ELSEVIER

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

Architecture, cost-model and customization of real-time monitoring systems based on mobile biological sensor data-streams☆

Alfredo Goñi[a,∗,1], *Alfredo Burgos*[a,1], *Lacramioara Dranca*[b,1], *Jimena Rodríguez*[a,1], *Arantza Illarramendi*[a,1], *Jesús Bermúdez*[a,1]

[a] *University of the Basque Country, Paseo Manuel Lardizabal 1, 20018 Donostia-San Sebastian, Spain*
[b] *University of Zaragoza, Maria de Luna 3, 50015 Zaragoza, Spain*

ARTICLE INFO

Article history:
Received 4 December 2008
Received in revised form
8 April 2009
Accepted 13 April 2009

Keywords:
Data stream management
Monitoring applications
Arrhythmia monitoring
Apnea monitoring

ABSTRACT

Innovation in the fields of wireless data communications, mobile devices and biosensor technology enables the development of new types of monitoring systems that provide people with assistance anywhere and at any time. In this paper we present an architecture useful to build those kind of systems that monitor data streams generated by biological sensors attached to mobile users. We pay special attention to three aspects related to the system efficiency: selection of the optimal granularity, that is, the selection of the size of the input data stream package that has to be acquired in order to start a new processing cycle; the possible use of compression techniques to store and send the acquired input data stream and; finally, the performance of a local analysis versus a remote one. Moreover, we introduce two particular real systems to illustrate the suitability and applicability of our proposal: an anywhere and at any time monitoring system of heart arrhythmias and an apnea monitoring system.

1. Introduction

Innovation in the fields of wireless data communications, mobile devices and biosensor technology enables the development of a new type of monitoring systems that provide people with assistance anywhere and at any time. Each one of the previous technologies presents a wide spectrum of possibilities that allows for different kind of monitoring systems to be built.

There are several possibilities of sending data using wireless communications: through cellular networks and through wireless networks. Examples of cellular networks are General Packet Radio Service (GPRS) [20] that provides moderate bandwidth (theoretically up to 171.2 kbit/s) but very broad coverage (theoretically world coverage), or Universal Mobile Telecommunications System (UMTS) [7] that offers better bandwidth (a few Mbit/s) and also world coverage, at least in theory. Those cellular networks provide the technology needed to build wireless wide area networks. Examples of wireless networks are WiFi [21], a technology that allows to build wireless local area networks with a bandwidth of several Mbits/s but only a few meters (around 50) of coverage, or WiMAX [41], a

☆ This work was mainly supported by the Basque Government (IT-427-07), Diputación Foral de Gipuzkoa (cosupported by the European Social Fund), and Spanish Ministry of Education and Science under grant TIN2007-68091-C02-01.
∗ *Corresponding author*. Tel.: +34943 015066.
E-mail address: alfredo@ehu.es (A. Goñi).
1 http://siul02.si.ehu.es/.

doi:10.1016/j.cmpb.2009.04.010

technology that allows building metropolitan area networks with much bigger bandwidth (70 Mbits/s) and several kilometers of coverage. With regards to mobile devices, there are Personal Digital Assistants (PDAs) with powerful computing capabilities and with memory capacities sufficient to develop complex applications. Those PDAs support also interfaces to connect to wireless networks like GPRS, UMTS, WiFi and even WiMAX. Moreover, there are some other type of mobile devices like smartphones which are mobile phones that, at the same time, integrate the functionality and computing power of PDAs. Finally, with respect to biosensors, several non-invasive biosensors have recently appeared. By non-invasive it is meant that no incision into the body or removal of biological tissue needs to be made in order to measure the vital signs, biosignals, or measurements in general. Among them, we can find *pulseoximeters* that measure pulse and the amount of oxygen in blood, *ECG sensors* that measure the electrical voltage in the heart, *EEG sensors* that measure electrical activity of the brain, and others like *temperature* and *blood pressure* sensors. There is also a great interest in developing non-invasive *glucose sensors* that measure the level of sugar in blood.

In particular, by taking advantage of the previous technologies we have designed and implemented two *anywhere* and at *any time* monitoring systems. One of them capable of detecting in real-time heart arrhythmias which we call MOLEC [35]. MOLEC has already been implemented and validated with the MIT-BIH Arrhythmia Database.[2] This system is thought as an improvement of holters [15] that only record ECG of patients for several hours but cannot react if high risk arrhythmias occur. The second system is able of monitoring in real-time apnea episodes based on SpO_2 signal. The system has also been implemented and validated with the Apnea-ECG Database[3][9]. Although the definitive diagnosis of apneas has to be made by way of expensive polysomnographies (PSG) carried out in hospitals, it is required to define reliable and cheaper monitoring systems that diminish the number of PSGs to be performed. Moreover, if a very long apneic episode is detected, then a sound can be played so that user wakes up. Both systems have been tested with about 20 patients in the Policlinica of Gipuzkoa and Hospital Amara in San Sebastian. For the apnea monitor, our system also exports the signal data in such a format so that the pneumologists with which we are collaborating can analyze the signals with the same software they use daily at their hospital.[4]

The usefulness of these systems is beyond any doubt since the number of people with cardiological problems is on the rise and because SAHS is a high prevalence illness on general population that is considered as a very important public health problem.

However, not only features of *anywhere* and at *any time* monitoring are necessary for valid systems: they have to be also *efficient* and able of running in *real-time*. In this paper, we want to present some of the key ideas we learnt during the design and implementation of those systems so that they can be considered if another similar monitoring system needs to be built.

In this paper, after a section where related works are discussed, we present an architecture for Mobile Biological Sensor Data Streams (MBSDS) useful for several monitoring systems; then, two particular monitoring systems: an ECG monitoring system and an apnea monitoring system based on our proposal of a MBSDS architecture; next, a general cost model for MBSDS and the customization of that cost model for the two monitoring system; and finally, the conclusions.

2. Related work

Concerning related work, we classify them in two main groups. First of all, we review some other research works that provide solutions to systems that work with data streams, sensor networks and/or try to perform some data stream mining. Then, we mention different existing commercial and innovative systems that are being used in order to detect arrhythmias and apneic episodes.

2.1. *Related work: data stream*

As mentioned in the introduction, many different works in the data stream area concerned with monitoring systems. In this section we present some of those works that their main focus is one of the following aspects: management of data streams generated by sensor networks or data stream mining, respectively.

2.1.1. *Related work: data stream and sensor networks*

Classical Database Management Systems (DBMSs) are used with success to develop lots of business applications. However, for applications that have to work with data streams, it is widely accepted that they present some limitations. That is why some Data Stream Management Systems (DSMS) have been proposed. In general, DSMSs offer Query Languages for streams whose basic operators recall on the relational operators. So, in [5], they propose the CQL query language with operators such as *select*, *project*, *binary-join*, *mjoin*, *union*, *except*, *intersect*, *antisemijoin*, *duplicate-eliminate* and *aggregate* and other operators to transform streams into relations and vice versa. In [2] they present *filter* (some kind of selection), *map* (some kind of projection), *union, bsort, aggregate, join* and *resample* (similar to semi-join). In [12] they support selection, join, aggregation and stream merge. In [11], they also provide relational operators such as *join*, *selection*, *projection*, *grouping*, *aggregation*, *duplicate elimination* and the possibility of performing a join-like operator over a temporary repository. Nevertheless, for many monitoring applications, these relational operators do not fit well.

There are other works [8,30] that deal with Sensor Networks (SN), also called sensor databases, that try to support monitoring applications that involve data acquired by sensor and data about the sensor (location and/or related attributes). In [30], they present a query language which is a subset of SQL (only one table allowed in FROM clause, no subqueries and

[2] In particular, it provides a very good accuracy for classifying rhythms (100% for arrhythmias that require medical assistance in less than 3 min, 97.95% for arrhythmias that require medical assistance in less than an hour and 95% for arrhythmias that usually happen before a worse arrhythmia).

[3] It also offers an accuracy of 96% and a ROC–AUC of 99%.

[4] It is the DS-5 software developed by Konica Minolta Sensing Inc.

several restrictions in WHERE and HAVING clauses) and that also allows to specify restrictions about the duration of the samples obtained from the sensors. In [8], an SQL-like query language is also used in order to formulate the queries where appear expressions that retrieve data from the sensors and re-execution frequency. Sensors are modelled as Abstract Data Types (ADT) whose interfaces consist on the sensor's signal-processing functions. Nevertheless, SQL-like languages do not also fit very well with monitoring applications like the one supported by our ECG monitoring system.

There is another work [3] whose aim is to integrate both types of query processing systems (the query processing of the DSMS and the query processing of the SN). In fact, the context of that work is the development of a second-generation distributed stream processing engine called BOREALIS [1], that addresses new requirements like dynamic revision of query results, dynamic query modification and distributed optimization. Its query model is inherited from AURORA [2], with the possibility of changing dynamically the box (query operator) semantics.

Unfortunately, we cannot formulate a sensor continuous query like "Return repeatedly the abnormal high-risk arrhythmias measured by the ECG sensor" in terms of relational-like operators. Moreover, as far as we know, existing DSMS or SN systems cannot be run, or at least, they are not specifically designed for running into mobile devices like PDAs. Therefore, we have not been able to make use of some of the existing DSMSs or SNs in order to build our monitoring applications.

However, some of the issues dealt in these works make also sense in our scenario. In the case of continuous queries, a rate-based optimization has been proposed in order to work with infinite input streams [40]. In that case, it is possible to optimize the queries depending on the highest output rate or the least time to produce the output. In our case, we do not work with relational operators and therefore their proposals are not directly applicable. However, we think that the concept of finding the optimal granularity depending on the maximum allowed latency (the least time to output the current rhythm) recalls their rate-based optimization.

In the case of sensor networks, an in-network query processing has also been proposed, where some of the query operators are executed in the lower nodes of the hierarchy with the goal of minimizing the computation goal, the data transmission and managing energy in an efficient way [39,37]. Our proposal to perform local analysis in the mobile device attached to the sensor goes also in that direction of in-network processing (with only one sensor).

2.1.2. *Related work: data stream mining*

There can be found some works in the data mining literature about mining data streams [19,14,25]. Data stream mining consists on extracting knowledge (patterns, clusters, forecasts, etc.) in streams of information by applying different techniques such as clustering, classification, frequency counting and time series analysis. All those techniques require some more basic techniques to measure if data streams are similar, which also require some even more basic techniques: (a) to calculate distance between data streams (for example Euclidean distance and Dynamic Time Warping (DTW)), (b) to index data streams (for example, R-trees) and (c) to extract features by highlighting periodicities in the data streams (for example Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT), commonly used in the digital signal processing area). There is also an interest in finding cost-efficient mining techniques. In [17] they propose a date rate adaptation technique (algorithm output granularity) to mine data streams in an efficient way.

Moreover, there are other closer works to ours in the area of ubiquitous data mining, that are concerned with the process of performing analysis of data on mobile, embedded and ubiquitous devices [18]. In [24] they present a client/server system that monitors time-critical financial data from a PDA. Although in that system, the data mining is performed in the server part, more recent works [16] propose an architecture where it is possible to perform a light-weight data mining in the client part, and if the data stream rate goes faster than the data mining task, then requests can be made to a data mining server.

In [19], a list of research issues is addressed, among which, we can find some that we have considered in our work: handling the continuous flow of data streams, minimizing energy consumption of the mobile device, transferring data mining results over a wireless network with a limited bandwidth, data stream pre-processing, the needs of real world applications.

To sum up, we consider that our work is complementary to the previous ones and represents a step further in this exciting research area. It proposes a cost model that allows to calculate an optimal configuration as the result of achieving a trade-off among different factors: volume, autonomy and latency.

2.2. *Related work: monitoring systems*

If we consider all kind of systems that deal with ECG Monitoring, that is, those commercially available as well as research proposals, we can classify them into three groups: (A) Systems that record signals and *perform ECG classification off-line*; for example, the holters [15]. Although these kind of systems have the advantage that patients can continue living a normal life, they present a serious drawback: if the patient suffers from a serious rhythm irregularity, only recording is performed and not real-time classification of ECGs: the classification is performed off-line. (B) Systems that *perform remote real-time ECG classification*, for example, Vitaphone [13], MeditSense [29], QRS Diagnostic [34], Cardio Control [10], Active ECG [4] and MobiHealth [26]. Most of them make use of mobile telephones and/or PDAs to capture the ECG signal and send it to a control center where the real-time classification is performed. They continuously send ECGs through a wireless communication network. In spite of the advantages these kinds of systems provide in relation to recording devices, they still present certain limitations related to the fact that the classification is not performed in the place where the signal is acquired. In fact, there is a loss of efficiency in the use of a wireless network because normal ECGs are also sent (which implies a high cost); and, in case the wireless network is not available at some moment, there might be a loss of ECG signal with the corresponding risk of not detecting some anomalies (unless the signal is recorded in the mobile device and sent when the wireless network is available again). (C) Systems that *perform local real-time ECG classification*, for example @Home [36], PhMon [27] and Bai et al. [6] These systems make use of a classical indirect architecture in the mobile computing area: an architecture that includes an intermediary local computer between the sensors and the control center. Those intermediary computers perform some local real-time monitoring in order to detect some anomalies and send alarms to a control center or a hospital. However, they are not mobile.

With respect to apnea detection, there are some commercial systems [31,33] that help to diagnose apnea at home. However those systems only record the signals (heart rate, blood oxygen saturation or airflow) but do not perform any analysis in real-time. The recorded data is downloaded in computers where the analysis is performed off-line. Therefore, they are systems that *perform apnea classification off-line*.

Our monitoring systems are capable of performing remote real-time as well as local real-time classification. In this paper, we present a cost model that permits to decide, among other things, if it is worthy performing local or remote classification.

3. An architecture for mobile biological sensor data stream (MBSDS) systems

In this section we present our proposal for an architecture that can serve as the basic infrastructure for systems that monitor Mobile Biological Sensor Data Streams (MBSDS), that is, data streams generated by biological sensors (for example ECG, EEG, pulseoximetry, temperature, blood pressure, etc.) which are attached to mobile users. The main feature of these kind of systems is that they provide an *anywhere* and at *any time* monitoring. That is, the monitored users wearing the biological sensors do not have to be hospitalized, nor in a health center, nor at their homes so that they can be monitored. They may be living a normal life while feeling safe at the same time, because they know that if some dangerous situation were detected, then physicians would be notified and could make a decision in time, just as if they were in a hospital or in a health center.

The main tasks performed by these monitoring systems are: (1) *data acquisition* from the biological sensors, (2) *data analysis* in order to detect some anomalous situations, and in case they are detected, then (3) *alarm generation* and notification to the physicians so they can make the correct decision.

The main components of the three-tier architecture (see Fig. 1) are: (1) the biological sensors that measure some vital signs from persons; (2) the mobile device (cellular phone, smartphone, PDA, etc.) which is a small computer that presents several features: (a) it is small enough to be carried by any person; (b) it has its own battery; (c) it can be connected to the biological sensors that generate the data stream via a wired connection or a wireless one such as Bluetooth; (d) it can send data to another computer via a wireless link (GPRS or UMTS connection, for example). It is the mobile device which provides an anywhere and at any time monitoring, because it can be working while it is being carried by users; it is a *mobile computer*. And (3) the control center which is a computer where the alarms are shown to the physicians, and that is connected via a wireless connection to the mobile device.

Regarding to which component performs which task we can state that: (1) the data acquisition of the data generated by the biological sensors is made in the mobile device; (2) there are two possibilities of where to perform the data analysis and the alarm generation: in the control center, or, in the mobile device when it is a PDA (or a powerful smartphone) because it is capable of performing some kind of complex processing. (3) the notification of the alarms to the physician is obviously made in the control center; those alarms should be sent from the mobile device in case they were generated there.

In the next two sections, we present the applicability of the proposed architecture to a system that monitors ECG signals, and to a system that monitors apnea episodes.

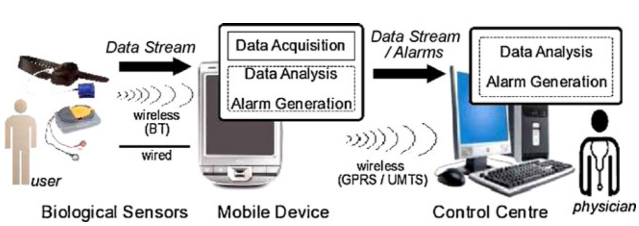

Fig. 1 – Architecture for MBSDS.

4. Case study 1: an Arrhythmia Monitoring System

The goal of our ECG monitoring system (see Fig. 2) is to provide an anywhere and at any time assistance to persons that suffer from heart arrhythmias. An arrhythmia happens when the heart rhythm – the sequence of muscle contractions or beats – is irregular, faster or slower than normal. Using MOLEC, the users wear ECG sensors that capture their ECG signals and are connected to a PDA. Notice that ECG sensors are a good representative of biological sensors because they generate much more data than other biological sensors such as pulseoximeters, temperature, etc. In the control centers there are cardiologists that have to be advised in real-time when high-risk arrhythmias are detected, so that they can verify them and act in accordance.

In the following we present the main tasks that are performed by MOLEC with the goal of showing details about the biological data stream considered and type of analysis made with them. Those main tasks are: ECG Data Acquisition, ECG Signal Preprocessing, Beat Classification, Rhythm Classification and Alarms Management. The goal of this section is not to show all the details of the ECG monitoring system (which can be found in [35]); the goal is to show that it is *feasible* to build such an ECG monitoring system, and to present the high-risk arrhythmia detection process as a data stream transformation problem.

4.1. *ECG data acquisition*

The first step to monitor high-risk arrhythmias is to acquire the electrical signals generated by the user's heart. For that purpose, ECG sensors need to be used. ECG sensors consist of a set of electrodes located at the person's chests that are

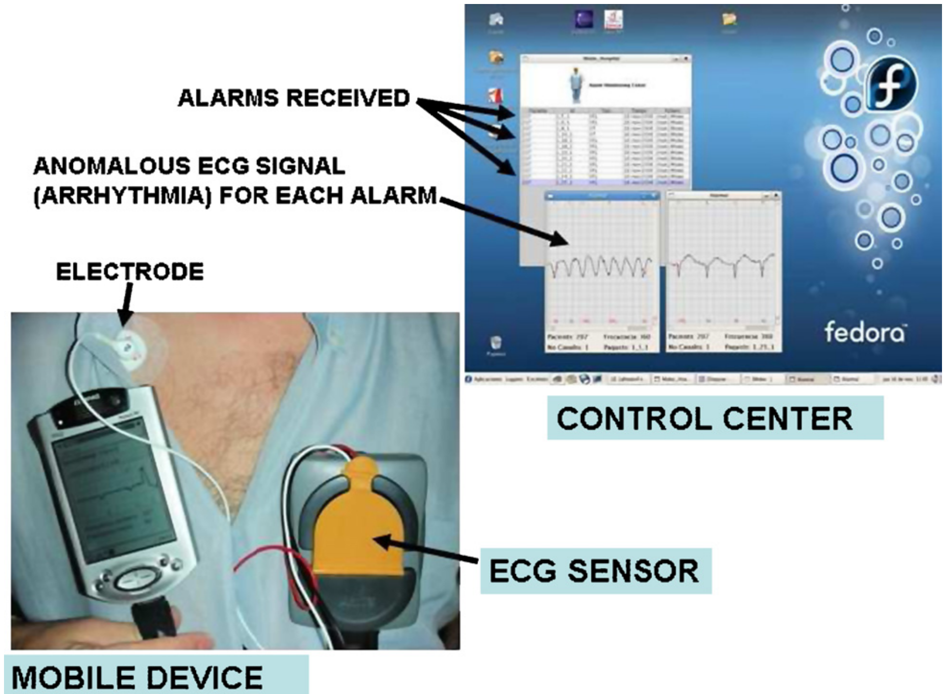

Fig. 2 – The ECG monitoring system.

connected through derivation cables to a device.[5] Those sensors acquire the sequence of samples that form the ECG signal S with a particular frequency F (usually between 100 Hz and 400 Hz). From the data management point of view, ECG sensors generate a continuous data stream $\{s_1, s_2, \ldots, s_i, \ldots\}$ where each s_i is a sample of the ECG signal S, i.e. a number (between 0 and 1024) that represents the voltage of the ECG signal at a particular moment. That data stream and the successive transformations over it, explained in the next subsections, can be seen in Fig. 4.

4.2. ECG signal preprocessing

In order to classify heart rhythms, the pure ECG signal, that is, the input data stream $\{s_1, s_2, \ldots, s_i, \ldots\}$ is not very useful. It has to be processed to identify a series of waves along with their morphology and timing information. A typical normal beat (see Fig. 3) is formed by the P wave, the QRS complex and the T wave (in fact the beginning, end and peak point of each wave to which we call wave events). Moreover, other timing information like P and QRS durations, PR, QT and RR intervals, level of the ST segment, etc. can be useful. Fig. 4

From the data management point of view, the input data stream is preprocessed and transformed into another stream containing those wave events and intervals:

$$preprocess(\{s_1, \ldots, s_i \ldots\}) = \{beat_1, \ldots, beat_j \ldots\}$$

where $beat_j = \langle p_j^b, p_j^p, p_j^e, qrs_j^b, qrs_j^p, qrs_j^e, t_j^b, t_j^p, t_j^e, \overline{pr}_j, \overline{qt}_j, \overline{rr}_j \rangle$ and p_j^b, p_j^p and p_j^e are the begin, peak and end of the P wave corresponding to the jth beat in the ECG signal, respectively; qrs_j^b, qrs_j^p, qrs_j^e, t_j^b, t_j^p and t_j^e are the wave events corresponding to the QRS and T waves; $\overline{pr}_j$, $\overline{qt}_j$ and $\overline{rr}_j$ are the duration of the PR, QT and RR intervals, respectively.

For this preprocessing task, typical techniques in the signal processing area are used [38,23]. First of all, in the original signal (or data stream), the baseline wander and the powerline interference are removed (by applying appropriated filters). Then the QRS complex is detected in three steps, by applying: (1) a linear filter that enhances the spectral content of the QRS and suppresses the corresponding to P and T waves; (2) a nonlinear transformation that converts the QRS complex into a single positive peak; (3) a decision rule (based on thresholds) that decides if something is a QRS complex. After the QRS complex is detected, then the T and P waves are found by looking for the wave boundaries (the

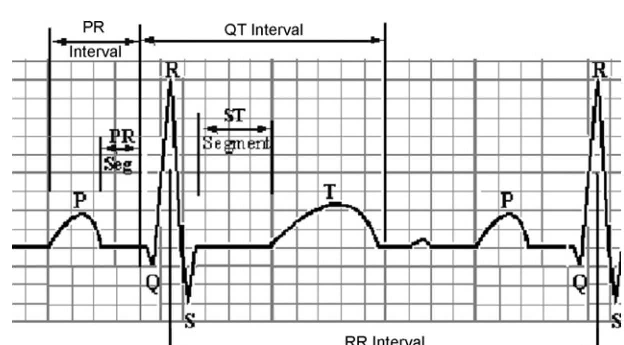

Fig. 3 – Wave events and intervals in a normal beat.

[5] In the current version of MOLEC we use a wired sensor called ActiveECG [4]. Nowadays, the same company also commercializes wireless ECG sensors that could have been used. However, what is wireless is the link between that device and the mobile device, not the link between the electrodes and the device.

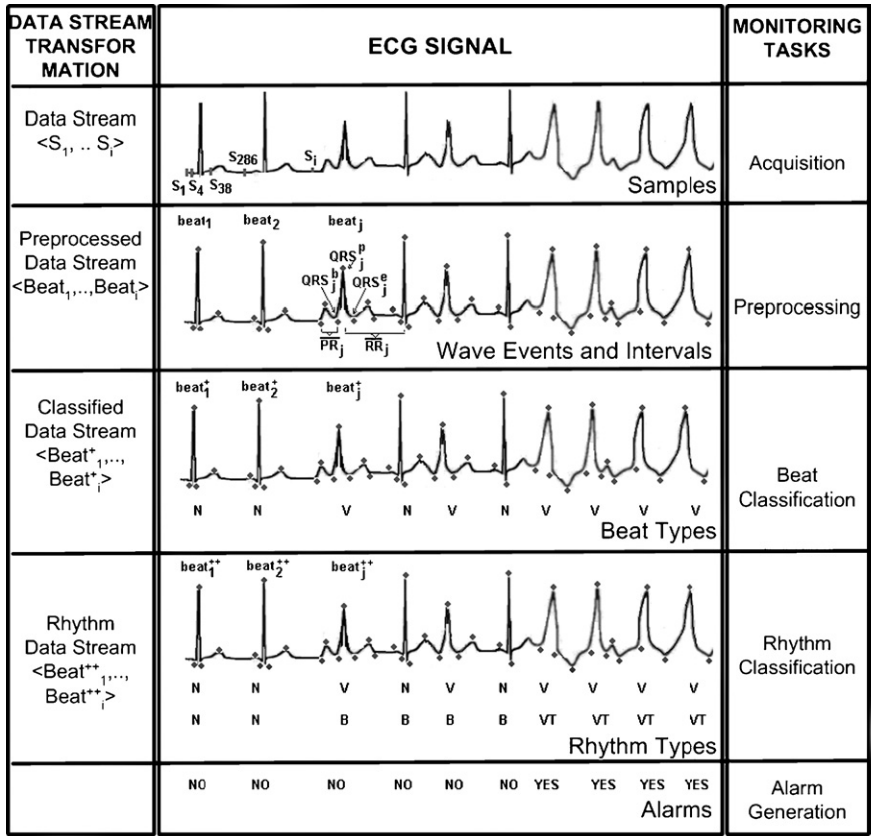

Fig. 4 – ECG monitoring tasks and data stream transformation.

time instants in which some amplitude changes occur) which is made with tests based on thresholds. Finally, once the sequence of wave events is obtained, an automata identifies the beats (by knowing the normal structure of a beat that appears in Fig. 3, but also by taking into account that there can be other abnormal beats that do not have P or T waves).

In fact, those signal processing techniques involve several scans of the whole set of samples in the analyzed ECG signal part. With the goal of realizing what applying a filter involves in terms of processing, we briefly explain what this is. To apply a filter to a signal s is to obtain another signal s^f where each sample s_n^f is calculated as a combination of the original sample s_n, the previous j samples of the original signal s_i and of the filtered signal s_i^f. All those samples are multiplied by some coefficients a_i and b_j that depend on the type of filter as described in the next formula:

$$s_n^f = b_n \times s_n + \sum_{i=n-j}^{n-1} (a_i \times s_i^f + b_i \times s_i)$$

However, this processing is not very time-consuming as we will see in Section 7.4.1, Table 5.

Two important points have to be made here: (1) the size of the input data stream is considerably greater than the size of the preprocessed data stream (18,000 numbers per minute versus 960 numbers per minute when heart frequency is 80 beats per minute and $F = 300$ Hz); (2) although the input data stream is not furthermore used to detect high-risk arrhythmias, it cannot be deleted because it may be queried by cardiologist if they want to see the original ECG signal.

4.3. *Beat classification*

For each beat, a classification process is performed that assigns a byte type to every beat $Bclassify(beat) = bt$, taking into account the beat features extracted during the preprocessing step. All the 14 different types of beats described in PhysioNet [32] are considered by MOLEC.

From the data management point of view, after this classification process the beat type is added to the data stream:

$$\{beat_1^+, \ldots, beat_j^+ \ldots\},$$

where

$$beat_j^+ = \langle p_j^b, p_j^p, p_j^e, qrs_j^b, qrs_j^p, qrs_j^e, t_j^b, t_j^p, t_j^e, \overline{pr}_j, \overline{qt}_j, \overline{rr}_j, bt_j \rangle$$

The size of the data stream is slightly greater now, 1040 numbers per minute when heart frequency is 80 beats per minute and $F = 300$ Hz.

For this classification task, data mining techniques have been used. In particular, techniques based on decision rules. Although the process of building those decision rules is expensive, the beat classifier is generated off-line, and the task of classifying a beat consists of applying a set of *if-then* rules which check conditions (thresholds and so on) over the wave events and intervals. Applying those rules is not very time-consuming (as we will see in Section 7.4.1, Table 5).

4.4. *Rhythm classification*

Every sequence of four beats is grouped, classified and a rhythm type assigned to it, taking into account the beat features extracted during the preprocessing step and the beat types assigned during beat classification. All the 13 different types of rhythms described in PhysioNet [32] are considered by MOLEC.

$$Rclassify(beat^{+}_{j-3}, beat^{+}_{j-2}, beat^{+}_{j-1}, beat^{+}_{j}) = rt_j,$$

From the data management point of view, after this classification process the rhythm type is added to the data stream:

$$\{beat^{++}_{1}, \ldots, beat^{++}_{j} \ldots\},$$

where

$$beat^{++}_{j} = \langle p^b_j, p^p_j, p^e_j, qrs^b_j, qrs^p_j, qrs^e_j, t^b_j, t^p_j, t^e_j, \overline{pr}_j, \overline{qt}_j, \overline{rr}_j, bt_j, rt_j \rangle$$

The size of the data stream is again slightly greater, 1120 numbers per minute when heart frequency is 80 beats per minute.

For this classification task, data mining techniques (based on decision trees) have been used, and also some cardiological rules have been codified. The classifier, that consists of a set of *if-then* rules, is generated *off-line*. Once obtained, applying those *if-then* rules in order to classify actual rhythms is not very time-consuming (as we will see in Section 7.4.1, Table 5).

Notice that there is a delay in identifying a rhythm: the current rhythm type corresponding to a beat rt_j cannot be established until the following 3 beats are detected and classified.

4.5. *Alarms management*

From the 13 different types of rhythms described in PhysioNet, only 5 are considered as high-risk arrhythmias [35]. These are the alarms that have to be notified immediately to the cardiologists. The reason is that some of them (VFL and IVR) require medical assistance in less than 3 min; other type (VT) requires medical assistance in less than an hour and others (T and NOD) show that the user is at high-risk of suffering a heart attack. From the data management point of view, an alarm is attached to a $beat^{++}_{j}$ when:

$$alarm = yes \Leftrightarrow rt_j \in \{VFL, IVR, VT, T, NOD\}$$

Moreover, a special feature related in general to ECG monitoring systems is that cardiologists do not have great confidence on those automatic classifiers (in fact, none of them gets a 100% fiability). Therefore, it is required that the classifier software provides cardiologists with the supposedly abnormal signal parts so that they can verify if they have been correctly classified. That means that every alarm will be accompanied by the signal detected as anomalous.

5. Case study 2: an apnea monitoring system

The goal of this apnea monitoring system is a bit different than the previous one, because it does not make much sense to detect apneas anywhere and at any time. As it is well known apneas are pauses in breathing during sleep. Those pauses imply getting less oxygen from air and, therefore, a reduction of the oxygen in blood. An apneic event usually happens when there is an at least 10-s interval between breaths and, an arousal takes place or when there exists an important blood oxygen desaturation.

In order to diagnose apneas, polysomnographies have to be made to sleeping people. Different signals such as airflow at the nose and mouth, breast movement, ECG, EEG, pulse and oxygen saturation in blood are recorded and then analyzed by physicians. Those polysomnographies have to be made in hospitals and are very uncomfortable for the people (who have to spend a night in a sleep lab) and also very expensive. Moreover, many of those polysomnographies are made to people that do not have apnea problems but only snoring problems.

The goal of our apnea monitoring system is to reduce the number of polysomnographies that have to be made in a hospital, by performing some light polysomnography that monitors only oxygen saturation in blood and that can be made at home (see Fig. 5). If apneas are detected in that case, then a complete polysomnography should be made in a hospital to be sure about that and with the goal of identifying the type of apnea: central, obstructive or mixed.

5.1. *SpO_2 data acquisition*

In our apnea monitoring system, the first step is to obtain the SpO_2 signal. That signal is a sequence of values that represent the oxygen saturation in blood, that is a relative measure of the amount of oxygen that is carried in the blood. That signal is obtained from a pulseoxymeter sensor with the following features: it is non-invasive and it uses Bluetooth communications in order to send the SpO_2 (and also pulse) values to the PDA.

In this case, the frequency does not need to be very high: 1 Hz frequency is enough. From the data management point of view, SpO_2 sensors generate a continuous data stream $\{s_1, s_2, \ldots, s_i, \ldots\}$ where each s_i is a sample of SpO_2 signal S, i.e. a number (between 0 and 100) that represents the percentage saturation of oxygen in blood.

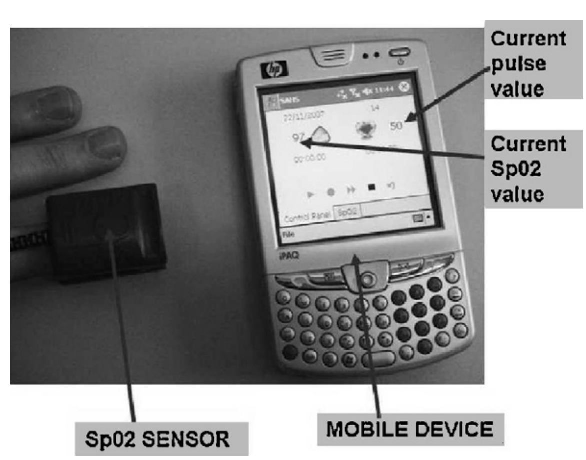

Fig. 5 – Apnea monitoring system.

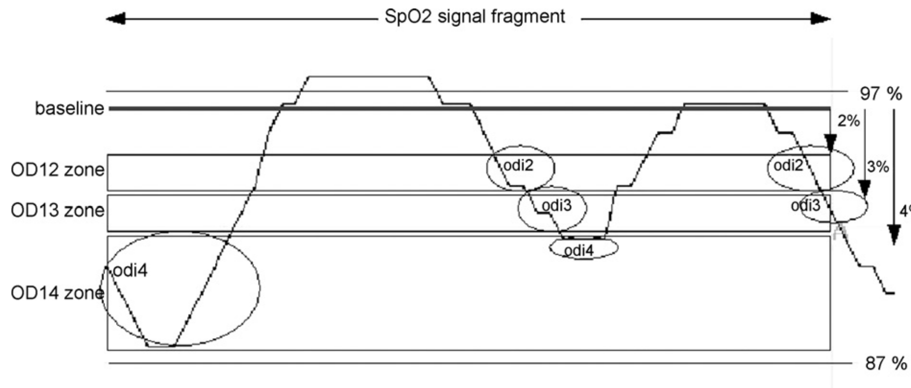

Fig. 6 – ODI events: falls in a zone relative to the baseline.

5.2. SpO_2 signal preprocessing

The isolated SpO_2 values do not say too much about if an apneic event is occurring (although values under 95 are not considered normal). It is necessary to process the SpO_2 signal in order to see if there exist important blood oxygen desaturations.

In fact, there are some processing tasks to be performed: (1) applying some filters in order to eliminate artifacts: SpO_2 values under 50 are usually considered as noise; and samples that have a variation greater than 10% with regard to the previous one are also considered as noise; (2) calculation of a moving baseline, that is, the reference level of oxygen saturation at each moment that is set to the greatest sample in every resaturation interval; (3) calculation of some measures: ODI4, ODI3, ODI2, TSA95, TSA90, TSA85, TSA80 and TSA70.

ODIx is the number of desaturation events in a considered fragment signal, that is, the number of falls in the SpO_2 signal with respect to the baseline. ODI4, ODI3 and ODI2 indicate the number of falls of 4%, 3% and 2% respectively with respect to the baseline (see Fig. 6).

TSA95, TSA90, TSA85, TSA80 and TSA70 indicate the time spent under a saturation of 95%, 90%, 85%, 80% and 70% respectively (see Fig. 7).

From the data management point of view, the input data stream is preprocessed and transformed into another stream containing the ODIs and TSAs values corresponding to a sequence of signal fragments:

$$preprocess(\{s_1, \ldots, s_i \ldots\}) = \{OT_1, \ldots, OT_j \ldots\}$$

where

$$OT_j = \langle ODI4_j, ODI3_j, ODI2_j, TSA95_j, TSA90_j, TSA85_j, TSA80_j, TSA70_j \rangle$$

that represent the ODI and TSA values corresponding to the jth fragment of the SpO_2 signal.

5.3. Apnea classification and alarms management

For each SpO_2 signal fragment, a classification process is performed that tells if that fragment is an apneic event or not. The classifier is in fact a Metaclassifier of type *Bagging* that uses an *ADTree* decision tree as base classifier. It has been built after a training process using the data annotated and available in the Apnea-ECG Database. A fragment of that annotated database can be seen in Fig. 8) where apneic events are annotated with the label A and normal ones with N. The accuracy of our classifier is greater than 97%. Although an alarm could be sent every time an apneic event were detected, it does not make much sense in this case because those events are not dangerous for the user's life; they do not require urgent medical assistance.

6. A general cost model for MBSDS

In Section 3 we have shown an architecture that supports anywhere and at any time monitoring systems. In that architecture, the mobile device, apart from being the key element, is also the most critical element, because, as it is well known, it has some limited resources like the battery, memory, screen size, etc.; it also uses some expensive and not always available resources (like the wireless network). Therefore, attention has to be paid to *efficiency* aspects of the monitoring system, so that the system can be considered *useful*.

The objective of this section is to present a model that can guide in the process of achieving an *efficient* monitoring system. With that purpose, we present the optimization goals in the system, the set of parameters that need to be tuned and then the cost model that will allow us to find the optimal configuration of the previous parameters, respectively.

6.1. Optimization goals

Three main features must be considered when looking for the system efficiency:

1. **Minimizing the use of the wireless link**. Taking into account that the cost of wireless communications of type GPRS/UMTS is high,[6] that these kind of networks are not very reliable, and that sometimes are not available, it is necessary to reduce the quantity of data to be transmitted through the wireless link.
2. **Diminishing latency in the alarms reception**. The alarms generated after an analysis process over the input data

[6] In some countries in Europe, 1 Mb may cost 1 euro.

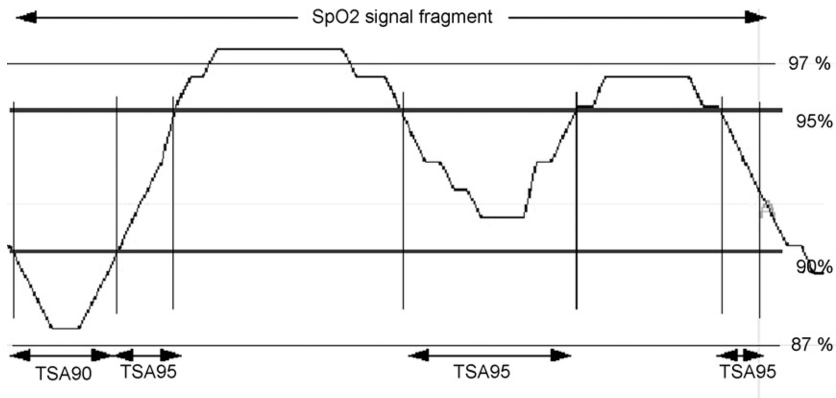

Fig. 7 – TSA measures: time spent under a saturation level.

stream must be communicated as fast as possible to the physicians located in the control center so that they can make the appropriate decision. Sometimes that may imply in saving the user's life.

3. **Rationalizing the use of the battery in the mobile device**. Considering that the battery life is short, it is very important to optimize its use. Alternatively, if the battery is flat, then the monitoring system will be off and no anomalous situations will be detected, stored or sent.

Although there are other well-known restrictions related to mobile devices such as small memory capacities, low processing capabilities or small screen sizes, we do not take them explicitly into account. In fact, current memory sizes are enough to store several hours of the input data stream (apart from the monitoring software), even in the case of the ECG that is one of the biological sensors that generate more data. Moreover, the techniques applied to "minimize the use of the wireless link" also achieve the goal of minimizing the use of the mobile device memory. With respect to the processing capabilities, of course they have an influence in the cost model, but the only way we see to improve the processing capabilities is to upgrade the mobile device processor. The last restriction, having small screen sizes is not a problem in these kind of monitoring systems because the interaction with the user is not required or it is very simple and those screens are enough to do it. Finally, the biological sensors may also need some batteries. It has been checked that those batteries have a much longer life than the batteries of the mobile device, and therefore, those batteries could be recharged whenever the mobile device battery needs to be recharged.

6.2. *Parameters to be tuned*

Three are the parameters that need to be tuned so that the previous optimization goals can be reached.

1. **Granularity**. The granularity of the system represents the frequency in which the input data stream is processed, or, in other words, the size of one package of the input data stream that has to be acquired in order to start a new processing cycle. Notice that this is different than the data

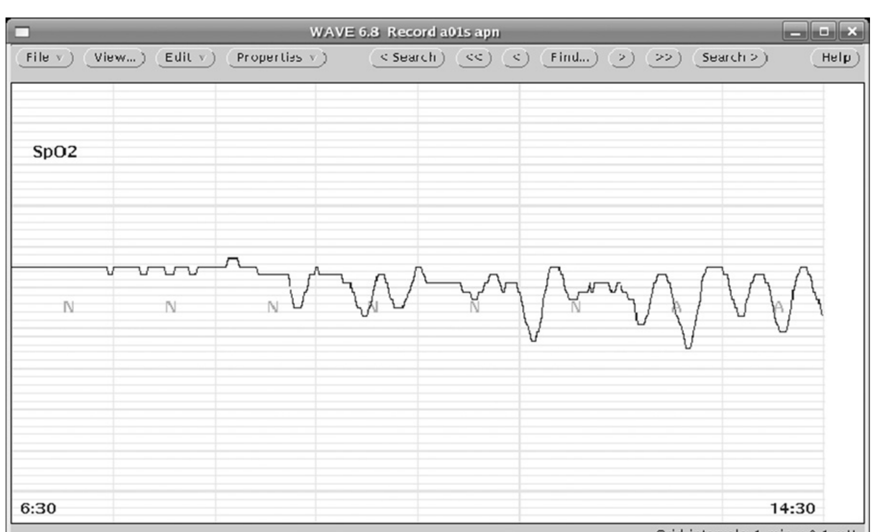

Fig. 8 – Fragment of the annotated SpO_2signal in Apnea–ECG database.

rate, the frequency with which the data are generated by the sensors. Those data are acquired by the mobile device according to the data rate; in the meantime, a new data package is formed, and once completed then another analysis process starts with it. It seems that the smaller the granularity is, the smaller the latency in detecting anomalies will be. But, on the contrary, the treatment of smaller packages will probably consume more battery. This is why this parameter needs to be adjusted. We will express the granularity in time units (i.e. seconds).

2. **Local or remote analysis**. In an MBSDS architecture, there are at least two possibilities of deploying the modules that perform the analysis: (1) The mobile device acquires the data from the biological sensors and sends them to the control center where the analysis is performed, and alarms are shown to the physicians. (2) The mobile device not only acquires the input data stream but it also performs the analysis. If alarms are detected then they are sent to the control center and shown to the physicians. We will call the first possibility *remote analysis* and *local analysis* the second.
3. **Compression**. Another important choice is to decide if it is worth compressing the input data stream or not. Compressing the input data will be a good option for the first goal (minimize the use of the wireless link), but probably bad for the other two goals (diminishing latency and optimizing battery), because extra processing needs to be performed that slows the monitoring process and consumes battery.

6.3. *Cost formulas*

In the following we show, on the one side, the formulas that describe the different factors that need to be optimized, and that depend on the parameters previously defined, and, on the other side, the cost function to minimize.

6.3.1. *Volume*

$V(g, c, a)$ is the function that calculates the volume of information to be transmitted by unit time from the mobile device to the Control Center when using a granularity g, depending on if compression is applied ($c \in \{yes, no\}$)and the type of analysis performed ($a \in \{loc, remote\}$):

$$V(g,c,a) = \begin{cases} P_{alarm} \times s & \text{if } (1) \\ s & \text{if } (2) \\ P_{alarm} \times s \times C(g) & \text{if } (3) \\ s \times C(g) & \text{if } (4) \end{cases}$$

$$(1) \equiv c = no \wedge a = local, \quad (2) \equiv c = no \wedge a = remote$$
$$(3) \equiv c = yes \wedge a = local, \quad (4) \equiv c = yes \wedge a = remote$$

where P_{alarm} is the probability of finding an anomalous situation in the input data stream, s is the size of the input data stream generated in a unit time, $C(g)$ is a rate that tells which is the size that remains after compressing an input data stream acquired during time g (in fact, it is $(1 - CR(g))$, 1 minus the compression rate)

6.3.2. *Latency*

$L(g, c, a)$ is the function that calculates the latency in receiving an alarm since the time in which is acquired by the biological sensors until it is shown in the Control Center to the physicians, when using a granularity g, depending on if compression is applied ($c \in \{yes, no\}$)and the type of analysis performed ($a \in \{local, remote\}$).

$$L(g,c,a) = \begin{cases} d + t_a^{local}(g) + t_s(g) & \text{if } (1) \\ d + t_a^{remote}(g) + t_s(g) & \text{if } (2) \\ d + t_a^{local}(g) + t_c(g) + & \\ +t_s(g \times C(g)) + t_d(g) & \text{if } (3) \\ d + t_c(g) + t_s(g \times C(g)) + t_a^{remote}(g) + t_d(g) & \text{if } (4) \end{cases}$$

where d is the input delay time, that is, it is the time required to acquire the minimum input data stream over which a complete analysis can be performed,[7] $t_a^{local}(g)$ is the time to analyze in the mobile device an input data stream acquired during time g, $t_a^{remote}(g)$ is the same but in the Control Center, $t_s(g)$ is the time to send from the mobile device to the Control Center an input data stream acquired during time g, $t_c(x)$ is the time to compress an input data stream of duration x, and $t_d(x)$ is the same but to decompress.

6.3.3. *Autonomy*

$A(g, c, a)$ is the function that calculates the autonomy of the PDA (that is, the life time of the battery) when using a granularity g, depending on if compression is applied ($c \in \{yes, no\}$)and the type of analysis performed ($a \in \{local, remote\}$). It can be defined as $A(g, c, a) = b/B(g, c, a)$, where b is the total power of the battery and $B(g, c, a)$ the battery consumption by unit time.

$$B(g,c,a) = \begin{cases} \dfrac{c_a^{local}(g) + P_{alarm} \times c_s(g)}{g} & \text{if } (1) \\ \dfrac{c_s(g)}{g} & \text{if } (2) \\ \dfrac{c_c(g) + c_a^{local}(g) + P_{alarm} \times c_s(g \times C(g))}{g} & \text{if } (3) \\ \dfrac{(c_c(g) + c_s(g \times C(g)))}{g} & \text{if } (4) \end{cases}$$

where $c_a^{local}(g)$ is the battery consumption required to analyze in the mobile device an input data stream acquired during time g, $c_s(x)$ is the battery consumption required to send an input data stream of duration x from the mobile device to the Control Center and $c_c(g)$ is the battery consumption required to compress it.

6.3.4. *Cost function*

We are interested in obtaining the configuration that minimizes the volume of data to be transmitted, with the lowest latency and highest autonomy. Unfortunately, these requirements get in conflict: the configuration that obtains the lowest latency will not probably be the one with the highest autonomy or the smallest volume. Moreover, some configuration may be very good for improving some of the requirements but completely unacceptable for others, and should be rejected. Therefore, the cost function must help in finding a config-

[7] In the ECG Monitoring case, 4 beats are required in order to classify the rhythm.

uration that represents a good trade-off among the three requirements.

This trade-off can be achieved through two mechanisms: *weights* and *thresholds*. On the one hand, in the definition of the general cost function, a weight is assigned to each of the factors (latency, autonomy and volume) that amplifies or attenuates their corresponding cost. With those weights it is possible to establish the relative importance of each of the cost factors in the total cost. On the other hand, some unacceptable configurations may be rejected by defining some thresholds for each of the factors, that is, some values (maximum latency, minimum autonomy and maximum volume by unit time) to which their corresponding cost value is considered as infinity.

In the following, we show the general cost function that permits finding the optimal configuration: the one that obtains the minimal value for that cost function.

$$C(\bar{p}) = \begin{cases} \infty, & \text{if } V(\bar{p}) > MV \vee L(\bar{p}) > ML \vee A(\bar{p}) < mA \\ w_{vol} \times V(\bar{p}) + w_{lat} \times L(\bar{p}) + \dfrac{w_{aut}}{A(\bar{p})}, & \text{else} \end{cases}$$

where $\bar{p} =< g, c, a >$, MV is the maximum volume by unit time allowed, ML is the maximum latency, mA is the minimum autonomy and w_{vol}, w_{lat} and w_{aut} are the weights corresponding to the volume, latency and autonomy, respectively.

7. Customization of the cost model for each monitoring system

In the previous section, we have presented a cost model that guides in the process of selecting an optimal configuration for a monitoring system, focusing on three aspects: best granularity, applying or not compression techniques and the kind of analysis (local or remote).

The proposed cost model is general enough to be valid for many kind of “real-time biological sensor data-stream monitoring system”. However, it has to be customized for each particular monitoring system that must be developed. This customization consists of providing concrete values to the constants and functions that appear in the general cost model. These constants and functions depend strongly on the type of monitoring performed, the type of signal or data-stream generated by the biological sensors, the user health, and some other machine factors like mobile device computing power, battery lifetime, or wireless communications used. Therefore, the semantics of the particular application must be taken into account in order to do the customization.

We propose to do the customization process in two steps: (1) studying and analyzing the monitoring process in order to assign values to the constants; (2) designing and performing a set of experiments and obtaining values with which to interpolate the functions. Those experiments are of three types: experiments that measure compression rates, experiments that measure the battery consumption (in compressing, analyzing and sending tasks) and experiments that measure the latency factors (also in compressing, analyzing and sending tasks).

In the rest of this section we give more details about how the assignment of constants has been made and also about the experiments performed with our monitoring systems: arrhythmia and apnea. The mobile device used for the experiments has been the next one: An iPaq 3970 PDA with a 400-MHz XScale processor, 64-MB SDRAM, and 48-MB Flash memory. The PC configuration for the Control Center was a Pentium IV (512 MB RAM, 2.4 GHz). The communication between the PDA and the Control Center is made via WiFi.[8] With the goal of working with ECG signals that may generate alarms, we have used one of the ECG records available in the MIT-BIH Arrhythmia database. It is the 207 record that corresponds to a 89-year old woman with 14 anomalous episodes that last 7 min (over a 30 min recording). Therefore, for these experiments, we have substituted the ECG sensors by another computer connected to the PDA that sends that ECG with the same frequency used by the sensors. Finally, we have to say that some of these elections (WiFi for communications and a record containing many arrhythmias) are not a priori good for local analysis[9] because communications between the PDA and the Control Center will be faster with WiFi than with GPRS/UMTS, and the more high-risk arrhythmias there are in the ECG Signal the more communications between the PDA and the Control Center will be needed.

7.1. *Assignment of weights and thresholds*

In order to do this task, it has to be established, on the one hand, the relative importance (or weight) of the three main factors involved in the cost model (volume, latency and autonomy) for the particular type of monitoring; and, on the other hand, if there are some values prohibited for those factors. After that, the values corresponding to the next constants in the cost model formulas: w_{vol}, w_{lat} and w_{aut}, and MV, ML and mA, respectively, must be provided.

7.1.1. *Customization of arrhythmia monitoring system*

In the arrhythmia monitoring case, we know that to reduce the latency is better than to reduce the volume because in the first case, some life could be saved, and in the second one, only money would be saved. For the same reason, to increase the battery life is also better than to reduce the volume. However, it is not clear if it is better to reduce the latency than to increase the autonomy because in that case, if the autonomy is too little then the monitoring system could be *out of order* just when a high-risk arrhythmia happened. Therefore, the same weight could be assigned to both factors; let us say that $w_{aut} = w_{lat} = 1$. The maximum latency should assure that it is still possible to react in time in case a high-risk arrhythmia happens. After consulting with cardiologists we assigned the value $ML = 13$ s. A minimum value for autonomy could be the corresponding to 5 h ($mA = 18{,}000$) s. It may be reasonable to consider that within that time the user will be at home, or near an electrical plug and will have the opportunity to charge the battery.

Moreover, if the latency is small enough and the autonomy is high enough, then, it may be good to reduce the volume (because to save money is also important!). In our case we have

[8] We have used WiFi because it was free of cost.

[9] In fact, local analysis is one of the original features of our ECG monitoring system, and we did not want to be unfair with the remote analysis approach.

assigned the weight $w_{vol} = 0.3$ and, with the goal of not invalidating all the configurations that use remote analysis and non-compression, we have assigned a maximum volume of 360 ($MV = 360$), supposing that it is affordable to pay for sending 30 Mb, the size of a 24-h ECG signal when using a 360 Hz acquisition frequency.

7.1.2. Customization of apnea monitoring system

In the apnea monitoring case, the autonomy is not really a problem because while the user is sleeping, the PDA can be connected to an electrical plug. With respect to alarms, nobody dies due to suffering an apnea, so the latency of detecting apneas is not very critical. Moreover, it is not really required to send the apnea information in real-time, and therefore the volume of data is not a problem either taking into account that the size of a 10-h SpO_2 signal is of 36 Kb using a 1 Hz acquisition frequency. Notice that we do not say that it is not important to detect apnea episodes in real-time. In case a very prolonged apnea would be detected, then it would be a good idea to wake up the user by using a sound alarm from the PDA. Nothing should be sent to the physician in order to do that. In summary, we can consider that the three factors are equally important: $w_{aut} = w_{lat} = w_{vol} = 1$. With the goal of not invalidating all the configurations that use remote analysis and non-compression, we can assign a maximum volume of 1 ($MV = 1$), a minimum autonomy of 1 h ($mA = 3600$) in case of an electrical power blackout would happen and a maximum latency of $ML = 62\,s$, a couple of seconds after the last 1-min signal.

7.2. Assignment of monitoring- and user-dependent constants

Some of the monitoring-dependent constants are d (the input delay time) and s (the size of the input data stream acquired by unit time) and must also be provided. The input delay time, d, represents the time required to acquire the minimum input data stream over which a complete analysis can be performed, and depends on the type of monitoring.

Another user-dependent constant is P_{alarm}, the probability of an alarm going off. That constant is user-dependent because not all the people have the same probability of suffering an anomalous episode.

7.2.1. Customization of arrhythmia monitoring system

In our ECG arrhythmia monitoring case, a sequence of 4 beats is required in order to classify its rhythm and then d is[10]: $d = g \times \lceil 240/g \times hr \rceil$ where hr is the heart rate (the number of beats per minute). Notice that hr is the average heart rate corresponding to the user.

s (the size of the input data stream acquired by unit time) is 360 bytes, by working with a frequency of 360 Hz and a unit time of 1 s.

[10] $4 \times 60/hr$ is the number of seconds required to occur 4 beats; $\lceil 4 \times 60/g \times hr \rceil$ is the number of entire packages of granularity g (in seconds) that contain 4 beats; and $g \times \lceil 4 \times 60/g \times hr \rceil$ is therefore the number of seconds that need to be waited until the packages that contain 4 beats are acquired.

The probability of an alarm going of, a high-risk arrhythmia, the P_{alarm} is extremely high (0.23) because the signal corresponds to the 207 patient where 7 min of a 30 min recording correspond to anomalous signal.

7.2.2. Customization of apnea monitoring system

In the apnea monitoring case, d is considered to be 60 s and s is 1 byte. A frequency of 1 Hz is enough. The probability of occurring an apneic episode, the P_{alarm}, is 0.05, taking into account that the 20% of the signals appearing in the Apnea–ECG Database contain apneic episodes.

7.3. Experiments related with compression rate

The goal of these experiments is to approximate the function $C(g) = 1 - CR(g)$, which defines the compression rate obtained when compressing an input data stream acquired during time g. Notice that it depends on several features related to the input data stream: acquisition frequency, size of each sample or datum in the data stream, the range and concrete values of those samples, and even the type of analysis required by it.

In order to choose good compression algorithms, we started studying some compression algorithms commonly used in the signal processing area [22]. In that area, some of the proposed algorithms are based on lossy compression, meaning that after decompression the original signal is not exactly obtained. However, that is not a great problem if the loss of information is not very significant. There are some measures, like the Percentage Root mean-square Difference (PRD), that can be used in order to quantify that loss of information. The closer to zero PRD is, the more similar are the original and compressed signals. Moreover, both signals do not have to be exactly equal, only the results of the analysis over the original and compressed signals should be equal. In that case, the lossy compression could be a reasonable option.

Moreover, we also tried to apply other well known lossless compression algorithms typically used in order to compress any kind of files in many computer science areas. Typical algorithms without loss of information are the following ones: ARJ, LHA, RAR, ZIP, GZIP and GIF.

Finally, we also tried to apply both types of algorithms: lossless and lossy compression algorithms.

Two are the experiments designed and performed: one to find the best compression algorithm, and, the other one, to obtain the compression rates for different input data stream sizes.

7.3.1. To find the best compression algorithm in the arrhythmia monitor

In this case, we saw that lossy compression methods with a PRD smaller than 0.5 compress the ECG signal and do not produce morphological changes that affect to the rhythm ECG classifier. In Table 1 we show the compression rates and PRD values obtained with two lossy compression algorithms: Differential Pulse Code Modulation (DPCM) and Fast Fourier Transform (FFT) applied over signals of several minutes.

Another interesting point is that the obtained results working with lossless compression algorithms were in general better than the compression algorithms used in the signal processing area. In Table 1 we show the compression rates for the

Table 1 – To find the best compression algorithm in the arrhythmia monitor.

	10 min	30 min	60 min	120 min
DPCM (PRD)	43.8% (0.0002)	43.6% (0.0001)	43.6% (0.0101)	43.6% (0.0124)
FFT (PRD)	42.9% (0.38)	24.% (0.31)	24.5% (0.31)	24.4% (0.31)
GZ	74.5%	75.6%	75.6%	75.6%
DPCM + GZ	82.6%	83.0%	83.0%	83.0%
DPCM + GZ2	83.2%	83.7%	83.0%	83.0%
DPCM + BZ2	87.7%	88.2%	88.2%	88.2%
FFT + BZ2	76.7%	69.4%	69.1%	68.8%

Table 2 – To find the best compression algorithm in the apnea monitor.

	10 min	30 min	60 min	120 min	480 h
GZ	90.3%	88.6%	88.7%	88.4%	88.9%
BZ2	93.2%	92.2%	92.7%	92.9%	93.7%
DPCM					16.6%
DPCM + BZ2					92.6%

lossless compression algorithm GZ (around 75% compression rate, much better than the 45% compression rate for DPCM).

Finally, we learnt the following property: (in fact it constitutes an original result) the combination of lossless compression and lossy compression algorithms outperforms both types of algorithms. The best option is to compress the input data stream with a lossy compression algorithm, and then to compress the previous result with a lossless compression algorithm. In Table 1 we can see that the compression rate for the combination of a lossless compression algorithm (BZ2) and a lossy compression algorithm (DPCM) obtains a really good compression rate (around 88%).

7.3.2. *To find the best compression algorithm in the apnea monitor*

In this case, we saw that a lossy compression methods like Differential Pulse Code Modulation (DPCM) does not produce a good compression rate (16.66% for an 8-h recording). Notice that with those low compression rates it is normal that the combination of a lossy compression algorithm (DPCM) and lossless compression algorithms (BZ2) is worse than BZ2 algorithm, that is the best compression algorithm in this case (see Fig. 2), with a 93.7% compression rate.

7.3.3. *To obtain the compression rates*

Once the best compression algorithm has been identified, it is required to obtain the compression rates depending on the input data-stream size. In the previous experiment, the compression rates were obtained when compressing relatively large data streams (in the matter of minutes). However, when monitoring data-streams in real-time, the granularity is much thinner and, therefore, the compression rate may be lower. In Table 3 we show which are the compression rates obtained by compressing ECG signals with combination of the DPCM lossy compression algorithm and BZ2 lossless compression algorithm, in the arrhythmia monitor case, and SpO_2 signals with the BZ2 algorithm. It can be seen that the bigger the granularity, the bigger the compression rate. Moreover, a compression rate very close to the optimal (88%) can be obtained with a granularity of around 30 s, and with granularities of only few seconds, the compression rate may be considered as very high (around 85%). In the apnea monitor case, compression rate for 60 s-signals (around 67%) are a bit far from the optimal rate (around 93%), for which 10-min signals are required.

Finally, the corresponding $C(g)$ functions can be obtained by interpolating the values appearing in Table 3.

7.4. *Experiments related with latency*

The goal of these experiments is to approximate several functions that appear in the cost model formulas: those corresponding to analyzing an input data stream acquired during time g in the PDA and in the Control Center ($t_a^{local}(g)$ and $t_a^{remote}(g)$), and, the functions that define the time to compress, send and decompress it ($t_c(g)$, $t_s(g)$ and $t_d(g)$). Particular attention must be paid to assure that the real-time processing is possible.

7.4.1. *To obtain the minimum granularity for real-time processing in the arrhythmia monitor*

The real-time monitoring of mobile sensor data-streams cannot be guaranteed if it is not possible to process an input package before the next input package is acquired. This depends strongly on the size of that input package, that is, the granularity. As, in many cases, the processing times change from execution to execution, then it is better to repeat the experiments several times and to work with the mean (μ) and standard deviation (σ). In general, granularities g such that $\mu(t_a(g)) \geq g$ must be rejected. Moreover, it is even safer to reject granularities such that $(\mu(t_a(g)) + \sigma(t_a(g))) \geq g$. In Table 5, we

Table 3 – Compression rates for different granularities (in s).

g (s)	1	2	5	30	60	120	300	600
$CR(g)$ using DPCM + BZ2 with ECG signals	80.41	83.36	85.58	88.18	88.33	88.35	88.13	87.73
$CR(g)$ using BZ2 with SPO_2 signals	−1233		−180	44.44	67.22	81.67	90.8	93.06

Table 4 – Analysis time (mean and standard deviation) for the arrhythmia monitor.

g	$\mu(t_a^{local})$	$\sigma(t_a^{local})$	$\mu(t_a^{remote})$	$\sigma(t_a^{remote})$
1	0.75	0.77	0.066	0.056
2	1.33	0.93	0.134	0.068
3	1.88	1.23	0.182	0.078
4	2.44	1.57	0.22	0.091
5	2.94	1.76	0.238	0.114
9	4.98	2.29	0.317	0.123

show the times required to analyze an ECG signal of size g in the PDA of our ECG monitoring system. We can see that the minimum granularity should be of 4 s (if we relax a little bit the previous restriction because $(\mu(t_a^{local}(4)) + \sigma(t_a^{local}(4))) = 4.01$). The minimum granularity if the analysis is remote could be of only 1 s because $(\mu(t_a^{remote}(1)) + \sigma(t_a^{remote}(1))) = 0.122$. No smaller granularities are taken into account because the heart frequency is usually greater than 60 beats per minute, what means that the probability of finding a beat if the granularity is smaller than 1 is really low.

Tables 4 and 5

7.4.2. *To obtain the minimum granularity for real-time processing in the apnea monitor*

In fact, in this case, the real-time processing does not constitute a problem, because it can be achieved even if the analysis process is performed every second, that is, every time a new sample is sent from the sensor (the input data frequency is of 1 Hz). Although that is the frequency of analysis, the analysis is performed with the last 60-s signal. Moreover, the obtained results show that the analysis process in the PDA is slower when the frequency is 1 Hz, because the CPU workload is bigger in that case. However, that does not happen when the analysis process is performed in the Control Center. In that last case, the more workload there is in the CPU, the faster the analysis is performed what might mean that the CPU clock frequency is automatically reduced if the workload is low.

7.4.3. *To obtain the compression, decompression and sending times*

Once the minimum granularity is obtained, then some experiments need to be performed in order to approximate the functions that define the time to compress, send and decompress it ($t_c(g)$, $t_s(g)$ and $t_d(g)$). In Table 6 we show the obtained results for granularities greater than 1 (that was the minimum granularity for remote analysis) in the arrhythmia monitor. For the apnea monitor case, we only show the compression, sending and decompression times for frames of 30 and 60 s. Independently of the analysis frequency, the frame size to be analyzed is of 60 s.

Table 7

Table 6 – Compression, sending and decompression times in the arrhythmia monitor.

g	$t_c(g)$	$t_s(g)$	$t_d(g)$
1	0.18	0.11	0.007
2	0.3	0.13	0.008
3	0.4	0.14	0.009
4	0.5	0.15	0.010
5	0.6	0.17	0.011
6	0.8	0.18	0.011
12	1.6	0.3	0.018

Table 7 – Compression, sending and decompression times in the apnea monitor.

g	$t_c(g)$	$t_s(g)$	$t_d(g)$
30	0.62	0.10	0.005
60	0.63	0.11	0.006

7.5. *Experiments related to autonomy*

The goal of these experiments is to approximate the functions that appear in the cost model formulas corresponding to battery consumption required to analyze, compress and send in the PDA an input data-stream acquired during time g ($c_a^{local}(g)$, $c_s(g)$ and $c_c(g)$). Before performing those experiments, simple techniques to reduce battery consumption need to be applied.

7.5.1. *To reduce the battery consumption*

There are two simple but effective things that can be done in order to reduce the battery consumption: to disable brightness and sound in the PDA, and to adjust the frequency of the PDA processor.

It is a very good option to disable brightness and sound of the PDA because the power savings are really important (up to 100% better in many cases).

Another technique that can be applied is to adjust the frequency of the PDA processor in order to minimize the battery consumption. The idea is to reduce the CPU frequency (what means a power saving in the battery) without increasing the latency, that is, without increasing the times to compress, analyze and send the input data-stream. In our case, we tried with several frequencies, and also with a utility call LAMP (Linux and Managed Power) [28], a simple user-level tool to adapt CPU

Table 5 – Analysis time (mean and standard deviation) for the apnea monitor.

g	$\mu(t_a^{local})$	$\sigma(t_a^{local})$	$\mu(t_a^{remote})$	$\sigma(t_a^{remote})$
1	0.014	0.002	0.0004	0.003
2	0.013	0.002	0.0007	0.004
5	0.013	0.002	0.002	0.011
15	0.012	0.002	0.005	0.015
30	0.010	0.002	0.008	0.011
60	0.008	0.001	0.016	0.01

Table 8 – Battery consumption in the arrhythmia monitor.

g	$c_a^{local}(g)$ (in mA)	$c_s(g)$	$c_c(g)$
2	0.06	0.04	0.10
3	0.09	0.06	0.22
4	0.12	0.06	0.23
5	0.16	0.10	0.24
9	0.19	0.16	0.41
120	0.29	1.50	2.00

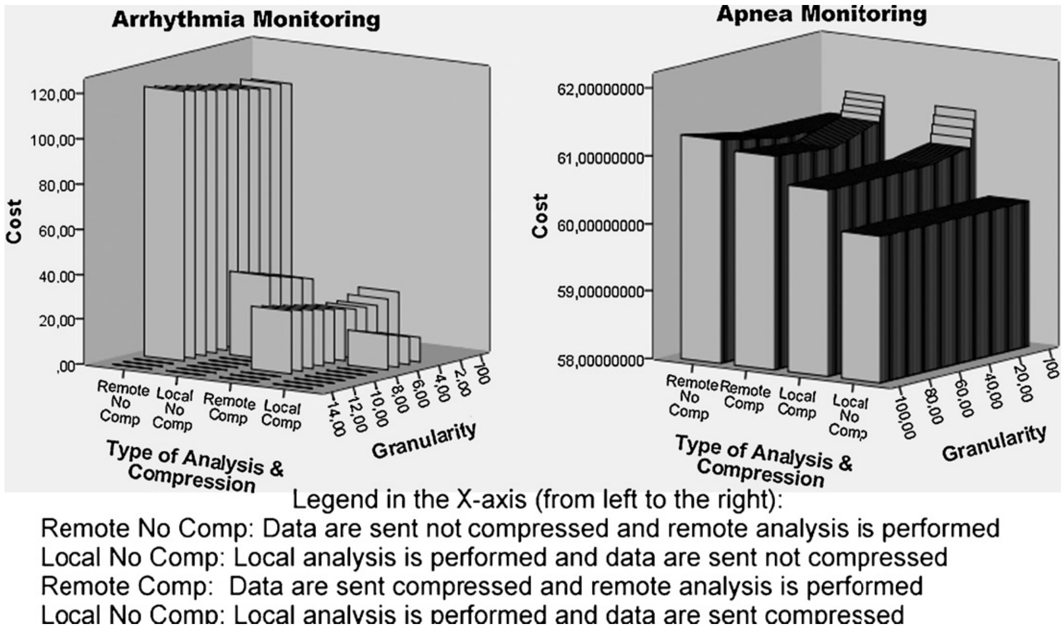

Fig. 9 – Different configurations for the monitoring systems.

frequency to CPU load and we realized that with this last tool some power savings (of about 10% more) were gained.

7.5.2. *To obtain the battery consumption to analyze, compress and send*

Finally, some experiments were performed in order to approximate the functions $c_a^{local}(g)$, $c_s(g)$ and $c_c(g)$, that calculate the battery consumption required to analyze, compress and send in the PDA an input data-stream acquired during time g. Notice that these experiments were performed by applying the basic techniques explained in previous subsection: brightness and sound disabled and the LAMP utility enabled. In Table 8 we show the battery consumptions obtained for some of the granularities.

In the apnea monitor, we also performed these experiments in order to know the battery consumption. However, as the input data size is always of 60 s we only show the average consumption in mA required for signal input packages of 60 s: $c_a^{local}(60) = 0.11$, $c_s(60) = 0.15$ and $c_c(60) = 0.09$. Moreover, this time is not so critical due to fact that the PDA can be connected to the electrical plug while the user is sleeping.

7.6. *Global experiments: the optimal configuration*

Once all the constants and functions in the cost model have been set, then an optimal configuration can be calculated. We have written a module that calculates the cost for all the possible configurations (a set of different granularities, with or without compression, with local or remote analysis) and identifies the optimal configuration. For the arrhythmia monitor, the optimal configuration is the one that uses granularity of 4 s, and applies compression and local analysis (with a cost of 1,080,578,377). However, for the apnea monitor, the optimal configuration is the one that uses a granularity of 2 s (every 2 s, it analyses the last 1-min signal), and does not apply compression but uses local analysis (with a cost of 6,015,441,132).

In Fig. 9 we can see which are those best configurations, and also other interesting data. For example, we can see, for the arrhythmia case (graph on the left), that applying compression techniques is in general better than not applying them. And also, for the apnea monitoring case graph on the right), that it is not worthy applying compression techniques when performing local analysis.

Finally, we can say that the optimal configuration can be calculated dynamically in runtime. Depending on the actual conditions: battery power that is left, availability of a non-expensive and more reliable network like WIFI, WiMAX, or even a wired one instead of GPRS/UMTS, mobile device plugged to a power supply, etc. then the weights of the different factors w_{aut}, w_{vol} and w_{lat} could be changed and an optimal configuration could automatically be calculated by the optimization module and set up. Of course, some of the user-dependent values such as P_{alarm}, average heart rate hr, could be continuously updated depending on the actual number of high-risk arrhythmias and apnea episodes detected and the heart rate.

8. Conclusions

The goal of this paper is twofold. First, to present our proposal: an architecture for Mobile Biological Sensor Data Streams (MBSDS) that supports monitoring systems that manage data streams generated by biological mobile sensors. Moreover, we have proved its suitability with two real systems: an ECG monitoring system whose goal is to provide an anywhere and at any time assistance to persons that suffer from heart arrhythmias, and an apnea monitoring system whose goal is to detect apneas in real-time. Those systems have already been implemented, validated with the MIT-BIH arrhythmia and Apnea–ECG databases and offer accurate results.

Second, knowing that, in order to be useful, these kind of systems need to be efficient, we have also proposed a cost model that takes into account the different factors that can be optimized (volume, latency and autonomy) and that guides in finding an optimal configuration for the monitoring system. Moreover, several parameters and techniques are taken into account by the cost model such as the optimal granularity, measuring if local analysis is worth being applied, compressing the data streams by working with different compression algorithms, adapting frequency of processor, or disabling brightness and sound in the PDA. This cost model can be applied to any monitoring system that follows an MBSDS architecture. We have also explained how that model can be customized for each particular monitoring system, and we have illustrated the customization process for the arrhythmia and apnea monitoring systems.

REFERENCES

[1] D. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J. Hwang, W. Linder, A. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, S. Zdonik, The design of the Borealis stream processing engine, in: CIDR Conference, 2005.

[2] D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, C. Erwin, E. Galvez, M. Hatoun, J. Hwang, A. Maskey, A. Rasin, A. Singer, M. Stonebraker, N. Tatbul, Y. Xing, R. Yan, S. Zdonik, Aurora: a data stream management system, in: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'03), San Diego, June 2003.

[3] D. Abadi, W. Linder, S. Madden, J. Schuler, An integration framework for sensor networks and data stream management systems, in: Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004.

[4] ActiveECG Active ECG, 2009, http://www.activecenter.com.

[5] A. Arasu, B. Babcock, S. Babu, M. Datar, K. Ito, R. Motwani, I. Nishizawa, U. Srivastava, D. Thomas, R. Varma, J. Widom, STREAM: the Stanford stream data manager, IEEE Data Eng. Bull. 26 (1) (2003) 19–26.

[6] J. Bai, Y. Zhang, D. Shen, L. Wen, C. Ding, Z. Cui, F. Tian, B. Yu, B. Dai, J. Zhang, A portable ECG and blood pressure telemonitoring system, IEEE Eng. Med. Biol. Mag. 4 (18) (1999) 63–70.

[7] E. Berruto, G. Colombo, P. Monogioudis, A. Napolitano, K. Sabatakakis, Architectural aspects for the evolution of mobile communications toward UMTS, IEEE J. Select. Areas Commun. 15 (8) (1997) 1477–1487.

[8] P. Bonnet, J. Gehrke, P. Seshadri, Towards sensor database systems, in: Proceedings of the International Conference on Mobile Data Management, 2001, pp. 3–14.

[9] A. Burgos, A.G. Ni, A. Illarramendi, J. Bermúdez, Real-time detection of ECGs on a PDA, IEEE Trans. Inform. Technol. Biomed., submitted for publication, in revision.

[10] CardioControl, Cardio Perfect CE, 2009, http://www.cardiocontrol-us.com/.

[11] S. Chandrasekaran, O. Cooper, A. Deshpande, M. Franklin, J. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss, M. Shah, TelegraphCQ: An architectural status report, IEEE Data Eng. Bull. 26 (March (1)) (2003).

[12] C. Cranor, T. Johnson, O. Spataschek, V. Shkapenyuk, Gigascope: a stream database for network applications, in: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, ACM Press, New York, NY, USA, 2003, pp. 647–651.

[13] N. Daja, I. Relgin, B. Reljin, Telemonitoring in cardiology-ECG transmission by mobile phone, J. Ann. Acad. Stud. (2001) 4.

[14] C. Faloutsos, Sensor data mining: similarity search and pattern analysis, in: Tutorial in Proceedings of the International Conference on VLDB, 2002.

[15] Farreras and Rozman Medicina Interna, in: Aula Magna (Ed.), 13rd ed. in CD-ROM, 2001, pp. 395–523.

[16] M. Gaber, S. Krishnaswamy, A. Zaslavsky, A wireless data stream mining model, in: Proceedings of the Third International Workshop on Wireless Information Systems (WIS), 2004.

[17] M.M. Gaber, S. Krishnaswamy, A. Zaslavsky, Cost-efficient mining techniques for data streams, in: E. ACS (Ed.), Australasian Workshop on Data Mining and Web Intelligence (DMWI2004), Dunedin, New Zealand. CRPIT, 32, Purvis, M., 2004.

[18] M.M. Gaber, S. Krishnaswamy, A. Zaslavsky, Ubiquitous data stream mining, in: Proceedings of the Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining, Sydney, Australia, May 2004.

[19] M.M. Gaber, A. Zaslavsky, S. Krishnaswamy, Mining data streams: a review, SIGMOD Rec. 34 (June (2)) (2005).

[20] B. Ghribi, L. Logrippo, Understanding GPRS: the GSM packet radio service, Comput. Netw. 34 (5) (2000) 763–779.

[21] IEEE, WiFi, 2009, http://grouper.ieee.org/groups/802/11/.

[22] S. Jalaleddine, C. Hutchens, R.D. Strattan, W.A. Coberly, ECG data compression techniques a unified approach, IEEE Trans. Biomed. Eng. 4 (April (37)) (1990) 329–343.

[23] R. Jané, A. Blasi, J. García, P. Laguna, Evaluation of an automatic threshold based detector of waveform limits in holter ECG with the QT database, Comput. Cardiol. 24 (1997) 295–298.

[24] H. Kargupta, B. Park, S. Pittie, L. Liu, D. Kushraj, K. Sarkar, MobiMine: monitoring the stock market from PDA, SIGKDD 3 (2) (2002).

[25] E. Keogh, Tutorial 4: A decade of progress in indexing and mining large time series databases, VLDB Tutorial (2006).

[26] D. Konstansas, V. Jones, R. Hersog, MobiHealth—innovative 2.5/3g mobile services and applications for healthcare, in: Workshop on Standardization in E-Health, Geneva, Italy, 2003.

[27] C. Kunze, U. Grossmann, W. Stork, K. Müller-Glaser, Application of ubiquitous computing in personal health monitoring systems, in: Proceedings of the 36th Annual Meeting of the German Society for Biomedical Engineering, 2002.

[28] LAMP Linux and Managed Power, http://gandalf.ics.uic.edu/haldar/lamp/.

[29] P.M. Limited, ECG System, MeditSense 100, 2009, http://www.pulsemedical.co.uk.

[30] S. Madden, M.J. Franklin, J.M. Hellerstein, W. Hong, TinyDB: An acquisitional query processing system for sensor networks, ACM TODS 30 (March (1)) (2005) 122–173.

[31] Nonin, Nonin SpO_2 Sensors, http://www.nonin.com/.

[32] Physionet, The Research Resource for Complex Physiologic Signals, http://www.physionet.org/.

[33] Pulsox-300/Pulsox-300i, Pulsox-300/Pulsox-300i, 2009, http://www.konicaminolta.com/.

[34] QRSDiagnostic, PDAs-Mobile Diagnostic Workstations, qrs White Paper, http://www.qrsdiagnostic.com.

[35] J. Rodríguez, A. Goñi, Ni, A. Illarramendi, Real-time classification of ECGs on a PDA, IEEE Trans. Inform. Technol. Biomed. 9 (March (1)) (2005) 23–34.

[36] I. Sachpazidis, @Home: a modular telemedicine system, in: Second Workshop on Mobile Computing in Medicine, Heidelberg, Germany, 2002.

[37] M. Sharaf, J. Beaver, A. Labrinidis, P.K. Chrysanthis, Balancing energy efficiency and quality of aggregate data in sensor networks, VLDB J. (2004).
[38] L. Sörnmo, P. Laguna, Bioelectrical Signal Processing in Cardiac and Neurological Applications, Elsevier Academic Press, 2005.
[39] U. Srivastava, K. Munagala, J. Widom, Operator placement for in-network stream query processing, ACM PODS (2005).
[40] S. Viglas, J.F. Naughton, Rate-based query optimization for streaming information source, in: ACM SIGMOD, USA, 2002.
[41] WiMAX Forum, http://www.wimaxforum.org/home.