Due to the limited storage of the synchronized database, the oldest data is purged for storing incoming data.

Data compression helps to save network bandwidth. Although compressing and decompressing cost some resources and latency, they do not affect the performance of other services and only increase the total latency slightly [25].

Categorization service is used to categorize local users and external users. Particularly, the categorization service scans Wi-Fi devices around smart gateways. As a result, local devices are stored in smart gateways' database. When a user tries to connect to smart gateways, the system checks the database. If the user is a local user, the smart gateways send real-time data directly to the user's terminal without going through Cloud servers.

Push notification is one of the most important services in Fog. Push notification is used for informing abnormalities in real-time to responsible people such as system administrators or caregivers.

These services are explained in detail in our previous works [32,30,33]. Although these services are not the main focus of this paper, they are still implemented in the system.

In addition to mentioned services, there are vital services such as data analysis, data processing, interoperability, and security. Detailed information of these services is discussed in Section 4.

3.3. Cloud layer with end-user terminal

Cloud layer provides many benefits such as centralized global storage, scalability, data security and data processing. Heavy computational tasks, which cannot be run at Fog, can be processed smoothly in Cloud servers which are the core of the Cloud layer. For example, Cloud servers support modern machine learning services with pre-trained models. In addition, Cloud servers support powerful search, discovery and image analysis. E-health and other related data (e.g., records of patient's health status) can be stored at Cloud servers. Different technologies (e.g., WebSockets and JSON) can be installed at Cloud for hosting a comprehensive website showing real-time data in both textual and graphical interfaces [46,47]. Furthermore, Cloud servers support push notification sending the instant messages to an end-user in real-time. In the proposed system, push notification is used for informing abnormalities (i.e., related to patient health and the system's technical problems) to responsible people such as caregivers or system administrators. End-users can use terminals such as smart phone's app or web browsers to access both real-time data and the historical data. In addition, end-users such as caregivers can provide instructions or advice via the terminals.

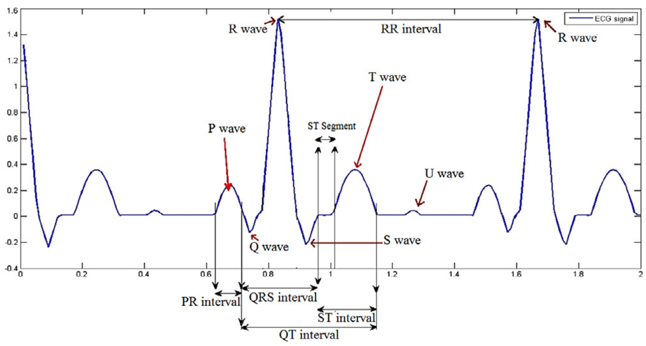


Fig. 2. ECG signal generated using Matlab (Ideal).

Algorithm 1 Algorithm to compute local maximum of a function $f(x)$

procedure MAX-ALGORITHM(x_i, t_i)

if ($x_{i-1} = 0$ and $x_i > 0$) **then**
$I_1 \leftarrow t_i$
$M \leftarrow x_i$

else if ($I_1 \neq 0$ and $x_i > 0$) **then**
$M = Max(x_i, M)$

else if ($I_1 \neq 0$ and $x_i = 0$) **then**
$I_2 \leftarrow t_i$
break

return M, $[I_1, I_2]$
where x: value of ECG
t: specific time(s)

4. Fog services

As mentioned, smart gateways in Fog computing can offer many advanced services (e.g., localhost, categorization, and push notification) and thereby potentially enhance the quality of healthcare services. In this paper, interoperability, security, data processing are investigated and explained as follows:

4.1. Data processing

Data processing and data analysis in Fog-assisted smart gateways play important roles in health monitoring systems. They not only help to reduce the burden of Cloud servers but also help to extract important information which can be used for real-time critical decision making and push notification. In this paper, heart rate, QT intervals, corrected QT intervals are extracted from an ECG waveform. The extracted information combined with other e-health data such as blood glucose level, body temperature, and body motion is used for detecting hypoglycemia and hyperglycemia in real-time.

4.1.1. Heart rate and the QT interval extraction algorithm

ECG can be defined as a periodic signal in which each normal ECG waveform represents the electrical events in one cardiac cycle. A normal ECG waveform, shown in Fig. 2, often consists of several waves named as P, Q, R, S, T, and U. If the baseline of the ECG is zero, the three waves P, R, and T often have positive peaks whereas the waves Q and S often have negative peaks.

From an ECG signal, heart rate can be calculated based on the formula: *Heart rate = 60/RR interval* [48]. *RR interval* can be easily calculated since R peaks have the highest amplitude among all waves.

Peak detection can be computed using a linear time algorithm that seeks the determination of local extreme. The algorithm to determine the QT-period starts by first locating the lowest interval I_P in which the P-wave reaches its maximum. Then, the same procedure is applied to compute I_P, I_R, I_S and I_T, where the subscript in I designed the type of the wave. The QT-length is then computed using $I_{QT} = I_R + I_Q + I_S + I_T$. The pseudo-algorithm to compute the lowest interval in which a function $f(x)$ reaches its local maximum is shown in Algorithm 1. The algorithm takes two inputs: $x_i = f(t_i)$, where t_i is the instant of time t_i. The algorithm is a stream type, consequently, does not consume memory which makes it suitable for tiny devices.

When QT interval is extracted, corrected QT (QTc) interval can be easily calculated by one of the formulas shown in Table 1. Currently, Bazett's QTc (QTcB) is used as a clinical standard but authors [54] showed that Fridericia's QTc (QTcFri) might become the next clinical standard replacing for QTcB. In this paper, Bazett's QTcB is applied for the experiments.

4.1.2. Activity status categorization and fall detection algorithm

It is inaccurate to analyze only ECG without considering an activity status because ECG signals change dramatically based on the current activity status. For example, ECG of a person during resting and running is different. Therefore, ECG and activity status must be monitored and analyzed simultaneously.

Activity status representing daily physical activities of a person can consist of three primary groups such as non-moving/resting, walking, and training exercises. Each group can have many activities (e.g., sleeping, lying, standing and sitting belong to a non-moving/resting group or running, push up, weight lifting and other heavy activities belong to a training group). Activities belong to the same group have similar effects to the ECG waveform. Miao et al. [55] show that ECG of a person during standing and sitting are almost similar. Therefore, the paper only focuses on standing, walking and running where each activity represents for each group of three mentioned groups, respectively.

A person's activity status can be detected by using camera or wearable motion sensors. In this paper, wearable motion sensors are used because it does not limit a person's activities. In addition, wearable motion sensors can be used to detect a human fall [56,23]. A person's activity status and a human fall are detected by the algorithm shown in Fig. 3.

The algorithm includes many steps such as the acquisition of 3-D acceleration and 3-D angular velocity, data filtering and fall detection. The algorithm uses both 3-D acceleration and 3-D angular velocity because according to Gia et al. [56], they help to improve the accuracy of a fall algorithm. Similar to ECG signals, motion-based signals are affected by surrounding noise. Therefore, noise must be removed by using filters (e.g., moving average and 50 Hz low pass) to achieve a high quality of signals. The filter data is used to calculate activity-related parameters by the following formulas [56]:

$$SVM_i = \sqrt{{x_i}^2 + {y_i}^2 + {z_i}^2} \tag{1}$$

$$\Phi = \arctan\left(\frac{\sqrt{y_i^2 + z_i^2}}{x_i}\right) * \frac{180}{\Pi} \tag{2}$$

SVM: Sum vector magnitude
i: sample number
x,y,z : accelerometer value or gyroscope value of x, y, z axis
Φ : the angle between y-axis and vertical direction

Table 1
Formulas for calculating corrected QT interval.

Algorithm	Formula
Bazett (QTcB) [49]	QTc = QT / ($\sqrt{RR}$)
Fridericia (QTcFri) [50]	QTc = QT / ($\sqrt[3]{RR}$)
Framingham (QTcFra) [51]	QTc = QT + 0.154 x (1 − RR)
Hodges (QTcH) [52]	QTc = QT + 0.00175 x ([60 / RR] − 60)
Rautaharju (QTcR) [53]	QTc = QT − 0.185 * (RR − 1) + k (k = +0.006 s for men and +0 s for women)

The calculated parameters are compared with a first set of thresholds including activity status categorization threshold and fall detection threshold. For example, activity status categorization threshold consists of 1.2 g and 30 deg/s for 3-D acceleration and 3-D angular velocity, respectively. For detecting a human fall, the first fall detection threshold is 1.6 g and 100 deg/s for 3-D acceleration and 3-D angular velocity, respectively. If the collected data surpasses the first set of thresholds, they are marked as "possibility" data and they are compared with both the second set of thresholds and the historical data. For instance, 1.6 g and 100 deg/s can be used as the second activity status categorization thresholds while 1.8 g and 130 deg/s can be used as the second fall detection thresholds for 3-D acceleration and 3-D angular velocity, respectively. If one of the values surpasses the second fall detection thresholds, a human fall is detected. Simultaneously, the activity status is detected based on the number of values surpassing the first and second activity status categorization threshold. In most of the cases, motion data only surpasses the first activity status threshold when a person is walking. When the person is running, the number of values which are higher than the second activity status threshold is large. The results of comparing the data with the history data help to categorize activity status more accurately. For instance, some periods of the 3-D acceleration should have a similar pattern (i.e., rising up to peak then decreasing) when a person walks or runs. In addition, the results help to detect the malfunctioned sensors. If one of the sensors (accelerometer or gyroscope) and its historical data do not provide the expected values (i.e., 1 g and 0 deg/s for acceleration and angular velocity, respectively when standing or lying in bed), the push notification service is triggered.

4.2. Interoperability

In general, most of the traditional monitoring systems merely support a specific type of sensor nodes such as Wi-Fi-based node for EMG monitoring [43], 6LoWPAN-based node for ECG monitoring [21,22], classic Bluetooth-based node for ECG, EMG monitoring [44] or BLE-based node for human fall detection [23]. Some other systems can support some of the wireless communication protocols such as Wi-Fi and BLE [25]. These systems are not suitable for sensor nodes using other communication protocols such as LoraWan, Zigbee or nRF. Fog computing with its capability offers interoperability to solve these challenges. The interoperability is a capability of supporting not only sensors from different manufacturers but also different communications protocols including wire and wireless protocols. For example, Fog-assisted smart gateways can support Ethernet, Wi-Fi, classic Bluetooth, BLE, nRF, and 6LoWPAN. Depending on applications, other wire or wireless communication protocols can be added into Fog-assisted smart gateways. For instance, LoraWan can be added for supporting long-range distance related applications. When the new hardware (i.e., LoraWan chip) is added to a smart gateway, the operating system of the gateway automatically detects a new device or component. A new thread can be created for transmitting the data via the added component. Sensor nodes in the Fog-based system can work both independently and cooperatively. The sensor nodes can communicate with each other via Fog-assisted smart gateways. In this work, smart gateways support Wi-Fi, BLE, Ethernet, and nRF.

4.3. Security with lightweight cryptography

In health monitoring IoT systems, the connection between sensor nodes and gateways is often the most vulnerable part of the system. The main reason is that the sensor nodes are wearable and resource-constrained devices. Therefore, they cannot run complex security algorithms. Even though complex security algorithms can be run successfully at sensor nodes, they are not applied because latency requirements of the system might be infringed and their battery is depleted. In many IoT systems [24,23,56], raw data is often transmitted for saving sensor nodes' battery life. This approach is dangerous because data can be listened by unauthorized parties. In the worst case, they can instruct commands to cause a harm to a patient. For example, Klonoff [57] uses his software to steal the security credential of the glucose monitoring system. As a result, he has a full control to an insulin pump. In order to avoid such cases, lightweight security algorithm must be run at sensor nodes. The algorithm must provide some levels of security while sensor node's battery life cannot be reduced significantly. In the paper, the AES algorithms [58] are applied. The AES algorithm consists of four basic primary operations (i.e., SubBytes, ShiftRows, MixColumns, and AddRoundKey). Each sensor node or a group of sensor nodes has its private keys for encrypting the data while a gateway has all private keys of all sensor nodes and groups of sensor nodes. The keys are hard-coded in the sensor node's firmware and their length can be 128, 192 or 256 bits. In detail, each sensor node has three different private keys where each private key has an ID and is used during a period of time (e.g., 1 h). Before a new key is applied, the sensor node sends messages to inform a corresponding gateway about the key ID. At a smart gateway, the encrypted data received will be decrypted by the correct private key which has been retrieved from a table of all private keys based on the received ID.

5. Test-bed and system implementation

A complete IoT-based system with Fog computing for continuous glucose, ECG, body temperature and body motion monitoring is implemented. The system includes 4 smart gateways, 6 contextual sensor nodes, 4 e-health sensor nodes, Cloud servers, and end-user terminals such as mobile apps. Two gateways are placed in two adjacent rooms while other two are placed at corridors. These gateways are connected to the Internet via Ethernet cables and supplied with power from wall sockets. Each of the rooms has 3 contextual sensor nodes placed at middle, top and back corners of the room. Each e-health sensor node is attached to a chest of volunteers who are about 30 years old and healthy male and female people. The e-health sensor node collects ECG data via electrodes placed at left arm, right arm, and left leg. The experimented rooms are office rooms consisting of computers and furniture such as tables and chairs. The detailed setup is shown in Fig. 4. Detailed information of the system's components are explained as follows:

5.1. Sensor layer implementation

The system has two types of sensor nodes consisting of contextual sensor nodes and e-health sensor nodes. Each sensor node has

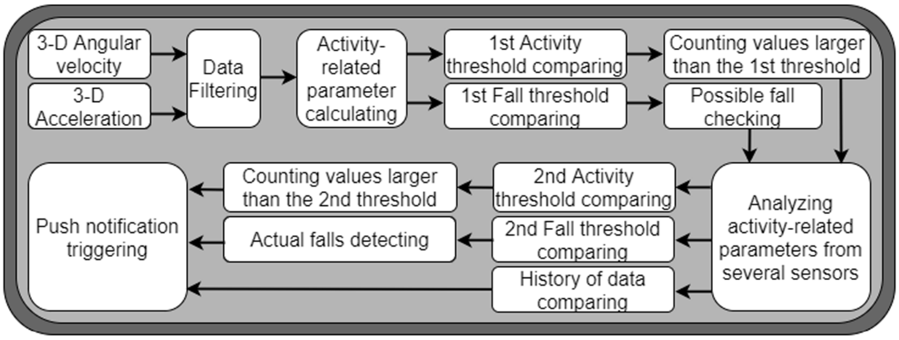


Fig. 3. Activity status and fall detection algorithm.

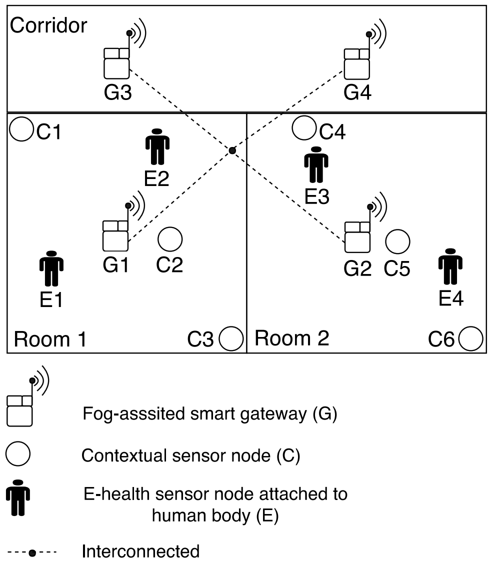


Fig. 4. Test-bed scenario.

five primary components including microcontroller, sensors, energy harvesting unit, power management unit and wireless communication chip. These sensor nodes are discussed in detail as follows:

An ultralow power ATmega328P-8-bit AVR microcontroller is used in sensor nodes. The microcontroller flexibly supports different frequencies (e.g., up to 20 MHz) and various sleep modes for saving energy. In the proposed system, a sensor node merely performs simple computational tasks whilst heavy computational tasks are processed at Fog. Therefore, the sensor node does not need to run at a high clock frequency for saving energy consumption. In the implementation, 1 MHz clock frequency is applied to all sensor nodes.

The microcontroller supports different communication interfaces such as SPI, I2C, UART and many GPIO ports (i.e., digital and analog ports). Correspondingly, it can connect to various sensors and components produced by different manufacturers. Furthermore, the microcontroller has 1 kB EEPROM and 2 kB internal SRAM. Hence, it is capable of supporting many libraries for collecting data from different sensors. In [56], authors show that SPI is more energy-efficient and has a higher bandwidth than other interfaces. Therefore, SPI is used in most of the cases such as the connection between the microcontroller and other components (i.e., wireless communication chip and sensors). In case of unavailable SPI, I2C is preferred.

Contextual sensor nodes are equipped with BME280, SNS-MQ2, SNS-MQ7, SNS-MQ135 for collecting room temperature, humidity, and air quality levels. DHT22 is a small size humidity and temperature sensor which outputs the calibrated digital signals. With a high operating range (i.e., 0%–100% for humidity and −40–80 Celsius degrees), the sensor can operate in harsh environments. The sensor has a high resolution (0.1% RH for humidity and 0.1 Celsius for temperature) and it is accurate (e.g., +−2%RH and +−0.5 Celcius). SNS-MQ2, SNS-MQ7, and SNS-MQ135 are air sensors for collecting LPG, propane, hydrogen, methane, CO, NH3, NOx, alcohol, benzene, smoke, CO2 from the air. These contextual sensor nodes are fixed in a room. Therefore, the contextual sensor nodes can use power from a large capacity battery (e.g., 3.7 V 10 000 mAh Lithium battery having a size of 5 x 120 x 90 mm) or from a wall socket with a voltage adaptor. The detailed information of power consumption of these contextual sensor nodes is presented in Section 6.

E-health sensor node can be categorized into low data rate, high data rate, and hybrid nodes where hybrid node consists of both low and high data rate sensors. Low data rate nodes are equipped with a glucose sensor and a body temperature sensor. The glucose sensor includes an implantable sensor under a patient's skin and a transmitter placed on a top of the skin. In the implementation, the transmitter is connected to the microcontroller via SPI. The glucose sensor collects glucose level every 5 min as the glucose level does not change rapidly. Similarly, the body temperature sensor (i.e., BME280 produced by Bosch) is connected to the microcontroller via SPI. The temperature data is collected every 2 min.

High data rate e-health sensor nodes are equipped with a motion sensor and an ECG analog front-end. A motion sensor (i.e., MPU-9250) is an ultralow power sensor for collecting 3-D acceleration, 3-D angular velocity, and 3-D magnetism. The data rate of the motion sensor is 50 samples/s. The low-power ECG analog front-end can be TI ADS1292 or AD8232. In the implementation, a data rate of 125 samples/s is used.

A power managing unit is a low-power Schmitt trigger based circuit having several super-capacitors. The power managing unit can detect the energy level of the battery, power, and current via INA226 which is a current shunt and power monitor produced by TI.

The wireless communication chip is nRF24L01 which is an ultralow power RF transceiver supporting many-to-many communications. The nRF24L01 chip supports up to 2 Mbps. However, 250 kbps is used for saving energy consumption. The chip can run with low-power, average-power or maximum power. In this paper, it is configured to run at low-power mode and is connected with the microcontroller via SPI.

5.2. Smart gateway and fog services implementation

A smart gateway of the system is built with Pandaboard which has a 1.2 GHz dual-core Arm Cortex microprocessor and 1 GB low-power DDR2-RAM. Pandaboard supports several communication interfaces such as Wi-Fi, Bluetooth, and Ethernet by built-in components. In addition, it supports a 32 GB SD-card which can be used for installing embedded operating systems. In the implementation, a lightweight version of Ubuntu based on Linux is used. Many services such as security (AES), data decompression, data processing and data analysis have built on the operating system.

For providing interoperability, several wireless communication components are added into Pandaboard. To receive data from the sensor nodes which are equipped with nRF, an nRF24L01 chip is connected to Pandaboard via SPI. The nRF24L01 chip in the gateway is similar to the nRF24L01 chip used in sensor nodes except that it has some extra circuits and uses a large external antenna. Using a large antenna costs higher energy consumption, but it increases the quality of collected signals. For supporting 6LoWPAN, a composition of a CC2538 module and a SmartRF06 board is added into Pandaboard. The detailed information of the connection can be seen in our previous works [21,22]. These components are connected to Pandaboard via Ethernet and USB ports because Ethernet can provide high transmission bandwidth. To support several BLE sensor nodes, the smart gateway can be equipped with BLE components (i.e., CYBLE-202007-01 provided by Cypress Semiconductor). The number of added BLE components depends on the available UART ports of Pandaboard. However, these UART ports are limited. To overcome the issue, an FTDI chip and an ATmega328P microcontroller are added to Pandaboard. These components can facilitate 7 BLE components which are connected via software-based or hardware-based UART. When the number of BLE sensor nodes is larger, more components can be added to Pandaboard via USB hubs. Due to a built-in Wi-Fi chip in Pandaboard, it can support high-speed Wi-Fi based sensor nodes.

Since sensor nodes run the AES algorithm for encrypting transmitted messages, the smart gateway has to run the AES algorithm for decrypting the received messages. For being compatible with other services, the AES algorithm run in the smart gateway is implemented in Python. Decrypted messages are stored in the smart gateway's database which is built from MongoDB and JSON objects. The database combines with HTML5, XML, Django, CSS, JavaScript to provide a localhost with user interface. When a user accesses the data, the system categorizes the user as a local user or an external user via the categorization service. When a user belongs to a local network, the system directly sends the data from Fog to the user. When a user does not belong to a local network, data is sent to the user via Cloud servers. This method helps to reduce the large latency of transmitting real-time data via Cloud servers when a user belongs to a local network. The categorization service is implemented by Python and Linux packages such as "iw" and "iwconfig".

For protecting the smart gateways, Iptables and a part of our advanced security methods presented in [28,59] are implemented in the system. Particularly, the part of these methods for protecting the connection between the smart gateways and Cloud servers is merely applied. Other parts are not utilized because they can cause an increase in latency and energy consumption of sensor nodes.

In the smart gateway implementation, the QT detection algorithm, data filtering, and data processing are implemented in Python because it remains the consistency with other services. For instance, moving average filters and 50 Hz low pass filters for removing noise out of ECG signals are implemented in Python.

5.3. Cloud servers and end-user terminals

In the implementation, Google Cloud servers, API, and Cloud's services are used for storing, processing data and providing advanced services. For instance, the push notification service of the system is primarily implemented at Cloud. Similar to localhost in Fog, Cloud servers host the global web-pages which can show both real-time data and the historical data in textual and graphical forms. For accessing data, end-users can use the global web-pages or a mobile app which is built by PhoneGap for supporting both IOS and Android.

6. Experimental results

At Fog-assisted smart gateways, collected e-health data such as acceleration, angular velocity, and ECG is processed with 3 primary steps, shown in Figs. 5 and 6, including data filtering, baseline detection, and baseline wander removal. As mentioned, raw data is filtered to eliminate noise from surrounding environment. In most of the cases, the filtered data has a different baseline than the reference baseline which is 1 g, 0 deg/s, and 0 voltage for acceleration, angular velocity, and ECG, respectively. Therefore, baseline detection and baseline wander removal are applied for shifting the signals' baselines into the expected ones. Fig. 5 does not show the baseline detection step because the signals and the detected baseline overlapped in several periods. Two different methods are applied for detecting the baseline of different signals. A mean value is applied for detecting the baseline of acceleration and angular velocity while Daubechies d4 wavelet transform is applied for detecting the baseline of ECG. The processed data has the same magnitude and waveform as the filtered data and are used as inputs for algorithms such as fall detection, heart rate calculation and QT wave's length extraction shown in Section 4.1.

Parameters of the experimented room environment are 22 degrees Celsius, 31% humidity, 0.6 ppm for CO, around 8 ppb for NO_2, and 6 ppb for SO_2. These values indicate that the room environment is good. Body temperature and glucose are collected but it is not used for the comparison because its value merely slightly changes during different activities. For instance, the collected body temperature and glucose of a volunteer are around 37 degrees Celsius and around 100 mg/dl for all activities except for training (e.g., running), respectively. When a volunteer intensively runs, the core temperature increases. The blood glucose level varies depending on the monitoring time. For instance, the glucose level in the morning is lower than in the afternoon and after lunch. In our experiments, the glucose level of that person fluctuates around 90–98 mg/dl for all measurement cases.

Fig. 7 shows acceleration, angular velocity, and ECG data from the experiment. The data is collected from different activities such as standing, lying in bed, walking and running. Data in the same category is presented in the same scale for fair comparisons. Data in both cases of lying in bed and standing is stable and similar (e.g., 1 g acceleration, 0 deg/s angular velocity, and stable ECG waveform). The ECG waveform has waves such as P wave, Q wave, R wave, S wave and T wave which are needed for algorithms (e.g., heart rate calculation and QT's length extraction). In case of lying and standing, results of the algorithms show that heart rate is 59 beats/minute, the length of QT and QTcB is around 390 and 387, respectively. In Fig. 7(a), acceleration and angular velocity slightly changes (i.e., 1.08 g acceleration and 10 deg/s angular velocity) at some moments (i.e., around 166–172th samples and 235–240th samples). These variations are expected as the user slightly moves his body two times during lying in bed. Fortunately, the ECG waveform in those moments remains stable (e.g., ECG before and during those moments is similar in terms of the number of waves, waves' magnitude, and shape of the waves such as P wave and QRS waves).

Although acceleration and angular velocity fluctuate during walking, the amplitude of the fluctuation is small when comparing to pre-defined thresholds (i.e., 2 g and 200 deg/s for acceleration and angular velocity, respectively) in the fall detection algorithm. However, the fluctuation helps to identify a walking status and calculate the number of steps of a user (e.g., by counting the number

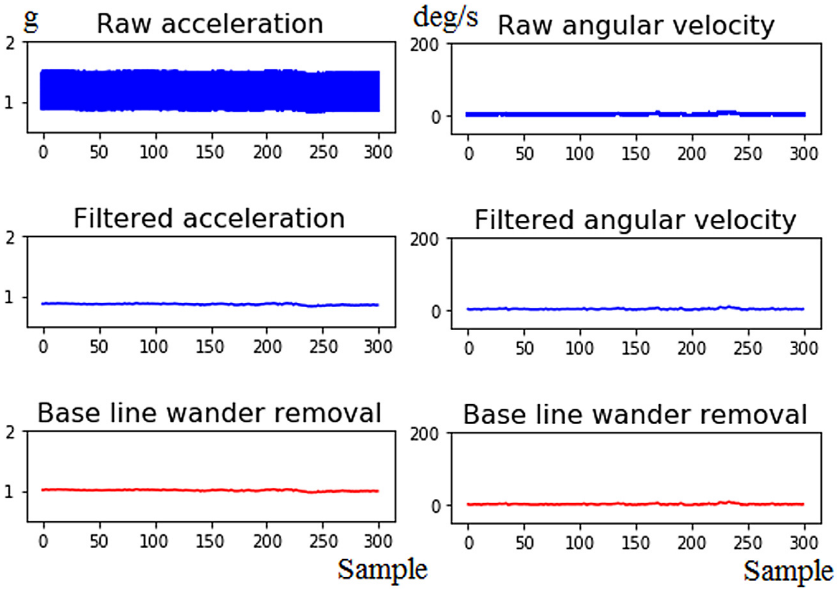


Fig. 5. Acceleration and angular velocity processing.

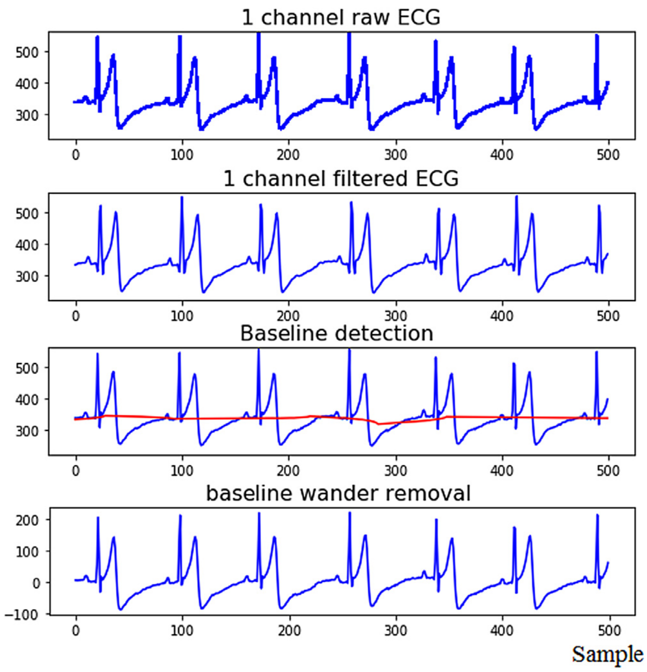


Fig. 6. ECG processing.

of top peaks of the fluctuation). Angular velocity can be used as the compliment parameter to distinguish different activities such as movement and non-movement. The shape of angular velocity waveform can be different depending on the walking or running style of the user (e.g., swinging arms and hands during walking). ECG moderately changes during walking. QRS wave, T wave, and QT's length can be detected in most of the ECG cycles whilst P wave merely appears in some of the ECG cycles (i.e., one per every 6–8 ECG cycles). In this case, QT's length and QTcB's length is around 395 and 392 ms, respectively. Comparing with ECG waves in standing and lying in bed, the ECG wave during walking is not as good as others in terms of stability.

In case of running, data fluctuates dramatically when compared with their baseline. Acceleration, in this case, shows the number of steps of a user (i.e., top peaks have much higher amplitude than the amplitude of the acceleration baseline which is 1 g). At a moment of 87–92th samples, the acceleration is higher than the predefined acceleration thresholds 2 g in the fall detection algorithm. However, the fall case is not detected by the system in this case because of two reasons. First, during 87–92th samples, angular velocity is not higher than angular velocity thresholds in the fall detection algorithm. Second, historical data shows that none of the sensors is malfunctioned. Similarly, the case of angular velocity at 58–64th samples is higher than angular velocity threshold but the fall event is not detected. ECG during walking is not as good as ECG in other statuses such as standing and lying in bed. However, P, Q, R, S and T waves appear in some of the ECG cycles (i.e., at 140–150th samples). The value of QT's length and QTcB's length varies dramatically. Therefore, it is not recommended to monitor ECG during intensive activities such as running or jumping.

In some of the experiments, a user falls in some random moments. There are 3 different falling cases during walking such as fall forward, fall back and fall aside. In this paper, fall cases during walking are focused because people are more likely to fall during activities than in static cases such as lying in bed and standing. In addition, when fall cases during activities (e.g., walking) can be detected, fall cases in other static statuses (e.g., lying in bed and standing) can be also successfully detected. Acceleration, angular velocity and ECG of these fall cases during walking are shown in Fig. 8. In most of the cases, a person tends to sit up or stand up after falling. When he sits up or stands up, the acceleration should increase to peak values. Correspondingly, two peaks including a falling peak and a standing/sitting up peak (i.e., having a higher amplitude compared to other peaks) are expected to appear in both acceleration and angular velocity waveform. In all experimented cases, these two peaks appear in the collected data. For example, two peaks of acceleration and two peaks of angular velocity appear 58–65th samples and 110–115th samples in the fall forward case. The first peak representing a fall moment often has the highest amplitude while the amplitude of the second peak varies depending on specific situations such as sitting up, standing up or crawling up. Therefore, the second peak can be smaller or larger than predefined thresholds. A distance between two peaks varies depending on different situations. In case of fall aside, at a moment after falling, angular velocity (at 125-1130th samples) does not change dramatically (i.e., reaching to a peak value) while acceleration

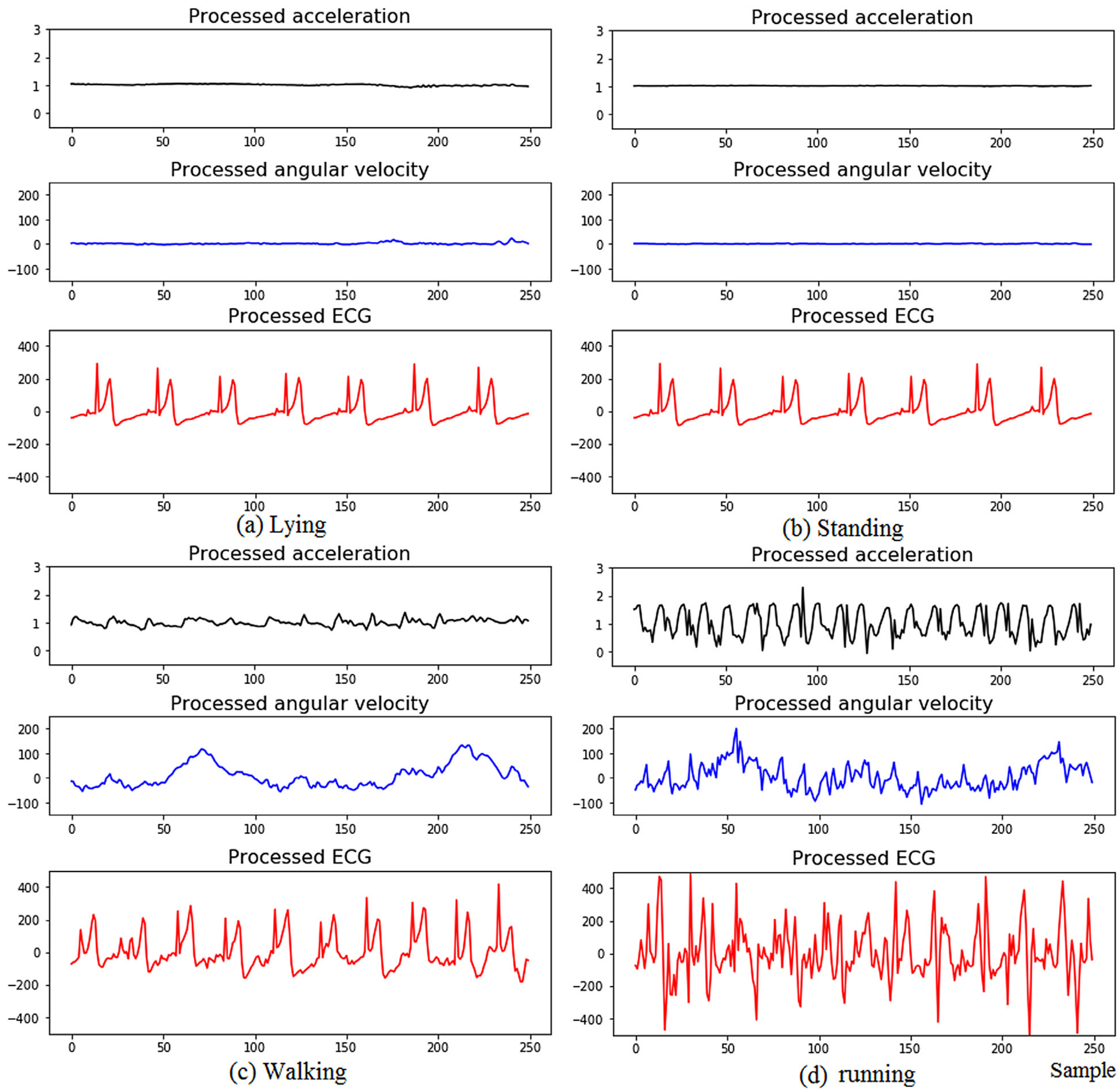


Fig. 7. Acceleration, angular velocity and ECG in different activity status.

reaches to a peak value. The reason is that a user slowly crawls and sits up after falling. It can be concluded that during fall moments, ECG fluctuates whilst ECG remains good during other moments.

In the experiments, power consumption of an e-health sensor node in different configurations is measured. In each configuration, a sensor or a group of several sensors is integrated into a sensor node. Data collected from the sensor(s) is transmitted to a gateway via nRF. Detailed information of the configurations and results of power consumption are shown in Table 2 and Fig. 9, respectively. In Table 2, the first four configurations (i.e., from Conf 1_E to Conf 4_E) are the configurations of high data rate e-health sensor nodes while the other three configurations (i.e., from Conf 5_E to Conf 7_E) are the configurations of low data rate e-health sensor nodes. The last configuration (i.e., Conf 8_E) is for the hybrid sensor nodes having low and high data rate sensors. Results show that high data rate sensors (i.e., motion sensor and ECG sensor) consume a large amount of energy while low data rate sensors consume a small amount. By using 1000 mAh Lithium battery (a size of 60 x 32 x7 mm), the low data rate sensor node (i.e., Conf 7_E) can be used up to 1639 h while the high data rate sensor node (i.e., Conf 4_E) can be used up to 173 h. In case of the hybrid sensor node, it can be used up to 157.5 h with the same battery.

Contextual sensor nodes collect and send data in every second to a gateway. Configurations and power consumption of the sensor nodes are showed in Table 3 and Fig. 10. Results show that these sensors for collecting air-related parameters (i.e., Mq2, MQ7, and MQ125) consume a large amount of power. When applying the 10 000 mAh battery having a size of 5 x 120 x 90 mm, contextual sensor nodes can operate up to 46 h. As mentioned, contextual sensor nodes are fixed in a room. Therefore, it is recommended that contextual sensor nodes should be supplied with wall-socket power. The battery is only used for a case of electricity cut.

In the experiments, sensor nodes in different configurations are applied with AES-256. Power consumption of the sensor nodes with and without is shown in Fig. 11. The results show that power consumption of the sensor nodes increases slightly (i.e., about 11% of total power of the e-health hybrid sensor node) when applying encrypting with AES-256. In this case, the hybrid sensor node can

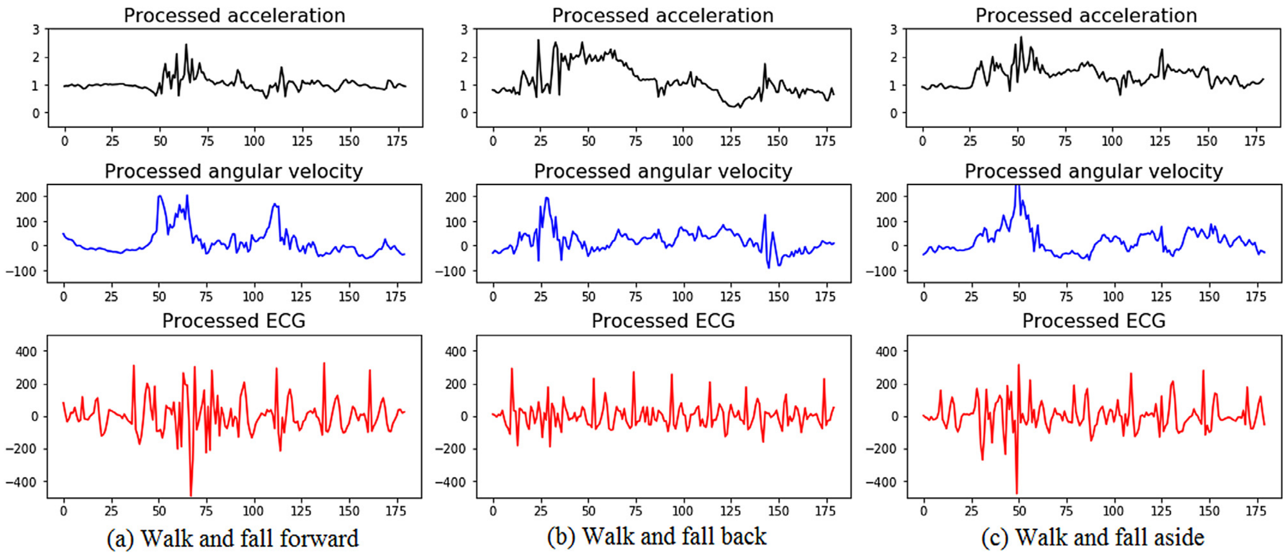


Fig. 8. Acceleration, angular velocity and ECG when a user falls in different cases.

Table 2
Configurations and average current draw of e-health sensor nodes.

Configuration	Conf 1_E	Conf 2_E	Conf 3_E	Conf 4_E	Conf 5_E	Conf 6_E	Conf 7_E	Conf 8_E
BME280					X		X	X
Samples/minute					1		1	1
Glucose sensor						X	X	X
Sample/minute(s)						1	1	1
MPU-9250			X	X				X
Sample/second(s)			50	50				50
AD8320	X	X	X	X				X
Sample/second(s)	60	120	60	120				120
Voltage (V)	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
Current (mA)	1.65	3.15	4.58	5.76	0.22	0.45	0.61	6.35

Table 3
Configurations and average current draw of contextual sensor nodes.

Configuration	Conf 1_C	Conf 2_C	Conf 3_C	Conf 4_C	Conf 5_C
MQ2	X				X
MQ7		X			X
MQ135			X		X
DHT22				X	X
Voltage (V)	3.3	3.3	3.3	3.3	3.3
Current (mA)	84.3	66.1	73.6	1.68	216.5

Table 4
Latency of sensor nodes and smart gateways when applying AES-256.

Device	Algorithm	Latency (μs)
Our sensor node	AES-256 encryption	1358
Smart gateway	AES-256 decryption	43
Smart gateway	AES-256 encryption	52
Cloud server	AES-256 decryption	10

operate up to 183 h. In case of contextual sensor nodes, power consumption of the nodes increases less then 0.01%. The contextual sensor nodes still operate up to 46 h when they are supplied with the 10000 mAh battery.

Results of latency for encrypting and decryption with AES-256 shown in Table 4 indicate that the latency of the sensor node increases slightly (i.e., about 1.358 ms). Correspondingly, the requirements of latency (e.g., less than 500 ms for ECG and approximately in seconds for glucose) are still fulfilled.

In this paper, power consumption of our sensor node is compared with other state-of-the-art nodes. Results shown in Table 5 indicate that our sensor node is one of the most energy-efficient

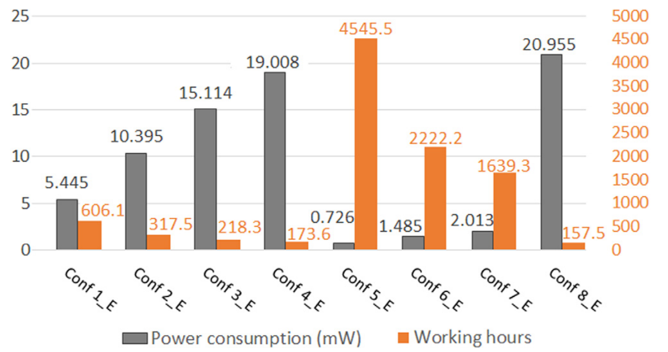


Fig. 9. Power consumption of e-health sensor node (in different configurations) with 1000 mAh battery.

sensor nodes even though our sensor node is equipped with many types of sensors for collecting motion-related data, ECG, body temperature, and glucose.

Table 5
Comparisons of the proposed sensor node and other state-of-the-art sensor nodes.

Sensor node	Microcontroller (MHz)	Flash (kB)	SRAM (kB)	Sensor (s)	Voltage (V)	Power consumption
in [60]	ATmega32L (8)	256	8	Motion	5	Low
in [61]	ATmega128L (8)	128	4	Motion	3	Medium
in [62]	MSP430F2617 (8)	92	8	Motion	3.7	Low
in [63]	MSP430 (8)	48	10	Motion	3	Low
in [64]	MSP430F1611 (8)	48	10	Motion	3.7	High
in [24]	ATmega328P (8)	32	2	Motion	3	Low (36.68 mW)
in [21]	Arm Cortex M3 (24)	512	32	ECG	3.3	Ultralow
in [65]	MSP430 (8)	48	10	ECG	3.3	Low (36 mW)
in [66]	ATmega328 (8)	32	2	ECG	3.3	Low
in [67]	MSP430 (8)	48	10	ECG	3.3	Medium (64 mW)
in [30]	ATmega328P-PU (8)	32	2	Motion, ECG, body temperature	3	Ultralow (21.3 mW)
in our work	ATmega328P-PU (1)	32	2	Motion, ECG, body temperature, glucose	3.3	Ultralow (23.4 mW)

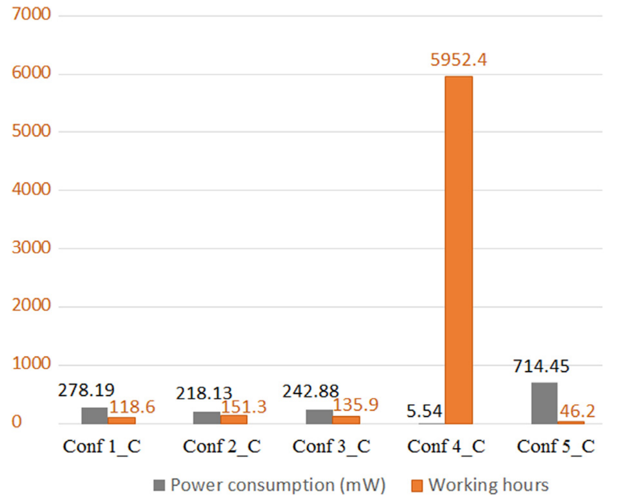


Fig. 10. Power consumption and working hours of contextual sensor nodes (in different configurations) with 10 000 mAh battery.

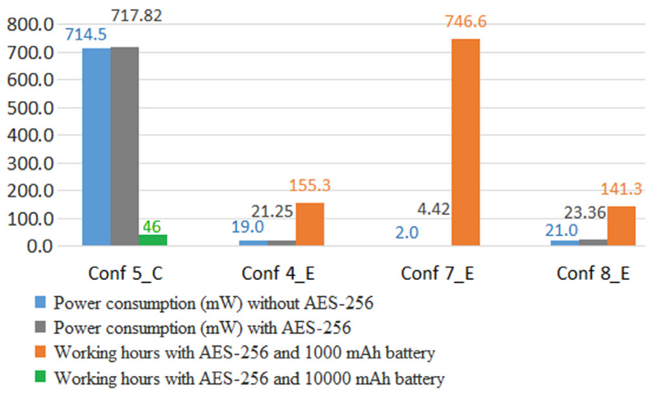


Fig. 11. Power consumption and working hours of sensor nodes in different configurations with AES-256.

7. Conclusion and future directions

In this paper, we presented a novel and smart Fog-based system for continuous, remote monitoring glucose, ECG and other signals in real-time. The complete IoT system consisting of sensor nodes, smart gateways with Fog computing and a back-end server was implemented. By simultaneous monitoring different types of signals from bio-signals (i.e., glucose, ECG, and body temperature) to contextual signals (i.e., air quality, room humidity and temperature), the accuracy of disease analysis was improved. By leveraging smart gateways and Fog computing in the system, loads of sensor nodes were alleviated whilst augmented services (e.g., local data storage, security, interoperability) were provided. In addition, we proposed algorithms for calculating the duration of QT length, fall detection, and activity status detection, respectively. These algorithms combining with the push notification service helped to improve quality of healthcare services. Results from the experiments showed that the complete sensor node for gathering glucose, ECG, motion-related signals and body temperature is one of the most energy-efficient sensor nodes and it can operate in a secured way up to 157.5 h with a 1000 mAh Lithium battery.

The future directions of this work involve diverse field. As mentioned above the sensor node battery lasts around a week. Frequent battery replacement is undesirable especially with wearable sensors as this might cause inconvenience, discomfort and even pain in case of implanted sensors.

Energy harvesting of ambient sources can be exploited for powering, recharging or extending the time between recharging of wearable micro-power sensor nodes. It involves converting the ambient energy inherent in the sensor node's environment into electrical energy. By doing so, a sensor node will have the opportunity to extend its life to a range determined by the failure of its own components rather than by its previously limited power supply. Several sources have been investigated for supplying energy to wearable sensors nodes such as solar, electromagnetic waves, indoor light, and even harvesting energy from the body of the person wearing the sensor, like piezoelectric energy from footfalls and thermal energy from temperature gradients on the skin. In previous works [31], the feasibility of RF energy harvesting has been investigated as a source for powering to the sensor. The targeted frequency band for harvesting was 925 MHz GSM band, and due to their low threshold voltage (0.2–0.3 V), Schottky diodes were utilized as rectifying element. Despite this low turn-on voltage, the rectifier will not be able to deliver any power to the load unless a voltage of 0.2 V or higher is available for forward driving the Schottky diode. For this reason, in the current work in progress, low threshold voltage diodes connected transistors are being equipped with mini solar panels attached to the transistors' gates aiding in securing the required turn-on voltage for the transistor rendering the harvesting circuit more sensitive and able to operate even at very low RF signals available at its antenna, −15 dBm [68]. While RF energy harvesting is currently able to directly power the sensor in a standard alone scenario, it can be exploited along with an efficient power management unit to recharge the battery and extend the life of the sensor node. On the other hand, the contextual sensor nodes and gateways can be completely powered autonomous when being powered by a solar along with a much simple power management unit consisting of a boost converter, a buck converter and a voltage regulator.

The use of flexible wearable and printed sensors is also being investigated, other than low fabrication cost, lightweight, better mechanical and thermal properties compared to rigid non-flexible sensors, they are more convenient and comfortable when being used for monitoring on bendable surfaces like arms and thighs.

Acknowledgments

The authors would like to thank Academy of Finland, Nokia Foundation, Finland, University of Turku Graduate School (UTUGS), the doctoral programme in mathematics and computer sciences (MATTI), Finnish Foundation for Technology Promotion (TES), Finland for the financial support this project.

References

[1] R.T. Robinson, et al., Mechanisms of abnormal cardiac repolarization during insulin-induced hypoglycemia, Diabetes 52 (6) (2003) 1469–1474.
[2] R. Vedanthan, et al., Sudden cardiac death in low-and middle-income countries, Global Heart 7 (4) (2012) 353–360.
[3] Atrial fibrillation fact sheet. Available: https://www.cdc.gov/dhdsp/data_statistics/fact_sheets/fs_atrial_fibrillation.htm [Accessed: May 2018].
[4] H.T. Nguyen, et al., Detection of nocturnal hypoglycemic episodes (natural occurrence) in children with type 1 diabetes using an optimal bayesian neural network algorithm, in: IEEE Engineering in Medicine and Biology Society 2008, IEEE, 2008, pp. 1311–1314.
[5] D. Pickham, E. Flowers, B.J. Drew, Hyperglycemia is associated with qtc prolongation and mortality in the acutely ill, J. Cardiovasc. Nurs. 29 (3) (2014) 264.
[6] L.L. Nguyen, S. Su, H.T. Nguyen, Effects of hyperglycemia on variability of rr, qt and corrected qt intervals in type 1 diabetic patients, in: IEEE Engineering in Medicine and Biology Society 2013, IEEE, 2013, pp. 1819–1822.
[7] K.D. Kochanek, et al., Deaths: Final data for 2014, Natl Vital Statist. Rep. 65 (4) (2016).
[8] WHO. Diabetes. 2017. Available: http://www.who.int/mediacentre/factsheets/fs312/en/ [Accessed: Jan 2018].
[9] Rates of new diagnosed cases of type 1 and type 2 diabetes on the rise among children, teens. Available: https://www.nih.gov/news-events/news-release/rates-new-diagnosed-cases-type-1-type-2-diabetes-rise-among-children-teens [Accessed: May 2018].
[10] C.D. Mathers, D. Loncar, Projections of global mortality and burden of disease from 2002 to 2030, PLoS Med. 3 (11) (2006) e442.
[11] D.A. Sterling, J.A. O'connor, J. Bonadies, Geriatric falls: injury severity is high and disproportionate to mechanism, J. Trauma Acute Care Surg. 50 (1) (2001) 116–119.
[12] J.A. Stevens, et al., The costs of fatal and non-fatal falls among older adults, Inj. Prev. 12 (5) (2006) 290–295.
[13] CDC. Important facts about falls. 2017. Available: https://www.cdc.gov/homeandrecreationalsafety/falls/adultfalls.html [Accessed: Jan 2018].
[14] S. Volpato, et al., Risk factors for falls in older disabled women with diabetes: the women's health and aging study, J. Gerontol. A Biol. Sci. Med. Sci. 60 (12) (2005) 1539–1545.
[15] Cardiovascular Disease & Diabetes. Dec.2017. Available: https://www.heart.org/HEARTORG/Conditions/More/Diabetes/WhyDiabetesMatters/CardiovascularDisease-DiabetesUCM313865Article.jsp/ [Accessed: Jan.2018].
[16] M.S. Kirkman, et al., Diabetes in older adults, Diabetes Care 35 (12) (2012) 2650–2664.
[17] N. Paul, T. Kohno, D.C. Klonoff, A review of the security of insulin pump infusion systems, J. Diabetes Sci. Technol. 5 (6) (2011) 1557–1562.
[18] T.N. Gia, et al., Fog computing in healthcare internet-of-things : A case study on ECG feature extraction, in: IEEE International Conference on Computer and Information Technology (CIT'15), IEEE, 2015, pp. 356–363.
[19] T.N. Gia, et al., Fog computing in body sensor networks: An energy efficient approach, in: Proc. IEEE Int. Body Sensor Netw. Conf.(BSN), 2015, pp. 1–7.
[20] Tuan Nguyen Gia, Amir M Rahmani, Tomi Westerlund, Pasi Liljeberg, Hannu Tenhunen, Fog computing approach for mobility support in internet-of-things systems, IEEE Access 6 (2018) 36064–36082.
[21] T.N. Gia, et al., Customizing 6lowpan networks towards internet-of-things based ubiquitous healthcare systems, in: NORCHIP, 2014, IEEE, 2014, pp. 1–6.
[22] T.N. Gia, et al., Fault tolerant and scalable IoT based architecture for healthcare monitoring, in: Sensors Applications Symposium (SAS), IEEE, 2015, pp. 1–6.
[23] I. Tcarenko, et al., Energy-efficient iot-enabled fall detection system with messenger-based notification, in: International Conference on Wireless Mobile Communication and Healthcare, Springer, 2016, pp. 19–26.
[24] T.N. Gia, et al., Iot-based fall detection system with energy efficient sensor nodes, in: Nordic Circuits and Systems Conference (NORCAS), 2016 IEEE, IEEE, 2016, pp. 1–6.
[25] A.M. Rahmani, et al., Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems, in: Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE, IEEE, 2015, pp. 826–834.
[26] A.J. Jara, et al., An internet of things-based personal device for diabetes therapy management in ambient assisted living (aal), Pers. Ubiquitous Comput. 15 (4) (2011) 431–440.
[27] R.S.H. Istepanian, et al., The potential of Internet of m-health Things ḿ-IoT f́or non-invasive glucose level sensing, in: Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE, IEEE, 2011, pp. 5264–5266.
[28] S.M. Moosavi, et al., Sea: a secure and efficient authentication and authorization architecture for iot-based healthcare using smart gateways, Procedia Comput. Sci. 52 (2015) 452–459.
[29] S.M. Moosavi, et al., Session resumption-based end-to-end security for healthcare internet-of-things, in: CIT 2015, IEEE, 2015, pp. 581–588.
[30] T.N. Gia, et al., Low-cost fog-assisted health-care iot system with energy-efficient sensor nodes, in: Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International, IEEE, 2017, pp. 1765–1770.
[31] T.N. Gia, et al., IoT-based continuous glucose monitoring system: A feasibility study, Procedia Comput. Sci. 109 (2017) 327–334.
[32] A.M. Rahmani, et al., Exploiting smart e-Health gateways at the edge of healthcare internet-of-things: A fog computing approach, Future Gener. Comput. Syst. 78 (2018) 641–658.
[33] B. Negash, et al., Leveraging fog computing for healthcare iot, in: Fog Computing in the Internet of Things, Springer, 2018, pp. 145–169.
[34] H. Dubey, et al., Fog data: Enhancing telehealth big data through fog computing, in: Proceedings of the ASE BigData & SocialInformatics 2015, ACM, 2015, p. 14.
[35] Y. Cao, et al., Distributed analytics and edge intelligence: Pervasive health monitoring at the era of fog computing, in: Proceedings of the 2015 Workshop on Mobile Big Data, ACM, 2015, pp. 43–48.
[36] I. Azimi, et al., Medical warning system based on internet of things using fog computing, in: Big Data and Information Security (IWBIS), International Workshop on, IEEE, 2016, pp. 19–24.
[37] R. Craciunescu, et al., Implementation of fog computing for reliable e-health applications, in: Signals, Systems and Computers, 2015 49th Asilomar Conference on, IEEE, 2015, pp. 459–463.
[38] Y. Cao, et al., Fast: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation, in: Networking, Architecture and Storage (NAS), 2015 IEEE International Conference on, IEEE, 2015, pp. 2–11.
[39] O. Fratu, et al., Fog Computing System for Monitoring Mild Dementia and COPD Patients-Romanian Case Study, IEEE, 2015, pp. 123–128.
[40] A.M. Elmisery, et al., A fog based middleware for automated compliance with oecd privacy principles in internet of healthcare things, IEEE Access 4 (2016) 8418–8441.
[41] X. Masip-Bruin, et al., Will it be cloud or will it be fog? f2c, a novel flagship computing paradigm for highly demanding services, in: Future Technologies Conference (FTC), IEEE, 2016, pp. 1129–1136.
[42] P. Davies, I. Maconochie, The relationship between body temperature, heart rate and respiratory rate in children, Emerg. Med. J. 26 (9) (2009) 641–643.
[43] M. Jiang, et al., Iot-based remote facial expression monitoring system with semg signal, in: Sensors Applications Symposium (SAS), 2016 IEEE, IEEE, 2016, pp. 1–6.
[44] V.K. Sarker, et al., Portable multipurpose bio-signal acquisition and wireless streaming device for wearables, in: Sensors Applications Symposium (SAS), 2017 IEEE, IEEE, 2017, pp. 1–6.
[45] 2.4GHz RF - Ultra low power 2.4GHz RF ICs/solutions. Available: https://www.nordicsemi.com/eng/Products/2.4GHzRF [Accessed: May 2018].
[46] Websocket. Available: https://developer.mozilla.org/en-US/docs/Web/API/WebSocket [Accessed: May 2018].
[47] Introducing to json. Available: https://www.json.org/ [Accessed: May 2018].
[48] K.W. Goh, et al., A pda-based ecg beat detector for home cardiac care, in: IEEE-EMBS 2005. 27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005., IEEE, 2006, pp. 375–378.
[49] H.C. Bazett, An analysis of the time-relations of electrocardiograms, Ann. Noninvasive Electrocardiol. 2 (2) (1997) 177–194.
[50] L.S. Fridericia, Die systolendauer im elektrokardiogramm bei normalen menschen und bei herzkranken, J. Intern. Med. 53 (1) (1920) 469–486.
[51] A. Sagie, et al., An improved method for adjusting the qt interval for heart rate (the framingham heart study), Amer. J. Cardiol. 70 (7) (1992) 797–801.
[52] M. Hodges, Bazett's qt correction reviewed: evidence that a linear qt correction for heart rate is better, J. Am. Coll. Cardiol. 1 (1983) 694.
[53] P.M. Rautaharju, et al., Assessment of prolonged qt and jt intervals in ventricular conduction defects, Amer. J. Cardiol. 93 (8) (2004) 1017–1021.
[54] B. Vandenberk, et al., Which qt correction formulae to use for qt monitoring? J. Amer. Heart Assoc. 5 (6) (2016) e003264.
[55] F. Miao, et al., A wearable context-aware ecg monitoring system integrated with built-in kinematic sensors of the smartphone, Sensors 15 (5) (2015) 11465–11484.
[56] T.N. Gia, et al., Energy efficient wearable sensor node for iot-based fall detection systems, Microprocess. Microsyst. 56 (2018) 34–46.
[57] D.C. Klonoff, Cybersecurity for connected diabetes devices, J. Diabetes Sci. Technol. 9 (5) (2015) 1143–1147.

[58] N.F. Pub, Advanced encryption standard (aes), Fed. Inf. Process. Stand. Publ. 197 (441) (2001) 0311.
[59] S.M. Moosavi, et al., End-to-end security scheme for mobility enabled healthcare internet of things, Future Gener. Comput. Syst. 64 (2016) 108–124.
[60] O. Biros, et al., Implementation of wearable sensors for fall detection into smart household, in: Applied Machine Intelligence and Informatics (SAMI), 2014 IEEE 12th International Symposium on, IEEE, 2014, pp. 19–22.
[61] S.Z. Erdogan, T.T. Bilgin, A data mining approach for fall detection by using k-nearest neighbour algorithm on wireless sensor network data, IET Commun. 6 (18) (2012) 3281–3287.
[62] P. Pivato, et al., A wearable wireless sensor node for body fall detection, in: Measurements and Networking Proceedings, 2011 IEEE International Workshop on, IEEE, 2011, pp. 116–121.
[63] Y. Li, et al., Accelerometer-based fall detection sensor system for the elderly, in: Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on, Vol. 3, IEEE, 2012, pp. 1216–1220.
[64] F. Wu, et al., Development of a wearable-sensor-based fall detection system, Int. J. Telemed. Appl. 2015 (2015) 11.
[65] R. Dilmaghani, et al., Wireless sensor networks for monitoring physiological signals of multiple patients, IEEE Trans. Biomed. Circuits Syst. 5 (4) (2011) 347–356.
[66] S. Mahmud, et al., An inexpensive and ultra-low power sensor node for wireless health monitoring system, in: E-Health Networking, Application & Services, 2015 17th International Conference on, IEEE, 2015, pp. 495–500.
[67] S. Lee, W. Chung, A robust wearable u-healthcare platform in wireless sensor network, J. Commun. Netw. 16 (4) (2014) 465–474.
[68] M. Ali, et al., Autonomous patient/home health monitoring powered by energy harvesting, in: GLOBECOM 2017 - 2017 IEEE Global Communications Conference, 2017, pp. 1–7.

Tuan Nguyen Gia received his Master (Tech) from the University of Turku (Turku, Finland) in 2014. He received a 4-year funded position in University of Turku and a grant scholarship from Nokia Foundation for an efficient and excellent researcher in 2015. In addition, he receive a grant scholarship for excellent research from the Finnish Foundation for Technology Promotion. Currently, he works at the Internet-of-Things for Healthcare (IoT4Health) research group in Future Technologies department, University of Turku. His research field is Fog computing, Internet-of-Things, healthcare, FPGA, and autonomous system.

Imed Ben Dhaou (S'97-M'02, SM'2011) is an assistant professor at both Qassim University, Saudi Arabia and University of Monastir, Tunisia. He received Master's degree in Electrical Engineering from the Tampere University of Technology, Tampere, Finland, and the Ph.D. degree from the Royal Institute of Technology, Stockholm, Sweden, in 1997, and 2002, respectively. His research interests are in the areas of green data center, IoT for the smart grid, interconnect optimization, low-power circuit design, high-level power estimation, arithmetic for cryptosystems, Intelligent Transportation Systems, and VLSI for DSP and wireless systems. He has authored and co-authored more than 50 journals, book chapters, technical reports and conference papers in these areas. Dr. Ben Dhaou received the Best Paper Award from the 1997 Finnish Symposium on Signal Processing and a travel grant from the Ph.D. forum at DAC, Los Angeles in 2000. Since September 2014, Dr. Ben Dhaou has been serving as editor to the microelectronics journal, Elsevier. He was the Guest Editor for the special issue of on-chip signaling in deep-submicron technology, Journal of Analog Integrated Circuits and Signal Processing. Dr. Ben Dhaou is a vice-chair for the IEEE ComSoc, Saudi Section.

Mai Ali is currently an Instructor of Electrical Engineering at Alfaisal University, KSA. She obtained BSc degree (honors) from University of Khartoum, Sudan in 2008, and MSc in electrical engineering (with distinction) from the American University of Sharjah, UAE in 2011. Her research interests are in the areas of autonomous wireless sensor networks, energy harvesting,and low power RFIC design. She is a recipient of several awards and grants including College of Engineering research award, 2017 from Alfaisal University, Student research competition award form 41st European Microwave week, Manchester, 2011 and best student poster award from International Symposium of Mechatronics Applications, Sharjah ,2010, and 2016–2018 Internal Research Grants from Alfaisal University.

Amir M. Rahmani received his MSc from Department of ECE, University of Tehran, Iran, in 2009 and Ph.D. degree from Department of IT, University of Turku, Finland, in 2012. He also received his MBA jointly from Turku School of Economics and European Institute of Innovation \& Technology (EIT) Digital, in 2014. He is currently Marie Curie Global Fellow at University of California Irvine (USA) and TU Wien (Austria). He is also an adjunct professor (Docent) in embedded parallel and distributed computing at 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 is the author of more than 150 peer-reviewed publications. He is a senior member of the IEEE.

Tomi Westerlund received his PhD (Tech) 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 acted yearly as a visiting scholar at 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 MSc and PhD degrees in electronics and information technology from the University of Turku, Turku, Finland, in 1999 and 2005, respectively. He received Adjunct professorship in embedded computing architectures in 2010. Currently he is working as a professor in University of Turku in the field of Embedded Systems and Internet of Things. At the moment his research is focused on Internet of Things, Fog computing, biomedical engineering, self-aware systems, approximation computing and health technology. In that context he has established and leading the Internet-of-Things for Healthcare (IoT4Health) research group. Liljeberg is the author of more than 270 peer-reviewed publications.

Hannu Tenhunen received the diplomas from the Helsinki University of Technology, Finland, 1982, and the PhD degree from Cornell University, Ithaca, NY, 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 department director. Since 1992, he has been a professor at the Royal Institute of Technology (KTH), Sweden, where he also served as a dean. He has more than 600 reviewed publications and 16 patents internationality. He is a member of the IEEE.

Turku Centre for Computer Science
TUCS Dissertations

1. **Marjo Lipponen**, On Primitive Solutions of the Post Correspondence Problem
2. **Timo Käkölä**, Dual Information Systems in Hyperknowledge Organizations
3. **Ville Leppänen**, Studies on the Realization of PRAM
4. **Cunsheng Ding**, Cryptographic Counter Generators
5. **Sami Viitanen**, Some New Global Optimization Algorithms
6. **Tapio Salakoski**, Representative Classification of Protein Structures
7. **Thomas Långbacka**, An Interactive Environment Supporting the Development of Formally Correct Programs
8. **Thomas Finne**, A Decision Support System for Improving Information Security
9. **Valeria Mihalache**, Cooperation, Communication, Control. Investigations on Grammar Systems.
10. **Marina Waldén**, Formal Reasoning About Distributed Algorithms
11. **Tero Laihonen**, Estimates on the Covering Radius When the Dual Distance is Known
12. **Lucian Ilie**, Decision Problems on Orders of Words
13. **Jukkapekka Hekanaho**, An Evolutionary Approach to Concept Learning
14. **Jouni Järvinen**, Knowledge Representation and Rough Sets
15. **Tomi Pasanen**, In-Place Algorithms for Sorting Problems
16. **Mika Johnsson**, Operational and Tactical Level Optimization in Printed Circuit Board Assembly
17. **Mats Aspnäs**, Multiprocessor Architecture and Programming: The Hathi-2 System
18. **Anna Mikhajlova**, Ensuring Correctness of Object and Component Systems
19. **Vesa Torvinen**, Construction and Evaluation of the Labour Game Method
20. **Jorma Boberg**, Cluster Analysis. A Mathematical Approach with Applications to Protein Structures
21. **Leonid Mikhajlov**, Software Reuse Mechanisms and Techniques: Safety Versus Flexibility
22. **Timo Kaukoranta**, Iterative and Hierarchical Methods for Codebook Generation in Vector Quantization
23. **Gábor Magyar**, On Solution Approaches for Some Industrially Motivated Combinatorial Optimization Problems
24. **Linas Laibinis**, Mechanised Formal Reasoning About Modular Programs
25. **Shuhua Liu**, Improving Executive Support in Strategic Scanning with Software Agent Systems
26. **Jaakko Järvi**, New Techniques in Generic Programming – C++ is more Intentional than Intended
27. **Jan-Christian Lehtinen**, Reproducing Kernel Splines in the Analysis of Medical Data
28. **Martin Büchi**, Safe Language Mechanisms for Modularization and Concurrency
29. **Elena Troubitsyna**, Stepwise Development of Dependable Systems
30. **Janne Näppi**, Computer-Assisted Diagnosis of Breast Calcifications
31. **Jianming Liang**, Dynamic Chest Images Analysis
32. **Tiberiu Seceleanu**, Systematic Design of Synchronous Digital Circuits
33. **Tero Aittokallio**, Characterization and Modelling of the Cardiorespiratory System in Sleep-Disordered Breathing
34. **Ivan Porres**, Modeling and Analyzing Software Behavior in UML
35. **Mauno Rönkkö**, Stepwise Development of Hybrid Systems
36. **Jouni Smed**, Production Planning in Printed Circuit Board Assembly
37. **Vesa Halava**, The Post Correspondence Problem for Market Morphisms
38. **Ion Petre**, Commutation Problems on Sets of Words and Formal Power Series
39. **Vladimir Kvassov**, Information Technology and the Productivity of Managerial Work
40. **Frank Tétard**, Managers, Fragmentation of Working Time, and Information Systems

86. **Sanna Ranto**, Identifying and Locating-Dominating Codes in Binary Hamming Spaces
87. **Tuomas Hakkarainen**, On the Computation of the Class Numbers of Real Abelian Fields
88. **Elena Czeizler**, Intricacies of Word Equations
89. **Marcus Alanen**, A Metamodeling Framework for Software Engineering
90. **Filip Ginter**, Towards Information Extraction in the Biomedical Domain: Methods and Resources
91. **Jarkko Paavola**, Signature Ensembles and Receiver Structures for Oversaturated Synchronous DS-CDMA Systems
92. **Arho Virkki**, The Human Respiratory System: Modelling, Analysis and Control
93. **Olli Luoma**, Efficient Methods for Storing and Querying XML Data with Relational Databases
94. **Dubravka Ilić**, Formal Reasoning about Dependability in Model-Driven Development
95. **Kim Solin**, Abstract Algebra of Program Refinement
96. **Tomi Westerlund**, Time Aware Modelling and Analysis of Systems-on-Chip
97. **Kalle Saari**, On the Frequency and Periodicity of Infinite Words
98. **Tomi Kärki**, Similarity Relations on Words: Relational Codes and Periods
99. **Markus M. Mäkelä**, Essays on Software Product Development: A Strategic Management Viewpoint
100. **Roope Vehkalahti**, Class Field Theoretic Methods in the Design of Lattice Signal Constellations
101. **Anne-Maria Ernvall-Hytönen**, On Short Exponential Sums Involving Fourier Coefficients of Holomorphic Cusp Forms
102. **Chang Li**, Parallelism and Complexity in Gene Assembly
103. **Tapio Pahikkala**, New Kernel Functions and Learning Methods for Text and Data Mining
104. **Denis Shestakov**, Search Interfaces on the Web: Querying and Characterizing
105. **Sampo Pyysalo**, A Dependency Parsing Approach to Biomedical Text Mining
106. **Anna Sell**, Mobile Digital Calendars in Knowledge Work
107. **Dorina Marghescu**, Evaluating Multidimensional Visualization Techniques in Data Mining Tasks
108. **Tero Säntti**, A Co-Processor Approach for Efficient Java Execution in Embedded Systems
109. **Kari Salonen**, Setup Optimization in High-Mix Surface Mount PCB Assembly
110. **Pontus Boström**, Formal Design and Verification of Systems Using Domain-Specific Languages
111. **Camilla J. Hollanti**, Order-Theoretic Mehtods for Space-Time Coding: Symmetric and Asymmetric Designs
112. **Heidi Himmanen**, On Transmission System Design for Wireless Broadcasting
113. **Sébastien Lafond**, Simulation of Embedded Systems for Energy Consumption Estimation
114. **Evgeni Tsivtsivadze**, Learning Preferences with Kernel-Based Methods
115. **Petri Salmela**, On Commutation and Conjugacy of Rational Languages and the Fixed Point Method
116. **Siamak Taati**, Conservation Laws in Cellular Automata
117. **Vladimir Rogojin**, Gene Assembly in Stichotrichous Ciliates: Elementary Operations, Parallelism and Computation
118. **Alexey Dudkov**, Chip and Signature Interleaving in DS CDMA Systems
119. **Janne Savela**, Role of Selected Spectral Attributes in the Perception of Synthetic Vowels
120. **Kristian Nybom**, Low-Density Parity-Check Codes for Wireless Datacast Networks
121. **Johanna Tuominen**, Formal Power Analysis of Systems-on-Chip
122. **Teijo Lehtonen**, On Fault Tolerance Methods for Networks-on-Chip
123. **Eeva Suvitie**, On Inner Products Involving Holomorphic Cusp Forms and Maass Forms
124. **Linda Mannila**, Teaching Mathematics and Programming – New Approaches with Empirical Evaluation
125. **Hanna Suominen**, Machine Learning and Clinical Text: Supporting Health Information Flow
126. **Tuomo Saarni**, Segmental Durations of Speech
127. **Johannes Eriksson**, Tool-Supported Invariant-Based Programming

128. **Tero Jokela**, Design and Analysis of Forward Error Control Coding and Signaling for Guaranteeing QoS in Wireless Broadcast Systems
129. **Ville Lukkarila**, On Undecidable Dynamical Properties of Reversible One-Dimensional Cellular Automata
130. **Qaisar Ahmad Malik**, Combining Model-Based Testing and Stepwise Formal Development
131. **Mikko-Jussi Laakso**, Promoting Programming Learning: Engagement, Automatic Assessment with Immediate Feedback in Visualizations
132. **Riikka Vuokko**, A Practice Perspective on Organizational Implementation of Information Technology
133. **Jeanette Heidenberg**, Towards Increased Productivity and Quality in Software Development Using Agile, Lean and Collaborative Approaches
134. **Yong Liu**, Solving the Puzzle of Mobile Learning Adoption
135. **Stina Ojala**, Towards an Integrative Information Society: Studies on Individuality in Speech and Sign
136. **Matteo Brunelli**, Some Advances in Mathematical Models for Preference Relations
137. **Ville Junnila**, On Identifying and Locating-Dominating Codes
138. **Andrzej Mizera**, Methods for Construction and Analysis of Computational Models in Systems Biology. Applications to the Modelling of the Heat Shock Response and the Self-Assembly of Intermediate Filaments.
139. **Csaba Ráduly-Baka**, Algorithmic Solutions for Combinatorial Problems in Resource Management of Manufacturing Environments
140. **Jari Kyngäs**, Solving Challenging Real-World Scheduling Problems
141. **Arho Suominen**, Notes on Emerging Technologies
142. **József Mezei**, A Quantitative View on Fuzzy Numbers
143. **Marta Olszewska**, On the Impact of Rigorous Approaches on the Quality of Development
144. **Antti Airola**, Kernel-Based Ranking: Methods for Learning and Performace Estimation
145. **Aleksi Saarela**, Word Equations and Related Topics: Independence, Decidability and Characterizations
146. **Lasse Bergroth**, Kahden merkkijonon pisimmän yhteisen alijonon ongelma ja sen ratkaiseminen
147. **Thomas Canhao Xu**, Hardware/Software Co-Design for Multicore Architectures
148. **Tuomas Mäkilä**, Software Development Process Modeling – Developers Perspective to Contemporary Modeling Techniques
149. **Shahrokh Nikou**, Opening the Black-Box of IT Artifacts: Looking into Mobile Service Characteristics and Individual Perception
150. **Alessandro Buoni**, Fraud Detection in the Banking Sector: A Multi-Agent Approach
151. **Mats Neovius**, Trustworthy Context Dependency in Ubiquitous Systems
152. **Fredrik Degerlund**, Scheduling of Guarded Command Based Models
153. **Amir-Mohammad Rahmani-Sane**, Exploration and Design of Power-Efficient Networked Many-Core Systems
154. **Ville Rantala**, On Dynamic Monitoring Methods for Networks-on-Chip
155. **Mikko Pelto**, On Identifying and Locating-Dominating Codes in the Infinite King Grid
156. **Anton Tarasyuk**, Formal Development and Quantitative Verification of Dependable Systems
157. **Muhammad Mohsin Saleemi**, Towards Combining Interactive Mobile TV and Smart Spaces: Architectures, Tools and Application Development
158. **Tommi J. M. Lehtinen**, Numbers and Languages
159. **Peter Sarlin**, Mapping Financial Stability
160. **Alexander Wei Yin**, On Energy Efficient Computing Platforms
161. **Mikołaj Olszewski**, Scaling Up Stepwise Feature Introduction to Construction of Large Software Systems
162. **Maryam Kamali**, Reusable Formal Architectures for Networked Systems
163. **Zhiyuan Yao**, Visual Customer Segmentation and Behavior Analysis – A SOM-Based Approach
164. **Timo Jolivet**, Combinatorics of Pisot Substitutions
165. **Rajeev Kumar Kanth**, Analysis and Life Cycle Assessment of Printed Antennas for Sustainable Wireless Systems
166. **Khalid Latif**, Design Space Exploration for MPSoC Architectures

167. **Bo Yang**, Towards Optimal Application Mapping for Energy-Efficient Many-Core Platforms
168. **Ali Hanzala Khan**, Consistency of UML Based Designs Using Ontology Reasoners
169. **Sonja Leskinen**, m-Equine: IS Support for the Horse Industry
170. **Fareed Ahmed Jokhio**, Video Transcoding in a Distributed Cloud Computing Environment
171. **Moazzam Fareed Niazi**, A Model-Based Development and Verification Framework for Distributed System-on-Chip Architecture
172. **Mari Huova**, Combinatorics on Words: New Aspects on Avoidability, Defect Effect, Equations and Palindromes
173. **Ville Timonen**, Scalable Algorithms for Height Field Illumination
174. **Henri Korvela**, Virtual Communities – A Virtual Treasure Trove for End-User Developers
175. **Kameswar Rao Vaddina**, Thermal-Aware Networked Many-Core Systems
176. **Janne Lahtiranta**, New and Emerging Challenges of the ICT-Mediated Health and Well-Being Services
177. **Irum Rauf**, Design and Validation of Stateful Composite RESTful Web Services
178. **Jari Björne**, Biomedical Event Extraction with Machine Learning
179. **Katri Haverinen**, Natural Language Processing Resources for Finnish: Corpus Development in the General and Clinical Domains
180. **Ville Salo**, Subshifts with Simple Cellular Automata
181. **Johan Ersfolk**, Scheduling Dynamic Dataflow Graphs
182. **Hongyan Liu**, On Advancing Business Intelligence in the Electricity Retail Market
183. **Adnan Ashraf**, Cost-Efficient Virtual Machine Management: Provisioning, Admission Control, and Consolidation
184. **Muhammad Nazrul Islam**, Design and Evaluation of Web Interface Signs to Improve Web Usability: A Semiotic Framework
185. **Johannes Tuikkala**, Algorithmic Techniques in Gene Expression Processing: From Imputation to Visualization
186. **Natalia Díaz Rodríguez**, Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces. A Case Study on Ambient Assisted Living
187. **Mikko Pänkäälä**, Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS
188. **Sami Hyrynsalmi**, Letters from the War of Ecosystems – An Analysis of Independent Software Vendors in Mobile Application Marketplaces
189. **Seppo Pulkkinen**, Efficient Optimization Algorithms for Nonlinear Data Analysis
190. **Sami Pyöttiälä**, Optimization and Measuring Techniques for Collect-and-Place Machines in Printed Circuit Board Industry
191. **Syed Mohammad Asad Hassan Jafri**, Virtual Runtime Application Partitions for Resource Management in Massively Parallel Architectures
192. **Toni Ernvall**, On Distributed Storage Codes
193. **Yuliya Prokhorova**, Rigorous Development of Safety-Critical Systems
194. **Olli Lahdenoja**, Local Binary Patterns in Focal-Plane Processing – Analysis and Applications
195. **Annika H. Holmbom**, Visual Analytics for Behavioral and Niche Market Segmentation
196. **Sergey Ostroumov**, Agent-Based Management System for Many-Core Platforms: Rigorous Design and Efficient Implementation
197. **Espen Suenson**, How Computer Programmers Work – Understanding Software Development in Practise
198. **Tuomas Poikela**, Readout Architectures for Hybrid Pixel Detector Readout Chips
199. **Bogdan Iancu**, Quantitative Refinement of Reaction-Based Biomodels
200. **Ilkka Törmä**, Structural and Computational Existence Results for Multidimensional Subshifts
201. **Sebastian Okser**, Scalable Feature Selection Applications for Genome-Wide Association Studies of Complex Diseases
202. **Fredrik Abbors**, Model-Based Testing of Software Systems: Functionality and Performance
203. **Inna Pereverzeva**, Formal Development of Resilient Distributed Systems
204. **Mikhail Barash**, Defining Contexts in Context-Free Grammars
205. **Sepinoud Azimi**, Computational Models for and from Biology: Simple Gene Assembly and Reaction Systems
206. **Petter Sandvik**, Formal Modelling for Digital Media Distribution

207. **Jongyun Moon**, Hydrogen Sensor Application of Anodic Titanium Oxide Nanostructures
208. **Simon Holmbacka**, Energy Aware Software for Many-Core Systems
209. **Charalampos Zinoviadis**, Hierarchy and Expansiveness in Two-Dimensional Subshifts of Finite Type
210. **Mika Murtojärvi**, Efficient Algorithms for Coastal Geographic Problems
211. **Sami Mäkelä**, Cohesion Metrics for Improving Software Quality
212. **Eyal Eshet**, Examining Human-Centered Design Practice in the Mobile Apps Era
213. **Jetro Vesti**, Rich Words and Balanced Words
214. **Jarkko Peltomäki**, Privileged Words and Sturmian Words
215. **Fahimeh Farahnakian**, Energy and Performance Management of Virtual Machines: Provisioning, Placement and Consolidation
216. **Diana-Elena Gratie**, Refinement of Biomodels Using Petri Nets
217. **Harri Merisaari**, Algorithmic Analysis Techniques for Molecular Imaging
218. **Stefan Grönroos**, Efficient and Low-Cost Software Defined Radio on Commodity Hardware
219. **Noora Nieminen**, Garbling Schemes and Applications
220. **Ville Taajamaa**, O-CDIO: Engineering Education Framework with Embedded Design Thinking Methods
221. **Johannes Holvitie**, Technical Debt in Software Development – Examining Premises and Overcoming Implementation for Efficient Management
222. **Tewodros Deneke**, Proactive Management of Video Transcoding Services
223. **Kashif Javed**, Model-Driven Development and Verification of Fault Tolerant Systems
224. **Pekka Naula**, Sparse Predictive Modeling – A Cost-Effective Perspective
225. **Antti Hakkala**, On Security and Privacy for Networked Information Society – Observations and Solutions for Security Engineering and Trust Building in Advanced Societal Processes
226. **Anne-Maarit Majanoja**, Selective Outsourcing in Global IT Services – Operational Level Challenges and Opportunities
227. **Samuel Rönnqvist**, Knowledge-Lean Text Mining
228. **Mohammad-Hashem Hahgbayan**, Energy-Efficient and Reliable Computing in Dark Silicon Era
229. **Charmi Panchal**, Qualitative Methods for Modeling Biochemical Systems and Datasets: The Logicome and the Reaction Systems Approaches
230. **Erkki Kaila**, Utilizing Educational Technology in Computer Science and Programming Courses: Theory and Practice
231. **Fredrik Robertsén**, The Lattice Boltzmann Method, a Petaflop and Beyond
232. **Jonne Pohjankukka**, Machine Learning Approaches for Natural Resource Data
233. **Paavo Nevalainen**, Geometric Data Understanding: Deriving Case-Specific Features
234. **Michal Szabados**, An Algebraic Approach to Nivat’s Conjecture
235. **Tuan Nguyen Gia**, Design for Energy-Efficient and Reliable Fog-Assisted Healthcare IoT Systems

Turku Centre *for* Computer Science

http://www.tucs.fi
tucs@abo.fi

University of Turku
Faculty of Science and Engineering

- Department of Future Technologies
- Department of Mathematics and Statistics

Turku School of Economics

- Institute of Information Systems Science

Åbo Akademi University
Faculty of Science and Engineering

- Computer Engineering
- Computer Science

Faculty of Social Sciences, Business and Economics

- Information Systems

ISBN 978-952-12-3773-7
ISSN 1239-1883

Tuan Nguyen Gia

Design for Energy-Efficient and Reliable Fog-Assisted Healthcare IoT Systems

Tuan Nguyen Gia

Design for Energy-Efficient and Reliable Fog-Assisted Healthcare IoT Systems

Tuan Nguyen Gia

Design for Energy-Efficient and Reliable Fog-Assisted Healthcare IoT Systems