Smart e-Health Gateway: Bringing Intelligence to Internet-of-Things Based Ubiquitous Healthcare Systems

Conference Paper · July 2015

DOI: 10.1109/CCNC.2015.7158084

Amir M. Rahmani
University of California, Irvine

Nanda Kumar Thanigaivelan
University of Turku

Tuan Nguyen gia
University of Turku

Jose Granados
University of Turku

Smart e-Health Gateway: Bringing Intelligence to Internet-of-Things Based Ubiquitous Healthcare Systems

Amir-Mohammad Rahmani[1], Nanda Kumar Thanigaivelan[1], Tuan Nguyen Gia[1], Jose Granados[1], Behailu Negash[1], Pasi Liljeberg[1], and Hannu Tenhunen[1,2]
[1]Department of Information Technology, University of Turku, Turku, Finland
[2]School of ICT, Royal Institute of Technology (KTH), Stockholm, Sweden
Email: {amirah, nakuth, tunggi, behneg, josgra, pakrli}@utu.fi, hannu@kth.se

***Abstract*—There have been significant advances in the field of Internet of Things (IoT) recently. At the same time there exists an ever-growing demand for ubiquitous healthcare systems to improve human health and well-being. In most of IoT-based patient monitoring systems, especially at smart homes or hospitals, there exists a bridging point (i.e., gateway) between a sensor network and the Internet which 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 to offer several higher-level services such as local storage, real-time local data processing, embedded data mining, etc., proposing thus a Smart e-Health Gateway. By taking responsibility for handling some burdens of the sensor network and a remote healthcare center, a Smart e-Health Gateway can cope with many challenges in ubiquitous healthcare systems such as energy efficiency, scalability, and reliability issues. A successful implementation of Smart e-Health Gateways enables massive deployment of ubiquitous health monitoring systems especially in clinical environments. We also present a case study of a Smart e-Health Gateway called UT-GATE where some of the discussed higher-level features have been implemented. Our proof-of-concept design demonstrates an IoT-based health monitoring system with enhanced overall system energy efficiency, performance, interoperability, security, and reliability.**

Keywords*—*Internet of Things, Healthcare, Smart Hospital, Home care, Smart Gateway, Sensor Network

I. Introduction

Internet of Things (IoT) is a promising paradigm to integrate several technologies and communication solutions. As defined by European Commission Information Society, the Internet of Things is a manageable set of convergent developments in sensing, identification, communication, networking, and informatics devices and systems [1]. Wireless Sensor Network (WSN), as a fundamental enabling technology of IoT, integrates a number of spatially distributed autonomous sensors into a network and cooperatively pass their data through wireless communication. This network can be connected to a higher level system via a network gateway. WSN is built of sensor/actuator nodes, from a few to several hundreds or even thousands nodes, where each node is typically portable, lightweight, low-cost, and simply deployable. WSN is extended to a network commonly called Ubiquitous Sensor Networks (USN) when it is integrated into a system of IoT [2].

USN technologies can provide significant possibilities for the field of healthcare. Intelligent wireless networks can assist patients and their caregivers by providing continuous medical monitoring, memory enhancement, control of appliances, medical data access, and emergency communication. The IoT is in the revolutionary road and it will remodel the healthcare sector on the way in terms of social benefits and penetration as well as economics. Enabled by ubiquitous computing, all the healthcare system entities (individuals, appliances, medicine) can be monitored and managed continuously [3], [4]. The IoT's connectivity provide a way to monitor, store and utilize health and wellbeing related data on a 24/7 basis [3] and enable the IoT related data and services to be ubiquitous and customized for personal needs [5], [4].

It has been predicted that in the following decades, the way healthcare is currently provided will be transformed from hospital-centered, first to hospital-home-balanced in 2020th, and then ultimately to home-centered in 2030th [6]. This essential transformation necessitates the fact that the convergence and overlap of the IoT architectures and technologies for smart spaces and healthcare domains should be more actively considered. In a smart home or hospital, where the mobility and location of patients are confined to the hospital facilities or the building, gateways can play a key role. Gateways generally act as a hub between body/personal/local area networks (BAN/PAN/LAN) and a remote health center. The stationary nature of such gateways empowers them with the luxury of being non-resource constrained in terms of processing power, power consumption and communication bandwidth. On the one hand, this valuable luxury can be exploited by outsourcing some burden of resource-constrained sensors/actuators to be performed on the gateways, and on the other hand, it can be used to add some levels of intelligence to its basic functionalities and extend its role to an intelligent embedded server.

In this paper, we present the concept of Smart e-Health Gateway capable of enhancing IoT architectures used for healthcare applications in terms of energy-efficiency, performance, reliability, interoperability, just to mention a few. In addition, we elaborate a Smart e-Health Gateway's features and its offered services from the viewpoint of cost-benefit analysis. In order to provide a proof of concept implementation, we also demonstrate our prototype of a Smart e-Health Gateway and discuss the design and implementation of our demonstrator.

The rest of the paper is organized as follows: In Section II, the related work and the motivation of this paper are discussed. Section III describes the architecture of an e-health platform using Smart e-Health Gateways. The concept and features of a Smart e-Health Gateway are presented in more details in Section IV. Demonstration of our Smart e-Health Gateways and experimental results are provided and discussed in Section V. Finally, Section VI concludes the paper.

978-1-4799-6390-4/15/$31.00 ©2015 IEEE

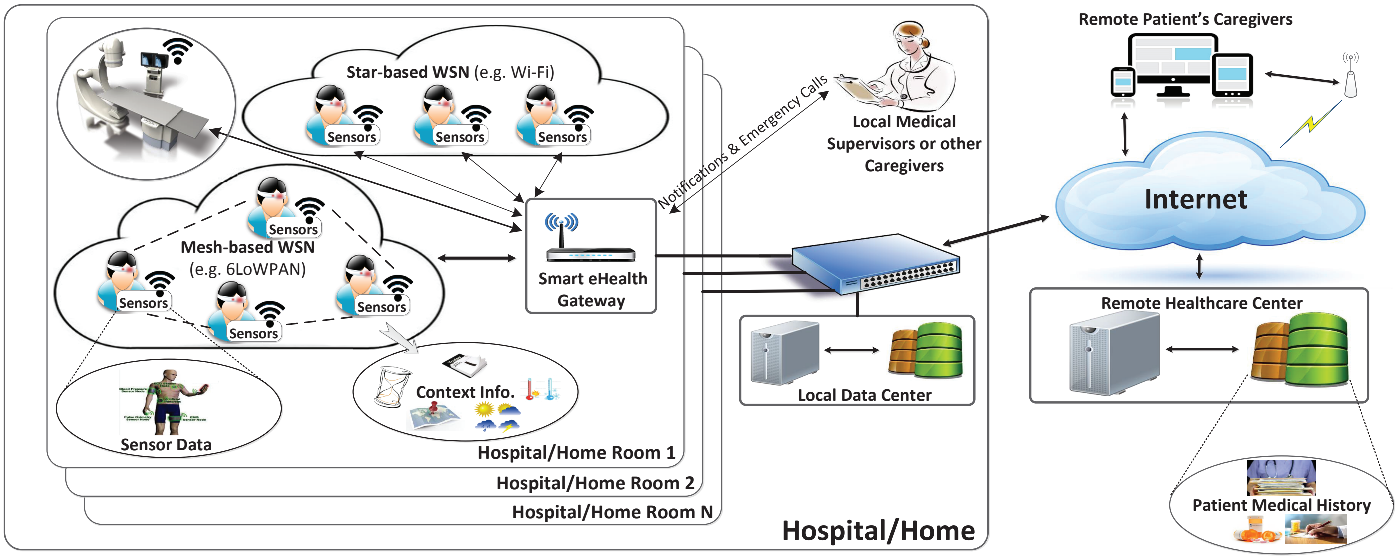

Fig. 1: IoT-based health monitoring architecture

II. Related Work and Motivation

There have been many efforts in designing gateways for one or several specific applications and architectural layers. For example, the works presented in [7], [8] propose gateways to transparently connect networks with different protocols such as ZigBee, Bluetooth, and Ethernet. However, these gateways have limited flexibility as they cannot be customized for different applications. In a different category of related work, Mueller *et al.* [9] present a gateway called SwissGate which handles and optimize the operation of the sensor network. They specifically apply SwissGate on home automation applications such as measuring heating, ventilation, and air conditioning control (HVAC) parameters. Bimschas *et al.* [10] aim to provide some levels of intelligence to gateways by enabling them to execute application code. They propose a middleware for the gateway to offer three possible services: protocol conversion, request caching, and intelligent caching and discovery.

Jong-Wan *et al.* [11] present a sensor network system comprising of a main server and several sensing servers acting as gateway and connecting with different sensor networks. Using network-dependent sensing servers instead of gateways results in high implementation and hardware cost as well as poor scalability, making this architecture inefficient for many IoT applications. In a work presented in [12], a plug-configurable-play service-oriented generic gateway is proposed in order to provide simple and rapid employment of various external sensor network applications. The gateway offers a proper level of interoperability by facilitating the bridging between heterogeneous sensor networks and homogenous resources. The middleware presented in their work lacks intelligence and runs on PC, limiting its advantages for many IoT applications. In a similar attempt, Guoqiang *et al.* [13] propose a smart general purpose gateway which provides i) pluggable architecture enabling the communication among different communication protocols, ii) unified external interfaces fitting for flexible software development, and iii) flexible protocol to translate different sensor data.

In order to save energy and reduce the cost of smart home, Bian *et al.* [14] present a new type of intelligent home control system, using Android Phone as a temporary home gateway instead of the default home gateway. Their aim is to automatically shut down the unused devices by predicting user behavior. Finally, in the work presented in [15], authors propose 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.

Satyanarayanan *et al.* [16] propose that mobile devices can use modern algorithms such as facial recognition and language translation to augment human cognition. However, shortcomings in processing power and long WAN latency are unacceptable. The authors then propose cloudlets which consist of a computer wireless connectivity in the vicinity of the mobile device. A base virtual machine (VM) is installed on the cloudlet from where VM instances are launched and configured using an overlay script from the mobile device with desired application. The VMs are designed to migrate from cloudlet to cloudlet to allow ubiquity. That way, user interaction is faster due to lower latencies an higher processing power. The authors face challenges regarding long launching times. Comparing with the smart gateway with the cloudlet, the gateway performs data mining algorithms on patient data in a similar way, however, there is no need for VMs since the gateway performs applications related to health only, which are the same for all patients. In that sense, there would not be any launching period, and the gateway is always available to start receiving and processing data.

Although such efforts have been greatly expanded in recent years, there are still limited improvements towards realizing smart gateways specifically for the healthcare domain. Most of the presented efforts focus on general purpose gateway design which minimizes the provided level of intelligence due to lack of information from the application domain. Some of these efforts limit their level of intelligence only to interpretability, plug-and-play ability, or reconfigurability. Some others just focus on specific domains such as smart home.

It should be noted that designing an efficient IoT-based healthcare systems is a difficult task as it faces the following challenges. First, the sensor networking mechanism must be resource-efficient and customized for e-Health applications as medical sensor nodes, especially implanted ones, have much lower processing power, memory, transmission speed, and energy supply than sensors in other sensor networks. Second, in spite of common sensor networks where interval-based data transmission is used (e.g., temperature and humidity monitoring), e-Health applications often need to manage streaming-

978-1-4799-6390-4/15/$31.00 ©2015 IEEE

based transmissions where realtime requirements need to be considered and consequently a considerable energy is dissipated for the transmission process (e.g., Electrocardiogram (ECG) signal transmission with 4kbps required bandwidth per channel). Third, in multi-patient applications for instance 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 parallel nature of the workloads. However, the existing general-purpose gateways have not designed for such scenarios. Fourth, reliability in e-Health application is of utmost importance and even short system unavailability cannot be tolerated. Thus, as we will discuss in the following sections, 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.

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 collecting health and context information from those networks and providing different services accordingly.

III. System Architecture

The architecture of an IoT-based health monitoring system which can be used in smart hospitals or home is shown in Figure 1. In such 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 Sensor Network:** Enabled by the ubiquitous identification, sensing, and communication capacity, biomedical and context signals are captured from the body/room used for treatment and diagnosis of medical states. The signal is then transmitted to the gateway via wireless or wired communication protocols such as Serial, SPI, Bluetooth, Wi-Fi or IEEE 802.15.4.
2) **Smart e-Health Gateway:** The gateway, which supports different communication protocols, acts as the 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:** The back-end of the system consists of the two remaining components, a local switch and a cloud computing platform that includes broadcasting, data warehouse and Big Data analytic servers, and finally Web clients as a graphical user interface for final visualization and apprehension. The collected health and context information represents a vital source of big data for the statistical and epidemiological medical research (e.g., detecting approaching epidemic diseases).

As can be noticed from the figure, the strategic location of the smart gateway can be exploited by offering many higher level services to enhance the system characteristics in many different aspects. The following section discusses the different features of a Smart e-Health Gateway.

IV. Smart e-Health Gateway

As mentioned before, the main requirement of a gateway is to support various wireless protocols and inter-device communication. In this section, we extend its role to support several features such as acting also as repository (i.e. local database) to temporarily store sensors' and users' information, and bringing intelligence by enhancing with data fusion, aggregation, and interpretation techniques, essential to provide preliminary local processing of sensors' data, becoming thus a Smart e-Health Gateway. In the following, we discuss these features in more detail. It is worth noting that the non-resource constrained local storage and processing characteristic of Smart e-Health Gateway can open the field of remote health monitoring to various research areas including both hardware and software perspectives by, for example, introducing innovative wearable technologies and wireless networking as well as novel data mining and machine learning algorithms.

A. Local Data Processing

1) Data Compression: The concept of data compression at sensor nodes has been researched a lot. For example, Touati *et al.* [17] present a lossy data compression technique at sensor nodes in wireless body area network as lossless compression demands high amount of computation and bandwidth. Many ECG data compression techniques for sensor node were extensively proposed in [18], [19], among them the techniques presented by Zhang *et al.* [19] and Rao *et al.* [19] show the best system characteristics in terms of energy consumption and data distortion. However, efficiency of BAN/PAN can considerably increase if data compression is extended at the gateway. Due to limitation of network bandwidth, and susceptibility of essential data in healthcare, data compression is considered as a competent feature for Smart e-Health Gateways as one of the services provided by the Local Data Processing module (Figure 2). When data is compressed, the communication cost is reduced significantly and the capacity of the communication channel increases. Furthermore, as the Smart e-Health Gateway is not a resource-constrained device, even lossless compression techniques can be efficiently executed on the gateway, therefore, strict requirements of healthcare data can be properly fulfilled. It should be noted that depending on e-health application requirements, compression can be enforced either on the node/gateway or both.

2) Data Fusion: The most recent definition described by Boström *et al.*, based on strengths and weaknesses of previous works, is "Information fusion is the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making" [20]. Data fusion can be implemented in both centralized and distributed systems [21]. In a centralized system, all raw data collected from sensors is sent to a central node while in a distributed system different data fusion methods are implemented on distributed sensors. Distributed and localized data fusion is more preferable [21]. However, data fusion on distributed sensors may give unsatisfactory outcomes and result in waste of resources, and misleading assessments in case of careless design [21]. Therefore, deploying data fusion in the gateway can overcome problems of imprudence in system designing. Furthermore, there is a limited battery capacity available in sensor nodes, therefore misusing data fusion in nodes may reduce their battery lifetime. Besides, data fusion in node is processed

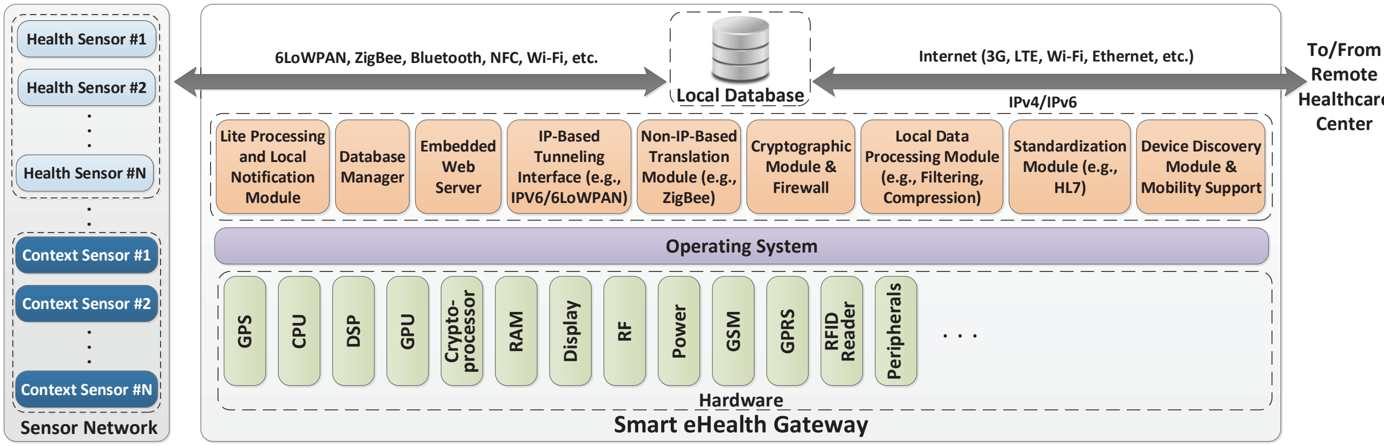

Fig. 2: Smart e-Health Gateway architecture

and localized by utilizing data provided by neighbor nodes, however, data might be fused by intermediate nodes before reaching to the final node. This problem can be solved when applying data fusion in the Smart e-Health Gateway due to direct availability of raw data collected from nodes. Data fusion in the Smart e-Health Gateway can be implemented in the Local Processing Module shown in Figure 2. Applying data fusion in gateways provides several advantages such as reduced data ambiguity, extended coverage in space and time, robustness and reliability, and increased quality of data. After data is fused, only final results are transmitted through the network so the network bandwidth can be also efficiently utilized and the system can be also more energy efficient.

3) Data Filtering: Physiological systems of the human body such as cardiovascular, nervous and muscular systems generate bio-signals that are the primary source of information for assessing the patient health status. Often the acquired signals have small amplitudes making them susceptible to interference or artifacts that are superimposed on the signal. Any kind of disturbance that affects the signal and distorts the information carried by it, is considered as noise and must be minimized. The sources of noise may be external such as the 60Hz electric line frequency or electromagnetic interference (EMI) or internal such as patient motion. For example, in the case of measuring ECG or EEG signals, the signal energy might arise from other sources different than the heart or brain such as eye blinking, muscle contraction, arterial blood pressure and patient respiration. Moreover, for biomedical applications the sole acquisition of the signal is not sufficient since the relevant information might be "buried" in it or may not be visible from the raw signal. Different signals have different bandwidths and depending on the required application, different frequency components of the signals must be extracted while others rejected. It is therefore required to apply analog or digital processing for feature extraction.

A Smart e-Health Gateway can interface analog and digital sensors directly and providing analog processing to increase the amplitude of the acquired signal as well as converting it to the digital domain. For the case of digital sensors, the gateway provides means to receive the generated signal via serial protocols such as USART, SPI and USB. The received signal is then processed by the Local Data Processing Module. Filtering of the signal can be done in the gateway by implementing the required finite impulse response (FIR) and infinite impulse response (IIR) filters for the removal of undesired low and high frequencies. Furthermore the gateway implements different algorithms for biomedical signal processing such as averaging, standard deviation, RMS values, frequency-domain analysis, Fast Fourier Transform (FFT), etc., to extract useful information. Therefore, the gateway can be designed to ensure the integrity of the information contained in the signal, as well as help making the bio-signal detectable for monitoring and diagnostics purposes. This will also enhance the system reliability by providing local feedbacks even during the times of unavailability of internet.

B. Local Storage

Data availability is highly critical in e-health applications. The availability factor can be easily affected during network unavailability which might negate the advantages of the entire application. Relying on nodes for availability is out-of-question and the only possibility is to take advantage of the position of the gateway. The gateway requires an operating system to performs tasks such as packet forwarding from node to remote server, therefore it can act as an intermediate storage for certain period of time and also during unforeseen circumstances with provision of additional storage.

The local repository is used to hold a copy of received data from the nodes temporarily along with the servers. These locally copied data will be deleted on regular basis which can be scaled depending on the availability of the installed memory either manually or automatically. During the unavailability of internet, these local repositories can be extended to hold until least availability of the provided memory. With the implementation of a lite web server or by a program, we can get the stored information from the local repository in different formats such as Health Level-7 (HL7) [22] whenever required. The local storage provides reliability and assurance to the system operation even during the times of network unavailability. The local repository is handled by the database manager as demonstrated in the Figure 2. It offers security features such as encrypting stored data to protect them from unauthorized access by any other means than the programs provided in the gateway.

C. Embedded Data Mining

The Smart e-Health Gateway has lite processing and local notification module, shown in Figure 2, at its core to realize the smartness by applying different machine learning and data mining techniques on the local data storage. The smart gateway can be considered as an intermediate computing layer between the sensor nodes and the cloud, also known as Fog computing layer in [23]. It allows realtime computing to provide efficient feedbacks and notifications based on rules and coefficients acquired locally and from the remote cloud server. This minimizes the latency of certain event notifications to nodes or any subscribers, gives better location and context information as it is close to the nodes, and facilitate handling of the mobility requirements of the nodes.

One simple case of the local processing is the validation of the incoming raw data from the nodes against a set of rules. If the data does not conform, the appropriate configured mechanism will be invoked to notify the (specified) user. The decision making rule can be enforced for specific node or for the entire network of nodes.

Another important part in the lite processing and local notification module is the local notification service. In healthcare applications, the delivery of critical information is highly essential. In most cases, the notification service can have realtime requirements that could be set depending on the type of patient and criticality of the notification. The gateway enables us to meet such requirements as compared to the time it could take using the cloud. Any malfunctions in the notification process may results in serious impact or loss to person or medical procedure. The results of the decision has to be notified to the user either by the remote server or by the gateway. Remote server will raise notification over any medium. Unlike remote server, the resources at the gateway are limited but the important aspect is that the gateway's notification will run independent of remote server. Even during the unavailability of remote server, the notification module running on the gateway can ensure the delivery of critical information to appropriate users and thus enhancing the efficiency and reliability of the system operations.

D. Security

Security is one of the important aspects of the system. Depending on the implementation of security mechanisms, we can predict the probability of unsecured ratio of the system. Most of the cases, the data received from sensor nodes are transmitted as plain text which can be accessed easily by third party. However, some cases where information is classified as personal, it should not be accessible by third party. Leaking information might have some serious impacts on the person involved specially in e-health applications where confidentiality is necessary.

As mentioned in Section 4.2, the gateway requires an operating system or at least a core (kernel) to carry out its operations. Currently, Linux is highly favored compared to freeBSD and other variants for using as core for router particularly in IoT due to its efficiency and customizable factors. The Linux kernel comes with basic setting called IPtables [24] and freeBSD comes with IPFW. These are simple security mechanisms yet offering robust security depending on the configuration (cryptographic module and firewall in Figure 2. With IPtables/IPFW, only very few ports will be opened for traffic and rest are closed. For instance, dedicated ports for gateway-to-node over UDP and database port over TCP/UDP are opened to carry out respective tasks. The gateway also act as web server during network unavailability or whenever needed. These web server communications are transferred over secure HTTPS and accessor identification to maintain the confidentiality and integrity of the system operation.

Apart from the above, the gateway can accommodate and perform several security related tasks because of its processing power and memory availability. In future, if we need any specific security mechanism, it can be achieved without the requirement of additional hardware and also, we can shift or share the load from the node to gateway which allows the nodes to perform complex tasks.

E. Interoperability and Reconfigurability

The other feature of the Smart e-Health Gateway is its ability to interoperate with a wide range of devices and the ability to reconfigure it to support more protocols and standards. One of the biggest challenges for the success of the IoT is the fact that different device manufacturers use different protocols and communication mechanism than the other and there is no defined standard as in the internet we currently have [4]. Interoperability in the general sense can mean different things in the high-level architecture shown in the Figure 1. It can be at the network level, sensors using different network technology can work together, or sensors using different standards can understand each others message.

As shown in Figure 2, the health sensors and the context sensors are connected with the Smart e-Health Gateway using either wireless network (e.g. ZigBee, 6LoWPAN, Bluetooth, Wi-Fi, etc.) or wired connections and they use different protocols to communicate with the gateway. They are also based on different operating systems, some might not even have one. The true smartness of the gateway comes here to easily integrate these heterogeneous networking technologies, protocols and standards thereby enabling them to interoperate at different levels.

1) Device Interoperability: The sensors which are made by different vendors support variety of network technologies and they can be arranged in different topologies, for instance, one category using ZigBee and the other using Bluetooth. These distinct devices can be made to communicate to each other using variety of ways by implementing techniques serving as adapters for each technology to convert it to the other. One typical scenario can be the use of sockets to handle the communication where it is possible. Sockets can also enable us to create a platform independent network. For instance, WebSockets can be one solution in specific domains. One of the advantages of such a smart gateway is also to make the device discovery process easier and also the process of joining in to the network with proper authentication.

2) Protocol Interoperability: The various networking technologies come with a range of protocols, both at the network/transport layer and application layer. Starting from the general category of IP based and non-IP based networks, we face the challenge all the way down in each layer. Tunneling protocol for instance can be used to encapsulate messages in the network layer. One common example of tunneling is the case when a 6LoWPAN edge router needs to tunnel between 6LoWPAN and IPv4/IPv6 protocols [25]. Protocol translators can also be used to buffer an incoming message and forward in another format and there are implementation of this technique in the cloud these days. A popular example is the ZigBee gateway which translates between the ZigBee and IP networks [26]. These functions are realized using the two modules, IP based tunneling interface and non-IP based translation module, shown in Figure 2.

Another opportunity in this realm is presented by the WebSocket protocol introduced in HTML5 which defines a duplex communication standard for web browsers and servers [27]. An embedded WebSocket server, Embedded web server module in Figure 2, enables the gateway to send messages to the client at any time after the WebSockets connection has been established. This open technology is supported by the most popular Web browsers and frameworks and has the ability to serve as layer for a higher level web-based communication protocol that in our case is customized for the managing and transmission of remote medical and context sensors data and health applications related information. This allows us to have heterogenous networks to interoperate regardless of the

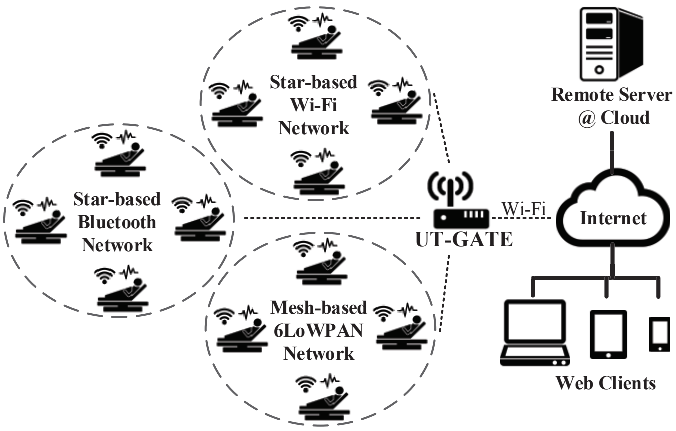

Fig. 3: Architecture Overview

topology used or the underlying protocol.

3) Data Interoperability: As medical data that is exchanged between different hospitals using different applications should be formatted according to Electronic Health Record (EHR) standards, such as HL7 [22], re-formatting data in different formats to the same standard is done by the smart gateway, in the standardization module shown in Figure 2. The discussion of data interoperability can be extended beyond the Electronic Health Record formatting standards to other technical standards such as data rate and precision correction as required. Since the formatting is done at the gateway level, the sensor nodes will 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 raw data to the gateway, where it is formatted to standards.

4) Reconfigurability: In practice, it is difficult to implement all the possible combination of protocols and platforms on the gateway and wait for any incoming device. However, the smart gateway should be reconfigured to enable or disable protocols to be used, which types of features to enable/disable, standards applicable under the given setup and other relevant settings. Reconfigurability of the smart gateway is one of the key features which makes the remaining features realizable from performance, security and scalability perspectives.

5) Device Discovery and Mobility Support: Device discovery has been mentioned in device interoperability from the point of view of the sensors becoming active after a certain timeout and trying to join the network. In relation to the mobility of a patient from one place to the other, device discovery plays the major role in discovering the destination gateway and handing over all the necessary information. As shown in Figure 2, device discovery and mobility support module determines the destination gateway from the context sensors and other relevant information, start negotiation to handover and complete the smooth transition to the next one. Data consistency between the previous and the new gateways is maintained using the shared cloud server which is connected to all the gateways as shown in Figure 1.

V. Demonstration and Evaluation

In this section, we first present the architecture of our system demonstration and evaluate the system characteristics including higher-level functions developed on our gateway. In our architecture, the system implementation is divided into three major phases: node implementation, smart gateway implementation, and back-end and user interface implementation. Our Smart e-Health Gateway, called UT-GATE, collaborates with sensor nodes, remote server, and clients. At the end, the platform based on the UT-GATE provide medical data collected from sensors to end-users through web browsers to their devices. The implemented system architecture is shown in Figure 3. To demonstrate the device and protocol level interoperability of UT-GATE, we have implemented different network topologies, mesh and star topologies, using several wireless sensor technologies, such as 6LoWPAN, Wi-Fi and Bluetooth, so that each sensor in each subnet utilize different platform but work 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 Figure 4, UT-GATE is constructed from combination of a Pandaboard [28] and Texas Instruments (TI) SmartRF06 board integrated with CC2538 module [29] and MOD-ENC28J60 Ethernet Module [30]. The Pandaboard is low-power, low-cost single-board computer development platform based on TI OMAP4430 system-on-chip (SoC) following OMAP architecture and fabricated using 45nm technology for providing high-performance multimedia applications such as streaming video, image and graphic processing, and 2D/3D mobile gaming [28]. OMAP4430 processor is composed of Cortex A9 microprocessor unit (MPU) subsystem including dual-core ARM Cortex-A9 cores with symmetric multiprocessing at up to 1.2GHz each. The Pandaboard device can support different operating systems such as Windows CE, WinMobile, Symbian, Linux, Palm and integrate on-chip memory, external memory interfaces, memory management, system and connecting peripherals. In our implementation, 8GB 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, the Pandaboard 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) in Pandaboard 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 activities in the Arduino Due are executed on FreeRTOS [31], an open source realtime 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 [32] 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 [33], 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 which will 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 [29].

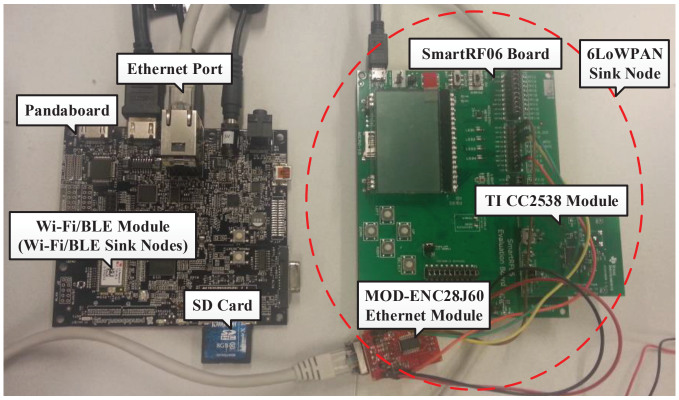

Fig. 4: UT-GATE: Our demonstrator of a Smart e-Health Gateway

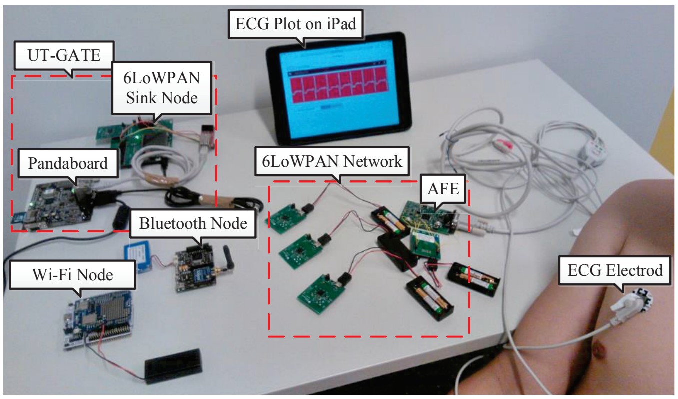

Fig. 5: IoT-based health monitoring system demonstration

As shown in Figure 4, 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 Real-time Programming Language (RPL) implementation in Contiki OS [34] which is an open source operating system focusing on low-power IoT devices.

The sensor nodes receive Electrocardiogram (ECG), Electromyogram (EMG) and Electroencephalogram (EEG) digital signals through 2 SPI connections from AFE devices (i.e., TI ADS1292 [35] and TI ADS1298 [36]). 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 master role while the ADS129x module act as a slave.

In our IoT-based health monitoring system, the remote server is responsible for handling client requests by providing the requested 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 Figure 5, we have established an IoT-based health monitoring system including the 6LoWPAN, Wi-Fi and Bluetooth subnetworks in our lab 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.

A. *Compression at UT-GATE*

UT-GATE can perform data compression to save network bandwidth and improve certain levels of efficiency. However, not all data compression algorithms and methods can be processed in the Smart e-Health Gateway due to different levels of accuracy and latency. Lossless and lossy algorithms can be applied for data compression; nevertheless, lossless algorithm seems to be an appropriate choice for our e-health application comparing with lossy algorithms. The computation needed for lossless algorithms is not an important issue for UT-GATE due to the availability of high-performance processing elements in the gateway. Power consumption is not also a problem for the UT-GATE as it is plugged in to the power supply.

Realtime processing is an essential issue in e-health applications. Thus, latency during data compression process must be considered. Due to IEEE 1073 real time requirements [37], maximum latency for ECG, EMG, EEG signals must be less than 500ms. However, after some experiments from applying different data compression algorithms including lossless and lossy, we decided to implement lossless data compression LZ4 and LZO [38] because these lossless compression algorithms are the fastest comparing to existing lossless data compression algorithms. We applied these compressions on UT-GATE with two different methods.

In the first method, we compress data collected from nodes at UT-GATE and then UT-GATE sends these compressed data to the remote server directly. In the second method, we compress data collected form nodes and saved compressed data into files and finally send the files to the server. Although the second method is more complex and time consumption than the first one, it gives more opportunity for reducing the lost package rate in transmission and increasing level of encryption and data management. With files sent over the network, the server can check the lost packages easily through handshaking in communication. If the package is lost in transmission, the server can request the file from UT-GATE. UT-GATE can easily resend the lost package as the package is temporarily stored in the gateway. A package is only deleted when the server acknowledges the successful transmission. Handshaking in communication may impact on network bandwidth and latency but it guarantees that all data are received at the server, which is extremely vital in healthcare applications as data loss has direct negative impact on the treatment procedure. Results of the two methods, LZO and LZ4, are shown in Table I and Table II, respectively.

Tables I and Table II show compression and decompression times and compressed data size with number of nodes varying from 1 to 50. Data collected from analog front-ends with 8 analog channels is accumulated until 30 samples obtained and then compressed. The main reason for accumulating data is to improve efficiency of the data compression algorithm. When the volume of data is larger, the data compression algorithm can perform more efficient. Accumulation process increases the latency, but in our gateway, accumulated data still fulfills real-time requirements of IEEE 1073. Each node with 8 channels needs 2ms for gathering 1 sample (18 bytes), so accumulating 30 samples costs about 60ms. This additional latency in compression and de-compression times, shown in Table I and Table II, is still less than 50% of the time required to meet the realtime deadline. The number of accumulated samples can be increased, but we reserved surplus time for retransmitting lost packages, therefore accumulating 30 samples is an appropriate choice in our demonstration to improve reliability at the same time. The compressed data size in the tables shows the achieved minimum data size with the maximum efficiency in both compression algorithms.

TABLE I: Compression results for the LZO algorithm

Number of nodes	Number of analog channels	Data size (1 sample)(Bytes)	Data size (30 samples)(Bytes)	Compressed data Size (Bytes)	Compression time (ms)	Decompression time (ms)
1	8	28	840	39	18	23
2	8	56	1680	41	20	19
5	8	140	4200	54	23	27
10	8	280	8400	66	30	37
50	8	1400	42000	193	30	40

TABLE II: Compression results for the LZ4 algorithm

Number of nodes	Number of analog channels	Data size (1 sample)(Bytes)	Data size (30 samples)(Bytes)	Compressed data Size (Bytes)	Compression time (ms)	Decompression time (ms)
1	8	28	840	22	60	40
2	8	56	1680	35	60	50
5	8	140	4200	48	50	60
10	8	280	8400	60	40	40
50	8	1400	42000	180	40	40

B. Filtering at UT-GATE

Once data is received from the sensor node, UT-GATE proceeds to filter the acquired raw signal. A set of filters for cleaning an ECG signal is implemented for evaluation purposes using the Python programming language together with the open-source SciPy and NumPy extensions. The algorithm generates the necessary order and coefficients for the low-pass finite impulse response (FIR) filter at 40Hz using a Kaiser window to limit ripple and width of the transition region. The resulting filter order is around 220 for a 80dB attenuation in the stop-band. To correct the DC bias known as baseline wandering, a first order infinite impulse response (IIR) filter is implemented with coefficient 0.992. A second set of filters was implemented using the same Python library functions for EMG signal cleaning. In this case the low-pass filter cutoff frequency was defined as 200Hz and the high-pass FIR filter with cut-off was specified at 25Hz. An additional notch FIR filter for removing power-line interference at 50Hz was added to the EMG test.

C. Local Storage, Notification, and Security at UT-GATE

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

TABLE III: XML Status code and description

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

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 III. 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 very basic 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.

D. 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 1.1KB/s. In our LAN setup, it takes 32ms 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.

VI. Conclusions

In this paper, we presented the concept of Smart e-Health Gateway. The gateway serves as a bridge for medical sensors and home/hospital building automation appliances to IP based networks and cloud computing platforms. We showed that by exploiting the unique strategic position of gateways in IoT architectures, a Smart e-Health Gateway can tackle many challenges in ubiquitous healthcare systems such as energy efficiency, scalability, interoperability, and reliability issues. We presented a proof of concept implementation of an IoT-based remote health monitoring system which includes our demo of a Smart e-Health Gateway called UT-GATE. UT-GATE provides efficient local services for health monitoring applications such as local repository, compression, signal processing, data standardization, WebSocket server, protocol translation and tunneling, firewall, and data mining and notification. The system demonstrator includes all the dataflow process from the bioelectrical signal acquisition at sensor nodes to the remote cloud-based healthcare center and web clients.

Acknowledgment

The authors wish to acknowledge the financial support by the Academy of Finland during the course of this project.

References

[1] European Commission Information Society. Internet of Things Strategic Research Roadmap 2009. *http://www.internet-of-things-research.eu/ [accessed 2014-08-04].*

[2] Ubiquitous Sensor Networks (USN). *ITU-T Technology Watch Briefing Report Series*, (4), 2008.

[3] A. Dohr et al. The internet of things for ambient assisted living. In *Proceedings of the International Conference on Information Technology: New Generations*, pages 804–809, 2010.

[4] M.D. Miorandi. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497 – 1516, 2012.

[5] Mari Carmen Domingo. An overview of the internet of things for people with disabilities. *Journal of Network and Computer Applications*, 35(2):584 – 596, 2012.

[6] C.E. Koop et al. Future delivery of health care: Cybercare. *IEEE Engineering in Medicine and Biology Magazine*, 27(6):29 – 38, 2008.

[7] K.A. Emara et al. A gateway-based framework for transparent interconnection between WSN and IP network. In *Proceedings of the EUROCON*, pages 1775–1780, 2009.

[8] Q. Zhu et al. IoT Gateway: BridgingWireless Sensor Networks into Internet of Things. In *Proceedings of the International Conference on Embedded and Ubiquitous Computing*, pages 347–352, 2010.

[9] R. Mueller et al. Demo: A Generic Platform for Sensor Network Applications. In *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–3, 2007.

[10] D. Bimschas et al. 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*, pages 8–14, 2010.

[11] J.-W. Yoon. Sensor Network Middleware for Distributed and Heterogeneous Environments. In *Proceedings of the International Conference on New Trends in Information and Service Science*, pages 979–982, 2009.

[12] L. Wu et al. Plug-configure-play service-oriented gateway - for fast and easy sensor network application development. In *Proceedings of the International Conference on Sensor Networks*, pages 53–58, 2013.

[13] S. Guoqiang et al. 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*, pages 720–723, 2013.

[14] J. Bian et al. 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*, pages 526–530, 2011.

[15] W. Shen et al. Smart Border Routers for eHealthCare Wireless Sensor Networks. In *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4, 2011.

[16] M. Satyanarayanan et al. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, 8(4):14–23, 2009.

[17] F. Touati and R. Tabish. U-Healthcare System: State-of-the-Art Review and Challenges. *J. Medical Systems*, 37(3), 2013.

[18] H. Mamaghanian et al. Compressed Sensing for Real-Time Energy-Efficient ECG Compression on Wireless Body Sensor Nodes. *IEEE Transactions on Biomedical Engineering*, 58(9):2456–2466, Sept 2011.

[19] Z. Zhang et al. Compressed Sensing for Energy-Efficient Wireless Telemonitoring of Noninvasive Fetal ECG Via Block Sparse Bayesian Learning. *IEEE Transactions on Biomedical Engineering*, 60(2):300–309, 2013.

[20] H. Boström et al. On the Definition of Information Fusion as a Field of Research. Technical Report HS- IKI -TR-07-006, University of Skövde, School of Humanities and Informatics, 2007.

[21] T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 2012.

[22] Health Level Seven Int'l. Introduction to HL7 Standards. *www.hl7.org/implement/standards [accessed 2014-08-04].*

[23] F. Bonomi et al. Fog Computing and Its Role in the Internet of Things. In *Proceedings of the MCC Workshop on Mobile Cloud Computing*, pages 13–16, 2012.

[24] netfilter/iptables - nftables project. *http://netfilter.org/projects/nftables/ [accessed 2014-08-04].*

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

[26] ZigBee Alliance. Understanding ZigBee Gateway: How ZigBee extends an IP network. (September):1–13, 2010.

[27] W3C technical reports. The WebSocket API. *http://dev.w3.org/html5/websockets/ [accessed 2014-08-04].*

[28] OMAP®4 PandaBoard System Reference Manual. *http://pandaboard.org [accessed 2014-08-04].*

[29] SmartRF06 Evaluation Board User's Guide. *http://www.ti.com/lit/ug/swru321a/swru321a.pdf [accessed 2014-08-04].*

[30] Olimex. MOD-ENC28J60 development board, Users Manual. *https://www.olimex.com/Products/Modules/Ethernet-/MOD-ENC28J60/resources/MOD-ENC28J60.pdf [accessed 2014-08-04].*

[31] R. Barry. *Using The FreeRTOS Real Time Kernel, Microchip PIC32 Edition*. FreeRTOS Tutorial Books, 2010.

[32] RTX41xx Low Power Modules. *http://www.rtx.dk/RTX41xx_Modules-4024.aspx [accessed 2014-08-04].*

[33] RTX Real-Time Operating System. *http://www.keil.com/rl-arm/kernel.asp [accessed 2014-08-04].*

[34] A. Dunkels et al. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *International Conference on Local Computer Networks*, pages 455–462, 2004.

[35] Texas Instruments. RTX41xx Low Power Modules. *http://www.rtx.dk/RTX41xx_Modules-4024.aspx [accessed 2014-08-04].*

[36] Texas Instruments. Low-Power, 2-Channel, 24-Bit Analog Front-End for Biopotential Measurements. *http://www.ti.com/lit/ds/sbas502b/sbas502b.pdf [accessed 2014-08-04].*

[37] IEEE Standard for Medical Device Communication, Overview and Framework. In *ISO/IEEE 11073 Committee*, 1996.

[38] K. Sayood. *Lossless Compression Handbook*. Elsevier Science, 2002.