Conference Paper

Feasibility of Gateway-Less IoT E-Health Applications

Carlos Pereira
Diana Guimarães
João Mesquita
Frederico Santos
Luís Almeida*
Ana Aguiar

*CISTER Research Centre

CISTER-TR-181104

2018/06/18

Feasibility of Gateway-Less IoT E-Health Applications

Carlos Pereira, Diana Guimarães, João Mesquita, Frederico Santos, Luís Almeida*, Ana Aguiar

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: dlguim@fe.up.pt, lda@fe.up.pt

http://www.cister.isep.ipp.pt

Abstract

Machine-to-Machine (M2M) communications are a key enabler of Internet of Things (IoT) applications. One domain with growing interest in M2M/IoT is e-health, either for self-monitoring, home monitoring, or hospital systems. However, current sensing devices in this domain rely on short-range communication protocols that require a gateway (GW) for Internet connection. Smartphones have been proposed as GWs in mobile M2M communications due to their enhanced connectivity and sensing capabilities. However, the GW functionality impacts on the smartphone usability, causing undesirable battery depletion and the smartphone itself increases the overall cost of e- health solutions. In this work, we propose converging e-health devices and Wi-Fi towards direct Internet access through the existing Wi-Fi infrastructure and by-passing current GWs. We use recent low-cost ultra low-power Wi-Fi modules and feature them with M2M capabilities supporting their integration in an interoperable e-health framework. We present results on end-to-end latency and power requirements within a concrete e-health use case that show the feasibility of the proposed GW-less solution.

Feasibility of Gateway-less IoT E-health Applications

Carlos Pereira*‡, Diana Guimarães*‡, João Mesquita‡, Frederico Santos*§, Luis Almeida*†‡, Ana Aguiar*‡
*IT - Instituto de Telecomunicações
†CISTER - Centro de Inv. em Sistemas de Tempo-Real
‡FEUP - Universidade do Porto, Porto, Portugal
Email: {dee12014, dlguim, up201305568, lda, anaa}@fe.up.pt
§ ISEC - Instituto Politécnico de Coimbra, Coimbra, Portugal
Email: fred@isec.pt

***Abstract*—Machine-to-Machine (M2M) communications are a key enabler of Internet of Things (IoT) applications. One domain with growing interest in M2M/IoT is e-health, either for self-monitoring, home monitoring, or hospital systems. However, current sensing devices in this domain rely on short-range communication protocols that require a gateway (GW) for Internet connection. Smartphones have been proposed as GWs in mobile M2M communications due to their enhanced connectivity and sensing capabilities. However, the GW functionality impacts on the smartphone usability, causing undesirable battery depletion and the smartphone itself increases the overall cost of e-health solutions. In this work, we propose converging e-health devices and Wi-Fi towards direct Internet access through the existing Wi-Fi infrastructure and by-passing current GWs. We use recent low-cost ultra low-power Wi-Fi modules and feature them with M2M capabilities supporting their integration in an interoperable e-health framework. We present results on end-to-end latency and power requirements within a concrete e-health use case that show the feasibility of the proposed GW-less solution.**

***Index Terms*—E-Health; Internet of Things (IoT); Machine-to-Machine (M2M) communications; Mobile devices; System performance; Wireless networks.**

I. INTRODUCTION

Machine-to-Machine (M2M) communications are leveraging the Internet-of-Things (IoT), making it suitable to diverse domains such as e-health, smart grids, or smart cities [1], [2], [3]. Particularly in e-health, patients are monitored and tracked using networked sensors that collect personal data and send it to medical or processing centers for monitoring of chronic conditions or for prophylactic reasons.

One situation in which deaths and serious adverse events that could be potentially prevented still occur is in emergency wards [4], [5]. This rose the awareness to the importance of continuous monitoring and fast response in wards as done in intensive care units [6], [7]. Currently, monitoring in wards can be done with wearable devices, e.g. wristbands, that collect vital signs like heart rate frequency and variability, blood pressure and body temperature. However, such devices use short-range communication protocols, typically Bluetooth, requiring a gateway (GW) for Internet connection. GWs are frequently implemented using smartphones for their usability and enhanced connectivity. On the other hand, the GW functionality increases power consumption on the smartphone, severely reducing its autonomy, and increases the cost of the overall e-health system.

This paper shows a first feasibility assessment of an interoperable e-health system with Wi-Fi-enabled wearable sensors taking advantage of recent low-cost ultra low-power ESP8266 Wi-Fi modules [8]. We consider a realistic scenario where heart rate is collected from patients that carry a wristband while inside an hospital for continuous ambulatory monitoring. The wristband would use an embedded ESP8266 module to send data through the Internet to an Electronic Health Record (EHR) service, providing information to medical personnel.

Bringing Internet to the sensors interface can also improve usability and openness of e-health systems. In fact, current efforts in e-health systems have led to disparate and rather specific vertical solutions [9], [10]. Conversely, we leverage an interoperable framework [11] that relies on two *de facto* standards, namely, oneM2M [12] for interconnecting devices and services and openEHR [13] for data semantics, storing, and making data available to the medical personnel. This work is also a contribution to that framework.

The remainder of the paper starts with the description of our motivating e-health use case in Section II and follows on with the framework architecture and implementation in Section III, the feasibility assessment with both a qualitative comparison and quantitative performance analysis, focusing on power requirements and end-to-end latency, in Section IV and ends with the conclusion in Section V.

II. SCENARIO

We envision a triage scenario at an emergency ward, where patients are given wristbands containing the usual color code

---

This work was funded by Instituto de Telecomunicações, ref. UID/EEA/50008/2013, and by project NanoSTIMA – Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics, NORTE-01-0145-FEDER-000016, supported by Norte Portugal Regional Operational Programme (NORTE 2020), through Portugal 2020 and the European Regional Development Fund.

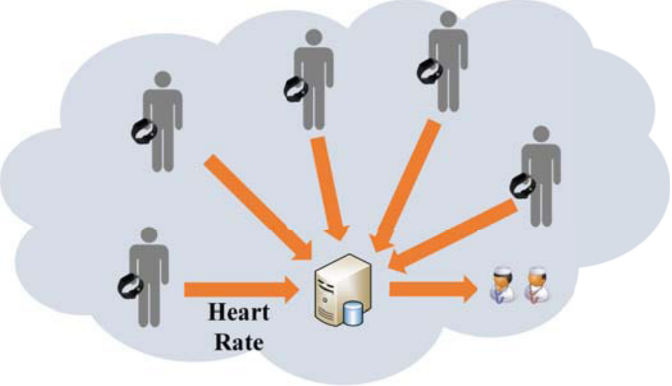

Fig. 1. Application e-health scenario with active wristbands sending vital signs data to an EHR following a publish-subscribe model.

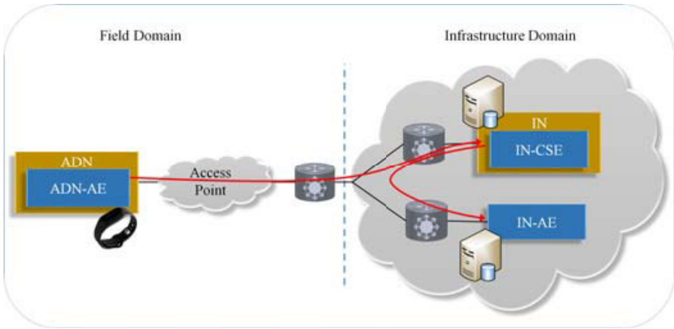

Fig. 2. System design using standardized oneM2M entities. Red arrows represent the user data flow.

indicating the severity of their condition [14]. However, instead of common passive wristbands, these ones are active, containing an optical sensor and a Wi-Fi interface that allows sending heart rate (and/or other vital signs) to an EHR without movement restrictions.

First, the triage personnel accesses the patient's EHR and registers the wristband the patient will use. This will signal the wristband to start collecting and transmitting data, and the EHR to subscribe to this data using a publish-subscribe model (Fig. 1). This allows the EHR to start receiving the wristband data either for storage in the patient's record or for online remote monitoring by medical personnel. When the patient leaves the hospital, the medical personnel signals the stopping of data collection in the EHR and recovers the wristband from the patient.

While the wristband is on, it continuously uploads sensor data through the Wi-Fi hospital network. Depending on the allowed delay for receiving the patients' data, transmissions can range from once every *x* minutes to once per second, or even faster. If the wristband loses Wi-Fi connectivity temporarily, it can buffer data for later transmission.

III. GATEWAY-LESS INTEROPERABLE E-HEALTH FRAMEWORK

The oneM2M standard is a key enabler of our interoperable framework. It supports both publish-subscribe and request-response communication models but we will use the former, only, given its higher efficiency for sensing and remote monitoring, while providing more flexibility and scalability.

A oneM2M system is divided in two domains: a *Field* containing Application Dedicated Nodes (ADN) that run Application Entities (ADN-AE) implementing the services logic; and an *Infrastructure* containing Infrastructure Nodes (IN) holding either a Common Service Entity (IN-CSE) or Application Entities (IN-AE) that are registered with the CSE. Both ADN-AE and IN-AE communicate over the Mca reference point with the IN-CSE.

In oneM2M, information is represented by resources, following a RESTful architecture style [15], which are mutable over time and uniquely addressable using a Universal Resource Identifier (URI). The IN-CSE hosts resources in a hierarchical tree structure, where information is maintained. Subscriptions are represented and can be operated upon as resources in the resource tree.

Fig. 2 shows how we support the previous use case scenario with a oneM2M standard-based ecosystem. The ADN-AE runs in the ESP8266 module and manages the wristband's internal sensor and data transmissions. For example, it can decide autonomously to transmit the data periodically, or adapting transmissions to save energy, or it can also receive remote actuation commands to set a given operational mode. The ADN-AE can also use buffers to hold data during periods without connectivity. The data is sent to the IN-CSE that acts as a broker in the communication model.

The IN-AE is a oneM2M interface to the openEHR service. Every time that there is an assignment of a wristband to a patient's EHR, the IN-AE makes a subscription to the respective resource in the IN-CSE, after which it starts to receive notifications of the data published by the device. For efficiency reasons, the ADN-AE is initially set to transmit only after receiving a command. The IN-CSE also controls every resource access, and TLS or DTLS can be used to ensure privacy and security.

The openEHR is a non-proprietary standard architecture aiming at interoperability and openness in e-health concerning data, models, and application programming interfaces for systems and components. It standardizes the EHR architecture following a multi-level modeling approach, separating information from knowledge. The specifications define a health information reference model, archetypes, and a query language [13]. Archetypes are the models for capturing the health information and usually specify a single clinical concept. The IN-AE receives the heart rate data from the wristbands and converts this information so that it can be interpreted and stored in the EHR.

Fig. 3 shows a normal message sequence between the entities. Initially, as the wristband is connected for the first time, the ADN-AE registers itself at the IN-CSE, at /~/in-cse/dartes in the figure, and creates the HR container where the data will be stored inside content instance resources by means of publications. The ADN-AE also creates a subscription under a

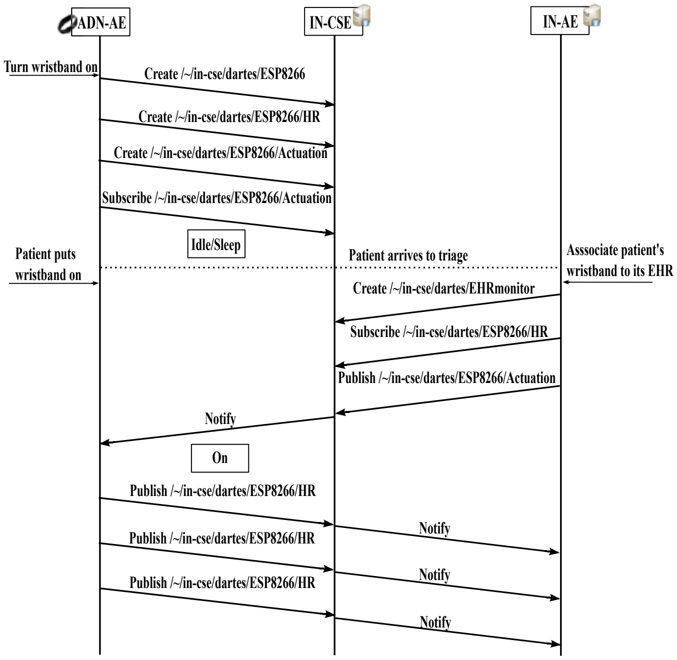

Fig. 3. Messages exchanged during the first time the device is turned on and when the device is associated to a patient.

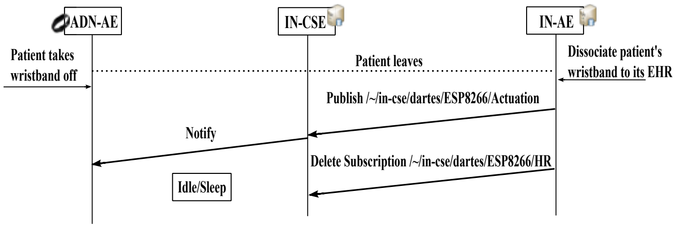

Fig. 4. Messages exchanged when the patient takes off the wristband and leaves the hospital.

specific container, called "Actuation" in the figure, to receive commands published by the IN-AE. The IN-AE subscribes at the IN-CSE all the information in which it is interested. The subscriptions are made directly to the HR container from the patient's ADN-AE which makes a mapping between the specific patient's wristband and the patient's EHR.

Fig. 4 shows the messages exchanged when the patient leaves the hospital. At that moment, the medical personnel collects the wristband from the patient, and signals in the openEHR the dissociation from the wristband. The IN-AE publishes a command to the ADN-AE to make it stop sending data and deletes its subscription from the IN-CSE.

IV. FEASIBILITY OF M2M-ENABLED E-HEALTH WEARABLES

We assess the feasibility of our proposal for Wi-Fi-based M2M-enabled wearable devices for e-health use cases considering both a qualitative analysis and an experimental characterization. The former aims at discussing the proposed system against the current approach of using smartphones as GWs to the wearable devices, connected with a short-range communication protocol. The latter aims at verifying in a lab environment whether the ESP8266 modules performance is sufficient for the desired use cases.

A. Qualitative analysis

To compare our GW-less with the common smartphone-based approaches we consider four important qualitative dimensions: capabilities, performance, ease of use, and cost.

Concerning **capabilities**, the features of current smartphones surpass by far most constrained devices, be it in terms of connectivity, memory, processing, and user interface. This grants smartphone-based architectures an advantage. Nevertheless, despite the scarce resources, the ESP8266 modules already present a reasonable processing capacity, with its 32-bit processor @ 80 (160) MHz, 36KB of on-chip SRAM, and up to 16MB of external SPI Flash memory [8]. This gives it enough power to handle the Wi-Fi interface with IEEE 802.11b/g/n/e/i compatibility (including transmission power and receiver sensitivity) while offering a TCP/IP stack with WPA and WPA2 security. These modules have several sleep modes that allow different trade-offs between time to wake-up and energy consumption, with all modes maintaining a clock running for wake-up control. Thus, despite its lower capabilities, the ESP8266 module can run M2M middleware with very low energy demands.

Concerning **performance**, note that smartphones' ubiquity is mainly powered by the use of cellular networks. These are known to have long and inefficient transitions between the different states of the network interface [16], adding undesired latency [17]. Moreover, additional delay can be introduced by the GW functionality, protocol conversion, M2M middleware overhead, etc. In M2M scenarios, the common recurrent transmissions pattern can impact negatively on the smartphones' usability, introducing undesirable battery depletion [18], [19]. In turn, the performance of the ESP8266 modules in M2M scenarios is still unknown, despite the promise for ultra low-power operation. This is assessed in the following section.

In what concerns **ease of use** the smartphone-based approach implies one additional device beyond the wearable devices, requiring extra configurations and extra weight. In this aspect, the standalone wristband has a clear advantage.

In terms of **cost**, the smartphone-based approach is significantly more expensive than a potential wristband wearable based on the ESP8266 module, giving another strong advantage to this option.

B. Experimental evaluation

1) Setup: The experiments were performed in our department, with the broker (IN-CSE) installed on a dedicated server running CentOS 6.9, on a *Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz* with 8GB of RAM, and with a wired connection of 100Mb/s with the openEHR subscriber client (IN-AE) installed in a machine running Ubuntu 16.04 LTS, on a *Intel(R) Core(TM) i7 - 4700HQ CPU @ 2.40GHz* with 8GB of RAM.

The ESP8266 publisher client (ADN-AE) is connected wirelessly with the broker via an access point ASUS RT-AC87U dual-band AC2400[1], using the default 802.11 protocol, and runs a FreeRTOS-based ESP8266 software framework[2]. This specific hardware version comprises a Tensilica 32-bit CPU @ 80MHz, 36KB of on-chip SRAM, and 4MB of external SPI Flash memory in this setup. The access point has a wired connection of 100Mb/s with the broker.

We installed the OM2M broker[3] as reference implementation for the oneM2M standard. Between the publisher and the broker, we used the CoAP protocol over UDP with non-confirmable messages. We used HTTP protocol between the broker and the subscriber. Publications and notifications are made with a POST method. A short token is added in every message as access control.

The ESP8266 publisher client was implemented in C, and the openEHR subscriber client was implemented in Java. Both the publisher and subscriber clients communicate with the broker using the Mca reference point.

2) Methodology: We measured the **current** consumed by the module using the Monsoon power monitor[4]. We tested three modes of operation: Normal mode, no sleeping involved; Modem-sleep mode, which switches off the Wi-Fi interface except the part needed to keep connectivity; and Light-sleep mode, which, in addition to the Wi-Fi interface, also switches off the CPU and its crystal-based clock, keeping just an RC-based clock working.

We focus on the impact of transmissions and neglect the sensor-related circuitry. We created representative sensor data that is stored in internal memory. Each heart rate measurement is published creating a new content instance resource in an individual container resource already existing in the broker representing a single ESP8266. The content instance's payload is marshaled in a JSON format.

For measuring the **end-to-end latency** we used global timestamps after synchronizing the publisher and subscriber with the same Network Time Protocol (NTP) [20] server. NTP time updates were done every 20s for higher precision. We define the end-to-end latency as the difference between the timestamps acquired at application-level when the notifications arrived at the subscriber and when the corresponding publications were issued by the publisher.

We considered 1s and 10s transmission periods, with a payload of 85B and 850B of CoAP data, respectively. The former represents a single heart rate measurement and the latter, 10 measurements sent together. The 85B should be similar to the size of data produced by common heart rate monitors [21].

Finally, to improve channel conditions we placed the ESP8266 module 1m away from the wireless access point. The access point was configured with delivery traffic indication map (DTIM) of 3 and the beaconing interval of 100ms. The measurements were sequential and we carried out and logged 100 publications in each configuration.

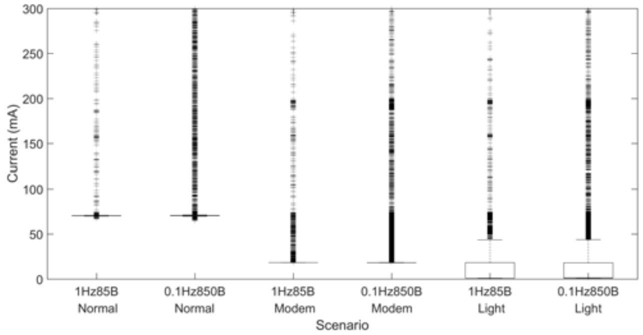

Fig. 5. Current measured for each scenario.

TABLE I
AVERAGE CURRENT AND EXPECTED BATTERY LIFETIME FOR EACH SCENARIO.

Scenario	Average Current (mA)	Battery lifetime (h)
1Hz85B Normal	70.9	12.7
0.1Hz850B Normal	70.8	12.7
1Hz85B Modem	27.0	33.4
0.1Hz850B Modem	23.6	38.2
1Hz85B Light	22.1	40.7
0.1Hz850B Light	15.3	58.7

3) Power Requirements Results: Fig. 5 shows the distribution of the current measured for each scenario. The Y-axis was truncated for better visualization. Table I presents the average current measured in each mode. We can see a significant reduction between the Normal and both sleep modes, specially in Light-sleep, as expected. Both sleep modes show lower current consumption when transmitting with longer period due to the overhead of enabling/disabling hardware submodules. The difference is larger in Light-sleep since more components are switched off and on.

Table I also shows estimated battery lifetime, considered as the time to reach 10% of battery capacity. With a battery capacity of 1000mAh and linear battery depletion, all sleep modes and periods lead to battery lifetimes above 24h, which seems adequate for the presented use case, where the wristbands would require a daily charge cycle.

4) Latency Results: Fig. 6 shows the evolution of the end-to-end latency measurements for all three modes and 1s period. We observe a relatively strong clock drift in the Light-sleep mode, with a rate of 0.73%, since the only clock that remains active is the RTC with low precision. It is corrected every 20s by NTP leading to lower latency values, that are more accurate. The growing values that follow are apparent, caused by the drift.

Fig. 7 shows the distribution of the end-to-end latency in all scenarios, including drift correction. When the transmission frequency is 1Hz the average end-to-end latency is 23.1ms, 27.4ms and 29.1ms for the Normal, Modem-sleep, and Light-sleep modes, respectively. With a transmission frequency of 0.1Hz these values increased marginally to 23.8ms, 27.6ms and 33.0ms, respectively. As expected, the two sleep modes

---

[1] https://www.asus.com/pt/Networking/RTAC87U/

[2] https://github.com/espressif/ESP8266_RTOS_SDK

[3] http://www.eclipse.org/om2m/

[4] https://www.msoon.com/

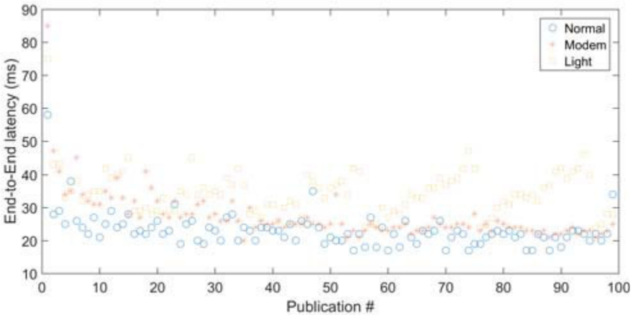

Fig. 6. End-to-end latency measured for publications of 85B CoAP data every second, showing clock drift in the Light-sleep mode.

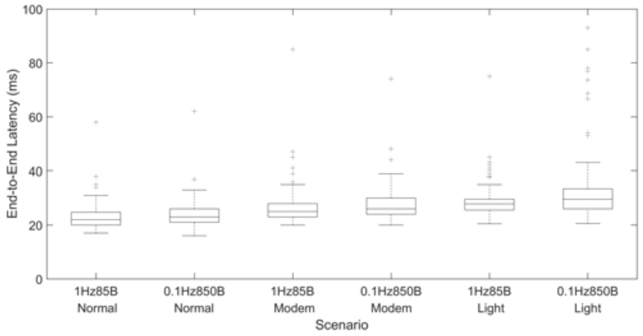

Fig. 7. End-to-end latency measured for each scenario.

introduce higher latency due to the switching on and off of the associated hardware modules.

The differences in latency when using a period of 1s or 10s are only significant in the Light-sleep mode, which is explained by the larger number of hardware modules it switches off.

For a real system, we would still need to add the sensor reading latency and the subscriber (IN-AE) to EHR latency. We expect not more than a few ms for both sensor and EHR, as long as it is placed in the same machine as the subscriber. Thus, an end-to-end latency from sensor to EHR below 50ms is feasible in building networks, even for the stronger sleep mode tested.

V. CONCLUSIONS AND FUTURE WORK

The rise of IoT e-health applications has been leveraged by M2M communications as key enabler for interoperable and efficient solutions. Smartphones have been used as M2M GWs for simple wearable devices but they introduce additional costs and battery life limitations that can hinder efficient operation of IoT scenarios. We propose using wearable sensors with new low-cost ultra low-power Wi-Fi modules, such as the ESP8266, that can improve efficiency and usability bypassing smartphones. In this paper we presented a realistic IoT e-health use case relying on an interoperable framework using oneM2M and openEHR standards together with Wi-Fi-enabled wearable sensors. We implemented it as proof-of-concept using ESP8266 modules and showed experimental evidence of the system feasibility.

As future work we will expand the experimental characterization of the Wi-Fi-enabled sensors varying other system parameters and trying a Deep-sleep mode that maybe bring strong advantages in scenarios of very low transmission frequency.

REFERENCES

[1] K. Chen, “Machine-to-machine communications for healthcare,” *Journal of Computing Science and Engineering*, vol. 6, no. 2, pp. 119–126, June 2012.

[2] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, “A survey on smart grid communication infrastructures: Motivations, requirements and challenges,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 5–20, Jan. 2013.

[3] A. Caragliu, C. Del Bo, and P. Nijkamp, “Smart cities in Europe,” in *Proceeding of the 3rd Central European Conference in Regional Science*, 2009, pp. 49–59.

[4] M. Cardona-Morrell, M. Prgomet, R. M. Turner, M. Nicholson, and K. Hillman, “Effectiveness of continuous or intermittent vital signs monitoring in preventing adverse events on general wards: a systematic review and meta-analysis,” *International Journal of Clinical Practice*, vol. 70, no. 10, pp. 806–824, Oct. 2016.

[5] L. J. Donaldson, S. S. Panesar, and A. Darzi, “Patient-safety-related hospital deaths in england: Thematic analysis of incidents reported to a national database, 2010–2012,” *PLOS Medicine*, vol. 11, no. 6, pp. 1–8, June 2014.

[6] D. Jones, I. Mitchell, K. Hillman, and D. Story, “Defining clinical deterioration,” *Resuscitation*, vol. 84, no. 8, pp. 1029 – 1034, Aug. 2013.

[7] M. Elliott and A. Coventry, “Critical care: the eight vital signs of patient monitoring,” *British Journal of Nursing*, vol. 21, no. 10, pp. 621–625, Aug. 2012.

[8] Espressif, “ESP8266 Technical Reference,” [Online]. Available: http://espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf, [Last access: 15-February-2018].

[9] G. L. Kreps and L. Neuhauser, “New directions in ehealth communication: Opportunities and challenges,” *Patient Education and Counseling*, vol. 78, no. 3, pp. 329 – 336, Mar. 2010.

[10] A. Dogac, “Interoperability in ehealth systems,” *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 2026–2027, Aug. 2012.

[11] C. Pereira, S. Frade, P. Brandao, R. Correia, and A. Aguiar, “Integrating data and network standards into an interoperable e-health solution,” in *2014 IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2014, pp. 99–104.

[12] oneM2M, “oneM2M Standardization,” [Online]. Available: http://www.onem2m.org, [Last access: 15-February-2018].

[13] OpenEHR, “openEHR,” [Online]. Available: http://www.openehr.org/, [Last access: 15-February-2018].

[14] I. Robertson-Steel, “Evolution of triage systems,” *Emergency Medicine Journal*, vol. 23, no. 2, pp. 154–155, Feb. 2006.

[15] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, University of California, Irvine, 2000.

[16] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, “A close examination of performance and power characteristics of 4g lte networks,” in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’12. ACM, June 2012, pp. 225–238.

[17] C. Pereira, A. Pinto, D. Ferreira, and A. Aguiar, “Experimental characterization of mobile iot application latency,” *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1082–1094, Aug. 2017.

[18] M. J. Morón, R. Luque, and E. Casilari, “On the capability of smartphones to perform as communication gateways in medical wireless personal area networks,” *Sensors*, vol. 14, no. 1, pp. 575–594, Jan. 2014.

[19] C. Pereira, J. Rodrigues, A. Pinto, P. Rocha, F. Santiago, J. Sousa, and A. Aguiar, “Smartphones as m2m gateways in smart cities iot applications,” in *2016 23rd International Conference on Telecommunications (ICT)*, May 2016, pp. 1–7.

[20] D. Mills, “Network Time Protocol (version 3): Specification, implementation and analysis. IETF RFC 1305,” [Online]. Available: https://tools.ietf.org/html/rfc1305, [Last access: 15-February-2018].

[21] Zephyr, “Zephyr HxM BT,” [Online]. Available: http://zephyranywhere.com/products/hxm-bluetooth-heart-rate-monitor/, [Last access: 15-February-2018].