Biomedical Monitoring of Non-Hospitalized Subjects using Disruption-Tolerant Wireless Sensors

Frédéric Guidec[1], Djamel Benferhat[1], and Patrice Quinton[2]

[1] IRISA, Université de Bretagne-Sud, France
Frederic.Guidec@univ-ubs.fr
[2] IRISA, ENS Cachan Bretagne, France

Abstract. The proliferation of private, corporate and community Wi-Fi hotspots in city centers and residential areas opens up new opportunities for the collection of biomedical data produced by sensors carried by mobile non-hospitalized subjects.
In this paper we investigate the possibility of using these many hotspots as gateways for biomedical data transmission. A disruption-tolerant application is presented, that can record biomedical data while the subject is not in the range of a Wi-Fi hotspot, and upload recorded data to a remote monitoring center whenever a hotspot is located nearby. Results of a field trial are presented, with a scenario involving a subject wearing an ECG-enabled sensor, walking in the streets of a residential area.

Keywords: biomedical monitoring,disruption-tolerant networking,Wi-Fi

1 Introduction

Wireless sensors open up interesting opportunities for biomedical monitoring, such as the long-term, continuous monitoring of subjects in a clinical environment or at home [1,2]. In a typical deployment scenario, one or several wireless sensors are attached to a subject, and a wireless base station is installed in this subject's surroundings. This base station can either record the data received from the sensors, or it can forward these data directly to a remote site, such as a physician's desktop computer or a hospital's monitoring center. In any case, since the sensors are wireless the subject can move freely around the base station, while an endless stream of data flows from the sensors he is carrying to the base station. This freedom of movement is however hampered by the short transmission range of current off-the-shelf wireless sensors. Indeed, most of these sensors include low-power radio transceivers, with which actual transmission ranges usually do not exceed a few meters. For this reason most research projects targeting health monitoring on mobile subjects rely either on dedicated base stations that must be deployed specifically for that purpose [3], or assume ubiquitous connectivity to 2.5/3G infrastructure [4].

The concept of *Disruption-Tolerant Networking*[3] *(DTN)* is a means to cope with challenging situations where continuous transmissions cannot be guaranteed. When considering a scenario involving mobile wireless devices, the general idea is to apply the *store, carry, and forward* principle: a device that is temporarily disconnected from the network can *store* data for a while in a local cache, carry these data while moving towards a location where network connectivity can be restored, and ultimately *forward* the data when circumstances permit.

Applying the store, carry and forward principle in wireless biomedical sensors is an appealing prospect. Indeed, a clinician monitoring a subject remotely does not necessarily need to receive data concerning this subject in real time. In most cases a time lag of a few minutes is perfectly tolerable. Using the store, carry and forward principle therefore makes sense in order to give greater mobility to the monitored subject. To the best of our knowledge this approach has not been investigated much so far, although disruption-tolerant solutions for *non-biomedical* sensor-based applications have already been proposed in the literature [5,6,7].

The solution we consider specifically in this paper consists in using any accessible Wi-Fi hotspot as a gateway for data uploading. Nowadays, Wi-Fi hotspots can be counted in millions. Besides corporate hotspots, most DSL providers distribute residential gateways that include a builtin Wi-Fi access point, which can operate a private and public Wi-Fi hotspot simultaneously. When a DSL subscriber accepts to enable the public hotspot service on his own residential gateway, he can in return access any other public hotspot deployed by the same DSL provider in the country. The millions of public hotspots managed by a single provider therefore constitute a wide community network that covers a significant part of urban areas. A subject should thus be able to attend to his daily business, while the sensing system he's wearing relies on nearby hotspots to upload data to a monitoring center.

The remainder of this paper is organized as follows. The SHIMMER platform we use in this project for data acquisition on mobile subjects is described in Section 2. Section 3 presents the main features of the transmission chain we designed in order to support the disruption-tolerant transmission of data between subjects and a monitoring center. In Section 4 we present the results of one of the field trials we conducted in order to validate this approach. Section 5 concludes this paper.

2 Overview of SHIMMER sensors

In this project we use SHIMMER platforms in order to acquire biomedical data on non-hospitalized subjects (see Fig. 1). The SHIMMER platform is a programmable lighweight wireless sensing system that can record and transmit physiological and kinematic data in real-time [8]. Data acquisition is performed on up to 8 channels through a 12-bit AD converter. Several kinds of expansion modules are available, including physiological sensors such as ECG (electrocardiography), EMG (electromyography) and GSR (galvanic skin response) sensors,

[3] The term Delay-Tolerant Networking is also used in the literature.

as well as kinematic sensors for 3-axis angular rate sensing and 3-axis low field magnetic sensing.

Fig. 1. A volunteer is equipped with an ECG-enabled SHIMMER sensor and a smartphone (which can be worn in an armband or in a pocket)

Two low-power radio transceivers operating in the 2.4 GHz ISM band are included in the platform: an IEEE 802.15.4/ZigBee compliant CC2420 transceiver, and a WML-C46A class 2 Bluetooth transceiver. Since none of these standards can be used to connect directly to a Wi-Fi (IEEE 802.11) access point, an additional wearable device is required to serve as a relay between a SHIMMER sensor and a Wi-Fi access point. In the solution we propose this relaying device is an Android smartphone, which can receive data continuously from the sensor through a Bluetooth RFCOMM link, and forward these data whenever possible to nearby Wi-Fi access points, while tolerating the transient connectivity to such access points.

3 Protocol for data acquisition and transmission

We developed specific code in nesC (a dialect of C) for the SHIMMER sensors, and a Java application for Android smartphones. The main features of this code are detailed below.

Data acquisition on a SHIMMER sensor. This acquisition is performed on two 12-bit channels, with a sampling frequency that can be adjusted as needed. The nature of the data depends on the kind of expansion module that is associated with the main unit. In any case the data stream produced by the sensor is transmitted on-the-fly to the smartphone through a Bluetooth RFCOMM link.

Transmission between sensor and smartphone. Each sensor must be paired with a specific smartphone, and two paired devices must of course be carried by the same subject. 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 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 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. The bundle's header includes an identifier of the source sensor and a timestamp. Its payload is simply a byte array that contains a sequence of data bytes received from the sensor. The size of this payload 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 12 kiB payload.

Hotspot discovery, selection, and authentication. When a subject enters a hotspot, the smartphone he is carrying must detect the Wi-Fi access point that serves this hotspot and attempt to associate with it. If the association succeeds, the smartphone must send a DHCP request in order to obtain IP parameters from a DHCP server. Finally, some form of authentication may be required before access to the Internet is granted.

In order to ensure hotspot discovery, selection, and authentication in our project we rely on Wi2Me, an Android-based application that has been designed specifically to support fast handover between hotspots in corporate and community networks [9].

Wi2Me notably uses active probe requests (rather than the standard's default passive scan) in order to locate nearby hotspots, and Kalman filters to characterize signal attenuation tendencies and select interesting hotspots accordingly. When connecting to a selected hotspot Wi2Me can authenticate with this hotspot using the standard WPA (Wi-Fi Protected Access) procedure, or through an HTTPS captive portal.

Transmission between smartphone and remote server. Once the Wi2Me application has managed to establish a connection and to authenticate with a nearby hotspot, the disruption-tolerant Android application we designed can start uploading data bundles to a server that is the entry point of the monitoring center. This application basically behaves like a client thread with respect to the server. Whenever a "Connected" notification is issued by the Wi2Me program, this thread attempts to open a TCP session with the server. If this attempt fails, the thread waits for a few seconds before initiating another attempt. Once a TCP session is established, bundles stored in the local cache are sent to the server sequentially, and each bundle received by the server is acknowledged explicitly at application level. This approach allows the client thread to detect and react to transmission failures, which typically occur when the subject carrying the smartphone moves out of the radio range of the current hotspot. The Wi-Fi connection is then broken unexpectedly, and the TCP session must be closed unilaterally on the client side while TCP segments (containing fragments of data

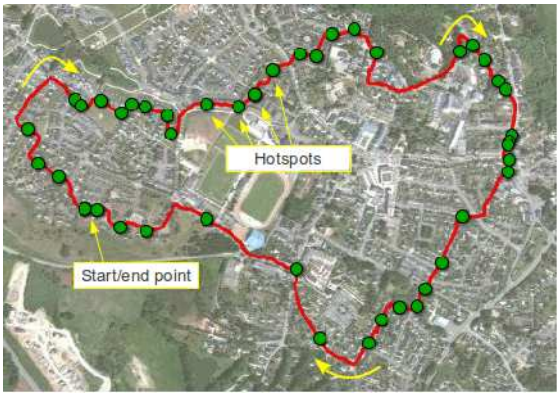


Fig. 2. Route followed by the volunteer subject during the experiment. The green circles represent community hotspots used by his smartphone to upload ECG data while he walked along streets and footpaths

bundles) are still pending in the client socket's send buffer. The client thread then waits for the next "Connected" notification from the Wi2Me program, and as soon as this notification is received it tries to open a new TCP session with the server. If a bundle sent during the previous TCP session has not been acknowledged yet, then this bundle is sent again. Afterwards, the client thread resumes its normal routine activity, which consists in sending available data bundles one after another, and waiting for an acknowledgement after each bundle.

Several strategies can be devised in order to determine which data bundles should be sent first when a smartphone establishes a connection with a new hotspot. An option is for example to preserve the chronological ordering of data bundles, uploading the oldest bundles first. In the current implementation we decided to favor the transmission of "fresh" data first, and to fill the gaps by uploading older bundles whenever possible. The application's client thread was therefore implemented in such a way that "real-time" bundles (i.e. those produced during a radio contact between the sensor and the 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 data concerning a subject, while allowing an operator to rewind the data stream in order to display past data if necessary.

4 Experimental results

In order to validate our approach, we conducted several field trials involving volunteers carrying SHIMMER sensors and HTC Wildfire S smartphones.

Several scenarios were considered, with subjects staying at home, going to work, shopping, etc. In this section we present the results observed with a sub-

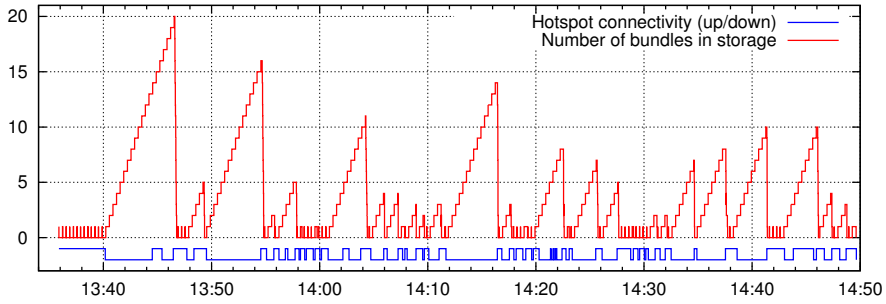


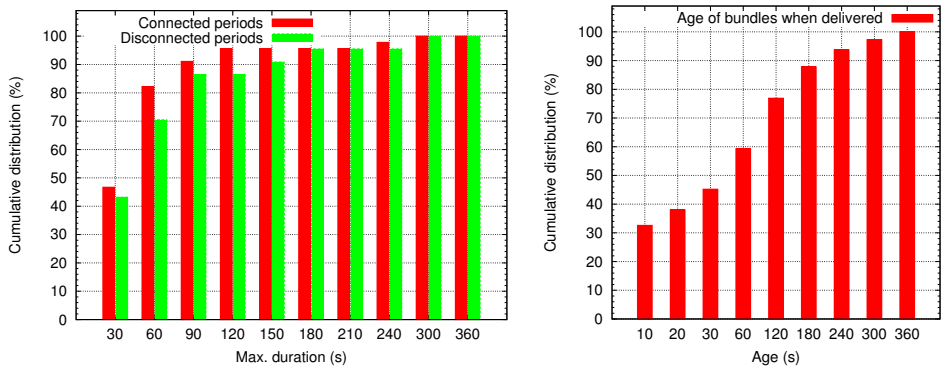
Fig. 3. Timeline of data storage and transmissions during the experiment

ject wearing an ECG-enabled SHIMMER sensor, walking along the streets and footpaths of a residential area. During this trial the smartphone was configured so as to split the ECG data stream received from the sensor in 16 kiB bundles, each bundle containing 20 seconds of recorded data. Additionally, the Wi2Me application running on the smartphone was configured so as to seek and connect to any community hotspot administered by either of the French network operators Free and SFR.

The subject was first equipped at home with an ECG-enabled sensor and a smartphone (Fig. 1). After a few minutes he went for a 4.6 km walk, and came back home about an hour later. Figure 2 shows the route followed by the subject during his walk, as well as the community hotspots his smarphone managed to connect to along that route. Figure 3 shows the timeline of data bundles storage and transmissions during this experiment, as well as the connectivity status of the smartphone.

At 13:35 both devices were switched on. The sensor immediately started monitoring the cardiac activity of the subject, while the smartphone connected to the subject's own Wi-Fi access point (the most accessible hotspot at that time). The bundles of ECG data produced were therefore transmitted in real time to the remote server. At 13:40 the subject went out. The smartphone then disconnected from the access point, and started storing data bundles. At 13:44 the smartphone connected to a second hotspot, but did not manage to upload bundles through that hotspot. At 13:46 it connected to a third hotspot, and this time it managed to upload through that hotspot the 20 bundles stored in its cache. Since the connection with that hotspot was maintained for almost one minute, the smartphone additionally managed to upload a couple of "fresh" bundles before the connection was broken. As the subject continued walking in the streets further connections were established with community hotspots, until the subject came back home around 14:47.

This trial lasted 74 minutes, and 45 hotspots were used for data uploading in the meantime. Figure 4.a shows the cumulative distribution of connected and



(a) (b)

Fig. 4. Cumulative distribution (a) of hotspot connectivity periods, and (b) of the age of bundles at delivery time

disconnected periods. It can be observed that about 80% of the connections to hotspots lasted less than one minute, which is consistent with the fact that the subject was mobile –and walking at a steady pace– most of the time during the period considered. Similarly, it can be observed that more than 90% of the disconnected periods lasted less than 2.5 minutes. This observation confirms that the density of community hotspots in a residential area is such that a disruption-tolerant application like ours can find many opportunities to upload data to a remote server. The delay before data bundles can reach the server of course depends on the frequency of connections with hotspots. As shown in Figure 4.b, during this trial most bundles reached the server in less than one minute, and no bundle was more than 6 minutes old when it reached the server.

During this trial the battery level on the smartphone dropped by 40%. Since the GPS receiver was enabled in order to record the route followed by the subject, this figure is not a good indication of the autonomy of the system. 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 data continuously on a Bluetooth RFCOMM link. A HTC Wildfire S smartphone maintaining a Bluetooth connection with a SHIMMER sensor and establishing episodic connections with nearby Wi-Fi hotspots depletes its battery in 5 to 7 hours, depending on the frequency of radio contacts with these hotspots. In contrast, it is worth mentioning that if the smartphone uses 3G transmissions instead of Wi-Fi transmissions its battery is depleted in less than 3 hours.

These figures confirm that with a combination of Bluetooth and Wi-Fi transmissions (with disruption tolerance on the Wi-Fi segment) a subject can be monitored during his daily activity, provided the smartphone's battery can be recharged at least once or twice during the meantime.

5 Conclusion

In this paper we investigated the possibility to collect biomedical data on non-hospitalized mobile subjects. The approach we propose involves off-the-shelf wireless sensors for data acquisition, and smartphones that can record data continuously and upload these data whenever possible to a remote monitoring center. Transient connectivity with personal, corporate, or community Wi-Fi hotspots is used by a disruption-tolerant application running on the smartphone to perform data uploading transparently and opportunistically.

Results of a field trial involving a subject walking in a residential area confirm that the density of community hotspots in such an environment is sufficient to ensure regular updates of the data collected by the monitoring center. Other trials conducted in different conditions (subject at work, shopping, practising sports, etc) have led to similar conclusions.

There are of course circumstances when the smartphone carried by a subject may be unable to connect to a Wi-Fi hotspot for long periods of time. In such circumstances it may be useful to resort to 3G transmissions, if only to upload a minimal set of data while waiting for the next hotspot connection. With this approach the major bottlenecks are the lower transmission rate (for upload) and the power-greedy nature of 3G transmissions. In future papers we shall propose a combination of these solutions, balancing transmission delays with cost and longevity.

References

1. Alemdar, H., Ersoy, C.: Wireless Sensor Networks for Healthcare: a Survey. Computer Networks **54**(15) (2010) 2688–2710
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. Babovic, Z., Crnjin, A., Racocevic, G., Stankovic, M., Peric, Z., Cirkovic, I., Damjanovic, I., Milutinovic, V.: Prosense Reaseach Activities in Belgrad (2009)
4. Konstantas, D., Jones, V., Herzog, R.: MobiHealth Innovative 2.5-3G Mobile Services and Applications for Healthcare. In: Proceedings of the Eleventh Information Society Technologies (IST) Mobile and Wireless Telecommunications. (2002) 43–52
5. Pisztor, 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
6. Jain, S., Shah, R., Brunette, W., Borriello, G., Roy, S.: Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks. MONET **11**(3) (2006) 327–339
7. 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
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

9. Castignani, G., Lampropulos, A.M., Blanc, A., Montavont, N.: Wi2Me: A Mobile Sensing Platform for Wireless Heterogeneous Networks. In: IEEE International Workshop on Sensing, Networking, and Computing with Smartphones (ICDCS 2012), Macau, China (June 2012)