Smart Data Synchronization in m-Health Monitoring Applications

Abdelghani Benharref
Faculty of Engineering and Information Sciences
University of Wollongong in Dubai
Dubai, United Arab Emirates

Mohamed Adel Serhani
College of Information Technology
United Arab Emirates University
Al Ain, United Arab Emirates

Rabeb Mizouni
Khalifa University of Science and Applied Research
Abu Dhabi, United Arab Emirates

Abstract-Nowadays, mobile applications/devices have become the trends, especially, when they were gradually shifted from basic communication services to supporting more sophisticated service provisioning. Mobile applications are usually very light, are nowadays likely to be often connected to the Internet, and can be used quite easily. However, these applications exhibit some challenges related to limited resources they have access to, including limited processing power, memory, storage size, battery power, and intermittent network connection. In fact, these considerations have to be taken seriously into consideration when developing mobile applications especially if those applications will be used for critical services, for example, to collect and report vital health data over a long period of time. In this paper, we study the use of mobile applications for monitoring patient's vital. Mobile devices, through an application, are connected to body-strapped biosensors to collect and synchronize these parameters with information systems. This synchronization should be done in such a way that the cost of synchronization is kept low and urgent readings are delivered as soon as possible. To optimize the synchronization process and reduce its cost, we propose and validate cost-oriented algorithms. A case study is developed to illustrate the applicability and effectiveness of our innovative techniques in making continuous monitoring an efficient process.

I. INTRODUCTION

Mobile applications are considered as a potential asset to contribute to the solution of an enormous worldwide challenge such as healthcare resources scarcity, mainly for patients with chronic diseases. In this context, it has been shown that keeping chronic disease patients out of hospitals as much as possible is cost-effective solution beneficial to all stakeholders including patients, their families, and health care professionals. Patients morally improve their wellbeing when spending most of their time with relatives, thus healthcare systems' resources were released and allocated to more critical emergency situations. However, outpatients do not benefit from the same thorough monitoring that inpatients have access to while in hospitals, which at some point might present a threat to their life. Mobile applications combined with wireless body sensors can be used as a compromised solution to remotely monitor outpatients while they are home. Wearable biosensors [1] collect health data, relay it to a mobile application deployed on smart phones, and then communicate these data to Hospital Information Systems (HIS) where around the clock surveillance activities are taking place.

Sensory data have to be communicated to patients and health professionals to help taking appropriate and timely decisions whenever the patient under monitoring encounters a severe health situation. Patients might need to be monitored continuously and for a long period of time depending on the severity of their health conditions. Such a process presents a big defy to mobile applications, and their underlying mobile devices. Heavy and continuous communication with biosensors generates a huge amount of data that makes storage management and synchronization with backend systems very costly in terms of network cost, storage space, and battery.

Due to the above-mentioned challenges, development of mobile applications for m-Health requires specific considerations all along the development and production life cycles. Available resources have to be used with great care to reduce the Total Cost of Ownership (TCO). These resources include screen space, battery, processor, memory, and network. We will limit the scope of this work to the battery and network components. Investigation on the other parameters is left to future work.

As mentioned previously, continuous outpatient monitoring over a long period of time is likely to generate a huge amount of data. For example, this size might be around few dozens of Mega bytes per hour for a representative ElectroCardioGram (ECG) signal [2]. If monitoring spans over a full working day, the size of collected data is in terms of hundreds of Mega bytes. These data have to be synchronized with HIS where professionals can have a real time view of patient's vital readings. Synchronizing all these data over 3G or while away from a charging station is likely to generate a high network cost and will quickly drain the battery. For such monitoring to be practical, data synchronization has to be smart enough to reduce the cost and offer a better view of the patient's health condition. The decision on what to synchronize, when to synchronize, and how to synchronize has to be smartly taken, on the fly, by the mobile application. This is dynamically done in order to give professionals sufficient, and recent data they need to take supported decisions but, at the same time, minimize the volume of data to synchronize mainly with low-bandwidth and/or pay-per-usage connections (e.g. Bluetooth, 3G) and low battery level.

The main objective of this work is to develop a smart and adaptive synchronization approach that can be implemented by m-Health applications. These applications will auto adjust dynamically to changes in the environment and continuously measure the battery level, required battery, and network status to make optimized data synchronization decisions. We develop and assess algorithms to evaluate the cost of network use and its ability of handling data synchronization in different situations, including network availability, the current network bandwidth, the urgency of collected data, and the status of battery level. These algorithms are embedded into the mobile application to decide on the fly between synchronizing all the data, only critical data, or store data momentarily locally on the mobile. The study of those algorithms will have three main stages:

- A thorough theoretical and analytical study of factors impacting the cost of data synchronization.
- Definition of a cost function to estimate the cost of each data synchronization operation.
- A pragmatic approach will be adopted in order to evaluate the effectiveness of our low cost-oriented approach and the dynamic decision process.

The remaining of this paper is organized as follows: next section surveys the existing work and solutions on mobile health monitoring and data synchronization. Section III starts by introducing a thorough theoretical study of factors impacting the cost of data synchronization then it describes the main algorithms we have developed to optimize data synchronization. Section IV details the experimentations and results using different scenarios to evaluate the effectiveness of our low cost-oriented scheme. Finally, we conclude the paper in section V and we point to some future research investigations.

II. Related Work

During the last decade, patient monitoring using mobile applications gained a lot of importance and many projects have been conducted to prove their feasibility. Each of them used/targeted a certain technology and tried to solve a specific problem. As an example, the authors in [3] presented a tool for the coordination of different actors (patient, emergency center, ambulances, and hospital) involved in urban medical emergency management based on wireless mobile technology. Service-Oriented Multi-Agent Systems (SOMAS) for Agent technology has been used to implement the application.

Airmed-Cardio [4] is a GSM-based platform targeting patients with chronic heart diseases who are in stable condition. The system objective is to provide attractive contact between patients and health-care agents directly involved in their care, and complete specifically defined protocols for follow-up and monitoring.

VirtualCRPs [5] (Cardiac Rehabilitation Program) is a web-based application used to deliver relevant information to cardiac patients in remote rural communities. Among the other services, the VirtualCRPs provides chat session with professionals and allows a self-reported data capture.

DITIS [6] is a collaborative medical team for home healthcare of cancer patients. It supports the dynamic creation, management and coordination of virtual medical teams, for the continuous treatment of the patient at home, and if needed for periodic visits to places of specialised treatment and back home.

Finally, MobiHealth [7] tried to bring together the technologies of Body Area Networks (BANs), wireless broadband communications, and wearable devices to provide mobile healthcare services for patients and health professionals. For patients, these technologies enable remote care services such as management of chronic conditions and detection of health emergencies. For health professionals, the technology offers access to information and communication services from a mobile device, thus enabling mobility for the individual professional and supporting the operation of distributed ‘virtual’ healthcare teams.

All these systems share a common interest, which is home delivering of health care to patients. Even though each of them has answered to some extent the need of particular patients, all were concerned mainly by proving the viability of such systems rather than the performance and the concrete TCO of such applications. Deploying these applications at large brought a new challenge to researchers: while these applications are feasible, they are resources’ consumers. The focus shifted from proving the viability to improving the performance.

In a recent work [8], MobiHealth was extended to take into consideration the context changes. They use learning to predict the QoS of the provided network and therefore, an intelligent mobile service can use the QoS prediction to proactively obtain best of best effort service.

The authors in [9] presented a task redistribution based adaptation middleware (MADE). In line with our approach, the authors consider battery and network as the context factor that drives this adaptation. However, since the task redistribution algorithm cannot be performed at the device itself, a middleware is responsible to do this task.

In terms of battery and network use in non–mHealth application, a few works have tackled this issue with Internet of Things (IoT). Some works are addressed in [10], [11], [12], [13], and [14]. However, wireless sensor devices in IoT and even in other domains (e.g meteorology) are quite different from those in m-Health:

- Most of IoT can sleep, hibernate, or standby for some time, until the next round of readings. M-Health sensors have to be up and running all the time so they don’t miss critical health deteriorations.
- Most of IoT sensors do not handle heavy amount of data as an ECG sensor for example.
- Data acquisition frequency and urgency of m-Health is usually higher than IoT.
- Passive RFID tags do not have batteries; they are however powered by the interrogation signal.

In addition to the aforementioned work on mobile health applications, many recent papers tackled the problem of data management for mobile devices ([15-17] [18]). In [15], the authors proposed a Synchronization Algorithms based on Message Digest (SAMD) algorithm based on message digest in order to facilitate data synchronization between a server-side database and a mobile database. The SAMD algorithm makes the images at the server-side database and the mobile database uses message digest tables to compare two images in order to select the rows needed for synchronization.

In [18], the authors introduced a unified messaging and data serving abstraction for mobile apps. The client side caching policies they are proposing reduce the network bandwidth use.

While the approaches discussed above were successful in performing a better data management, our purpose is to achieve a context driven data management. We look at the correlation between battery-network-urgency of data to guide the data synchronization.

III. SMART MOBILE APPS FOR PATIENT MONITORING

Mobile devices and applications are used to collect sensory data and allow anywhere and anytime monitoring of vital signs such as ECG, blood pressure, and body temperature. These applications should be sufficiently smart to tackle few challenges related to mobility, network disconnection, battery drainage, and limited processing capacity of mobile devices. Intelligent algorithms will be developed and executed on mobile applications to conciliate between these different challenges. Thus, minimize their impact on the efficiency and the accuracy of the overall monitoring activities.

In this section, we describe how mobile applications can adapt to network conditions/status and devices' battery level. This can be done by adding smartness to the mobile application to react/adapt to different network characteristics/profiles such as intermittency, limited bandwidth, disconnection, and high cost. We also, describe algorithms extending the mobile applications with features that handle data collection and processing in a smart and efficient way while optimizing the cost of battery drainage and network use by the mobile device.

A. Formal representation of smartness functions

In m-Health monitoring using mobile devices, the main challenges are the following:

1. Accurate readings: a reading of a vital sign is insignificant unless its accuracy is appropriate, that is, the error margin is within the thresholds approved by experts and relevant authorities.
2. Data synchronization: there are two data stores: one in the mobile device and one in the hospital backend. It is very important that, at any time, both stores reflect up-to-date and coherent data while minimizing the synchronization cost.

While the first challenge is out of the scope of this paper, we hereafter provide a formal description and formulation of the second challenge namely the data synchronization problematic involving mobile devices, sensors, and applications. Even though the ultimate goal is to have coherent stores of data, a sharp consideration has to be given to the overall cost of synchronization. This cost depends on a set of highly correlated criteria including, availability of network connection, capacity of that connection, type of network, size of data to transfer back and forth, and mobile device's battery level.

We formalize in this section the cost function of data synchronization. This function has two components: the network component and the device battery component. The network cost component consists of three sub-namely: network type, network bandwidth, and network usage charges. However, the device battery level is a linear measure of the battery that lies between 0 and 1, where values close to 0 represent an exhausted battery and values close to 1 represent a fully charged battery. There is high correlation between the needed battery, the network type, and its bandwidth. For example, the battery needed for a given synchronization operation is inversely proportional to the available network bandwidth; that is, when the bandwidth is low, the battery needed is higher and vice versa. Also, the battery needed depends on the type of network. For instance, a connection over a 3G network consumes more battery than a connection over a Wi-Fi network for the same synchronization operation.

- Let B a battery level of a mobile device, categorized by a percentage from 0% to 100%.
- Let N a network connecting the mobile application and the backend server.
- Let D a Data to be synchronized.

A Cost Function to synchronize data D (CFD) can be calculated based on the Battery Cost (BC) and the Network Cost (NC) as follows:

$$(1)\ CF_{\mathrm{D}} = f_{\mathrm{b}}\, BC + f_{\mathrm{n}}\, NC$$

Where f_b and f_n reflect the contributions of the battery level and the network in the overall cost function. Those contributions are mainly based on users' preferences. In fact, these contributions are estimated based on users' constraints on battery and/or networks. For example, if a user is always close to a charging station, either in their car, home, or office, the battery contribution to the cost function is very low and can be a fairly considered as zero. Similarly, if the user spends most of their days connected to a Wi-Fi network, the network cost can fairly be estimated to be zero.

1) Battery Cost (BC): The battery cost (BC) is affected by two main parameters:

- Battery required: this is the required battery to perform the synchronization of data D. This includes battery to prepare, send, receive, and process exchanged data.
- Battery level: this is the battery level of the device before synchronization of data D.

The battery level might have a higher or even the highest contribution into the overall battery cost. For example, if the cost of synchronization is low and the actual level of battery is 80%, the synchronization can be allowed. However, if the battery level is less than 10%, such synchronization, if allowed, will drain the battery leading to a power shortage. So, the contribution of battery level and battery cost are weighted, as defined in formula (2):

$$(2)\quad BC = w_{bl} * \frac{1}{BL} + w_{bc} * BC_{\mathrm{D}}$$

Where $w_{bl} + w_{bc} = 1$ and w_{bl} goes up when the battery level is low.

2) *Network Cost (NC):* The Network Cost NC can be decomposed into three components namely: network type, network bandwidth, and network charges.

1. The Network Type (NT) represents the underlying infrastructure capable of transferring data packets from a source to a destination. In most cases, NT is WiFi or 3G, but it can be Bluetooth or infrared as well.
2. The Network Bandwidth (NB) is estimated as the number of Megabytes/bits of data per second that the network can handle.
3. The Network charge (NC) represents the monetary cost of data transmission charged by a service provider. This is usually charged per megabytes unless the customer has an unlimited flat rate subscription.

Therefore, the NC function, for a given data D to be synchronized, can be formulated in (3) as follows:

$$(3)\ NC = Size(D) * (w_{nt}*NT + w_{nb}*\frac{1}{NB} + w_{nc}*NC)$$

Where $Size(D)$, w_{nt}, w_{nb}, and w_{nc} are the size of data D, the weight for the network type, the weight for the network bandwidth, and the weight for the network cost respectively. NC is inversely proportional to the network bandwidth as a low bandwidth usually incurs more effort to transfer data.

The overall cost can then be reasonably estimated by the following equation:

$$(4)\ CF_D = w_{bl} * \frac{1}{BL} + w_{bc} * BC_D + Size(D) * (w_{nt}*NT + w_{nb}*\frac{1}{NB} + w_{nc}*NC)$$

B. Smart use of network and battery

A network and communication infrastructure will allow connecting sensing devices, mobile devices, application servers, and visualization servers. However, the heterogeneity of network protocols, the unreliability and intermittency of mobile networks, and the limited bandwidth of body sensor networks will require some adjustments whenever a network is unavailable or of low bandwidth.

Two main network variables should be considered whenever data synchronization would take place. These parameters are highly related to the type of network used and are: network bandwidth and network cost. For Bluetooth and Wi-Fi, there are usually no fees. However, 3G and 4G LTE networks always apply some fees, which are quite high in some parts of the world. The network bandwidth parameter varies from one network to another: Bluetooth offers the lowest bandwidth, while Wi-Fi, 3G, and 4G LTE offer higher and varying bandwidth.

Body sensors might collect a tremendous amount of data. Storing and transferring all of these data to backend servers or mobile devices might be expensive. Also, some of the data might be inaccurate, out-dated, and/or without any added value. Consequently, a central need is to develop intelligent agents within mobile applications to decide on the following:

- Which data to retrieve and store?
- Which data should the mobile application decide to share/synchronize with backend server?
- How to assess the Cost of an operation in terms of battery and network costs?

In the next section, we develop required algorithms for smart data management, and a network and battery cost-driven decision-making approach to minimize the TCO of mobile monitoring solutions.

C. Algorithms description and analysis

We describe hereafter, the approach we propose to use to optimize data synchronization between a mobile application and HIS. **Fig. 1** illustrates the algorithms that get activated whenever there are new vital signs readings from a body sensor.

```
Data: data : readings ;                    // new readings to synchronize

procedure Main() begin
    Register listener for Wi-Fi network;
    Register listener for well-charged battery;
    Attach both listeners to the nonUrgentSynchronization procedure;

Label:  get new readings from sensor;
    determine the urgency of the new readings;
    if (not urgent) then
        save data locally;
        go to Label;
    else
        find available networks;
        if (No network available) then
            ask user to connect;
            wait for connection(s);
        assess costs of available networks;
        select network with least cost;
        synchronize data over that network;
        update synchronization costs statistics;

procedure nonUrgentSynchronization() begin
    if (battery level is low) then
        exit procedure;
    check cost of synchronisation;
    if (cost is low) then
        synchronize data;
        update synchronization costs statistics;
```

Fig. 1. Smart Synchronization of Vital Signs' Readings

The algorithm starts by registering two useful listeners with the operating system of the mobile application: one listener for Wi-Fi network and another listener for a well-charged battery. These listeners keep watching the assertion of their condition (Wi-Fi or battery) and activate the nonUrgentSynchronization procedure when the assertion(s) is(are) true. That is, when a Wi-Fi becomes available, the listener activates the procedure. The same happens with the second listener when the battery gets into a high charge level.

The mobile app starts acquiring readings from the sensor. When new readings are available, their urgency gets determined. This process is based on pattern recognition and rules developed by domains experts for each specific domain. For the ECG for example, there are a few reliable methods to recognize various heart beats patterns ([19], [20]).

The level of urgency of the new readings dictates what happens next. If this level is low, the mobile application saves the readings locally and looks for new readings. These non-synchronized data that are saved locally are going to be synchronized when any of the two listeners activates the nonUrgentSynchronization procedure. If the level of urgency is high (as per experts rules and patterns), the data should be synchronized immediately. The mobile application will automatically searches for all available networks. If there is no network, the user will be requested to connect to a network. If many networks are available, the cost of synchronization of each one has to be assessed using the cost function of section III.A. The synchronization then happens over the network with the least cost. Whenever synchronization is performed, important statistics on battery usage, data size, and network types are updated. These statistics will be used next time there is a need for synchronization cost estimation.

In case of Wi-Fi network, the network cost is mostly free, and the cost of battery usage is estimated based on previous statistics collected on Wi-Fi network battery consumption and the mobile manufacturer benchmarks. In case a cellular network is used, the network cost can be calculated per data unit as per service provider's billing and plans, and the battery cost is predicted using previous statistics on cellular network battery drainage and manufacturer benchmarks. Yet, if a Bluetooth network is used, the cost of network is free, but the battery cost is retrieved from pervious statistics of Bluetooth communications battery consumption and manufacturer benchmarks.

IV. Experimentation

To illustrate the effectiveness of the proposed cost function and smartness of synchronization, we have conducted a series of experimentations monitoring the ECG of a patient. To keep the focus on the data synchronization rather than acquiring data from the sensor, we have used a patient's data that have already been collected previously using Bioharness body sensor [21]. These data are replayed by a mobile application that has been developed to simulate the behavior of a sensor. Replaying data offers more flexibility for our experimentations as we have full control and tuning over readings' times, frequencies, synchronization time, availability of network, type of network, and available network bandwidth.

A. Test Bed

The test bed used for these experimentations is depicted in Fig. 2 and consists of: a database server, an application server for Web Services, a network traffic analyser, the main monitoring application that implements smartness, and another mobile application that replays vital signs readings.

The main mobile monitoring application offers, in the data synchronization section, three options as depicted in the lower section of **Fig. 3.** : 1) synchronize to backend, 2) synchronize from backend, and 3) synchronize both ways.

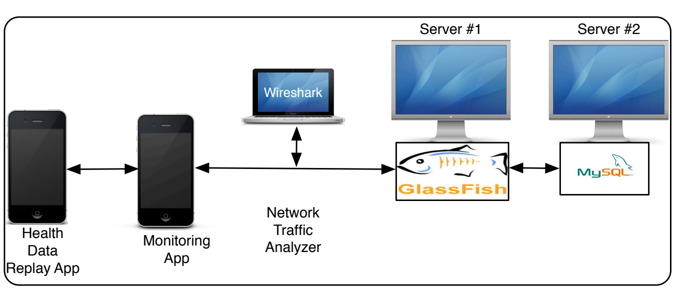

Fig. 2. Test bed

The upper section of the same figure shows a pre-synchronization summary after the user clicks the "Sync to backend" button. If the user clicks the "Cancel" button, the synchronization operation will be aborted. However, if the user still wants to go forward, the cost estimation procedures described in above sections will be invoked. In the example from **Fig. 3.** , the suggestions are presented in the alert view of

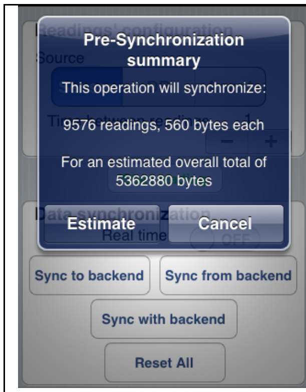

Fig. 3. Pre-synchronization summary

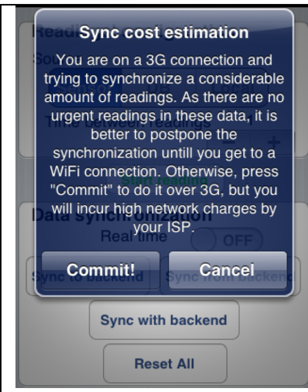

Fig. 4. Cost estimation function suggestion

B. Test Scenarios

To illustrate the robustness of the smart synchronization approach, we experimented different scenarios as follows:

- Non-smart individual synchronization: synchronization happens at real time over 3G, that is, whenever there is a reading, it is sent to the backend.
- Combined synchronization: readings are collected by the mobile application and then synchronized at once to the backend.
- Smart individual synchronization: real time synchronization over 3G is performed only if there is an urgent reading that needs to be sent to the backend as soon as possible. Non-urgent readings are synchronized when a Wi-Fi connection becomes available.

C. Results and discussions

The experiments we have conducted so far are focusing more on the network load than the battery. For the purpose of these experiments, values assigned to battery weights in equation 4 are entered by the user, BL is obtained from the operating system, and BCD is obtained from previous statistics. Moreover, weights for the network cost part are obtained from the operating system and previous statistics.

Our experimentation showed that the size of each ECG reading is around 560 Bytes and the size of a request to synchronize data is 320 Bytes. The size of a reading (i.e. 560 Bytes) includes the value of the reading itself, date of reading, time of reading, ID of reading as well as a few additional information required for the accurate handling of readings at the mobile application, HIS, and between them. In one of the author's data plan, the service provider charges \$0.14 for each megabyte downloaded or uploaded. **Fig. 5.** illustrates a summary of the results obtained from our ECG monitoring experimentations. The figure shows monthly and yearly synchronization costs for the three scenarios described above.

Fig. 5. shows that the highest cost is when all readings are individually synchronized over 3G. In fact, in such synchronization, the overhead represents more than the size of the reading being synchronized, generating then a higher cost. If it happens that readings are not urgent to an extent that they don't need real time synchronization, they are then grouped in one synchronization operation, which reduces the impact of the overhead. The smart synchronization represents by far the lowest network cost as only urgent readings are synchronized over 3G but non-urgent readings are kept on the mobile application until a Wi-Fi connection is available.

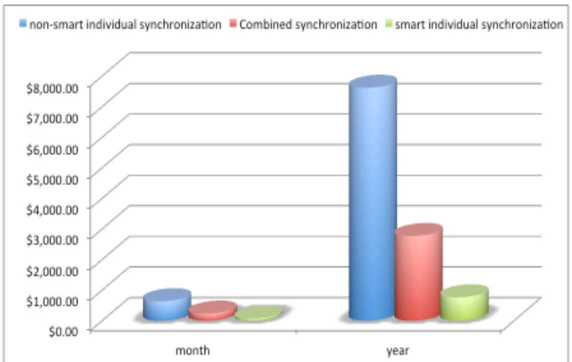

Fig. 5. Network cost of smart and non-smart synchronization

V. CONCLUSION

Continuous monitoring is very important for patients with chronic diseases. Unfortunately, observing the patients in hospitals is impracticable and not cost effective for both hospitals and patients. Fortunately, new emerging mobile and wireless technologies help in remotely monitoring patients while they are home. An appropriate body sensor, connected to a mobile application running on a smart mobile phone, can read vital body signs and send them to physicians who can have continuous monitoring of patients on the go. However, sensors might generate considerable amount of data that, when synchronized, can represent a higher TCO.

This paper presented a new approach for a smarter synchronization of health data between mobile applications and backend systems. This approach is based on a cost function that evaluates, on the fly, the cost of synchronization whenever there are some readings to synchronize back and/or forth. The cost function provides best suggestions to the user regarding the cost of synchronization and its urgency (or not). However, it is up to the user to take the final decision. The experimentations we have conducted showed a tremendous reduction in the network cost as charged by the patient's provider.

REFERENCES

[1] N. Nakajima, "Short-range wireless network and wearable biosensors for healthcare applications," *Proc. 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009*, 2009, pp. 1-6.

[2] R.N. Horspool and W.J. Windels, "An LZ approach to ECG compression," *IEEE Seventh Symposium on Computer-Based Medical Systems*, 1994, pp. 71-76.

[3] C. Roberto, F. Moser, B. Holger and O. Sascha, "Supporting Medical Emergencies by MAS," *Book Supporting Medical Emergencies by MAS*, Series Supporting Medical Emergencies by MAS, ed., Editor ed.^eds., Springer-Verlag, 2009, pp.

[4] C.H. Salvador, et al., "Airmed-cardio: a GSM and Internet services-based system for out-of-hospital follow-up of cardiac patients," *IEEE Transactions on Information Technology in Biomedicine*, vol. 9, no. 1, 2005, pp. 73-85.

[5] J. Maitland and M. Chalmers, "Self-monitoring, self-awareness, and self-determination in cardiac rehabilitation," *Proc. Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010*, 2010.

[6] A. Pitsillides, et al., "Ditis: A Collaborative Virtual Medical Team for Home Healthcare of Cancer Patients," *M-Health*, Topics in Biomedical Engineering International Book Series, R. S. H. Istepanian, et al., eds., Springer US, 2006, pp. 247-266.

[7] V. Jones, et al., "Mobihealth: Mobile Services for Health Professionals," *M-Health*, Topics in Biomedical Engineering International Book Series, R. S. H. Istepanian, et al., eds., Springer US, 2006, pp. 237-246.

[8] K. Wac and M. Ullah, "From Mobile Cognition to Cognitive Mobility: QoS-Aware, Mobile Healthcare Services," *Human Aspects in Ambient Intelligence*, Atlantis Ambient and Pervasive Intelligence 8, T. Bosse, et al., eds., Atlantis Press, 2013, pp. 135-152.

[9] H.M.a.B.J.F.v.B.a.I.A.W.a.V.M.J.a.H.J. Hermens, "Context-Aware Task Redistribution for Enhanced M-health Application Performance," *Mobile Health Solutions for Biomedical Applications*, Advances in Healthcare Information Systems and Informatics (AHISI) Series

, P. O. a. J.Tan, ed., 2009, pp. 23.

[10] L. Atzori, A. Iera and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, 2010, pp. 2787-2805.

[11] D. Giusto, A. Lera, G. Morabito and L. Atzori, *The Internet of Things*, Springer, 2010.

[12] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, 2013, pp. 1645-1660.

[13] M. Kranz, P. Holleis and A. Schmidt, "Embedded interaction: Interacting with the internet of things," *IEEE Internet Computing*, vol. 14, no. 2, 2010, pp. 46-53.

[14] S. Tozlu, M. Senel, W. Mao and A. Keshavarzian, "Wi-Fi enabled sensors for internet of things: A practical approach," *Communications Magazine, IEEE*, vol. 50, no. 6, 2012, pp. 134-143.

[15] C. Mi-Young, C. Eun-Ae, P. Dae-Ha, M. Chang-Joo and B. Doo-kwon, "A database synchronization algorithm for mobile devices," *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 2, 2010, pp. 392-398; DOI 10.1109/tce.2010.5505945.

[16] R. Singh and C. Dutta, "A Synchronization Algorithm of Mobile Database fpr Cloud Computing " *International Journal of Application or Innovation in Engineering and Management*, vol. 2, no. 3, 2013, pp. 6.

[17] B.S. Ramya, S.B. Koduri and M. Seetha, "A stateful Database Synchronization Approach for Mobile Devices " *International Journal of Soft Computing and Engineering*, vol. 2, no. 3, 2012, pp. 5.

[18] B.-G. Chun, et al., "Mobius: unified messaging and data serving for mobile apps," *Book Mobius: unified messaging and data serving for mobile apps*, Series Mobius: unified messaging and data serving for mobile apps, ed., Editor ed.^eds., ACM, 2012, pp. 141-154.

[19] C. Li, C. Zheng and C. Tai, "Detection of ECG characteristic points using wavelet transforms," *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 1, 1995, pp. 21-28.

[20] P. Hamilton, "Open source ECG analysis," *Computers in Cardiology*, 2002, pp. 101-104.

[21] Zephyr, "Bioharness 3," http://www.zephyr-technology.com/products/bioharness-3/.