An IoT based system for remote patient monitoring

Conference Paper · May 2016

DOI: 10.1109/CarpathianCC.2016.7501056

An IoT Based System for Remote Patient Monitoring

Alexandru Archip*, Nicolae Botezatu*, Elena Şerban*, Paul-Corneliu Herghelegiu* and Andrei Zală†
*"Gheorghe Asachi" Technical University of Iaşi, România
Faculty of Automatic Control and Computer Engineering
Email: {alexandru.archip,nbotezatu,eserban,pherghelegiu}@tuiasi.ro
†Moineşti Emergency Hospital, Bacău, România
Email: andrei_zala@yahoo.com

Abstract—Following a surgical procedure, patients are monitored in an ICU until physically stable, after which are discharged to a ward for further evaluation and recovery. Usually, ward evaluation does not imply continuous physiological parameters monitoring and therefore patient relapse is not uncommon. The present paper describes the steps taken to design and build a low-cost modular monitoring system prototype. This system aims to offer mobile support in order to facilitate faster and better medical interventions in emergency cases and has been developed using low-power dedicated sensor arrays for EKG, SpO2, temperature and movement. The interfaces for these sensors have been developed according to the IoT model: a central control unit exposes a RESTful based Web interface that ensures a platform agnostic behaviour and provides a flexible mechanism to integrate new components.

Index Terms—Embedded Systems, Internet of Things, E-health, RESTful Web Services, Remote patient monitoring

I. Introduction

Active post-surgery patient monitoring plays an important role in clinical services. A viable solution for this task must be cost effective, must solve the human resource availability problem and must have a positive impact on the patients' overall quality of life.

We devised a solution that includes modular sensor blocks and is mostly build from off-the-shelf hardware modules. These sensor blocks are connected to a central control unit. Communication between the sensors and the control unit is performed using low-power radio interfaces (i.e. IEEE 802.15.4). Furthermore, this solution has been developed according to the IoT model: it embeds a software component running on the control unit that is responsible for data acquisition; such data is then made available in real-time to external devices in JSON format through a RESTful based Web Service. Except for minimal data conversion, storing and buffering, the software component does not perform any other data processing tasks. We adopted this approach because 1) data processing may vary depending on the physiological parameter in focus and 2) it may induce supplemental delays that could nullify the real-time behavior of our prototype. Also, we have developed an application for mobile devices (i.e.tablets) that allows for real-time data visualisation.

The described prototype offers the following advantages. First of all, it allows for physicians to monitor the patients even when they are not physically present in front of an existing wired medical monitoring station. This is especially useful when the medical personnel is reduced to support or emergency medical services. A second importand advantage is that due to the mobile sensors used and their low-power consumption profile, this system allows for a greater degre freedom of movement for non-critical patients. Also, provid that adequate security is ensured, the system may be easily extended to allow for support for chronic patient remote monitoring (such as people suffering from asthma, chronic obstructive disease and overlap syndrome).

II. Related Work

Nowadays, telemedicine represents a very important research avenue. Significant effort has been deployed in assisting the patients in their everyday life through telehealth and telemedicine systems. A comprehensible review on the telemedicine systems and the communication technologies used has been presented by Custodio et al. [1]. Commonly, the telehealth systems focus on detecting health related ab normalities in elderly users. The related papers we analized describe similar systems at different levels of detail. Althou the majority present whole systems, the authors usually focus on three main topics: (1) data acquisition and low level sys design, (2) data transmission within the system and network related issues and (3) system integration in existing comm nication networks in the Internet-of-Things (IoT) pardigm.

An early system that uses a cell phone application for remote monitoring that uses a GPS, an accelerometer and a light sensor is described in [2]. The prototype in [3] allows physicians to remotely assess the patients health in and outdoors. Alongside sensors that acquire the data from the patient, their prototype also includes an alert system for emergency cases.

In [4] the authors propose a monitoring system based on Embedded Network Gateway Servers (ENGS). These devices can serve the acquired data via an embedded web server a have modular design in order to be interfaced to a variety o sensors. The authors of [5] detail the specific challenges po by the design of wearable patient monitoring devices. They address in detail power consumption, data acquisition and filtering issues. MEDiSN [6], a wireless sensor network for

monitoring physiological data in emergency rooms, combines a two-tier system architecture and a rate control protocol to successfully address QoS requirements in large scale netwo
Both real-time and historical patient data analysis and management are the objectives of the framework presented in [7]. This framework, intended to be lightweight, agile and scalable, utilizes a resource-oriented architecture (ROA) bas RESTful HTTP to connect wireless physiological sensors, wireless networks and a cloud computing platform.

Wanda, a remote health monitoring system for heart failure patients is presented in [8]. The system consists of a smartphone-based data collection gateway, an Internet-scale data storage and search system, and a backend analytics engine for diagnostic and prognostic purposes. The need and the benefits of electronic and IT services in the medical field is emphasised in [9]. The main purpose of the introduced framework is to provide better and more accurate data access for physicians and aid persons. A recent telemedicine monitoring system using ZigBee communications was developed by Yung et.al.[10]. It offers medical data monitoring services, patient tracking and paging services.

A common problem that occurs when using different types of sensors is their interoperability as they might use different communicaton standards. A gateway that aims to solve this problem was presented in [11].

A follow-up on the home telemonitoring approaches in Chronic Obstructive Pulmonary Disease has been presented [12]. The authors stress out that remote monitoring is cheaper that the clinical one. Another conclusion of the study is that adjustments to the telemonitoring systems have to be done to fulfill all user requirements. These adjustments should mostly focus on the usability of the systems.

An IoT paradigm integrating health related systems that was developed to track the patient movements was described in [13]. The authors introduce VIRTUS, a new middleware software system that provides interconectivity solutions between the sensors and the used mobile device.

To the best of our knowledge, there is no remote health monitoring system that provides support for real-time data acquisition and visual representation for multiple sensor types, while offering platform agnostic interfaces. The low-power profile and low cost of the modules used in our prototype are also key advantages of our solution.

III. System Design and Implementation

The prototype system we developed implements acquisition and primary processing components for physiologic parameters. It also exposes interfaces for local and remote patient data access. An overview of this prototype is presented in Fig. 1.

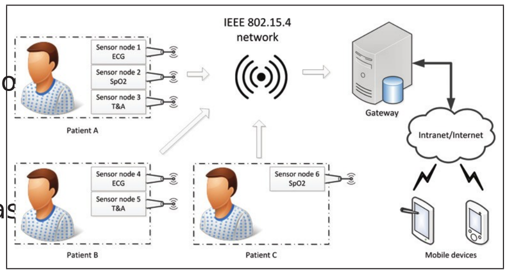

Fig. 1. Generic system architecture

A patient monitor implements at least ECG, respiratory, SpO2, NIBP and temperature functionality. Usually, a ward patient does not need all five parameters monitored, which leads to inefficient resources allocation (e.g. a patient admitted for a minor surgical intervention might need only temperature monitoring to detect subfebrile states which can indicate a nosocomial or superinfection). As a result, the system was designed with two other objectives in mind - flexibility and fair resource distribution.

These tasks are achieved by using two types of devices: sensor nodes and a Gateway. A sensor node monitors a sin physiological signal and patients can be equipped with more than one node, according to their needs (e.g. ECG node and one SpO2 node). The Gateway gathers the data from the sensor nodes and makes it available, through a minimal RESTful based Web interface, to the medical staff. This interface ensures the compliance with the IoT model.

A. Sensor nodes

The sensor nodes are mobile, battery powered devices. T are equipped with an IEEE 802.15.4 radio module used to transfer the acquired data to the Gateway.

The core of the node is the Zolertia Z1 platform, a Berke TelosB variant. It is based on a TI MSP430 16-bit microcontroller which runs at max. 16 MHz as is characterised by ri low-power capabilities.

As a proof of concept we have built a prototype system which includes the following node types:

- ECG node - based on Medlab EG04000, a four lead ECG module with 6 channels (I, II, III, aVR, aVL, aVF). The node is configured to use 3-6 channels at 50 samples p second with an amplitude of 0.03125 mV/LSB.
- SpO2 node - based on Medlab EG00352 module config ured to output the waveform at 50 samples per second. SpO2 values and perfusion index outputs are synchron with the heart rate.
- Temperature and acceleration (TA) node - a Maxim DS18B20 sensor is used for skin temperature measurement and is sampled every minute. An Analog Devices ADXL345 3-axis accelerometer is used to detect abnormal movements like seizures, excessive shivers or falls.

The application that runs on the nodes is based on the OpenWSN framework [14]. The framework enables the development of IoT solutions based on the 6LoWPAN, RPL an CoAP protocols.

Regardless of the node type, the application has a generic structure (Fig. 2). This facilitates the rapid development of new node types if specific module or sensor drivers are provide - the application uses a generic UART driver for the ECG and SpO2 nodes and for the temperature and acceleration

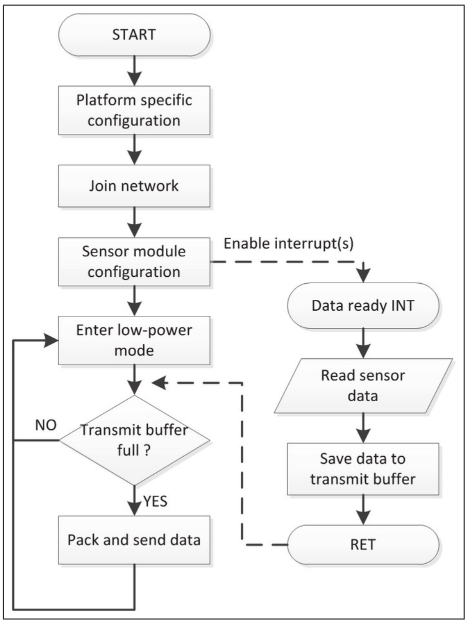

Fig. 2. Prototype system architecture at a glance

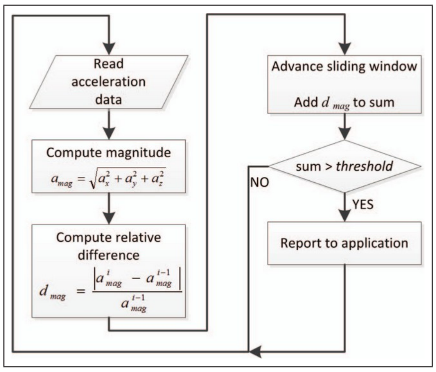

Fig. 3. Motion detection mechanism

it uses I2C (accelerometer) and 1-Wire (temperature sensor) drivers.

After the node executes the start-up code, it tries to join the network created by the Gateway and upon success, it receives the configuration parameters for the sensing module(s). The configuration step for the sensing module also enables one or more interrupts that are used for data reception. The main loop of the application enables the MSP430's LPM0 low-power mode which keeps active only selected peripherals (e.g. UART, I2C) and their clock sources. This mode is left when an interrupt is generated and upon return from the interrupt the node checks the transmit buffer and if necessary sends the stored data. Data is temporarily stored until at least 100 bytes of data are available (i.e. IEEE 802.15.4 packets have at most 128 bytes including headers).

No specific processing is done on the nodes (i.e. data is not interpreted), except for the motion detection mechanism (Fig. 3). Raw acceleration data sampled at 50 Hz would have increased the overhead of the radio link while bringing no substantial benefit to the system, so we aimed to implement the monitoring and detection mechanism without increasing the energy footprint of the node.

The simple mechanism is based on a 5 second sliding window used in conjunction with the difference between acceleration samples: 1) compute the magnitude of the cu acceleration sample; 2) determine the relative difference be- tween the magnitudes of two consecutive samples; 3) add difference to a running sum; if the value of the sum is larg than a certain threshold a message is added to the transm buffer.

B. Gateway

The Gateway is the main element of the system and is based on on a Raspberry Pi SBC system. The SBC is paired with a Zolertia Z1 platform used for data reception and se network management tasks. The data received from the s network is unpacked, stored and served upon request.

Fig. 4 presents the interactions between the software mo ules of the Gateway. The Manager module represents the logic between the other modules. The Web Server module provides an interface for the client applications. It is used for configuration and for accessing stored data (e.g. get moni patient status, get sensor type and status, get patient data history, login and user management). The DataSocket mod is used for real-time data access. This module streams the acquired from the sensors to the application. The File System module manages the access to a local database. The database stores patient (i.e. personal data, associated sensor nodes, data history) and sensor node (i.e. type, capabilities and settings) data. The Z1 platform accesses configuration information for the joining nodes and sends received data the connected ones.

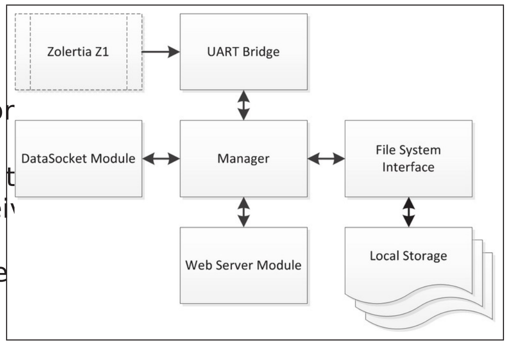

Fig. 4. Gateway software modules

C. The RESTful based Web Service

The RESTful based Web Service for our prototype has be designed with respect to the following considerations: 1) has to be lightweight such that it would offer quick response ti (i.e. the communication overhead induced by such an inte *must* be minimal); 2) must supply for quick access to its underlying resources; 3) the embedded processing must also be maintained to a possible minimum such that the energy consumption of the Gateway device is within accetable limi– – this is a mandatoy requirement since the Gateway is a low power profile device; 4) it must ensure real-time full access sensor data, in a platform agnostic manner.

In order to satisfy requirements 1) and 2) we have considered the lowest number of useful resources that are needed to implement the RESTful approach. Our system focuses on remote patient monitoring in hospital wards, following an IC discharge. Therefore the virtual resources of the RESTful We Service are the physician or nurse that would interpret the sensor data and the group of patients that he/she must keep under surveillance. At any given time, only one person from the medical staff may access a single Gateway device. This approach also complies with requirement 3), since it exclud the need for message routing and action isolation between multiple clients using the same system. Also, the RESTful Web Service simply delivers the data to interested parties without any supplemental processing. We considered serveral approaches to address requirement number 4): WebSockets, push notifications and basic TCP client-server communications. We chose to use basic TCP client-server communication. When a physician or a nurse chooses to closely survey a patient, the application on their device would provide a TCP server socket. The connection properties of the socket are forwarded to the Gateway device which creates a virtual resource that embeds the corresponding TCP client endpoint. The data exchanged between the two devices (the Gateway and the monitoring device) are represented as standard ASCII characters.

The complete set of resources and the allowed HTTP methods for our RESTful Web Service are given in Table I. All messages exchanged are represented using the JSON format.

TABLE I
THE RESTFUL API OF OUR PROTOTYPE

Virtual resource URI	*HTTP method*	*Brief description*
/ping	GET	query status of gateway
/login	POST	user LOGIN
/login/<token>	GET	read/query user token
	DELETE	user LOGOUT
/sensors?token=<token>	GET	list available sensors
/patients?token=<token>	GET	list available patients
/patient/set/<patient_id>	POST	sets active patient
/patient/history/<patient_id>?token=<token>	GET	reads active patient historic medical data
/patient/start	PUT	creates a new virtual resource embedding the TCP client
	DELETE	deletes the virtual resource embedding the TCP client

D. Communication Pattern

The communication pattern between the Gateway device and mobile device application (a tablet in our case) is exemplified in Fig. 5.

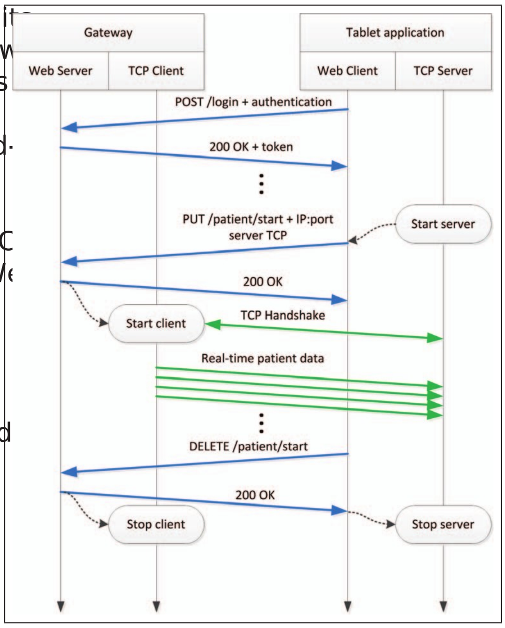

Fig. 5. Sample communication flow for real-time acquisition

The client application on the mobile device first performs a *POST* call to the RESTful API */login* resource in order to authenticate the physician or nurse that uses the system. Upon a succesful login, the user is presented with a list of patients that are connected to the Gateway device. In the next step, the medical operator may select a particular patient to monitor. The application on the mobile device opens a TCP server socket endpoint and transmits the socket properties to the Gateway through a *PUT* call. The Gateway uses this data and connects to the remote server on the mobile dev Upon a succesful connection, the Gateway application begins sending the corresponding sensor data to the mobile devic The application on the mobile devices then will receive the data from the Gateway and render it to the medical operat

Once the medical operator decides the patient no longer needs monitoring, he/she will issue a stop monitoring command which will be sent to the Gateway as a *DELETE* call and will close the corresponding TCP server socket. The Gateway devices will also close its corresponding TCP clien endpoint.

E. Mobile device application

The mobile device application was developed in Android Studio. The mobile device used for testing is a Sony xPeria Z3 Compact tablet. Following the steps presented in the subsection III-D, the main purpose of this application is to rend the sensor data it receives in real-time. In order to ensure behaviour, we have considered the following: we have wor with a maximum number of 4 sensors (ECG, accelerometer, SpO2 and temperature sensors); out of these 4 devices, the ECG sensor outputs the data on 3 distinct channels. To solve this challenge, the mobile application includes 4 distinct, h priority threads, that are assigned as follows: 3 threads ha the data from the 3 ECG channels and the last thread handles the rest of the data. The displayed data are read from the Gateway device as follows: 50 samples per second for ECG data and whenever new data is available for SpO2, temperature and pulse.

The ECG data is depicted by three separate graphics, one for each channel. The SpO2 data is displayed in a separate graphic. The other data types, like the temperature and pulse, are displayed as numerical values. Fig. 6 gives an example of the main view of the mobile device application. The accelerometer data is used to issue alerts whenever suspicious movements are detected (i.e. elder patients falls).

Fig. 6. Sample output on an Android based tablet

An image of the sensors and the Raspberry Pi B+ board (the Gateway) we have used may be observed in Fig. 7.

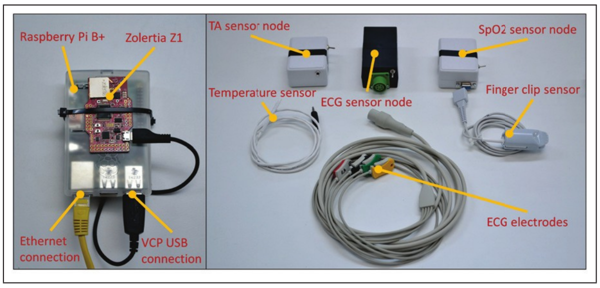

Fig. 7. Gateway (left) and Mobile sensors (right)

IV. Tests and Results

We performed two types of tests: one to evaluate the power consumption of the sensors and the mobile Gateway and the second one to asses the accuracy of the acquired data and the real-time capabilities of the remote monitoring system.

A. Power consumption tests and results

One drawback of battery powered devices is the need to frequently recharge or change the batteries which results in system downtimes and increased operational costs. As a result we set-up a test workbench in order to determine the power profiles of the sensor nodes.

The setup included two 2400 mAh NiMH rechargeable ce in a series configuration, a high-side current sensing circuit based on an Texas Instruments INA168 IC and a step-up D DC converter with a 3.3 V output. The step-up converter had to be used because of the ECG and SpO2 acquisition modu specifications that state an operating voltage of 3.3 V ±5% The results are presented in Table II. Excluding the initial-ization phase, the power rating of each node type can be s three or four states. Each node type spends most of the time in a low-power (LP) state; on a regular basis the sensors are sampled (SMP); and data is transmitted when the buffers a full (TX). One such example is given in Fig. 8.

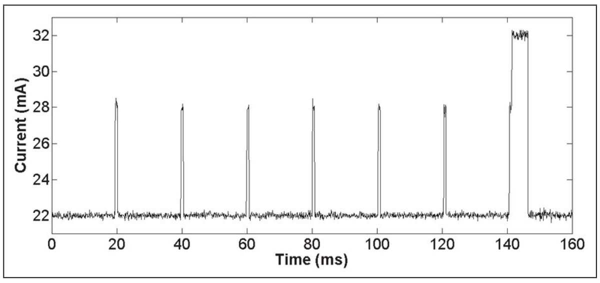

Fig. 8. Current trace for the ECG node. From left to right, the first six curre spikes indicate sampling (SMP) periods of the system, whereas the spike starts at 140 ms indicates a sampling period followed by the activation of radio module and the transmission of data.

For both the ECG and SpO2 nodes the biggest current consumers are the acquisition modules (ECG - 22 mA, SpO max. 20 mA) and the radio (< 17 mA). Based on the samp rate of the sensors and on the buffering of the data, the radio duty cycle was kept at 1.2% for the ECG node and 0.5% fo the SpO2 one. Based on these characteristics, in simulatio ECG node exceeded 4 days of operation and the SpO2 one exceeded 5 days. In our workbench the ECG node stopped operation after 85 hours, and the SpO2 one after 109 hours.

The lowest power consumption was achieved for the TA node due to the low current consumption of the two sensors and the low duty cycle of the radio module. The operating time of this node type was evaluated only is simulation an exceeded 1.3 years.

B. Feasibility tests and results

The prototype system has been briefly deployed and tes at the Moinesţi Emergency Hospital, Bacău county, in Roma-nia. These tests included sampling data from various patien that have undergone through various surgical procedures and were discharged in hospital wards until full recovery. The purpose had been to evaluate the feasibility and the real-t data acqusition capabilities of our prototype.

During these second group of tests, each patient has been equiped with the sensors. The Gateway devices have been deployed within the ward of the patients and configured to access the Wi-Fi network of the hospital. A physician had t sampled the data using an Android based tablet and monit each patient for 10 to 15 minutes. The system performed w supplying the desired data with no error to the physicians.

TABLE II
SENSOR NODES STATES AND AVERAGE POWER CONSUMPTION

State	Description	Power	Time slice
	ECG node		
LP	Low Power: MCU - processing core and main clock, radio module: disabled, communication module (UART), ECG module: enabled;	76 mW	95.8%
SMP	Sample: MCU, ECG module enabled; radio module disabled	92 mW	3%
TX	Transmit data: MCU, ECG module and radio module enabled	106 mW	1.2%
	SpO2 node		
LP	Low Power: MCU and radio module disabled, UART and SpO2 module enabled	59 mW	98%
SMP	Sample: MCU and SpO2 module enabled; radio module disabled	79 mW	1.5%
TX	Transmit data: MCU, SpO2 module and radio module enabled	92 mW	0.5%
	Temperature/acceleration node		
LP	Low Power: MCU, communication modules (I2C, 1Wire), temperature sensor and radio module disabled; accelerometer enabled	0.4 mW	96.9%
SMPA	Acceleration sample: MCU, accelerometer enabled; temperature sensor and radio module disabled	26 mW	2.9%
SMPT	Temperature sample: MCU, accelerometer, temperature sensor enabled; radio module disabled	31 mW	<0.1%
TX	Transmit data: MCU, accelerometer and radio module enabled; temperature sensor disabled	69 mW	<0.1%

Moreover, we did not notice any considerable delays between data acquisition and data visualisation on the mobile device.

V. CONCLUSIONS AND FURTHER DEVELOPMENT

In this paper we present a prototype system for remote patient monitoring. The main purpose of this prototype is to fill the gap in monitoring a patient's vital signs between ICU (after having undergone surgical procedures or other emergency treatments) and the actual hospital discharge. While ward evaluation may not necessarily require continuous physiological parameters assessment, patient relapse is not uncommon. The system we described may prove to be extremely useful in preventing such relapses since it allows the medical personnel to timely and accurately evaluate the ward patients.

The system is comprised of low-power profile sensors that communicate wirelessly with a Gateway type device implemented using the Raspberry Pi B+ board. The overall functionality of these devices is made available through a RESTful based Web Service. Therefore, this present prototype may be easily integrated within more complex IoT based medical applications simply by calling the RESTful API we have designed.

At present, we are actively involved in developing an iOS based application that would be a replica of the Android app and extend the functionality to iPad devices. The next steps involve improving the overall security of the prototype and automatic sensor discovery.

ACKNOWLEDGMENT

This work was supported by the Structural Funds - Sectoral Operational Programme "Increase of Economic Competitiveness" as the RDI component of project SMIS ID 49786.

REFERENCES

[1] V. Custodio, F. J. Herrera, G. Lpez, and J. I. Moreno, "A review on architectures and communications technologies for wearable heal… monitoring systems," *Sensors*, vol. 12, no. 10, p. 13907, 2012.
[2] V. Pigadas, C. Doukas, V. P. Plagianakos, and I. Maglogiannis, "Enabli… constant monitoring of chronic patient using android smart phones," … *Proceedings of the 4th International Conference on PErvasive Techno… gies Related to Assistive Environments*, ser. PETRA '11. New York, NY, USA: ACM, 2011, pp. 63:1–63:2.
[3] D. Vassis, P. Belsis, C. Skourlas, and G. Pantziou, "A pervasive archite… tural framework for providing remote medical treatment," in *Proceed… of the 1st International Conference on PErvasive Technologies Relate… to Assistive Environments*, ser. PETRA '08. New York, NY, USA: ACM, 2008, pp. 23:1–23:8.
[4] J. Neuhaeuser and L. D'Angelo, "Collecting and distributing wearable sensor data: An embedded personal area network to local area network gateway server," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, July 2013, pp. 4650–4653.
[5] R. Lupu, A. Stan, and F. Ungureanu, "Patient monitoring: Wearable device for patient monitoring," in *Advances in Electrical Engineering and Computational Science*, ser. Lecture Notes in Electrical Engineer… S.-I. Ao and L. Gelman, Eds. Springer Netherlands, 2009, vol. 39, pp. 659–668.
[6] J. Ko, J. H. Lim, Y. Chen, R. Musvaloiu-E, A. Terzis, G. M. Masson, T. Gao, W. Destler, L. Selavo, and R. P. Dutton, "Medisn: Medical emergency detection in sensor networks," *ACM Trans. Embed. Comp… Syst.*, vol. 10, no. 1, pp. 11:1–11:29, Aug. 2010.
[7] S. Singh, J. Jayasuriya, C. Zhou, and M. Motani, "A restful web networking framework for vital sign monitoring," in *Communications (ICC), 2015 IEEE International Conference on*, June 2015, pp. 524–529.
[8] M. Lan, L. Samy, N. Alshurafa, M.-K. Suh, H. Ghasemzadeh, A. Macabasco-O'Connell, and M. Sarrafzadeh, "Wanda: An end-to-end remote health monitoring and analytics system for heart failure patie… in *Proceedings of the Conference on Wireless Health*, ser. WH '12. New York, NY, USA: ACM, 2012, pp. 9:1–9:8.
[9] M. Fengou, G. Mantas, D. Lymberopoulos, N. Komninos, S. Fengos, and N. Lazarou, "A new framework architecture for next generation e-health services," *Biomedical and Health Informatics, IEEE Journal of*, vol. 17, no. 1, pp. 9–18, Jan 2013.
[10] H. C. Tung, K. F. Tsang, K. L. Lam, H. Y. Tung, B. Y. S. Li, L. F. Yeung, K. T. Ko, W. H. Lau, and V. Rakocevic, "A mobility enabled inpatient monitoring system using a zigbee medical sensor network," *Sensors*, vol. 14, no. 2, p. 2397, 2014.
[11] R. Ramirez-Ramirez, M. Cosio-Leon, D. Ojeda-Carreno, M. Vazquez-Briseno, and J. Nieto-Hipolito, "Designing a gateway ieee1451-hl7 for e-health telemonitoring services," in *Computing Systems and Telema… (ICCSAT), 2015 International Conference on*, Oct 2015, pp. 1–6.
[12] J. Cruz, D. Brooks, and A. Marques, "Home telemonitoring in copd: A systematic review of methodologies and patients adherence," *Inter… national Journal of Medical Informatics*, vol. 83, no. 4, pp. 249 – 263, 2014.
[13] M. Bazzani, D. Conzon, A. Scalera, M. Spirito, and C. Trainito, "Enabling the iot paradigm in e-health solutions through the virtus middleware," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, June 2012, pp. 1954–1959.
[14] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, and K. Pister, "Openwsn: a standards-based low-power wireless development environment," *Transactions on Emerging Telec… munications Technologies*, vol. 23, no. 5, pp. 480–493, 2012.