

Context-aware energy-efficient wireless sensor architecture for body activity recognition

Tifenn Rault, Abdelmadjid Bouabdallah, Yacine Challal, Frédéric Marin

▶ **To cite this version:**

Tifenn Rault, Abdelmadjid Bouabdallah, Yacine Challal, Frédéric Marin. Context-aware energy-efficient wireless sensor architecture for body activity recognition. IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Mar 2014, Budapest, Hungary. pp.203-206, 10.1109/PerComW.2014.6815201 . hal-01009394

HAL Id: hal-01009394
https://hal.science/hal-01009394

Submitted on 18 Jun 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Context-aware energy-efficient wireless sensor architecture for body activity recognition

Tifenn Rault, Abdelmadjid Bouabdallah, Yacine Challal
Université de Technologie de Compiègne
Heudiasyc UMR CNRS 7253
60205 Compiègne, France

Frédéric Marin
Université de Technologie de Compiègne
BMBI UMR CNRS 7338
60205 Compiègne, France

***Abstract*—In this paper, we present EEWAA, a new Energy-Efficient Wireless sensor Architecture for Activity recognition Applications. Contrary to existing approaches, our solution aims at optimizing the lifetime of every battery-operated devices that may contribute to the activity recognition, including sensors but also smart phones or tablets. Depending on the environment opportunities, our approach balance the load among multiple base stations (mobile phone, tablet, laptop) and when beneficial, enables the energy-limited base stations and/or sensors to offload a part of their tasks to back-end servers. In order to do this, we propose a new decision metric to select the best execution configuration, considering the latency and accuracy requirements, as well as the energy of every device involved in the computation.**

I. Introduction

The recognition of human activities has received a significant attention in the medical field, because it is a task of interest for many applications, including assisted living of elderlies, physical assessment, and supervision of patient with chronic diseases such as diabetes, obesity, cognitive disorder or arrhythmia [1]. In a typical scenario, wearable sensors like accelerometers or ECG are attached to the patient to measure inertial and physiological data. These data are further analyzed to provide a feedback to caregivers who can assess the efficiency of a new treatment, adjust medication, better study a disease or supervise a patient behavior [2]. In practice, such systems face a major challenge that limit their widespread adoption : the significant energy consumption of battery-operated devices for continuous sensing and communications. Indeed, radio transmissions, data processing and sensing tasks actively stress the nodes reserves, whereas they are expected to operate during the course of the day without maintenance.

To tackle this problem, many solutions have been proposed to save energy in activity recognition applications. Among them we can cite sensor set selection [3]–[5], deactivation of power-hungry sensors [2], [6], [7], adaptive sampling rate [8], [9], communications reduction [10], duty-cycling [11] and resource sharing [12], [13]. However, these works are usually interested in the energy consumption of the sensors but never consider that the base station can also be energy-constrained. Traditionally, powerful laptop acted as base station, but for mobility consideration most of the works now consider mobile phone as base station. Although phones are more powerful devices compared to wearable sensors, the truth is that they also face energy consumption problems [14].

We believe that activity recognition architectures can take advantage of the proliferation of smart devices (phones, tablets, laptops) and their daily usage to balance the load among base stations. For example, when at home and at the office, body sensors can communicate with a laptop or an access point. When outdoor, a mobile phone or a tablet can alternatively take the relay depending on their availability and battery state. Furthermore, additional energy can be saved at a base station through computation offloading, that is, a base station can execute (a part of) its tasks on a back-end server. This approach has proven to be energy-efficient under some conditions [13], [15], [16], and the usage of devices with higher computing, storage and energy capabilities allows to incorporate more complex algorithms and models. As computation offloading is a trade-off between energy and performances, we propose a new metric to determine when offloading is beneficial, based on the energy consumption of every devices involved in activity recognition, the latency and the classification accuracy. Besides, our solution is self-adaptive to the environment, that is the balancing and the offloading strategies are highly dependent on the available connections and devices, as well as their internal state. Finally, our architecture is complementary to the energy-efficient mechanisms already proposed in the literature, as we are still able to implement them to further enhance the system lifetime.

In what follows, in Section II we study traditional architectures for activity recognition applications. In section III, we present the general idea of our EEWAA architecture. In section IV, we propose a new metric for offloading decisions. Section V concludes the paper and highlights challenges.

II. Activity recognition

The main stages for activity recognition include *sensing*, *processing* and *analysis*, as illustrated in Figure 1. More specifically, sensing corresponds to the data acquisition at sensor nodes. Data processing generally consists in filters and features extraction to obtain relevant characteristics of the signal. The analysis refers to the activity recognition which is done through classification. Depending on where these tasks are executed, the energy/performance profile of the body sensor network varies. For example, the sensors have the possibility to directly store the raw data, leaving all the processing for later. The drawbacks of this method

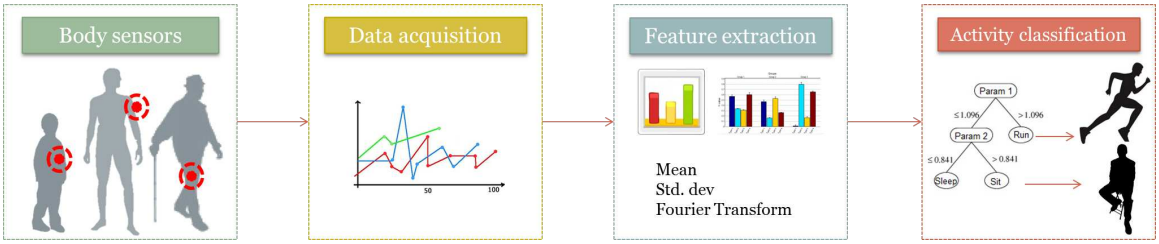


Fig. 1. The main stages of activity recognition applications.

is that it prevents real-time feedback to the patient and it requires a lot of memory, even if to save space, the sensor can store extracted features. Another solution consists in sending the raw data to the base station, but it results in a huge amount of energy consumption due to transmissions. In the meantime, using a more powerful base station allows to have access to more complex algorithms and models. Furthermore, to save communications, the sensor can send only the extracted features, the compressed signal or the results, which decreases communication cost but increases the computation load. From the energy point of view it may be optimal, but sensor scarce resources limit the implementation to lightweight algorithms. Actually, some classification algorithms are very expensive in terms of memory and computation requirements which makes them not suitable for sensor platforms [1].

Wang et al. [17] compare the energy/performance trade off between an on-node scheme where the classification is performed on the accelerometer, and an off-node scheme where the accelerometer sends its raw data to the base station. They show that with the on-node scheme 40% energy can be saved with 13% reduction in classification accuracy. So, their results suggest that it is more energy-efficient to perform all the tasks on the sensors. However, we can make several observations to nuance this statement. First, as mentioned by the authors, if the user stays in active state or if he/she changes activity often, on-node scheme may consume more energy due to additional processing and data communications. Second, the study only consider accelerometers, but other kind of sensors may require more complex processing like ECG. In this case, on-node signal processing may not be possible, or at the cost of accuracy loss. Third, computation offloading is more energy-efficient than local computation in some scenarios. For example, in SociableSense [16] it is more energy-efficient to offload the speaker identification task when using a microphone. In METIS [16], sensing offloading can save up to 35% energy over pure phone sensing. Fourth, energy is not the unique metric of interest and some applications may not accept a loss in accuracy. In these cases, leveraging energy consumption for more accurate classification may be necessary to maintain the desired performance.

As we can see, a static architecture can not satisfy the requirements of every body sensor networks. Indeed, activity recognition applications are constrained by the types of sensors and processing to perform, the patient behavior and the resource availability. That's why we propose a more general architecture that optimizes the energy/performance trade-off depending on the applications requirements and the environment opportunities. The idea is to switch from an operating mode to another when there is a change in the context. For example, if a patient leaves its office to go at a meeting, the sensors can stop sending their data to the PC and redirect their transmissions towards the patient's phone. If the phone has enough energy to last until the end of the day, it can process the received signals. Otherwise, the phone will only carry out the classification task. So, EEWAA supports many execution scenarios thanks to context-aware adaptation, which enables us to leverage energy consumption for accuracy and vice versa.

III. EEWAA ARCHITECTURE

Our architecture is composed of:

- A set of body sensors (e.g. accelerometer, EMG, gyroscope, ECG).
- A set of ambient sensors (e.g. a microphone in a room, an RFID tag on objects, a GPS in a phone).
- A set of candidate base stations, mobile or not, with different energy, storage and computing capabilities (e.g. laptop, mobile phone, tablet, access point).
- A back-end server.

At a given time t, only a subset of base stations may be available (possibly none), for two reasons : a base station may not be in the patient environment or it is in the patient environment but there is no wireless connections between the sensors and the base station. Consequently, the connection to the back-end server does not always exists. Similarly, there may not be ambient sensors in the environment. We denote by $D_t = \{d_1, d_2, .., d_q\}$ the set of devices available at time t. Each device (sensor/base station/server) can execute a set of tasks, like sensing, signal processing, classification or result display. For a given environment, there exist several execution configurations, depending on where each task is executed and resulting in different energy/performance profile. Through the load balancing and offloading strategies, the objective of EEWAA is to select the configuration that achieves a satisfactory tradeoff. We give in section IV a new metric for configuration selection.

In Figure 2 we illustrate an execution configuration. In this scenario, the phone is elected as base station (BS). The phone is equipped with a microphone, but it offloads its sensing task to a microphone embedded in the environment. The BS

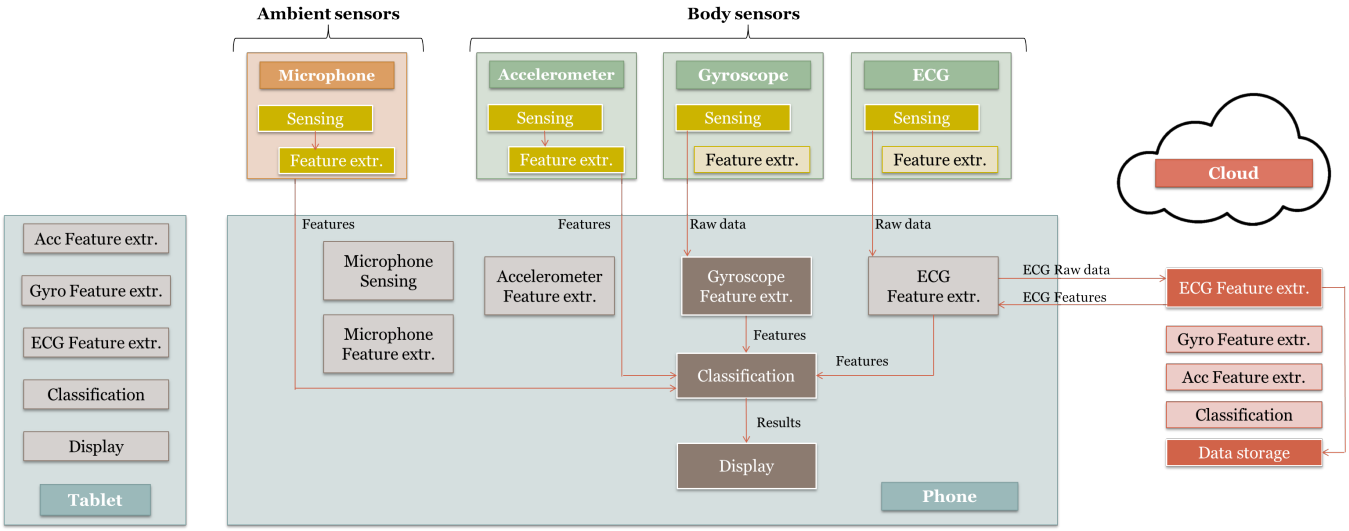


Fig. 2. An example of an execution configuration.

receives extracted features from the accelerometer and the microphone, and it receives raw data from the ECG and the gyroscope. The BS computes the features for the gyroscope but it offloads the task for the ECG. So, the back-end server perform the feature extraction of ECG data, send the result back to the phone and store them in the data base. Finally, the phone is able to perform the classification using the features extracted from every signal and display the result. We can imagine that after a while, if the phone battery has decreased below a threshold, the tablet can take the relay as base station.

We first construct a matrix of constraints, where (i, j) represents the costs of computing a task $j \in [1, n]$ on a device $i \in [1, q]$, as illustrated in Table I. If the device i cannot execute the task j, then the cost is set to infinity. The cost may be a tuple of different costs, e.g. (e_{ij}, l_{ij}) which are respectively the energy and latency costs of executing the task j on device i. We then periodically select a configuration using the metric defined in section IV, so that every task is assigned to one and only one device.

TABLE I
MATRIX OF CONSTRAINTS

	task 1	task 2	...	task n
d_1	(e_{11}, l_{11})	(e_{12}, l_{12})	...	∞
d_2	∞	∞	...	(e_{2n}, l_{2n})
:	:	:	:	:
d_q	(e_{q1}, l_{q1})	∞	...	∞

IV. OFFLOADING STRATEGIES

In this section, we present a new metric for computation offloading decision in EEWAA. Our objective is to decide where to execute each task, based on energy-efficiency, latency and accuracy requirements. It is often assumed a client/server architecture, where a mobile device can offload a part of its tasks to a back-end server. In this scenario, only the energy consumption of the mobile is of interest. This is the case of SociableSense [16], an architecture dedicated to activity recognition for sociable interactions, where a phone with integrated sensors can offload its tasks to the cloud. SociableSense selects the configuration c_i that maximizes a weighted additive utility function (1), which takes into consideration the energy consumption, the latency and the amount of data sent over the network.

$$u_{c_i} = w_e u_{e_i} + w_l u_{l_i} + w_d u_{d_i} \quad (1)$$

$$\text{with } u_{x_i} = \frac{x_{min} - x_i}{x_i} \; x \in (e, l, d) \quad (2)$$

where e_i is the energy consumption of the architecture when choosing the configuration c_i, l_i is the resulting latency and d_i is the amount of data transmitted over the network. Depending on the phone battery state, more prevalence can be given to the energy metric by adjusting the weights w_i dynamically.

Few works have considered that mobile devices can offload their data between them [18], [19]. In this case, the energy of every device is of interest. Mtibaa et al. [19] maximize the lifetime of the whole network, which is define as the time until the first mobile runs out of energy. They take into consideration the cost for computing locally E_{comp} and the costs for transmitting and receiving E_{comm}. A node u offloads a task to a neighboring node v if the response time of migrating the task to v is within the task deadline and the estimated remaining energy is maximized as in Eq. (3) [19].

$$max_{v \in G}\{(E_t^u - E_{comm(u,v)}) + (E_t^v - E_{comp(v)} - E_{comm(v,u)})\} \quad (3)$$

where E_t^u is the remaining energy of node u. After selecting v, if $v == u$ the node executes its task locally, otherwise it offloads the task. The results show that this approach manages to fairly distribute the load among nodes.

Shi et al. [18] maximizes the lifetime of the first depleted node under the constraint that tasks complete before their deadline. They minimize the following utility function to take into consideration the energy consumption of all nodes

involved in the remote computing and the residual energy available to these nodes:

$$u(T) = \sum_{i \in N_T} \frac{e_{Ti}}{R_i} \quad (4)$$

where N_T is the set of nodes involved in the remote computing of task T, e_{Ti} is the energy consumption of node i for task T, and R_i is the residual energy of node i [18].

We propose a new metric for decision offloading in EEWAA, that takes into account the application accuracy, the latency, and the energy spent by every battery-operated devices involved in the activity recognition. We aim at maximizing a weight additive utility function as in [16], while considering :

- δ_i, the error accuracy of the classifier used in configuration c_i.
- l_i, the latency of the application in configuration c_i.
- e_i, the energy consumption of the architecture in c_i.

$$u_{c_i} = w_e u_{e_i} + w_l u_{l_i} + w_\delta u_{\delta_i} \quad (5)$$

We can define e_i as the sum of the energy spent in every battery-operated devices involved in the computation over their residual energy as in [18]. However, it is not fair because phones are not only dedicated to the health supervision application, so the fraction of their energy dedicated to the activity recognition application must not be as high as the one of the sensors. In other words, a part of phones energy must be preserved to be dedicated to other tasks, like calling or text messaging. That's why we propose to introduce a factor α_j that represents the desired implication of the device j in the application. We redefine e_i as in Eq (6):

$$e_i = \sum_{j \in N_i} \alpha_j \frac{E_{ij}}{R_j} \quad (6)$$

where N_i is the set of devices involved in the computation in configuration c_i, E_{ij} is the energy spent by the device j in configuration c_i, R_j is its residual energy and α_j is a weight that represents how much a device can dedicate its energy to the activity recognition application. An α_j equal to 1 means that the device is exclusively dedicated to the healthcare application, while an higher value means that the device will not spend all its energy in the application. The energy consumption E_{ij} of a device j in c_i includes the energy spent for sensing, computing and communications. Notably, the energy spent in communications will depend on the size of data to be transferred [15], the bandwidth availability [20] and the communication technology (Wi-Fi, 3G) [21].

V. Conclusion

This paper describes the work in progress towards the development of EEWAA, an Energy-Efficient Architecture for Activity recognition Applications. EEWAA adapts its operating mode to context changes in order to take advantage of the opportunities provided by the environment. Many challenges need to be addressed. First, it will be necessary to manage data fusion and synchronization when signal processing are distributed over multiple devices, like in the scenario of Figure 2. Moreover, the presence of a remote server can be exploited to implement more complex classification algorithms based on prediction. It would be possible to progressively construct of a data base of the patient habits and infer additional knowledge.

Acknowledgments

This work was carried out and funded in the framework of the Labex MS2T. It was supported by the French Government, through the program "Investments for the future" managed by the National Agency for Research (ANR-11-IDEX-0004-02).

References

[1] O. D. Lara and M. A. Labrador, "A Survey on Human Activity Recognition based on Wearable Sensors," *IEEE Communications Surveys and Tutorials*, 2013.

[2] K. Lorincz, B. rong Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: a wearable sensor network platform for high-fidelity motion analysis," *ACM SenSys*, 2009.

[3] H. Noshadi, F. Dabiri, S. Meguerdichian, M. Potkonjak, and M. Sarrafzadeh, "Energy optimization in wireless medical systems using physiological behavior," *Wireless Health*, 2010.

[4] P. Zappi, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Troster, "Activity recognition from on-body sensors by classifier fusion: sensor scalability and robustness," *IEEE ISSNIP*, pp. 281–286, 2007.

[5] L. Gao, A. K. Bourke, and J. Nelson, "Activity recognition using dynamic multiple sensor fusion in body sensor networks," *EMBS*, 2012.

[6] F.-T. Sun, C. Kuo, and M. Griss, "PEAR: Power efficiency through activity recognition (for ECG-based sensing)," *PervasiveHealth*, 2011.

[7] T. Park, J. Lee, I. Hwang, C. Yoo, L. Nachman, and J. Song, "E-Gesture: a collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices," *ACM SenSys*, 2011.

[8] X. Qi, M. Keally, G. Zhou, Y. Li, and Z. Ren, "AdaSense: Adapting Sampling Rates for Activity Recognition in Body Sensor Networks," *IEEE RTAS*, 2013.

[9] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, "Energy-Efficient Continuous Activity Recognition on Mobile Phones: An Activity-Adaptive Approach," *ISWC*, 2012.

[10] L. Wang, T. Gu, X. Tao, and J. Lu, "A hierarchical approach to real-time activity recognition in body sensor networks," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 115–130, feb 2012.

[11] L. Au, M. Batalin, T. Stathopoulos, A. Bui, and W. Kaiser, "Episodic sampling: Towards energy-efficient patient monitoring with wearable sensors," *IEEE EMBS*, 2009.

[12] M. Keally, G. Zhou, G. Xing, and J. Wu, "Remora: Sensing resource sharing among smartphone-based body sensor networks," *IEEE/ACM International Symposium on Quality of Service - IWQoS*, 2013.

[13] K. Rachuri, C. Efstratiou, I. Leontiadis, C. Mascolo, and P. Rentfrow, "METIS: Exploring mobile phone sensing offloading for efficiently supporting social sensing applications," *IEEE PERCOM*, 2013.

[14] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," *ACM SIGCOMM IMC*, 2009.

[15] K. Kumar and Y.-H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy," *Computer*, 2010.

[16] K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow, "SociableSense: exploring the trade-offs of adaptive sampling and computation offloading for social sensing," *ACM MobiCom*, 2011.

[17] N. Wang, G. V. Merrett, R. G. Maunder, and A. Rogers, "Energy and Accuracy Trade-Offs in Accelerometry-Based Activity Recognition," *International Workshop on Sensor Networks*, 2013.

[18] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: enabling remote computing among intermittently connected mobile devices," *ACM MobiHoc*, 2012.

[19] A. Mtibaa, A. Fahim, K. A. Harras, and M. H. Ammar, "Towards resource sharing in mobile device clouds: power balancing across mobile devices," *ACM SIGCOMM MCC*, 2013.

[20] R. Wolski, , S. Gurun, C. Krintz, and D. Nurmi, "Using bandwidth data to make computation offloading decisions," *IEEE IPDPS*, 2008.

[21] E. Cuervo, A. Balasubramanian, D. ki Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: making smartphones last longer with code offload," *MobiSys*, 2010.