



**HAL**  
open science

## **Cardiac Monitoring of Marathon Runners using Disruption-Tolerant Wireless Sensors**

Djamel Benferhat, Frédéric Guidec, Patrice Quinton

► **To cite this version:**

Djamel Benferhat, Frédéric Guidec, Patrice Quinton. Cardiac Monitoring of Marathon Runners using Disruption-Tolerant Wireless Sensors. 6th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI'12), Dec 2012, Vitoria-Gasteiz, Spain. pp.395-402. <hal-00763319>

**HAL Id: hal-00763319**

**<https://hal.science/hal-00763319v1>**

Submitted on 10 Dec 2012

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.



HAL Authorization

# Cardiac Monitoring of Marathon Runners using Disruption-Tolerant Wireless Sensors

Djamel Benferhat\*, Frédéric Guidec\*, and Patrice Quinton\*\*

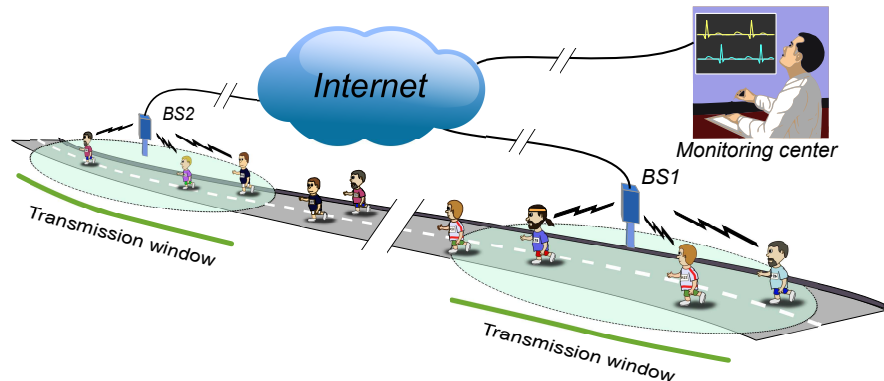
IRISA, \* Université de Bretagne-Sud, \*\* ENS Cachan Bretagne, France

**Abstract.** In most current biomedical monitoring applications, data acquired by sensors attached to a patient are either transmitted directly to a monitoring console for real-time processing, or they are simply recorded on the sensor unit for deferred analysis. In contrast collecting and transmitting biomedical data continuously over long distances in outdoor conditions is still a challenge. In this paper we investigate the possibility of using disruption-tolerant wireless sensors to monitor the cardiac activity of runners during a marathon race, using off-the-shelf sensing devices and a limited number of base stations deployed along the marathon route. Preliminary experiments conducted with a few volunteers running around a university campus confirm that this approach is viable, and suggest that it should scale up to a real marathon.

## 1 Introduction

In this paper we investigate the possibility of using off-the-shelf wireless sensors to monitor the cardiac activity of runners during a marathon race. In such a scenario the runners must cover a long distance during the race, and that distance clearly exceeds the very limited radio range of the low-power IEEE 802.15.4 (ZigBee) or IEEE 802.15.1 (Bluetooth) radio transceivers available on most current sensors. However, since runners in a marathon all follow exactly the same route, a number of base stations (BS) with broadband connectivity to the Internet can be deployed on the roadside (see Figure 1). Whenever a runner passes by a BS, ECG data received from this runner's ECG sensor can be forwarded to a remote site, such as the closest medical aid station or a physician's desktop, laptop, or smartphone.

In previous work we considered using IEEE 802.15.4 (ZigBee) transceivers to implement the wireless link between sensors and roadside stations [1]. Each roadside station then took the form of a netbook or laptop with a ZigBee interface. Yet this approach proved ineffective and unable to support a large number of runners, mostly because of the low bitrate and low transmission range allowed by the IEEE 802.15.4 standard. In the current work the base stations take the form of standard Wi-Fi access points. Since off-the-shelf ECG sensing devices do not feature Wi-Fi interfaces, each runner must carry a smartphone (additionally to the ECG sensor) that can serve as a relay between the sensor and base stations. With this configuration the ECG data stream produced by an ECG sensor



**Fig. 1.** Illustration of disruption-tolerant ECG monitoring of marathon runners

is transmitted directly and continuously to the associated smartphone through a Bluetooth RFCOMM link. The smartphone processes this data stream, and forwards the resulting data bundles whenever possible to roadside base stations using Wi-Fi wireless links.

Current biomedical applications usually either perform data recording for deferred analysis, or they depend on continuous connectivity for real-time transmission and processing. For example projects Mobihealth and HYGEIAnet assume continuous network connectivity, as they aim at developing new healthcare mobile services using 2.5/3G technology for physiological data collection [2,3]. Data recording for deferred analysis is notably used in [4,5], which rely on wearable recording sensors for health monitoring during sports activities. To the best of our knowledge the utilization of disruption-tolerant techniques to monitor athletes in outdoor conditions has not been investigated much so far, although disruption-tolerant solutions for *non-biomedical* sensor-based applications have already been proposed in the literature [3,6].

The remainder of this paper is organized as follows. In Section 2 we estimate the number of base stations required to would be required to cover the full length of a marathon route, and show that a sparse deployment of base stations can be sufficient to cover this route satisfactorily. The SHIMMER sensor we use in this project for ECG data acquisition is described in Section 3. Section 4 presents the main features of the transmission chain we designed in order to support the disruption-tolerant transmission of ECG data between runners and a monitoring center. In Section 5 we report on a field trial we conducted on our university campus in order to validate this work. Section 6 concludes this paper.

## 2 Rationale for a sparse coverage of the marathon route

In outdoor conditions, the transmission range between a mobile smartphone and a fixed Wi-Fi access point can sometimes exceed 300 meters. Of course this distance is often diminished because of obstacles or radio interferences. Let us

assume that the connectivity between mobile smartphone and a base station (basically an access point) is guaranteed on an average distance of 100 meters. A base station deployed along a marathon route would thus cover about 200 meters of that route, and at least 200 base stations would be required to ensure a full coverage of the 42.2 km route.

Since deploying hundreds of base stations along a marathon route is hardly an option for organizational and financial reasons, our approach is based on the premise that only a sparse coverage of the route needs to be ensured, using a reasonable number of base stations. A disruption-tolerant solution for data gathering must therefore be implemented, using the store-carry-and-forward principle. This principle is the foundation of *Disruption-Tolerant Networking (DTN)*: a mobile node that is temporarily disconnected from the rest of the network can *store* data (or messages) in a local cache, *carry* these data for a while, and *forward* them later when circumstances permit [7]. In our scenario, the ECG sensor carried by a runner captures data continuously and transmits these data to the smartphone, which stores these data locally. Whenever the runner passes by a BS, a transient radio contact occurs between the smartphone and that BS. This contact is exploited by the smartphone to upload bundles of data to the BS, which in turn can forward these bundles to the monitoring center (see Fig. 1). The distance between successive base stations and the speed of the runner determine how often “fresh” data can be sent to the monitoring center. According to cardiologists, a physician monitoring the cardiac activity of marathon runners should receive updated data for each runner at least every 5 to 10 minutes, in order to be able to detect arrhythmias and prevent incidents. Considering the pace of an average runner this implies that base stations should be placed about 1 to 2 km apart. With this approach, ten to twenty base stations are sufficient to cover a marathon route satisfactorily.

### 3 Overview of SHIMMER sensors

In this project we use SHIMMER platforms in order to acquire ECG data on each runner, and HTC Android smartphones as disruption-tolerant relays between SHIMMER sensors and roadside base stations.

The SHIMMER platform is a programmable lightweight wireless sensing system that can record and transmit physiological and kinematic data in real-time [8]. Data acquisition on the SHIMMER platform is performed on up to 8 channels through a 12-bit AD converter. Several kinds of expansion modules are available: the ECG module we use provides RA-LL (Right Arm - Left Leg) and LA-LL (Left Arm - Left Leg) input leads. The RA-LA (Right Arm - Left Arm) lead can be calculated based on the first two leads.

Two built-in radio transceivers operating in the 2.4 GHz ISM band are available: an IEEE 802.15.4/ZigBee compliant CC2420 transceiver, and a WML-C46A class 2 Bluetooth transceiver.

## 4 Protocol for data acquisition and transmission

We developed specific code in nesC (a dialect of C) for the SHIMMER sensors, and in Java for Android smartphones.

### 4.1 Data acquisition on a SHIMMER sensor

The acquisition of ECG data on each SHIMMER sensor is performed on two 12-bit channels (RA-LL and LA-LL leads), with a sampling frequency that can be adjusted as needed. The data stream hence produced is transmitted on-the-fly to the smartphone through a Bluetooth RFCOMM link.

### 4.2 Transmission between SHIMMER sensor and Android smartphone

Each SHIMMER sensor must be paired with a specific smartphone, and two paired devices must of course be carried by the same marathon runner. The Java application we designed for Android smartphones allows a user to locate nearby SHIMMER sensors, and to pair the smartphone with one or several of these sensors, using secured pairing if desired. The possibility for a single smartphone to collect data from several sensors is a provision for future work: several sensors may thus be attached to a single athlete, so different kinds of data can be collected simultaneously.

Once a smartphone is paired with a sensor, an RFCOMM link is established between them. Through this link the smartphone can control the sensor, and send simple commands in order to adjust the sampling frequency or resolution, to start or stop the data acquisition, etc. When data acquisition is enabled on a sensor, a continuous data stream is sent to the smartphone through the RFCOMM link.

The code we designed for both SHIMMER sensors and Android smartphones can tolerate transient disruptions in RFCOMM links. For example, if paired sensor and smartphone get disconnected for a while, they strive to re-establish the connection, and data transmission (if enabled) resumes as soon as the connection is re-established.

The data stream received by the smartphone is packetized in small bundles, which are then stored in the smartphone's SD-card, awaiting for transmission to the monitoring center. Each bundle consists of a header and a payload. The header includes an identifier of the source sensor and a timestamp. The payload is simply a byte array that contains a sequence of data bytes received from the sensor. The size of this byte array depends on the data acquisition frequency and resolution on the sensor, as well as on the period set for data bundling. For example, data acquisition on two 12-bit channels with 200 Hz sampling produces a continuous data stream at 4.8 kbps. Assuming a bundle is produced every 20 seconds on the smartphone, each bundle contains a payload of 12 kiB.

### 4.3 Transmission between SHIMMER sensor and roadside base station

As mentioned in the former section a base station is typically a standard Wi-Fi access point with broadband connectivity to the Internet. When a runner passes close to a base station, the smartphone he/she is carrying detects the access point and tries to associate with it. If the association succeeds, the smartphone authenticates if needed, and sends a DHCP request in order to obtain IP parameters from a DHCP server. Once the IP connectivity is obtained, the Android application we designed starts uploading data bundles to a remote server that is the entry point of the monitoring center.

The transmission of data bundles between smartphone and server is performed using UDP datagrams. Each bundle of ECG data easily fits in a single datagram, so no fragmentation is required at this level. An ARQ (Automatic Repeat reQuest) mechanism is however implemented in order to prevent any loss of data bundle.

*Rate control.* When a smartphone carried by a runner establishes a connection with a base station these devices are still far away from each other (see Fig. 1). The quality of the wireless link between them is usually quite low, so sending data bundles too hastily in such conditions could yield a high level of data loss. In order to prevent this problem our Android application implements a rate control mechanism that is inspired from TCP's slow-start mechanism. At the beginning of a contact window –when the smartphone has just received IP parameters from the DHCP server– a simple stop-and-wait method is used: the smartphone only sends one bundle, and waits for an acknowledgement that this bundle has been received by the server. Once this acknowledgement is received the next data bundles are sent with a go-back-N method: up to N bundles can be sent in a row, before receiving the acknowledgement of the first of these bundles. The width of the sliding window (N) can be adjusted dynamically as the runner moves closer to the access point, then farther from this access point. Experience confirms that the combination of both ARQ methods allows an efficient use of the contact window between smartphone and base station, and does not induce a high level of transmission failures at the beginning of a contact window.

*Uploading strategy.* Several strategies can be devised in order to determine which data bundles should be sent first when a smartphone establishes a connection with a nearby base station. An option is for example to preserve the chronological ordering of data bundles, uploading the oldest bundles first. For the field trial reported in the next section we decided to favor the transmission of “fresh” ECG data first, and to fill the gaps by uploading older bundles whenever possible. The transmission algorithm running in the smartphone's application was therefore implemented in such a way that “real-time” bundles (i.e. those produced during a radio contact between sensor and base station) get uploaded to the monitoring center first, and the time remaining during a contact window is used to upload “older” bundles, that is, bundles that are stored on the smartphone's micro-SD

card and that have not been uploaded to the monitoring center yet. A graphical application running in the monitoring center can thus display the latest cardiac activity of a runner, while allowing an operator to rewind the ECG stream in order to display past events if necessary.

*Power consumption.* According to measurements we performed in our laboratory, a SHIMMER sensor with an ECG expansion module can run for almost 10 hours on its built-in battery, while sending ECG data continuously on a Bluetooth RFCOMM link. An HTC Wildfire S smartphone maintaining a Bluetooth connection with a SHIMMER sensor and establishing episodic connections with nearby Wi-Fi access points depletes its battery in 5 to 7 hours, depending on the frequency of radio contacts with these access points. These figures confirm that with a combination of Bluetooth and Wi-Fi transmissions (with disruption tolerance on the Wi-Fi segment) a runner can be monitored during the whole duration of a marathon race. In contrast, it is worth mentioning that a smartphone using 3G transmissions instead of Wi-Fi transmissions to upload ECG data to a monitoring center depletes its battery in about 3 hours. If using 3G instead of Wi-Fi there is therefore a risk that the smartphone shuts down before the runner has reached the finish-line.

## 5 Experimental results

Before trying to capture the ECG stream of runners during a real marathon, we decided to run field trials at a much smaller scale in order to validate our approach. An experiment was thus conducted on our university campus. Three volunteers were equipped with ECG-enabled SHIMMER sensors and HTC Wildfire S smartphones, and two base stations (BS1 and BS2) were placed about 1 km apart along the running route. These base stations were standard Wi-Fi access points. They were both placed on a window-ledge, and connected to the campus LAN. The runners had to run around the campus, passing twice close to each base station.

During the race our prime motivation was to observe if the data bundles produced continuously on each runner could actually be transmitted when the runner's smartphone established radio contact with one or another base station. Figure 2 shows the timeline of transmissions between the smartphones carried by the three runners (S1 to S3) and the two base stations. The rectangles in dotted lines depict a radio contact between a smartphone and a base station. The grey background shows the actual running time of a runner during the trial.

Let us examine the transmission timeline for S1. This smartphone was installed together with a SHIMMER sensor on a runner around 10:58. Both devices were activated immediately, so S1 started collecting bundles of data from that time on. Once the three runners were ready to go, they walked together to the start line. Since BS1 was located near that line a connection S1 established a connection with B1, and started uploading to the remote server all the bundles it had recorded since its activation. At 11:04, the three runners started running.

The connection between S1 and BS1 was therefore interrupted, after a 105 second contact window during which 21 data bundles had been uploaded to the server. Around 11:10 S1 established a connection with the second base station. This new contact window lasted 40 seconds, and this time 19 bundles were uploaded by S1 (17 of these bundles had been produced since S1 lost contact with BS1, and 2 new bundles were produced while S1 was in contact with BS2). As the runner carrying S1 continued running around the campus, S1 later established a connection again with BS1 (around 11:19), and then with BS2 (around 11:28), which was installed close to the finish-line.

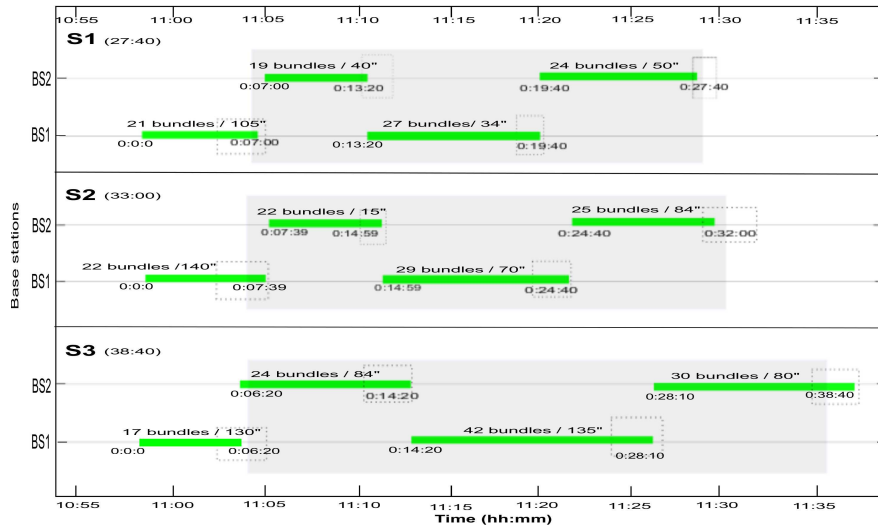


Fig. 2. Timeline of data transmissions during the race for the three runners

During this trial no data bundle was lost, or failed to reach the remote server. These results therefore confirm that the protocol we implemented on SHIMMER sensors and Android smartphones can indeed tolerate episodic connectivity to roadside base stations, with no loss of data. Further experiments involving a larger number of runners covering a longer distance should of course be conducted, but considering the high bandwidth available with Wi-Fi transmissions it can be expected that dozens of runners can be monitored simultaneously using this approach.

## 6 Conclusion

In this paper we investigated the possibility to monitor the cardiac activity of runners during a marathon race. Unlike most current monitoring applications,

which imply either indoor real-time data streaming or ambulatory data recording, the solution we propose makes it possible to get updates about each runner's health regularly during the race, using episodic transmissions between sensors carried by runners and base stations deployed sparsely along the marathon route. Disruption-tolerant transmission techniques are used to cope with the partial coverage of the route.

A preliminary field trial has been conducted with volunteers running around our university campus, using SHIMMER sensors for data acquisition and Android smartphones for Wi-Fi transmission to roadside base stations. This trial confirms that capturing and transmitting ECG data during a running race is indeed feasible with such devices and technologies. In future work we plan to increase both the number of monitored runners and the duration of the field trials in order to check the scalability of this approach. Ultimately we would of course like to demonstrate that dozens (or possibly hundreds) of runners can be monitored during a real marathon race.

## References

1. Benferhat, D., Guidec, F., Quinton, P.: Disruption-Tolerant Wireless Biomedical Monitoring for Marathon Runners: a Feasibility Study. In: 1st International Workshop on Opportunistic and Delay/Disruption-Tolerant Networking (WODTN'11), Brest, France, IEEE Xplore (October 2011) 1–5
2. Konstantas, D., Herzog, R.: Continuous Monitoring of Vital Constants for Mobile Users: the MobiHealth Approach. In: 25th Annual International Conference of the IEEE EMBS. (2003) 3728–3731
3. Nayebi, A., Sarbazi-Azad, H., Karlsson, G.: Routing, Data Gathering, and Neighbor Discovery in Delay-Tolerant Wireless Sensor Networks. In: 23rd IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2009, Rome, Italy, May 23–29, 2009, IEEE CS (2009) 1–6
4. Kugler, P., Schuldhaus, D., Jensen, U., Eskofier, B.: Mobile Recording System for Sport Applications. In Jiang, Y., Zhang, H., eds.: Proceedings of the 8th International Symposium on Computer Science in Sport (IACSS2011), Liverpool (2011) 67–70
5. Chelius, G., Braillon, C., Pasquier, M., Horvais, N., Pissard-Gibollet, R., Espiau, B., Azevedo Coste, C.: A Wearable Sensor Network for Gait Analysis: A 6-Day Experiment of Running Through the Desert. *IEEE/ASME Transactions on Mechatronics* **16**(5) (2011) 878–883
6. Pasztor, B., Musolesi, M., Mascolo, C.: Opportunistic Mobile Sensor Data Collection with SCAR. In: In Proc. IEEE Int'l Conf. on Mobile Adhoc and Sensor Systems (MASS07), IEEE Press (2007) 1–22
7. Fall, K.: Messaging in Difficult Environments. Technical report, Intel Research Berkeley (2004)
8. Burns, A., Greene, B., McGrath, M., O'Shea, T., Kuris, B., Ayer, S., Stroiescu, F., Cionca, V.: SHIMMER : A Wireless Sensor Platform for Noninvasive Biomedical Research. *IEEE Sensors Journal* (9) (2010) 1527–1534