# Fault Tolerant and Scalable IoT-based Architecture for Health Monitoring

Tuan Nguyen Gia[1], Amir-Mohammad Rahmani[1], Tomi Westerlund[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: {tunggi, amirah, tovewe, pakrli}@utu.fi, hannu@kth.se

***Abstract*— A novel Internet of Things based architecture supporting scalability and fault tolerance for healthcare is presented in this paper. The wireless system is constructed on top of 6LoWPAN energy efficient communication infrastructure to maximize the operation time. Fault tolerance is achieved via backup routing between nodes and advanced service mechanisms to maintain connectivity in case of failing connections between system nodes. The presented fault tolerance approach covers many fault situations such as malfunction of sink node hardware and traffic bottleneck at a node due to a high receiving data rate. A method for extending the number of medical sensing nodes at a single gateway is presented. A complete system architecture providing a quantity of features from bio-signal acquisition such as Electrocardiogram (ECG), Electroencephalography (EEG), and Electromyography (EMG) to the representation of graphical waveforms of these gathered bio-signals for remote real-time monitoring is proposed.**

***Keywords—Internet of Things, e-Health, 6LoWPAN, Wireless Sensor Network (WSN), Remote Patient Monitoring, Fault Tolerance, Scalability***

## I. INTRODUCTION

In the near future, it is predictable that many physical objects, in addition to computers, sensor actuators and such, will be distributed with unique addresses and the ability to transfer data. This technology, called Internet of Things (IoT), provides an integration approach for all these physical objects that contain embedded technologies to be coherently connected and enables them to communicate and sense or interact with the physical world, and also among themselves. As a result, information of any object and service will be accessible in a systematic way [1].

A low-cost and smart IoT enabled healthcare system which has the ability to monitor patients' health remotely using wireless sensors has a possibility to improve the quality of healthcare, potentially save patients' lives and reduce the overall costs in healthcare. Reducing the cost of healthcare is a significant concern around the world; therefore, the efficient utilization of resources and the unit cost of healthcare services and devices have to be considered. Solely, the obesity among the people increased exponentially over past three decades and reached 500 million, which resulted in several illnesses such as cardiovascular and diabetes [2]. Also the aging of population and hereditary ailments increase healthcare expenses. For instance, $1.5 trillion was spent annually for medical care in United States [3].

Wireless sensor networks (WSNs) are ubiquitously used in a large number of applications including home automation, entertainment, industry, and healthcare [4][5]. Wireless sensor networks can be constructed by applying different technologies such as Wi-Fi, Ethernet and IEEE 802.15.4. Especially, IEEE 802.15.4 standard [6] is broadly used due to its low-power consumption and low-cost. Various protocols and technologies based on IEEE 802.15.4 such as Zigbee, MiWi and other protocols are presented and applied in quantities of applications. However, these protocols and technologies cannot be considered as the most prominent candidates for remote health monitoring applications because of their problems of high power consumption, low adaptability, low scalability, and non-IP based connection. In order to cope with these issues, IPv6 Low-Power Wireless Personal Area Network (6LowPAN) [7] was proposed which extends IP to low-power WSNs. 6LoWPAN provides a quantity of advantages such as high reliability and adaptability, energy efficiency, mobility and low-cost. However, fault tolerance and healthcare services are not explored in a great number of 6LoWPAN healthcare applications. Our proposal in this paper is motivated by these advantages of 6LoWPAN and non-envisaged 6LoWPAN aspects in remote health monitoring systems.

In this paper, we present a customized 6LoWPAN architecture for healthcare environments. The aim is to implement an enhanced gateway to provide a method for solving bottleneck at edge routers due to 250kbps data rate limitation of 6LoWPAN and improving network fault tolerance. Furthermore, we provide a complete architecture for healthcare monitoring starting from bio-signal acquisition by using Analog Front End (AFE) devices integrated in 6LoWPAN medical sensor nodes to the final representation health and contextual data stored in a cloud server to end-users. The key contributions of this work are as follows:

- A complete IoT-based healthcare system supporting high data rate bio-signals
- An enhanced gateway with scalability and fault tolerance capability
- A customized tunneling gateway for routing packets from nodes to a server on the Internet.

This paper is organized as follows. In Section II, the related work and the motivation are discussed. Section III provides our e-Health system architecture based on 6LoWPAN. Section IV presents the system implementation in more details, while Section V demonstrates the experimental results. Finally,

978-1-4799-6117-7/15/$31.00 ©2015 IEEE

Section VI concludes the paper and discusses some directions for future work.

## II. Related Work and Motivation

In the near future, healthcare applications based on IoT will have important roles in hospital environments and also in everyday life. Healthcare applications based on wireless sensor networks are used by doctors/caregivers for real-time remote monitoring of health related bio-signals such as ECG, EEG, EMG, body temperature, peripheral capillary oxygen saturation (SpO2), blood pressure, respiration, glucose and contextual data. Strict requirements of data rate defined by the IEEE 1073 group [8], is shown in Table 1. In addition to that, other requirements of reliability, connectivity, user interaction and moderate costs must also be accomplished. Furthermore, when the number of old people and patients increases over the years, healthcare applications require expandability to serve all patients.

**TABLE I. Data rate of various bio-medical signals**

| Bio-medical Signal | Latency | Data Rate |
|---|---|---|
| Blood pressure | < 3 s | 80 - 800 bps |
| Pulse / Heart Rate | < 3 s | 80 - 800 bps |
| Glucose | < 3 s | 80 - 800 bps |
| Temperature | < 3 s | 80 - 800 bps |
| Respiration | < 300 ms | 50 - 120 bps |
| SpO2 | < 300 ms | 50 - 120 bps |
| ECG | < 300 ms | 3-lead (2.4 kbps), 5-lead (10 kbps), 12-lead (72 kbps), |

Many researches based on Wi-Fi, ZigBee, and RFID technologies have been carried out to partly implement architectures fulfilling these requirements. For instance, Zhang *et al.* [9] propose healthcare application based on ZigBee standard for disease monitoring, personal wellness monitoring and personal fitness monitoring. In another effort, Wu *et al.* [10] present a mobile health monitoring system using RFID ring-type pulse sensor. That application obtains patients pulse and temperature and then sends data to a smartphone through Bluetooth. Shanko *et al.* [11] propose a micro-controller application for real-time health monitoring using Wi-Fi and GPRS. These discussed applications are implemented for health monitoring but bio-medical signals such as ECG, EMG, and EEG which play important roles in diagnosing and treating several diseases are not properly considered.

There exist also some works presenting architectures based on 6LoWPAN. For instance, Bag *et al.* [12] discuss energy and bandwidth efficient architecture based on 6LoWPAN. The architecture focuses on mobility of network allowing mobile nodes (MNs) to be able to connect to multi-hop in different personal area networks (PAN) without changing software stack. Shin *et al.* [13] present a platform providing high level of mobility for large scale IP-based sensor networks. The platform based on 6LoWPAN gives solutions for scenarios in which a node moves in a single WPAN domain and in different WPAN domains. Tabish *et al*. [14] propose a 3G/Wi-Fi-enabled 6LoWPAN-based u-healthcare system for ubiquitous real-time monitoring and data logging. The proposed system uses a 6LoWPAN edge router connected to a computer for performing as a gateway to obtain ECG and body temperature. However, the system has some limitations such as i) there is no fault tolerance in 6LoWPAN networks, ii) it is inefficient in terms of cost and size when forming a gateway by a base station, iii) the system is not completely independent when using a free online SensorMonkey service. Scalability and fault tolerance, which are major challenges in 6LoWPAN due to 250kbps data rate and limited resources, are not studied in these approaches. Few applications based on 6LoWPAN in healthcare environments consider fault tolerance. Jara *et al.* [4] present HWSN6 Hospital Wireless Networks based on 6LoWPAN technology including mobility and fault tolerance management. The presented 6LoWPAN architecture supports node mobility in medical environments and fault tolerance in case of failing connection between a node and a sink node caused by mobility. The proposed architecture cannot guarantee to maintain system stability when connection failures in the network occur by virtue of malfunction of sink node hardware or other reasons.

The main motivation of this paper is to provide a novel IoT architecture supporting scalability, fault tolerance and healthcare services based on customized 6LoWPAN for medical environments. The major difference from previous works is that we present a fault tolerant approach with backup routing from a node to other sink node and an advance service mechanism to maintain connectivity in case of the occurrence of a connection failure between the node and the sink node. The fault tolerance covers many fault situations from malfunction of a sink node's hardware, to a communication bottle neck at a sink node due to a high receiving data rate. Furthermore, we introduce a method of extending the number of sink nodes at a single gateway which, in turn, increases the number of sensing nodes in a gateway sub-network. In addition, we provide a complete system comprising bio-signal acquisition such as ECG, EEG and EMG and plotting graphical waveforms of these gathered bio-signals to remote terminal. Our architecture is more advanced than other conventional platforms in terms of energy efficiency, scalability, fault tolerance, and quality of services.

## III. System Architecture

The architecture of our IoT-based health monitoring system comprises of star-based 6LoWPAN nodes, a gateway and a back-end part as shown in Fig. 1. A 6LoWPAN medical sensor node is integrated with AFE to obtain bio-signals such as ECG, EEG, EMG, and SpO2. The AFE device is a compact board including MCU, ADC and other components to acquire analog signals through electrodes attached at patients' skin in specific positions, and to convert these analog signals to digital forms to be transmitted from a node to a sink node. In addition, contextual data such as temperature, location, and humidity, which helps medical doctors in diagnosing patients' diseases more accurately, can be gathered through sensors at a node.

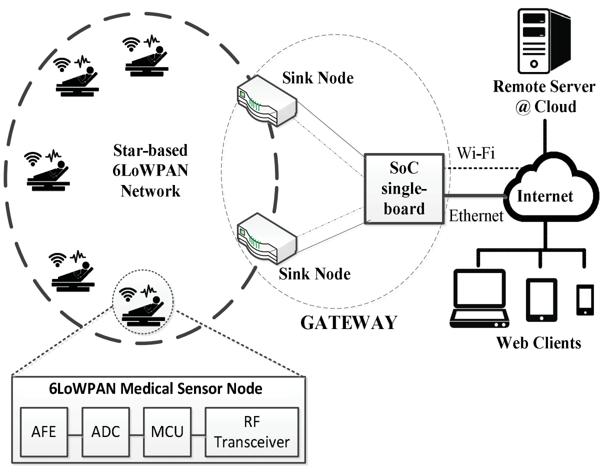


**Fig. 1.** **IoT-Based Healthcare System Architecture**

The gateway plays a vital role in our architecture such as collecting bio and contextual data from 6LoWPAN nodes, transmitting them to a remote server, providing many services for maintaining connectivity and enhancing the whole system. The gateway operational structure includes a hardware layer, an embedded operating system and a service layer as shown in Fig. 2. The hardware layer, which communicates with physical components such as micro controller (CPU), SD Card, Wi-Fi and Ethernet module, defines means of transmitting raw data. Operating system acts as an administrator in the gateway for managing resources, scheduling, and connecting between physical and service layers. Depending on running services and resources availability, different hardware modules can be assigned and utilized. The service layer, which provides gateway features and functionality, includes the following services.

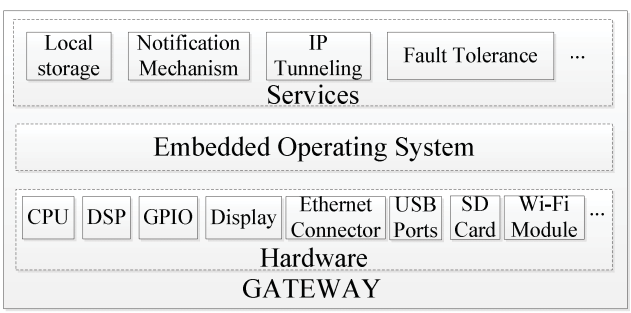


**Fig. 2.** **Gateway Operational Structure**

***Local Storage****:* A local storage consists of a synchronized storage for temporary bio-data sent by sensor nodes and a static look-up storage recording database and threshold values for notification mechanisms. The synchronization between a gateway's local database and a cloud server's database is regularly done with only synchronized storage. This ensures that bio-data in the cloud database is up-to-date all the time and can be used in patient monitoring. Up-to-date cloud database also ensures that all the gathered data is available and safe. Indeed, in medical applications the safety and security of data are essential. Another important point is limitations of embedded devices. This enforces us to purge gateway's synchronized local storage in every 5 minutes. By purging the synchronized storage, we avoid the problem of database becoming full which decreases the performance of other essential services. The static look-up storage, which is configured by system administrators, is used as references for different services and mechanisms; therefore, it is kept intact.

***Notification mechanism***: The lead-off notification service guarantees that gathered bio-signals have good qualities. The service informs caregivers/doctors when the connection between an electrode lead and a human body has loosened. A loose connection causes high impedance which is one of the reasons for a low quality signal. The temperature notification informs the system administration when a sensor node's temperature has reached the critical temperature. The temperature notification service prevents damages to a patient who wears a sensor node on the body as well as damages to the sensor node itself.

***IP tunneling:*** The IP tunneling is used for connecting between 6LoWPAN and IPv4/IPv6 protocols providing seamless extension of the existing IP networks to encompass resource-constraint sensor networks. IP tunneling includes the *gogoc* service and router advertisement daemon which provide configuration methods to modify a tunnel acquired from a tunnel server. This accompanies with the tunnel setup protocol and link-local advertisement of IPv6 router addresses respectively. IP tunneling can be reconfigured for serving many sink nodes due to the maximum of 256 sub-networks of 64bits in IP tunneling.

***Fault tolerance:*** Especially in health and well-being applications, robustness has an intrinsic role for the reliability and creditability of an application. To enhance the robustness of our IoT architecture, we implement extra measures to ensure that all sensor nodes, which are connected to a gateway, are active and working properly. The extra measures are implemented in a gateway and they are triggered by the inactivity of a sensor node. The inactivity is observed by a gateway by monitoring the data flow of a sensor node. If a sensor node has been inactive, that is, a gateway has not received data from that node, for a predefined period of time, the gateway will initiate a protocol to discover the reason for the inactivity.

The first step in the protocol is to send a *connection-status message* to a sensor node from which a gateway has not received data for a while, for example, within 1 or 2 minutes. The period of time can be changed according to the quality of service (QoS) requirements of the application. After receiving the connection-status message sent by the sink node to which the sensor is connected to, the inactive sensor node immediately transmits a *reply message* back to the gateway. If the gateway receives the reply message, the system is working correctly. However, if the gateway does not receive the reply message within one minute owing to, for example, a lost connection or a faulty sensor node, it broadcasts a *warning* *message*. The warning message is sent through another sink node within a gateway (see Fig. 1). By sending through another sink node, the possibility of a faulty sink node can be eliminated. Each of the sensor nodes within the range of the broadcasting sink node reacts to the warning message and verifies if the warning message is targeted for it or not by checking the identification number (Id) in the warning message.

An Identification number is assigned for each sensor node when they register to a sink node. It is a unique number that is based on a unique MAC address defined by the producer of the sensor node. The unique Id is stored in a register table. Each sink node within a gateway has its own register table. The table is used for keeping track of connections between sensor nodes and sink nodes.

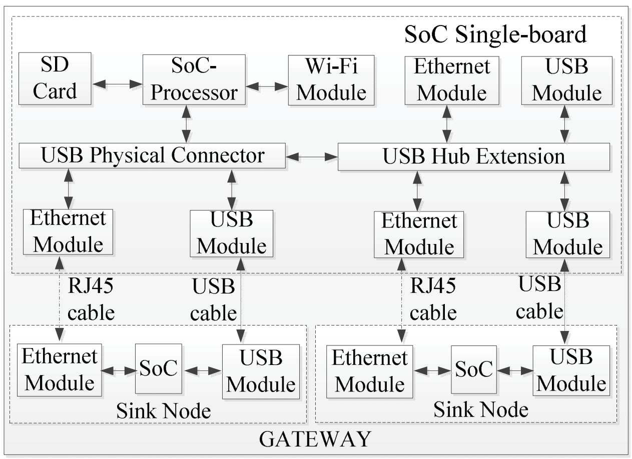


**Fig. 3. Gateway Physical Structure**

The physical structure of the gateway provides connections between a single-board system-on-chip (SoC) and sink nodes, shown in Fig. 3. Single-board SoC includes different components such as a SoC processor, an extendable SD card, a USB physical connector and a Wi-Fi module. Extendable SD card, which is used as hard drive for storing an operating system, MySQL database, and data, can be expanded depending on application demands. The USB connector provides a platform for integrating USB modules such as an Ethernet module, a serial module and a USB hub extension which offers means to expand a number of USB modules connected to a single-board SoC. There are three sets of Ethernet and USB modules in the single-board SoC, see Fig. 3. However, as long as the number of hub tiers is less than or equal to tier 7 which is the maximum hub extension tier, the number of module sets can be extended depending on application specifications. The single-board SoC connects to the Internet through either Wi-Fi or Ethernet module, which improves the flexibility of our gateway. The Ethernet connection, in some cases, is better than 3G/4G in term of stability because in some areas, 3G/4G network may not work properly. Other Ethernet and USB modules are utilized for tunneling between the single board and sink nodes.

## IV. System Implementation

In this section, we present the system implementation of our IoT-based healthcare system in three phases including sensor nodes, gateway, and back-end system.

### *A. Sensor nodes*

The core of the medical sensor node is TI CC2538 SoC [15] which has a powerful ARM Cortex M3-based 2.4 GHz MCU, IEEE 802.15.4 compliant transceiver radio, 512 kB flash, 32 kB RAM, 32 GPIO, and SPI. It also has battery monitor and temperature sensor making it ideal for our system.

The MCU sub-module uses SPI buses to connect with the ADS1292 AFE [16] which is a low-power, 2-channel, 24bit AFE with 2 low-noise PGAs and 2 high-resolution ADCs. AFE obtains analog signals from electrodes attached to patient's skin and then convert them to a digital form through fast, accurate, and low noise analog digital converter (ADC). 2 MHz clock and the *continuous* mode supported in AFE for sending ready digital data after ADC were applied to obtain data from AFE. When digital data in AFE is ready, interrupt in CC2538 is triggered; as a result, the sensor node can get up-to-date signal. The implementation of the medical sensor node with ADS1292 is illustrated in Fig. 4.

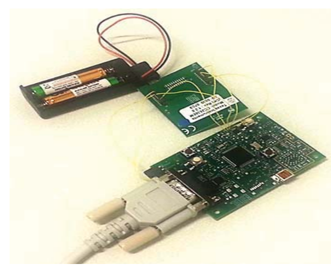

**Fig. 4. Medical sensor node based on TI CC2538 with TI ADS1292**

In our system, we used the Contiki operating system [17]. Contiki is developed for low-power IoT devices, making it ideal for our system. The operating system manages and schedules all tasks including communication related tasks such as reading data from the AFE devices and, generating and receiving UDP packets over 6LoWPAN networks.

Notification services are accompanied in the gateway. Our sensor node informs three important levels of battery: 10%, 5%, 1%, and two levels of temperature: 50 and 70 degrees of Celsius When battery or temperature reaches to these levels, interrupts in a sensor node are triggered and messages containing these data are sent to the gateway. Depending on messages' content and a notification mechanism in the gateway, different alerts with different priorities are sent to doctors/caregivers. Similarly, a lead-off notification was done by interrupt service routine.

### *B. Gateway*

Our gateway was implemented by integrating a Pandaboard device [18] and sink nodes which in turn are compositions of a CC2538 module and a SmartRF06 board as shown in Fig. 5.

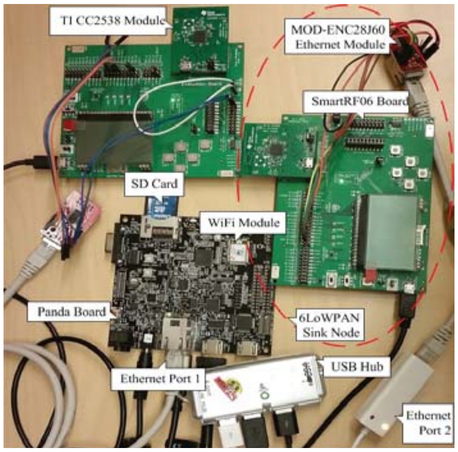


**Fig. 5. Gateway Implementation**

The Pandaboard is the core for our gateway, based on an OMAP4430 platform comprising of integrated on-chip memory, external memory interfaces, peripherals interfaces and Cortex A9 microprocessor unit including dual core ARM Cortex A9 cores with symmetric multiprocessing at up to 1.2 GHz each. It supports different network standards such as Ethernet, 802.11b/g/n and Bluetooth, and operating systems such as Windows CE, Symbian and Linux. The Pandaboard is connected to a sink node through an Ethernet port and a serial port, integrated in the Pandaboard. However, the Pandaboard originally has one Ethernet port and two serial ports. In order to extend the number of sink nodes connected to the Pandaboard, extra Ethernet and serial ports must be provided by applying a USB hub with USB to Ethernet port converter into the Pandaboard. The proposed method provides scalability because the number of extra ports and drivers are handled by the operating system installed in the system.

The CC2538 module is attached to a TI SmartRF06 board - an ARM Cortex-M based System-on-Chip (SoC) - to form a sink node on the ground that the CC2538 module does not offer adequate I/O port interface for all required connections. As mentioned above, a sink node requires an Ethernet port to connect to the Pandaboard. However, the set of the CC2538 module and TI SmartRF06 does not include an Ethernet port. In order to equip the Ethernet port for a sink node, the Olimex Ethernet module [19] is applied to establish the Ethernet connection between the SmartRF06 board and the Pandaboard to enable data transfer between them. The USB connection between the SmartRF06 board and the Pandaboard is used for providing power supply to the set of the CC2538 module and the SmartRF06 board.

The embedded operating system empowers the system to accomplish its functionalities by providing a platform for implementation and also enables us to have several applications. Other than UDP server, we implemented all mentioned services such as MySQL database to store the copy of received data temporarily in the gateway. The notification framework tables in the gateway are created using a federated engine to allow us to create reference for the records available in the server without database replication. Furthermore, it provides automatic data synchronization while giving less priority to manual synchronization. This allows the notification service to continue its operation in case of the Internet unavailability. Customizing IP tunnelling was done by configuring *gogoc* and *router advertisement daemon* for allowing sub-networks and routing advertising in many Ethernet interfaces.

In parallel with the physical setup mentioned above, different configurations and applications were implemented and customized in order to achieve fault tolerance in our gateway. We implemented the socket application for enabling real-time event-based communications between different sink nodes and the single-board SoC. We also customized a *tunslip* application provided in the embedded operating system for initiating different virtual *tun* interfaces and then creating serial line internet protocol (SLIP) tunnels between physical serial interfaces and virtual tun interfaces. In addition, database required for the fault tolerance mechanism, and sink node's applications were also created in MySQL and Contiki, respectively. These applications together with the fault tolerance mechanism mentioned create a novel solution for fault tolerance.

### *C. Back-end System*

For the server, we used a free hosting service for our demonstration. The back-end is MySQL database, PHP used as server-side scripting and JavaScript (JQuery) for HTML content generation such as plotting charts. We also developed an Android application for notification and appropriate web services in the server for information retrieval. The notification web services receive request and return XML as response. The received XML is parsed in the Android application and a notification is raised based on the status section of XML response.

## V. Experimental Results and Demonstrator

In our demonstration, to analyse the impact of environment and distances among nodes for the network power consumption, system reliability, scalability and fault tolerance, we setup four different scenarios in which our 6LoWPAN-based network is comprised of 7 nodes and a standalone gateway (see Fig. 1), but with different configurations. These scenarios are illustrated in Fig. 6.

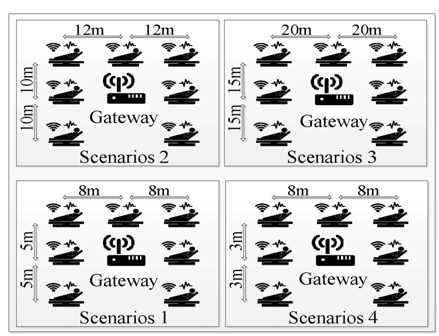


**Fig. 6. Experimental setup scenarios**

The sensor nodes and the gateway are located in a room with distances varying from 5 to 20 meters. All sensor nodes are configured to send a set of health data including 3-lead ECG, SpO2, Blood Pressure, Heart Rate, Temperature, Respiration and Glucose, which together needs 8.7kbps of data rate. This data rate is calculated according to the specifications of the IEEE 1073 standard. The power consumption is measured based on 3.3V power supply. As results, the average power consumption of a node ranges from 6 to 9mW. As expected, the 6LoWPAN network offers a low power consumption characteristic as compared to other IP based networks such as Wi-Fi where the average power consumption for the same scenarios is in the range of 14-20mW (when low-power RTX4140 Wi-Fi module [20] is used).

In order to verify fault tolerance in our architecture, we disabled one of the sink nodes in our gateway and applied these scenarios. As a consequence, bio-data was obtained properly, see Fig. 7. In the figure, a 3-lead ECG data is being captured by the ADS1292 analog front-end device, and sent to the gateway where the data is manipulated and updated in the remote virtual server at the cloud. Finally, a graph of the real-time ECG data is plotted in a PDA device. On the other hand, we added more sink nodes at our gateway to verify system scalability. Consequently, we got similar results as a test case of fault tolerance.

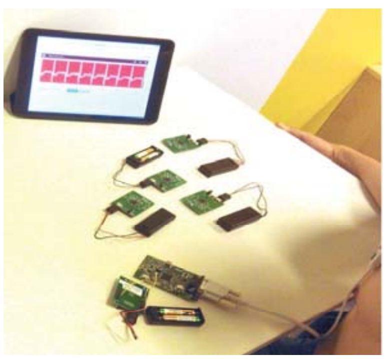

**Fig. 7. Demonstration of our IoT-based healthcare system**

## VI. CONCLUSIONS AND FUTURE WORK

Especially in healthcare, robustness has an intrinsic role for the reliability and credibility of the monitoring systems. In this paper, we presented a novel Internet of Things based architecture supporting scalability and fault tolerance. Fault tolerance is achieved by having advanced extra measures that maintain the connectivity between sensor nodes and a gateway. The presented fault tolerance approach covers many fault situations such as malfunction of sink node hardware and traffic bottleneck at a node due to a high receiving data rate. Part of fault tolerance, we presented a method to extend the number of medical sensing nodes at a single gateway. Finally, we presented a complete system architecture from bio-signal acquisition to remote real-time monitoring of these signals. The bio-signals are, for example, ECG, EEG and EMG. Part of the complete IoT based healthcare monitoring system is the capability to present the bio-signals in a graphical waveform on caregivers' handheld devices. Energy efficiency has been part of the design process, and therefore the wireless system is built on top of 6LoWPAN energy efficient communication infrastructure to maximize the operation time.

## REFERENCES


[1] European Commission Information Society (2008). Internet of Things in 2020: a Roadmap for the Future [online]. Available from: www.iot-visitthefuture.eu.

[2] "Obesity and overweight", World Health Organization, 2014. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs311/en/ [accessed 2014-8-5].

[3] C. Roehrig, G. Miller, C. Lake, J. Bryant, "National Health Spending by Medical Condition, 1996-2005" "Health Affairs", February 2009, w258-376.

[4] A.J. Jara et al. "HWSN6 Hospital Wireless Sensor Network based on 6lowpan technology: mobility and fault tolerance management", *In Proc. IEEE ICCSE* 2009 pp.-879-884

[5] G.T. Nguyen, N.K. Thanigaivelan, A-M. Rahmani, T. Westerlund, P. Lijeberg, and H. Tenhunen, "Customizing 6LoWPAN Networks towards Internet-of-Things Based Ubiquitous Healthcare Systems", NORCHIP, 2014

[6] 802.15.4-2003, IEEE Standard, "Wireless medium access control and physical layer specifications for low-rate wireless personal area networks", May 2003.

[7] Z. Shelby and C. Bormann, "6LoWPAN: The Wireless Embedded Internet", Wiley, 2009.

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

[9] B. Zhang, S. Coll, "Health care applications based on ZigBee standard", Computer Design and Applications (ICCDA), 2010 International Conference on (Volume:1)

[10] Wu et al., "A mobile health monitoring system using RFID ring-type pulse sensor", 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, pp. 317-322, 2009

[11] E.J. Shanko, and M.G. Papoutsidakis, "Real Time Health Monitoring and Wireless Transmission: A µController Application to Improve Human Medical Needs", The 4th IEEE International Conference on E-Health and Bioengineering - EHB 2013

[12] G. Bag, M.T. Raza, H. Mukhtar, A.H. Akbar, S.M.S. Shams, K-H. Kim, Y. Seung-Wha, and K. Donghwa, "Energy-aware and bandwidth-efficient mobility architecture for 6LoWPAN", Military Communications Conference 2008, pp.1-7

[13] M.K. Shin, H.J. Kim, "Mobility Support in IP-based Sensor Networks", ICT Convergence (ICTC) 2011

[14] Tabish et al., "A 3G/WiFi-enabled 6LoWPAN-based U-Healthcare System for Ubiquitous Real-Time Monitoring and Data Logging", Middle East Conference on Biomedical Engineering (MECBME), 2014

[15] CC2538 SoC for 2.4-GHz IEEE 802.15.4, 6LoWPAN and ZigBee Applications (Rev.A). [Online]. Available: http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=cc2538&fileType=pdf [accessed 2014-8-10]

[16] Low-Power, 2-Channel, 24-Bit Analog Front-End for Biopotential Measurement (Rev.B). [Online]. Available: http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=ads1292&fileType=pdf [accessed 2014-8-10]

[17] 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.

[18] Resources, pandaboard.org. [Online]. Available: http://pandaboard.org/content/resources/references [accessed 2014-8-10].

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

[20] RTX41xx Low Power Modules, retrieved on Jul 17 2014 from http://www.rtx.dk/RTX41xx_Modules-4024.asp