Power- and Delay-Aware Mobile Application-Data Flow Adaptation: the MobiHealth System Case Study

Katarzyna Wac [1, 2], Mortaza S. Bargh [3], Pravin Pawar [1], Bert-Jan van Beijnum [1], Arjan Peddemors [3], Richard Bults [1]

[1] University of Twente, EWI/BSS group, P.O. Box 217, 7500 AE Enschede, the Netherlands
[2] University of Geneva, CUI/ASG, 24 rue General-Dufour, 1211 Geneva, Switzerland
[3] Telematica Instituut, P.O. Box 589, 7500 AN Enschede, the Netherlands
{wac, pawar, beijnum, bults}@cs.unige.ch, {bargh, peddemors}@telin.nl

Abstract — Emerging healthcare applications rely on personal mobile devices to monitor patient vital signs and to send it to the hospitals-backend servers for further analysis. However, these personal mobile devices have limited resources that must be used optimally in order to meet the requirements of healthcare applications end-users: healthcare professionals and their patients. This paper reports on a case study of a cardiac telemonitoring application delivered by the so-called MobiHealth system. This system relies on a commercial personal mobile device with multiple (wireless) network interfaces (NI). The study focuses on how the choice of a NI affects the end-to-end application's data delay (extremely important in case of patient's emergency) and the energy consumption of the device (relating to the service sustainability while a patient is mobile). Our results show the trade-off between battery savings and the delay achieved by various NI activation strategies in combination with application-data flow adaptation. For a given mobile device, our study shows a gain of 40-90% in battery savings, traded against the higher delays (therefore applicable mainly in non-emergency cases). The insights of our studies can be used for application-data flow adaptation aiming at battery saving and prolonging device's operation while patients being mobile.

Key words—mobile device connectivity management, energy efficiency, end-to-end delay, application adaptation, mobile healthcare

1 Introduction

The emergence of new wireless broadband networks and the increased diversity of miniaturized and personalized networked devices give rise to a variety of new mobile interactive applications in our daily life. Examples of these are, on one hand, applications supporting traditional users as information-*consumers*, e.g. news, leisure and entertainment content delivery. On the other hand, mobile users are no longer only passive information and content consumers, but on a growing scale they take the role of information and content *producers*. Examples of these applications are especially ones supporting social networking and interactions. However, another emerging application domain, in which a user acts as a content producer, is a mobile healthcare domain, where a mobile patient's vital signs can be telemonitored by his healthcare professional in the healthcare center. In this paper we focus on this particular application example.

The above mentioned applications are ultimately envisaged to be delivered to the user on the move: anywhere anytime and under different conditions, while fulfilling his *Quality of Service* (QoS) *requirements*. These requirements are, for example, low application delays, long device battery life and seamless user mobility support along with low monetary cost of networks usage. However, as applications operate in a heterogeneous networking environment, consisting of a variety of wireless and wired networks owned by different parties, the QoS provided by this environment is one of the most critical factors influencing the assurance of the QoS provided by the application to the user. In this paper, the *QoS provided by an application* is defined as consisting of an *application-level throughput* (in kbps) and an *application-level delay* (in milliseconds).

There exists close relation between the provided application-level QoS and the provided network-level QoS. Particularly, the provided application-level throughput and delay depend respectively on throughput and data delay while using particular underlying (wireless) network over the given network interface (NI) on the mobile device. Moreover, the device battery life depends on a given application, given NI, and on how *application-data flow* is offered to this NI. Particularly, this flow is described in terms of its volume, i.e., size and rate of the data offered to the NI. By changing the size and the rate parameters we change volume of data to be send; in such a way we can adapt the application-data flow to suit better the provided network-level QoS and to obtain better application-level QoS.

This paper focuses on 1) an choice of NI and its activation strategy (as available on a mobile device) and 2) an application-data flow adaptation, and relations of these two with a) a device's energy consumption and b) an application-data delay. The NI activation strategy (parameter 1) assumes

This work is part of the Freeband AWARENESS project (http://awareness.freeband.nl). Freeband is sponsored by the Dutch government under contract BSIK 03025.

that a NI can be in an OFF state, or an ON-IDLE state, in which it is connected to the network, but it does not send/receive application-data, or in an ON-ACTIVE state while sending/receiving application-data. In this paper therefore we study the relation of these four parameters to the user's required QoS for a health telemonitoring application.

Applications in the mobile healthcare domain, like telemonitoring or teletreatment [1], pose strict QoS requirements, since a patient can be in an emergency, requiring an immediate system response, e.g., dispatch of an ambulance. In this paper we consider a mobile health telemonitoring application. Specifically, we study cardiac telemonitoring application delivered by the so-called MobiHealth system [2].

The rest of this paper is organized as follows. A description of the MobiHealth system is given in Section 2. In Section 3 we present a mobile device's NI states, and possible NI state selection criteria. In Section 4 we explain our approach for energy consumption and application-level delay measurements for a commercial mobile device used in the MobiHealth system. Section 5 summarizes and analyzes the measurement results, based on which we define NI activation strategies. Related work is discussed in Section 6. Based on measurements results, in Section 7 we provide the conclusions and recommendations for the MobiHealth system usage and some future work areas.

2 The MobiHealth system

2.1 System Overview

The MobiHealth system is a distributed system for telemonitoring of a patient's health condition.

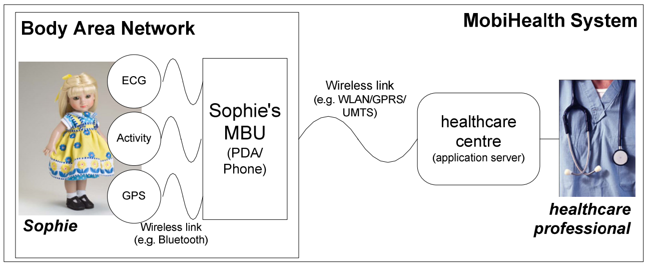
Body Area Network
MobiHealth System
ECG
Activity
GPS
Sophie's MBU (PDA/ Phone)
Wireless link (e.g. WLAN/GPRS/ UMTS)
healthcare centre (application server)
Sophie
Wireless link (e.g. Bluetooth)
healthcare professional

Figure 1. The MobiHealth system overview

In the MobiHealth system (Figure 1), a patient is wearing a *Body Area Network* (BAN), consisting of a sensor-set and a *Mobile Base Unit* (MBU). The sensor-set usually consists of specialized sensors monitoring the patient's vital signs, an alarm button that can be pressed by the patient in emergency, and a location sensor (e.g. a GPS receiver) for location determination of this patient. The sensor-set is specific for a patient's health condition. Example health conditions are: the respiration insufficiency, cardiac problems, epilepsy, chronic pain or muscle spasticity.

The MBU is the central unit of a BAN, usually in the form of a mobile phone or PDA. The MBU has three responsibilities: collecting sensor data, processing it (e.g. filtering, shaping, correlating) and sending (processed) data to a remote application *backend-server* located in a healthcare center. It is specific for the MobiHealth system is that all these tasks are performed in real time. Once the sensor data has been send to the backend server, it is made available to other applications, for instance for storing the data, displaying the data or for medical decision support systems. These consumers get near real-time access to the vital sign data.

Note that emergency can be defined differently for each patient, based on the patient's health condition. Moreover emergency can be determined by a) patient's pressing the alarm button or, based on the patient's vital signs analysis, b) the patient's BAN or b) the backend-server.

The BAN uses the intra-BAN communication network, like Bluetooth (BT) for sending data from the sensor-set to the MBU, and an extra-BAN communication network, like WLAN or 2.5G/3G (i.e. GPRS/UMTS) for exchange of the application and control data between the MBU and the backend-server.

The application execution is supported by a proprietary *MSP-Interconnect protocol (MSP-IP)* [3] that is a TCP/IP protocol-stack-based protocol, facilitating application-data-plane and control-plane[1] data. This application protocol, and the overall system architecture conforms the Jini Interconnect specifications as extensively presented in [4]. Interested readers we also refer to more detailed description of the MobiHealth system and its architecture given in [2, 5].

2.2 Telemonitoring Application-Data Flow

For the purpose of this paper, we consider the telemonitoring application for cardiac patients in a non-critical condition, i.e., with a small probability of an emergency. This assumption affects further possible application-data flow adaptation cases, which will be different for an emergency and non-emergency (Section 2.3). This telemonitoring application runs continuously at the MBU.

The sensor-set acquires patient's heartrate (HR), oxygen saturation (SO_2) and plethysmogram (pleth), an alarm button activity, and a control-data. The sensor-set has a sampling frequency of 128 Hz; each sample consists of a 5 Bytes of application-data. A unit of data that the application collects consists of one second aggregated sensor-set data, so, in total of 640 Bytes. Every unit of data is deflated (using lossless compression algorithm) before being send by the extra-BAN communication network. The data compression factor, i.e., the reduction in size relative to the uncompressed size, is 80-85 %. However, this factor strongly depends on the actual values of the measured vital signs; it decreases as variability in measured vital signs increases. The MSP-IP introduces a 10 Bytes overhead per an aggregated and compressed data. Hence, the protocol stack overhead is 64 Bytes for WLAN (MSP-IP/TCP/IP/Ethernet) and 58 Bytes for GPRS (MSP-IP/TCP/IP/PPP). The resulting data unit is sent over the data-

[1] BAN control-plane data consists of the MBU management lifecycle and aliveness (Keep-Alive) messages

plane. The overall volume of data sent by the NI comprises of the data-plane and control-plane[2] data; in total around 1.2-1.5 kbps[3].

2.3 QoS Requirements

The end-users of the telemonitoring applications and healthcare professionals are their patients, however mainly the former ones are in charge to define the QoS requirements [6]. The QoS requirements are related to the performance of application-data exchange a) its reliability in terms of a lossless and error-free exchange, and b) a minimum application-data delay (in case of a patient's emergency) from the sensor-set to the backend-server. The use of TCP/IP protocol stack and the use of local storage of the data in case when there is no network available to send the data or a real-time sending is not required, ensure system recovery in case of data loss and encountered data-errors. Further study of application reliability is not in scope of this paper.

Concerning the application-data delay requirement, we focus on the extra-BAN data delay, as a major contributor to the application-data delay in the MobiHealth system. Particularly the MobiHealth system performance is managed based on an *application-level Round Trip Response* (AppRTT)[4] times. The AppRTT is a time period it takes for a control message (i.e., a MBU Keep-Alive message[5] [4]) originated from the MBU, to be bounced by the backend-server (without being processed there) and received back by the MBU. This AppRTT strongly depends on the choice of the extra-BAN communication network sending data, i.e. the NI choice at the MBU, and the volume of the application-data being sent.

Moreover, The AppRTT reflects the delays provided by the underlying networks, as it is composed of the processing delays in the protocol stacks at the MBU and the backend-server side as well as an uplink (MBU to the backend-server) and downlink (backend-server to the MBU) network delays.

As we already indicated in the introduction, the considered cardiac telemonitoring application's delay requirements strongly depend on the actual health condition of a patient. In emergency, patient vital signs data needs to be continuously send (with a minimum possible delay) to the backend-server in the healthcare center, where it is made available for a healthcare professional in real-time. For non-emergency, it is possible that the MBU acquires all the application-data (for data-plane and control-plane), stores it locally, and sends it to the backend-server later (i.e., in bursts), e.g. when a cheap, high-throughput WLAN network is available. It is also possible that in non-emergency, the (real-time) BAN data is sent continuously to the backend-server together with the previously stored BAN data.

Another QoS requirement for MobiHealth is a maximum lifetime of the BAN. In this paper we focus on the MBU's power consumption for extra-BAN communication, as its contribution to the BAN's power consumption. We denote the MBU power consumption as $power_{MBU}$. It depends on the NI used for extra-BAN communication and the volume of the application-data being sent.

In our study we also consider an additional user's requirement resulting from the fact that a patient needs to use his MBU as a regular (WWAN) phone and needs to be WWAN-*reachable*, especially by his healthcare professional, for voice/data communication. Assurance of this requirement may not be favorable from the power perspective[6]; however in our study we consider this requirement.

We note additionally, that the MBU power consumption depends also on a user's mobility level and the MBU configuration parameters like backlight level, other running applications, or MBU location with respect to the network's access point/base station (i.e., MBU's received signal strength). However, in our study, we assume that a patient wearing BAN is in a fixed location (i.e. not being mobile).

3 Network Interfaces Activation

3.1 Network Interface States

The existing wireless technologies accessible by commercial mobile devices can be divided into two categories: WWANs that provide a low-throughput and high-delay service over a wide geographic area (e.g. GPRS or UMTS) and WLANs that provide a high-throughput and low delay service over a narrow geographic area (e.g. WiFi) [7]. We consider a device NI state model for mobile devices with GPRS or/and UMTS as WWAN interface and WiFi as WLAN interface. A NI is in one of the following states:

- OFF,
- ON-IDLE: an IP-idle state, where the mobile device has IP connectivity to the Internet. However it does not send/receive application level data-plane or control-plane IP packets (i.e., IP packets carrying application-data).
- ON-ACTIVE: an IP-active state, where mobile device is sending or receiving application level IP packets through this NI.

From the telemonitoring application perspective the following cases are possible:

1) Application-data being distributed for sending via the WLAN and WWAN NIs
2) Application-data being send via the WLAN NI, while the WWAN NI is OFF or ON-IDLE
3) Application-data being send via the WWAN NI, while the WLAN NI is OFF or ON-IDLE
4) Application data being stored locally, while the WWAN (or WLAN) NI is OFF or ON-IDLE

[2] of a negligible size comparing to the data-plane

[3] Data-plane calculation for compression factor of 80%: a) WLAN: [(640 *0.2)+64]*8 bps = 1536 bps b) GPRS: [(640 *0.2)+58]*8 bps = 1488 bps;
Data-plane calculation for compression factor of 85%: a) WLAN: [(640 *0.15)+64]*8 bps = 1280 bps b) GPRS: [(640 *0.15)+58]*8 bps = 1232 bps

[4] Reliable one-way delay measurements would only be possible if the clocks of MBU and backend-server were synchronized, which is hardly possible in the operational system

[5] KeepAlive message of size of 41 Bytes

[6] Additional power is consumed if WWAN is in ON-IDLE state without sending data

Note that the NI that is used for sending data could be in ON-ACTIVE state continuously or could alternate between the ON-ACTIVE and the ON-IDLE/OFF states (the latter implies data send in bursts).

3.2 Criteria for a Choice of a NI State

Based on the scope of our study and on the requirements posed by the MobiHealth users (Section 2.3), we conclude that a NI state (i.e., to be OFF, ON-IDLE or ON-ACTIVE) depends on criteria:

(i) application-data delay requirement posed by the current health condition of a patient (i.e., emergency or non-emergency)

(ii) the power$_{MBU}$ consumption while using that NI

(iii) the provided AppRTT while using that NI.

4 Measurements

4.1 MobiHealth System Setup

The MobiHealth sensor-set is based on Mobi5-3e1as [8], with only the NONIN finger clip attached (for HR, SO_2 and pleth data). As a MBU we have used Qtek 9090 with Intel® PXA263 400 MHz processor (32b), 128 MB RAM, firmware version 1.31.00 WWE (from 13.12.2004), radio version 1.06.02, protocol version 1337.38 running Windows Mobile 2003 SE PocketPC OS edition version 4.21.1088. The Qtek's battery is of a standard type, rechargeable Li-ion Polymer of capacity of 1490 mAh (3.7V, model PH26B). The Qtek has a TFT touch screen display of size of 53x71 mm (214 x 320 pixels, 65K colors) and its backlight level was set to zero.

The MBU has the WWAN-GPRS (GSM 850/900/1800/1900 Hz, class 10: 4+1/3+2 slots) and WLAN-WiFi (IEEE 802.11b, with 'best-battery' setting in the OS) as NIs for extra-BAN communication. The BT NI is used continuously for intra-BAN communication for sensor-set data acquisition. The MBU uses GPRS network provided by Sunrise mobile operator (received signal strength of 100%) and WLAN provided by the University of Geneva, Switzerland (received signal strength of 50%), where the MBU was placed such that the received signal strength has been maximized along the measurements).

The backend-server used is a standard high-performance server dedicated to MobiHealth telemonitoring services. The server was placed at Twente University, the Netherlands.

The MobiHealth telemonitoring application software version is a release from 17 October 2007.

Power and delay measurements instrumentation

The MobiHealth system was configured such that during the execution of the telemonitoring application, we collected the measurements logs at the MBU and backend-server. To measure the energy consumption of the MBU, we logged the remaining battery percentage in 5 seconds intervals. For the purpose of delay measurements, the MBU was instructed to log the AppRTT in intervals of 10 seconds continuously along the telemonitoring application delivery.

To obtain high application-data flow volumes, not feasible in the current state of the MobiHealth system, and especially important for our measurements over (high-throughput) WLAN NI (cases 3 and 4); we have used the NetPerf application [9]. This application is generating TCP traffic and measuring a unidirectional throughput between the MBU and the backend-server. These measurements were done for the same conditions as the other measurements; however, the MobiHealth application was NOT running in the background of the NetPerf application. By using this application, we attempted to simulate a case where MBU sends previously stored patient vital signs data. In the NetPerf measurements we obtained only the power$_{MBU}$ values.

Along the measurements, we assumed the MobiHealth system to be in the steady-state representing the behavior of the system usage for a typical system user, i.e. a cardiac patient, whose vital signs are being monitored. Measurements have been done over a time span of two weeks, always at same location (our University of Geneva office) but at different hours.

4.2 Measurements Cases

For the purpose of our research, we considered various measurements cases based on the following two parameters: application-data flow and NIs states. Each case represents the combination of states of the MBU WLAN and GPRS NIs (and BT ON-ACTIVE for intra-BAN communication) during our experiments. These cases represent typical execution of a health telemonitoring application in the MobiHealth system, and they are:

0. WLAN OFF, GPRS OFF
1. WLAN OFF, GPRS ON-ACTIVE
2. WLAN ON-IDLE , GPRS ON-ACTIVE
3. WLAN ON-ACTIVE, GPRS OFF
4. WLAN ON-ACTIVE, GPRS ON-IDLE
5. WLAN OFF, GPRS ON-IDLE
6. WLAN ON-IDLE, GPRS OFF
7. WLAN ON-IDLE, GPRS ON-IDLE

Note: Theoretically, it is also possible to have the WLAN in ON-ACTIVE and GPRS in ON-ACTIVE state; however, because this case is not implemented yet in the MobiHealth system (would require substantial application changes) we have not included it in our study.

Case 0 represents application 'base' energy consumption, i.e. for intra-BAN communication and the MBU processing and local storage of application-data (no extra-BAN communication). Additionally to this case, cases 5-7 represent application 'base' energy consumption increased of energy consumption for maintaining one (or both) NI in an ON-IDLE state.

Along the measurements execution we discovered that the case 2, i.e. where GPRS is ON-ACTIVE and WLAN is ON-IDLE was not possible to be executed on the given MBU. It is because the Qtek 9090 is preconfigured such that, if both GPRS and WLAN are available, it will always send data over the WLAN NI rather than leaving the choice of the usage of the NI to the user.

The last three cases (5-7) represent continuous application execution and local data storage cases, i.e., no data being send over a NI.

The telemonitoring application-data flow represents the volume of application-data send over the NI that is in the ON-ACTIVE state. Note that our healthcare application produces 1.2-1.5 kbps of data at the NI, i.e., at the physical layer (Section 2.2). The measured application-data volumes are therefore:

- 1.2-1.5 kbps corresponding to the continuous application execution and real-time sending of application-data (used in emergency and non-emergency)
- 5.2 or 7.7 kbps corresponding to the continuous application execution and delayed data send (i.e., sending data in burst, where 4-6 seconds of patient vital signs data are sent together). This can be used only in non-emergency.

Due to the limited processing capacity of Qtek (i.e., experienced system crashes) it was not possible to increase application-data volume beyond 7.7 kbps in case 1 and beyond 5.2 kbps in cases 3 and 4. Therefore we obtained volumes of 1.2-1.5 kbps, 5.2 kbps and 7.7 kbps for case 1, and volumes of 1.2-1.5 kbps and 5.2 kbps for cases 3 and 4.

5 Measurements Results

5.1 MBU Power Consumption (*power$_{MBU}$*)

We have executed measurements for cases as given in Section 4.2. We measured MBU’s remaining battery capacity in percents, and we transformed the results into the *normalized average power consumption values* indicating the decay rate of battery capacity over minutes. These normalized values facilitate comparison of the *relative energy cost* for WLAN and GPRS NIs in a particular device. Table 1 summarizes these normalized values for the Qtek device in different NIs states. We observed that in each experiment the remaining battery capacity decreased linearly with time; i.e., the normalized average power consumption values are constant.

The first 4 rows of Table 1 represent cases 0, 5-7, in which data was not sent, but locally stored at the device. Rows 1a, 3a and 4a correspond to cases of continuous application execution and sending of application-data in real-time. The other rows (1b, 1c, 3b, 3c, 4b, and 4c) correspond to cases of continuous application execution, but local data storage with delayed sending of data.

Table 1: NI’s normalized average power$_{\text{MBU}}$ values.

Case No.	Measurement case (Note: BT ON-ACTIVE for all cases)	Normalized power consumption [1/min]
0	WLAN OFF, GPRS OFF	0.00092
5	WLAN OFF, GPRS ON-IDLE	0.00487
6	WLAN ON-IDLE, GPRS OFF	0.00568
7	WLAN ON-IDLE, GPRS ON-IDLE	0.00963
1a	WLAN OFF, GPRS ON-ACTIVE (1.2-1.5 kbps)	0.00721
1b	WLAN OFF, GPRS ON-ACTIVE (5.2 kbps)	0.00874
1c	WLAN OFF, GPRS ON-ACTIVE (7.7 kbps)	0.00897
3a	WLAN ON-ACTIVE, GPRS OFF (1.2-1.5 kbps)	0.00873
3b	WLAN ON-ACTIVE, GPRS OFF (5.2 kbps)	0.00911
3c	WLAN ON-ACTIVE, GPRS OFF (NetPerf, 3.45 Mbps)	0.00982
4a	WLAN ON-ACTIVE, GPRS ON-IDLE (1.2-1.5 kbps)	0.00960
4b	WLAN ON-ACTIVE, GPRS ON-IDLE (5.2 kbps)	0.00974
4c	WLAN ON-ACTIVE, GPRS ON-IDLE (NetPerf, 3.95 Mbps)	0.00947

As we observe from Table 1, WLAN in ON-IDLE state consumes comparably the same energy as in ON-ACTIVE state (cases 4a, 7). This can be explained by the Qtek configuration, in which we did not instruct it to get into WLAN *power-save* mode when being ON-IDLE state. In this case, the WLAN NI continuously receives and processes all the data broadcasted between the Access Point and other WLAN devices.

Moreover, we conclude from that table, that from the power perspective, it is always better to use the GPRS NI and keep the WLAN OFF. If WLAN NI is used, it is always better to keep GPRS OFF.

5.2 Application-Data Delay (*AppRTT*)

We have executed measurements cases as given in Section 4.2 and observed, as we have previously expected, that AppRTT depends on the NI(s) used and the data volume being sent. Table 2 summarizes the results, with emphasis on the AppRTT mean value. Note that these results are reported only for the telemonitoring application execution, i.e., not for the cases when we have used the NetPerf.

Table 2: NI’s AppRTT values.

AppRTT [ms] & case No.	**Mean**	stdev	min	max	median
1a. WLAN OFF, GPRS ON-ACTIVE (1.2-1.5kbps)	3739	2005	1979	20856	3318
1b. WLAN OFF, GPRS ON-ACTIVE (5.2 kbps)	5505	2627	2767	20702	4706
1c. WLAN OFF, GPRS ON-ACTIVE (7.7 kbps)	6693	3954	2322	28220	5589
3a. WLAN ON-ACTIVE, GPRS OFF (1.2-1.5 kbps)	2753	1769	530	23807	2706
3b. WLAN ON-ACTIVE, GPRS OFF (5.2 kbps)	3513	2863	587	36819	3290
4a. WLAN ON-ACTIVE, GPRS ON-IDLE (1.2-1.5 kbps)	1806	1082	556	15756	1553
4b. WLAN ON-ACTIVE, GPRS ON-IDLE (5.2 kbps)	2211	1084	379	13609	2204

As we observe, from the delay perspective, the best is, whenever possible, to use a WLAN ON-ACTIVE and keep GPRS ON-IDLE (cases 4a and 4b). If WLAN is not available and it is necessary to use GPRS, it is better to use lower data volumes (case 1a), or, if patient is not in an emergency, gather the data for a local storage, and send it at the maximum possible volume later over WLAN (case 4b). An interesting observation is that for the real-time application-data sending, the GPRS has higher delay but slightly lower delay variation (i.e. stdev value) comparing to the WLAN (case 1a of 53% vs. 3a of 64% of a mean value). Moreover, the WLAN has lower delay and delay variation when GPRS being ON-IDLE (case 4a) than when GPRS being OFF (case 3a). That may be related to the internal NI management of the mobile device used (the real reasons are unknown for us, and to the best of our knowledge similar results have not been published so far).

5.3 NI activation strategies

In this section, we define the *basic* MBU NI activation strategies as ones, in which sending of patient vital signs data is done through an ON-ACTIVE NI in a real-time, i.e., without application-data buffering. These strategies are S_{EM}, S_1 and S_2 defined correspondingly to cases 4a, 3a and 1a and are to be used in emergency, but can also be used in non-emergency. These strategies are ordered by their AppRTT in Table 3, with the most delay-efficient strategy S_{EM} (and hence most recommended in emergency) and the least efficient S_2. The power consumed by the strategy S_{EM} is considered as our reference point for comparing the power efficiency of other strategies. The power efficiency of strategy S_X is then defined by ($power_{MBU}$ (S_{EM}) – $power_{MBU}$ (S_X)) / $power_{MBU}$ (S_{EM}); the bigger the resulting value, the more efficient the strategy is. The last row in Table 3 indicates if the strategy fulfills the requirement of a user being reachable on his/her mobile device via the WWAN-GPRS network.

Table 3: Performance of the basic NI activation strategies.

Strategy	S_{EM} (4a)	S_1 (3a)	S_2 (1a)
WLAN	ON-ACTIVE	ON-ACTIVE	OFF
GPRS	ON-IDLE	OFF	ON-ACTIVE
AppRTT [ms]	1806	2753	3739
power efficiency	0	9	25
WWAN reachability	yes	no	yes

For cases where the larger AppRTTs are acceptable, i.e. in non-emergency, the MBU may adapt application-data flows by acquiring *n-1* (n>1) seconds of the patient vital signs data, temporarily storing this data, and sending it in a burst in the n^{th} second (together with the n^{th} second data sample) to the backend-server via a chosen NI. The entries in Tables 2 and 3 for cases where data volumes achieve 5.2 kbps (1b, 3b, 4b) and 7.7 kbps (1c) make our basis to consider n=4 (thus achieving 5.2 kbps) and n=6 (thus achieving 7.7 kbps).

Table 4 summarizes the comparison results for three distinctive application-data flow adaptation cases extrapolated from measurements cases: 1b, 3b, 4b and 1c. The power efficiency of a strategy is again defined against the S_{EM}. The following relations hold in that table:

AppRTT = (n-1)*1000 + measured AppRTT [ms]

Normalized power =

$$1/n\ [(n-1)\ power_{MBU}\ (NI_1\text{=ON-IDLE},\ NI_2\text{=s}) + power_{MBU}\ (NI_1\text{=ON-ACTIVE},\ NI_2\text{=s})]$$

where NI_1 represents the NI through which the data is sent, while NI_2 is being in a state s.

Table 4: Performance of strategies related to the application-data flow adaptations

Strategy	S_4 (4b, n=4)	S_5 (3b, n=4)	S_6 (1b, n=4)	S_7 (4c, n=6)
WLAN	alternates: ON-IDLE ↔ ON-ACTIVE	alternates: ON-IDLE ↔ ON-ACTIVE	OFF	OFF
GPRS	ON-IDLE	OFF	alternates: ON-IDLE ↔ ON-ACTIVE	alternates: ON-IDLE ↔ ON-ACTIVE
AppRTT [ms]	3000 + 2211	3000 + 3513	3000 + 5505	5000 + 6693
normalized power	0.00966	0.00654	0.00584	0.00555
power eff.	-0.6	32	39	42
WWAN reachability	yes	no	yes	yes

From the Table 4 we conclude that for a patient in non-emergency, strategies S_6 and S_7, where data is sent in burst through the GPRS NI, are more power efficient than those where data is send through WLAN NI, however less AppRTT-efficient. The result for strategy S_4 shows that this strategy is a bit less power-efficient comparing to S_{EM}. This is due to the high power consumption of MBU for WLAN ON-IDLE state (as we explained for Table 1).

For the cases with larger bursts (i.e. larger n), we use the results for NetPerf measurements to extrapolate the efficiency, as presented in Table 5. Hereto, we estimate the maximum:

AppRTT ≈ (n-1) + C [s],

where C is a constant, with a slight dependency on n, in the order of a few seconds and approximately represents the time of n data samples. Similarly, the normalized power is computed as:

normalized average power ≈

$$1/n\ [(n-1)\ power_{MBU}\ (\text{WLAN=ON-IDLE, GPRS=s}) + power_{MBU}\ (\text{WLAN=ON-ACTIVE, GPRS=s})]$$

where s is a given state of the GPRS NI. For large values of n, the normalized average power approaches the $power_{MBU}$ for (WLAN=ON-IDLE, GPRS=s) case.

Strategies S_8 and S_9 as defined in Table 5, disclose large difference for the WLAN NI alternating between ON-IDLE and ON-ACTIVE states, and GPRS being in ON-IDLE or OFF states. If n is large enough, one may switch the WLAN NI between OFF and ON-ACTIVE states[7] resulting in strategies S_{10} and S_{11}.

[7] These NI state changes impose a fixed power penalty higher than that in case of strategies S8 and S9. This penalty is negligible as n increases.

Table 5: Asymptotic performance of the extrapolated application-data flow adaptation and NI activation strategies.

Strategy	S_8 (large n)	S_9 (large n)	S_{10} (large n)	S_{11} (large n)
WLAN alternates:	ON-IDLE↔ ON-ACTIVE	ON-IDLE↔ ON-ACTIVE	OFF↔ ON-ACTIVE	OFF↔ ON-ACTIVE
GPRS	ON-IDLE	OFF	ON-IDLE	OFF
AppRTT [ms]	≈ n-1 + C	≈ n-1 + C	≈ n-1 + C	≈ n-1 + C
normalized power	≈ (n-1)/n 0.00963	≈ (n-1)/n 0.00568	≈ (n-1)/n 0.00487	≈ (n-1)/n 0.00092
power eff.	-0.3	41	49	90
WWAN reachability	yes	no	yes	no

As can be seen from Table 5, strategy S_{10} is slightly more power efficient than S_7 while it induces very large AppRTT. Only the power efficiency of strategy S_{11} is significantly higher with respect to that of the strategy S_7, but the drawback is that the mobile device is not WWAN-reachable. The results of Table 5 indicate that adapting patient vital signs data and sending it in large bursts (i.e. with a large n) is not power efficient enough to motivate having a very long AppRTT or being WWAN-unreachable.

6 Related Work

Related work on NI activation strategies is mainly theoretical, and moreover focuses mainly on applications in which mobile user acts as an occasional data consumer and does not produce data, as in the MobiHealth system. For example, authors of [10-13] consider NI strategies together with methods for local or proxy-based caching data for users of email application and web-services. The work reported in [14] reduced energy consumption by introducing a NI ON-IDLE stand-by state, at which the mobile device is wakened-up if there is an incoming network event, e.g. a call.

Considering the impact of applications on NI power consumption, the authors of [15] studied the WLAN NI energy consumption for different multimedia data streaming applications like Microsoft (Windows media), Real (Real media) and Apple (Quick Time) content. They considered only WLAN NI and downlink data streams. Similarly, but from the NI perspective, authors of [16] measured NI energy consumption of use/and alternating between BT and WLAN NIs for a multimedia content download.

Furthermore, there exist general research frameworks, in which NI activation strategy is considered as one of multiple features. For example, the research reported in [17, 18] considered a simultaneous operation of NIs in multi-homed mobile hosts, and introduced a Basic Access Network to carry out signalling for network discovery, NI selection, inter-network handover, location updates, paging, authentication, authorization, and accounting. Authors tackled the NI activation strategy objective only theoretically. Similarly, the theoretical framework proposed in [19] focuses specifically on the WLAN NI activation strategy, based on the WLAN network availability, network state (throughput, delays and reliability), as well as application QoS requirements. Their NI strategy assumes that the UMTS network is always ON and available. However, they do not consider the NI energy consumption in their framework.

Authors of [20] aimed to estimate WLAN network availability and conditions without powering a NI up - based only on historical data. They have simulated healthcare application by data for 3 leads ECG; however they neither include BT power consumption for sensor-set nor adapted application-data flow being sent by network (i.e. it was fixed at 5 minutes).

And finally, in our previous work [21], we have studied the NI activation strategies based on its relative energy cost. We measured energy costs while sending dummy TCP packets over a given NI. However, the data range send was 25 kbps (GPRS) and 2 Mbps (WLAN), and the mobile devices, as well as measurements conditions were different, which made these results unusable for the MobiHealth case study presented in this paper.

We would like to emphasize the contribution of our research as an extensive case study of the existing system for telemonitoring of patient's health conditions. Based on our study we provide extensive and valuable recommendations for the system users, i.e. healthcare professionals and their patients.

7 Conclusions and the MobiHealth System Recommendations

Based on our measurements, we derive some conclusions and recommendations for the MobiHealth system and its cardiac telemonitoring application, concerning the most efficient and effective NI activation strategies along the power and the delay QoS requirements. Particularly, we have observed that the GPRS and WLAN NIs have complementary power and delay profiles. For GPRS, there is lower energy cost to maintain connectivity and lower energy for sending data, but higher delay. On the other hand, the energy cost of a WLAN data send can be higher, but delay is lower. Minimal power is used in strategies where data is stored and send later it bursts (S_8-S_{11}), resulting in the highest delay (as they include long local storage time). Maximum power is used by S_{EM} (comparing to the other strategies where data is send in real-time: S_1 and S_2), while the delay is minimal.

In an emergency case, the WLAN ON-ACTIVE and GPRS ON-IDLE NI activation strategy should be used, as it provides the system with the lowest patient vital signs data delay. However, if WLAN is not available, GPRS ON-ACTIVE and WLAN-OFF case should be used.

In non-emergency, when the user needs to be reachable, the data can be send in burst and power needs to be optimized, we recommend the use of the WLAN ON-ACTIVE and GPRS ON-IDLE strategy. The recommended burst size is the one corresponding to n=4 seconds of patient vital signs data. However, if WLAN is not available, GPRS should be used with WLAN OFF with a recommended burst size corresponding to n=6 seconds patient vital signs data.

Another conclusion derived from our studies is that the device used as the MobiHealth's MBU – Qtek 9090 is not necessarily the best choice from the power efficiency perspective for GPRS and WLAN interfaces. As one of the future work areas, we recommend execution of measurements for another device as an MBU and comparison of results with ones obtained from this study.

Moreover, future work encompasses work on more elaborated NI activation strategies methods, like those including multiple periodic application-data flows with different delay requirements (i.e. different delay defined per application-data flow). Moreover, the NI strategy should include network's monetary cost usage and a network's security level required by the MobiHealth users. Finally, we plan to extend our study of the power- and delay application-data flow adaptation from the user's stationary position to different mobility levels, where data is send over the different WWAN networks (GPRS, UMTS, or HSPA) as available at a given user's geographical location and time.

References

[1] Tachakra, S., X. Wang, et al. (2003). Mobile e-Health: the Unwired Evolution of Telemedicine. *Telemedicine Journal and e-Health*, 9(3): 247-257.

[2] van Halteren, A., Bults, R., et al. (2004). Mobile Patient Monitoring: The MobiHealth System. *The Journal on Information Technology in Healthcare* 2(5): 365-373.

[3] Dokovsky, N., A. van Halteren, et al. (2004). BANip: Enabling Remote Healthcare Monitoring with Body Area Networks. Intl Workshop on Scientific Engineering of Distributed Java Applications (FIJI03), Luxembourg, Springer Verlag.

[4] Pawar, P., van Beijnum, B. J., et al. (2007). Context-Aware Middleware Support for the Nomadic Mobile Services on Multi-homed Handheld Mobile Devices. *IEEE Symp. on Comp. & Comm.*, Portugal, IEEE.

[5] Wac, K., Bults, R., et al. (2004). Mobile Health Care over 3G Networks: The MobiHealth Pilot System and Service. *Global Mobile Congress*, Shanghai, China.

[6] Broens, T., Huis in't Veld, R., et al. (2007). Determinants for successful telemedicine implementations: a literature study. *Journal for Telemedicine and Telecare*, 13(6): 303-309.

[7] Bernaschi, M., Cacace, F., and Iannello, G. (2004) Vertical Handoff Performance in Heterogeneous Networks. Intl Conf. on Parallel Processing Workshops (ICPPW04), Montreal, Canada.

[8] Twente Medical Systems Intl., www.tmsi.com, retrieved on 09/12/2007.

[9] Netperf homepage: www.netperf.org, retrieved on 25/11/2007.

[10] Flinn, J. and Satyanarayanan, M. (1999). Energy-aware adaptation for mobile applications. *ACM Symp. on Operating Systems Principles*, USA. ACM, New York, NY, 48-63.

[11] Armstrong, T., Trescases, O., Amza, C., and de Lara, E. (2006). Efficient and transparent dynamic content updates for mobile clients. *MobiSys'06*, Sweden. ACM, New York, NY, 56-68.

[12] Lufei, H. and Shi, W. (2006). e-QoS: energy-aware QoS for application sessions across multiple protocol domains in mobile computing. *QShine'06*, Canada. v.191. ACM, New York, NY.

[13] Anand, M., Nightingale, E. B., and Flinn, J. (2005). Self-tuning wireless network power management. *Wirel. Netw.* 11(4), 451-469.

[14] Shih, E., Bahl, P., and Sinclair, M. J. (2002). Wake on wireless: an event driven energy saving strategy for battery operated devices. *MobiSys'02*, USA. ACM, New York, NY, 160-171.

[15] Chandra. S., (2003). Wireless network interface energy consumption. Implications for popular streaming formats, *Multimedia Systems*, Springer-Verlag, 9(2), pp. 185-201.

[16] Pering, T., Agarwal, Y., Gupta, R., and Want, R. (2006). *CoolSpots*: reducing the power consumption of wireless mobile devices with multiple radio interfaces. *MobiSys'06*, Sweden. ACM, New York, NY, 220-232.

[17] Inoue, M. Mahmud, K., Murakami, H. Hasegawa, M. and Morikawa, M. (2004). Novel Out-of-Band Signaling for Seamless Interworking Between Heterogeneous Networks, *IEEE Wireless Comm.*

[18] Wu, G., Mizuno, M. and Havinga, P. (2002). MIRAI Architecture for Heterogeneous Network, *IEEE Comm. Magazine*, Feb 2002.

[19] Song, Q. and Jamalipour. A. (2005). Network Selection in an Integrated Wireless LAN and UMTS Environment Using Mathematical Modeling and Computing Techniques. *IEEE Wireless Comm.*, 12(3), IEEE press, 42-48.

[20] Rahmati, A. and Zhong, L. (2007). Context-for-wireless: context-sensitive energy-efficient wireless data transfer. *MobiSys'07*, Puerto Rico. ACM, New York, NY, 165-178.

[21] Bargh, M., A. Peddemors. (2006). Towards an Energy-Aware Network Activation Strategy for Multi-Homed Mobile Devices. *Intl Conf. on Pervasive Systems Computing*, USA.