Computer Communications 31 (2008) 3831–3842

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Performance evaluation of the context-aware handover mechanism for the nomadic mobile services in remote patient monitoring

Pravin Pawar [a,*], Bert-Jan van Beijnum [a], Marten van Sinderen [a], Akshai Aggarwal [b], Pierre Maret [c], Frédéric De Clercq [d]

[a] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands
[b] Department of Computer Science, University of Windsor, Ontario, Canada
[c] Institut National des Sciences Appliquées de Lyon, Lyon, France
[d] Computer Resources International, Luxembourg, S.A., Luxembourg

ARTICLE INFO

Article history:
Available online 9 May 2008

Keywords:
Handheld mobile devices
Multi-homing
Nomadic mobile services
Remote patient monitoring

ABSTRACT

Owing to the recent advances in the mobile middleware technologies, hardware technologies and association with the human user, handheld mobile devices are evolving into data producers and in turn acting as nomadic mobile service providers. For the nomadic mobile service hosted on a multi-homed handheld mobile device, context-awareness provides a capability of selecting the suitable network interface for the data transfer. This paper conducts a performance evaluation of the context-handover mechanism for the nomadic mobile services applied in the remote patient monitoring domain and hosted on a multi-homed handheld mobile device. The experimentation analyzes the suitability of a particular network for the data transfer, the effect of multi-homing on the remote patient monitoring application and the resource utilization on the mobile device. The performance analysis provides us useful insights, which are currently being exploited in the extended middleware architecture for the vertical handover support to the nomadic mobile services.

1. Introduction

Due to the convenience offered by the mobility, portability and connectivity, mobile devices are becoming increasingly popular and they have become an integral part of everyday life. Over the past few years mobile devices such as mobile phones and Personal Digital Assistants (PDAs) have become more powerful in terms of the processing capabilities and power, and available memory. Moreover, todays mobile devices are often equiped with multiple network interfaces, typically we encounter some subset of GPRS, UMTS, WiFi, BlueTooth and Infrared interfaces. Alongwith the above-mentioned technological advances, applications of mobile devices have also evolved. The early generations of mobile devices provided only basic speech-based and text-based communication facilities. However, with the ability of mobile devices to connect to the Internet, a number of applications in the diverse areas such as e-commerce, information and entertainment are available to their users.

In this paper, we consider a service to be a unit of well-defined functional behavior (in syntax and semantics) that is offered by a software entity for use by other software entities [1]. The *Service Oriented Architecture* (SOA) paradigm allows the flexible service provisioning, service selection and service composition in the Internet. Traditionally, mobile devices take on a service consumer role. However, today these devices have sufficient resources to host services, and thereby have the ability to become a part of the service discovery network. A mobile device in the role of a service provider enables, amongst others, entirely new scenarios and end-user services. This paradigm shift from the role of service consumer to the service provider is also a step towards practical realization of various computing paradigms such as *pervasive computing, ubiquitous computing, ambient computing* and *context-aware computing*. For example, the applications hosted on a mobile device provide information about the associated user (e.g. location, agenda) as well as the surrounding environment (e.g. signal strength, throughput). Mobile devices also support multiple integrated devices (e.g. camera) and auxiliary devices (e.g. GPS receivers, printers). For the hosted services, it provides a gateway to make available its functionality to the outside world (e.g. providing paramedics assistance). In practice, the role of mobile devices as a service provider is being realized by a few technical platforms. For instance, in [2] a proxy-based middleware is presented for the development and deployment of application services hosted on a mobile device. In [3,4] a lightweight infrastructure is proposed to host web services on a mobile device. We name such a service as

* Corresponding author. Tel.: +31 53 489 2692; fax: +31 53 489 2287.
E-mail addresses: P.Pawar@utwente.nl (P. Pawar), B.J.F.vanBeijnum@utwente.nl (B.-J. van Beijnum), M.J.vanSinderen@utwente.nl (M. van Sinderen), akshaia@uwindsor.ca (A. Aggarwal), pierre.maret@liris.cnrs.fr (P. Maret), frederic.de.clercq@skynet.be (F. De Clercq).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.04.020

a *Nomadic Mobile Service* (NMS). A nomadic mobile service is hosted on the mobile host such as a handheld device, mobile phone or any type of embedded device capable of connecting to the Internet using wireless network. The mobile device roams from one mobile communication service to another which gives the services they host nomadic characteristics [2]. One of the applications of the NMSs is a *remote patient monitoring service* [5] in a (mobile) m-Health domain. Using a remote patient monitoring service, patient's mobile device acquires the vital signs data from the sensors attached to the patient's body, (pre-)processes the data locally at the mobile device, and sends the data to a m-Health *back-end system*. At the back-end system, the data is made available as services that can be used for any desired purpose, e.g. real-time retrieval by a qualified health professional.

Mobile devices equipped with multiple interfaces could connect to the Internet via different access technologies, and may even be capable to have multiple simultaneous connections, this capability is called *multi-homing* [6] and the utilization of it has been extensively reported in the literature [6–11]. Multi-homing is generally used to provide mobile devices with the high availability Internet connectivity, redundancy, load balancing, cost-based communication decisions, low latency handover and Quality of Service (QoS) improvement. However, to benefit from the multi-homing three main problems/issues need to be addressed. Firstly, in the case where multiple interfaces could be used, the information about these interfaces should be obtained. Secondly, in the case where a particular selected interface becomes unavailable; a handover to another interface is to be supported. Thirdly, in the case of the availability of multiple networks, a certain criteria should be used to select one of the available networks. For the nomadic mobile services on the multi-homed handheld mobile devices, the context-aware middleware presented in [12] addresses these aspects. The characteristics of the context-aware middleware proposed in [12] are that it provides the functionality for the application (HTTP) level handover and uses the communication context information (e.g. network availability in the form of cross-layer information). We refer to [12] for the detailed description of the issues in providing handover support to the nomadic mobile services on the multi-homed mobile devices.

One of the ways to capture the scope of this research is the analysis of scenarios and the identification of use cases from these scenarios [13]. Provided this, in the following, we present a motivating scenario that describes a possible day in the life of Mr. Janssen, an epileptic patient and later explain the aim and the goal of the paper from the observations.

> Mr. Janssen is a 46-year-old man suffering from epilepsy. Recently, he has been wearing a 24-hour seizure-monitoring system, consisting of a sensor set for collecting bio-signals and a Mobile Base Unit (MBU), for the transmission of the collected data. When Mr. Janssen is at home, a Wireless Local Area Network (WLAN) such as Wi-Fi is available to transfer his raw Electro Cardiogram (ECG) and activity information to the remote monitoring centre, e.g. the back-end system. Despite his epilepsy, Mr. Janssen is very particular about his routine and leaves for the office by his car. Since there is no WLAN network available in the car, the bio-signals of Mr. Janssen are being sent over the Wireless Wide Area Network (WWAN) network such as GPRS. However, since the uplink throughput of the GPRS network available to the mobile device is less than the requirements for a full set of vital signs, only the most important vital signs are being sent to the back-end system. Later, when Mr. Janssen is in his office, the WLAN network in the office is used to transmit the full set of vital signs. Mr. Janssen is used to drive to the nearby restaurant for lunch. During the course of driving, the vital signs are being transmitted over the WWAN network. On the way to the restaurant, a possible imminent epileptic seizure is detected and, very likely, is going to occur within a few seconds. Mr. Janssen is immediately warned, and stops driving. At the same time, an alarm and the geographic position of Mr. Janssen are sent to the monitoring centre and an ambulance with a health team is sent to Mr. Janssen. While driving, the team is constantly informed about the position of Mr. Janssen. When the team members arrive, they find Mr. Janssen in a status epilepticus, meaning that seizures keep following each other while Mr. Janssen is unconscious. Medical action is necessary and Mr. Janssen is transported to the nearest hospital by ambulance. As soon as the ambulance reaches the hospital premises, the available WLAN network is used for the vital signs transfer such that the doctors monitoring his signals could see a full set of vital signs on the display and prepare for the treatment accordingly.

By analyzing the above case study, it could be seen that the technical platform facilitating the vital signs delivery over the wireless networks should take into account on one hand the requirements imposed by the medical protocols and practices while on the other hand a variety of context information which is useful to make a decision on the network selection. Medical protocols and practices prescribe strict requirements on the quality of the vitals sign data set, such as sampling rate, sample size. Depending on the type of vital signs data to be obtained, in a realistic scenario, application data streams from 28–40 kbps typically occur. However, in practice, this amount of throughput may not be provided by the communication network, to which a patient is connected (e.g. GPRS). In the remote monitoring scenario, the caregivers or paramedics need to be informed about the critical conditions of a patient within a specified time such that the patient should not be engaged into the potentially dangerous activities (e.g. driving car). Hence low latency is desired from the selected network interface. The latency of the available wireless networks varies widely. For example, GPRS has a latency of 500–1000 ms while UMTS has a latency of 250–340 ms. For the streaming data, the latency during the vertical handover is also critical.

To realize a scenario like this, we proposed a context-aware middleware architecture in [12]. However, focusing on the design and architecture, [12] does not report any performance issues or results. NMS Applications like remote monitoring of patients impose strict requirements on the quality of the vitals sign data set (e.g. sampling rate and sample size) and the network characteristics (e.g. network throughput and delay). It is therefore necessary to evaluate the performance of the multi-homing mechanism proposed in [12] and to investigate related issues such as the latency of handover, network throughput, network delay and the resource utilization on the mobile device. The network interface selection mechanism reported in [12] considers only the link capacity (i.e. a static characterization of the available wireless links) of the wireless networks to take a handover decision However, this network selection mechanism could be enhanced by considering other needed information such as the collective throughput and latency requirements of all the running nomadic mobile services, the user's network preferences, device's processing capabilities, remaining batter power and the actual throughput offered by a particular network at a given place and time. Our longer term goal is to design such a mechanism. However, for this design we need a better understanding of the relationships between various parameters to be observed. E.g. considering a number of elements for the network selection, obtaining this information and related calculation may be CPU and memory intensive tasks. The handover latency is also an extra overhead when combined with the latency of the communication network. Motivated by this need, in this paper we come up with the performance evaluation objectives which provide us useful insights about the throughput offered by the communication networks, handover latency and the resource utilization on the mobile device. To evaluate these objectives, we conduct the experimental performance evaluation. Moreover, we analyse the obtained performance measurement results that provide an initial characterization of the system behaviour and that may provide inputs for the extension of the context-aware middleware architecture reported in [12].

The remaining of this paper is organized as follows: Section 2 discusses the related research and lists the distinguishing aspects of our research. Section 3 briefly explains the *Mobile Service Platform* (MSP) middleware which facilitates the development and deployment of nomadic mobile services and provides an overview of the remote-patient monitoring system. Section 4 elicits our performance evaluation objectives based on the motivating scenario illustrated in Section 1. Section 5 describes the experimental set-up of the performance evaluation, which includes the experimental network, additional implementation issues and the data collection points. Section 6 elaborates on the experimental runs and discusses the results of the experiments. Finally, the summary and future research are discussed in Section 7.

2. Related research

Most of the research related to the performance analysis of multi-homing and the corresponding handover has been reported for mobile devices such as laptops and mobile routers supporting mobile IP. The work illustrated in [6] exploits multi-homing for the low latency handover in heterogeneous networks. Using the simulation technique [6] conducts handover experiments between the Ethernet and Wireless Local Area Network (WLAN) using Network Simulator-2 (NS-2). The obtained results show that multi-homing in the mobile IPv6 network reduces handover latency and the corresponding packet loss. The work in [7] describes multiple network interfaces support by using policy-based routing for the mobile IPv6 multi-homed mobile nodes. The interfaces of a mobile node include Personal Digital Cellular network, WLAN and Ethernet. The reported results include the latency of interface switch, standard deviation and the TCP and UDP packets received on the wired and wireless interfaces. It shows some quantitative results of performance evaluation using laptops. The in-vehicle router system to support network mobility proposed in [8] aims at ensuring Internet connectivity and network transparency to a group of nodes connected to a mobile router in a moving vehicle. The performance analysis results consider the throughput and Round Trip Time (RTT) for the bulk file transfer operation over the WLAN and Personal Handy-Phone System (PHC). The research reported in [11] proposes a policy-based hybrid approach for IPv6 multi-homing. The performance evaluation results of the load balancing policies include the average load ratio on the network links, variance and standard deviation for each policy. The study conducted in [14] is on the vertical handover performance in Mobile IPv6 networks for Ethernet LAN, WLAN and GPRS networks. In the discussion on the results, the authors compare the results gathered and analyzed from measurements and model based estimates. The authors of [15] report their experiences with migrating TCP connections using vertical handover between GPRS and WLAN networks in the mobile IP environment. The results are reported for the handover detection and latency for various proposed handover optimization techniques. The IEEE 802.21 framework [16] is intended to provide methods and procedures that facilitate handover between heterogeneous access networks. These handover procedures can make use of the information gathered from both, the mobile terminal and network infrastructure to satisfy user requirements. There are several factors such as service continuity, application class and quality of service that may determine the handover decision [16].

Compared to the related work described herewith, our contribution differs in the following three aspects: (1) performance analysis in the m-Health domain for the remote patient monitoring service where the role of a mobile device changes from the service consumer to the service provider; (2) application level measurements of the handover, network and mobile device performance; (3) although powerful, the mobile devices (handhelds) we consider (in paticular PDAs) have limited resources available compared to the laptops and routers; and (4) as opposed to simulation, we have developed an advanced mobile services platform supporting nomadic mobile services and study vertical handover based on a test-bed and measurements.

3. Introduction to mobile service platform and remote patient monitoring system

The characteristics of nomadic mobile services impose some general challenges on their design. These include: (1) potentially lower availability of the nomadic services due to limitations on the battery power and intermittent connectivity; (2) frequent change of the network infrastructure used (e.g., changing between ad-hoc and managed networks) and mobile device reachability problems due to IP address changes; and (3) shifting of the functionality of the mobile device from a lightweight service consumer to a service provider. The *Mobile Service Platform* (MSP) middleware proposed in [2] deals with these challenges and provides a supporting infrastructure for the development and deployment of nomadic mobile services.

3.1. Mobile service platform

MSP extends the SOA paradigm to the mobile device. The design of MSP is based on the *Jini Surrogate Architecture Specification* [17], which enables the devices that cannot directly participate in a Jini Network to join a Jini network with the aid of a third party. MSP consists of an *HTTPInterconnect* protocol to meet the specifications of the Jini Surrogate Architecture and provides a set of custom APIs for building and running services on a mobile device. Using MSP, a service hosted on a mobile device participates as a Jini service in the Jini network.

A nomadic mobile service, realized using MSP, is composed of two components: (1) a service running on the mobile device (referred to as *device service DS*); and (2) a *surrogate service SS*, which is the representation of the device service in the fixed network. The surrogate service registers with the Jini lookup service so that the interested clients could discover the service. Fig. 1 shows these components. The surrogate service is hosted by a *Surrogate Host*. The surrogate host functions as a proxy for the device service and is responsible for providing a service to the clients. The clients in both the wireless network as well as in the fixed network com-

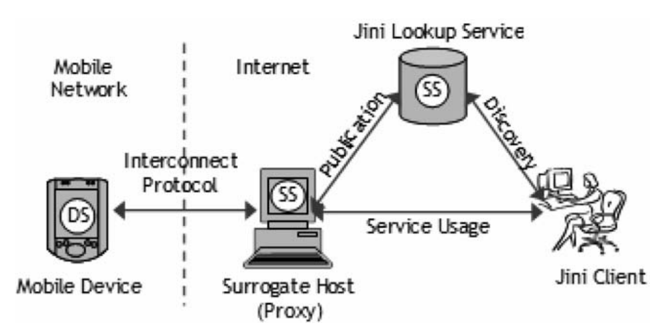

Fig. 1. Elements of the nomadic mobile service.

municate with the surrogate host to access the service. The device service runs on a mobile device and communicates with the surrogate host using an Interconnect protocol.

The MSP implementation consists of three modules: *Messages*, *Input–Output* (*IO*) and *Interconnect*. The *Messages* module defines the structure of messages exchanged between the device service and the surrogate host. The *IO* module which resides on the mobile device is responsible for the lifecycle of the device service and communication with the surrogate host. The *Interconnect* module which resides at the surrogate host is responsible for the surrogate management. MSP supports three interactions between the device service and the surrogate host. These are as follows: (1) *One-Way messaging* allows for the unacknowledged message delivery between the device service and its surrogate. (2) *Request-Response messaging* supports reliable message delivery. The request message must have a corresponding reply message. (3) *Streaming* interaction supports exchange of continuous data (streams) from the device service to the surrogate.

The MSP middleware uses dedicated control plane messages for the control, monitor and lifecycle management of the nomadic mobile services. An example of this message is the *Keep-Alive message*, which is sent by the device service at fixed intervals. The surrogate host acknowledges this message by sending a response. If the Keep-Alive message is not acknowledged in a certain time interval (*service timeout*), MSP deactivates the device service.

3.2. Remote patient monitoring system

In the health-care domain, new service scenarios are now within reach because of the use of mobile devices as service providers. Fig. 2 shows one of the applications of nomadic mobile services, namely the *remote monitoring of a patient* in the health-care domain. The following components contribute to this system:

- *Body Area Network* (*BAN*) *sensor set*: a BAN sensor set processes vital signs measured by the sensors attached to the patient's body, and outputs multiple channels of the patient vital signs data. Every vital sign is transmitted over an associated channel. It communicates with the mobile device using Bluetooth.
- *Remote monitoring service*: the remote monitoring service consists of two components: (1) *monitoring device service* on the mobile device; and (2) *monitoring surrogate* in the fixed network. The monitoring device service and its surrogate communicate with each other using the *context-aware MSP-IO* package. The monitoring device service consists of a *service buffer* which maintains the number of packets waiting to be processed by the MSP-IO. This number is mapped to the *fill level* (0–100) of this buffer. The buffer stores the vital sign data up to a predefined number of seconds (configured to approx. 60 s for the experimentation) till those are transmitted by MSP-IO.
- *Signal profile*: the signal profile informs the monitoring device service about the signals to be sent to the health-care professionals. This profile varies in accordance with the kind of treatment the patient is receiving. For example, there are two different signal profiles, one for *cardio-vascular diseases* and another for the *generic monitoring*.
- *Context-aware MSP-IO*: the context-aware MSP-IO delivers the vital sign data packets to the *back-end system* depending on the available transmission capacity (throughput) of the selected communication network (e.g. Wi-Fi, GPRS). The *Communication Context Source* (CCS) is an external element and it obtains a consistent cross-layer view of the available network resources, which we refer to as the *communication context*. It provides the context-aware MSP-IO with the information to select the access link/network to be used. CCS sends notifications on the communication context changes, for instance, a *join* to a network, or a *leave*. The components within the context-aware MSP-IO are:
  1. The *Context Processor* (CP) component subscribes to the CCS for the context change events. Such an event is triggered when a mobile device joins a new network or disconnects from one of the current networks. Based on the events reported by the notifications, the CP builds state information regarding the currently available network access links. This state information is further used by the *Context Reasoner*.
  2. The *Context Reasoner* (CR) uses the state information built-up by the CP, and the associated link characteristics, such as link capacity and Maximum Transmission Unit (MTU), and enforces a decision as to which network access link (and thus which device interface) to use for the communication with the surrogate host. Please note that, the network interface selection criteria in CR consider only the access link bandwidth capacity and availability; i.e. the access network with the highest bandwidth capacity (e.g. WLAN will always be selected over GPRS).
  3. The *Message Worker* component is responsible for receiving one-way and request-reply messages from the device service (surrogate) and sends them to its surrogate (device service). MSP-IO additionally uses the message worker to send control messages to the surrogate host.
  4. The *Stream Worker* component provides a buffer (within context-aware MSP-IO) to the device service to which a device service constantly writes the streaming data. The stream worker opens a connection to the surrogate of a device service via the selected network interface, reads the data from the buffer and transmits the data to the surrogate host. If the selected network interface is different than the one currently in use, the CR informs the message worker and the stream worker about the availability of a new network interface. In the case of unavailability of any network interface, the device service is notified of the unavailability of Internet connectivity.

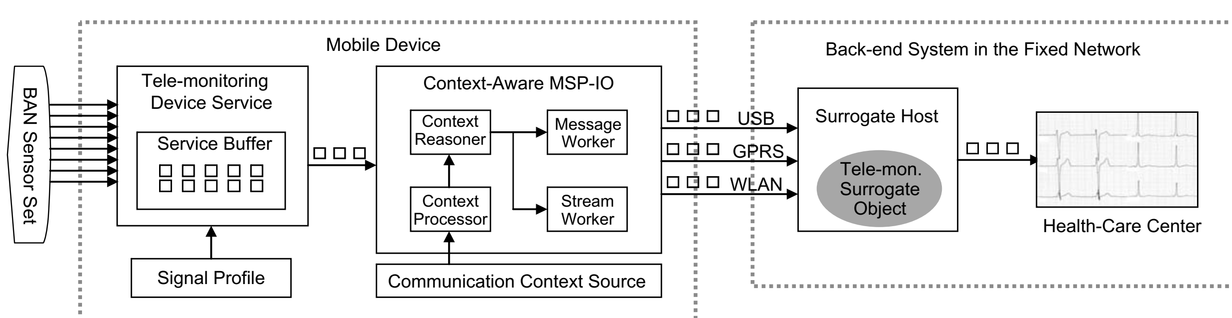

Fig. 2. Overview of the remote patient tele-monitoring system based on nomadic mobile services concept.

For the further details on the design and implementation of MSP, we refer to [2]. For the design and architecture of Context-Aware MSP, we refer to [12].

4. Performance evaluation objectives

According to the discussion and scenario described in Section 1, we classify our performance objectives in the following 3 groups: (1) bandwidth and latency characteristics of the selected network for the given signal profile; (2) vertical handover functionality performance; (3) resource utilization at the handheld mobile device during the selected network usage.

4.1. Network performance

The existing wireless communication technologies could be broadly divided into two categories: those providing a *low-bandwidth and high-latency service* over a wide geographic area (e.g. GPRS) and those offering a *high-bandwidth and low latency service* over a narrow geographic area (e.g. Wi-Fi) [14]. The remote monitoring service has varying bandwidth requirements depending on the signal profile. The multi-homing support to the nomadic mobile services (NMS) offers a higher probability that the mobile device is connected to the Internet in such a way that the requirements of the device service (i.e. required/desired application level throughput) are met. The fulfillment of the NMS requirements is bounded by the *end-to-end QoS* (e2eQoS) provided by the underlying heterogeneous networks [18]. In our case, the e2eQoS particularly encompasses the NMS-level throughput (in kbps) and delay (in milliseconds) of the underlying data communication path between the device service and the surrogate associated with it is placed. In most cases, the first hop in that path is a wireless (mobile) network, which is a bottleneck in the end-to-end path. Moreover, the observed bandwidth and latency at the application level differ from the bandwidth and latency offered by the lower layers [19]. If the wireless communication network provides higher throughput than the rate at which vital signs are generated, then the issue is how to measure the maximum available throughput. To solve this problem, it suffices if there is a adequate amount of data (more than the maximum available throughput) available in the service buffer awaiting its transmission. Summarizing this discussion, our network performance objectives include the following:

- *Monitoring service buffer fill level*: we observe the dynamics of the service buffer fill level during the experimentation to know the maximum throughput offered by the selected wireless communication network.
- *Vital signs delivery throughput* (B_v): this is the amount of the vital signs transferred for the given signal profile over the selected communication network during a remote patient monitoring session.
- *Keep-Alive RTT* (R_k): the latency of the selected communication network as well as the amount of data of the vital signals in transit affects the Keep-Alive RTT. The Keep-Alive RTT indirectly measures the delay of the selected network. Measuring this parameter also helps in configuring the proper value for the *service timeout* (We refer to Section 3.1 for these terminologies).

4.2. Vertical handover performance

The vertical handover process is composed of two phases: (1) Detection and handover triggering; (2) Handover execution. The detection, handover triggering and execution phases contribute to the handover latency and could be measured by the following [14]:

- *Delay for detecting the lower layer events*: Basically this is the delay between the occurrence of a particular event in the system and the notification reporting the event. In our case, the CCS detects the events such as availability of a new network and informs the context processor component.
- *Delay for configuring the new IP address* (D_t): This delay is defined as the time elapsed between the reception of network change event from the CCS and instructing the message worker and stream worker components of MSP-IO to use the new network interface.
- *Handover execution delay* (D_e): This is the delay incurred between the configuration of the new IP address in MSP-IO and the arrival of the data packets at the surrogate.

However, we will not focus on the delay for detecting the lower layer events because CCS is developed by a third party [20]. We are interested to measure D_t and D_e to know the vertical handover latency at MSP-IO.

4.3. Resource utilization on the mobile device

Though today's handheld mobile devices have enhanced capabilities, these devices still have limited resources compared to the desktop and notebook computers. Because of the number of buffers in the remote patient monitoring system (Fig. 2), a considerable amount of memory is used on the mobile device. The conversion of the signals obtained from the BAN Sensor System involves certain processing, which could be demanding for the mobile device [13]. Because of these factors, we are also interested to monitor the processor utilization and memory usage of the mobile device during the performance evaluation exercise.

5. System under test

This section describes the test-bed, lists the system implementation details, and provides an overview of the measurement data collection points.

5.1. Test-bed

The test-bed used for the performance measurement is shown in Fig. 3. In this test-bed, a handheld mobile device is equipped with a GPRS interface, a WLAN (Wi-Fi) interface and a USB interface. Hence, a selection out of three access networks could be potentially made. Regarding GPRS, the maximum uplink throughput is 26 kbps (using 2 time slots). The test-bed uses *IEEE 802.11b* for the WLAN. The BAN consists of a mobile device and a sensor set. The mobile device used is a QTEK 9090 pocket PC running the *Windows Mobile 2003 Second Edition* operating system with the Intel PXA263 400 MHz processor, 128 MB RAM and 32 MB flash memory. The mobile device communicates with the BAN sensor set using Bluetooth. The surrogate host runs on a server connected to the university's fixed network.

We use two signal profiles – the first profile for the cardio-vascular diseases referred to as *cardio signal profile*, and the second profile for the generic monitoring referred to as *generic monitoring signal profile*. These profiles are designed in the Health-Services 24 project [21] and they generate vital sign data at the rate of 25,880 bps (3235 Bps) and 36,864 bps (4608 Bps), respectively, from the BAN sensor set to the monitoring device service. For the description of sensors and sample size for the data generated by each sensor, we refer to [22].

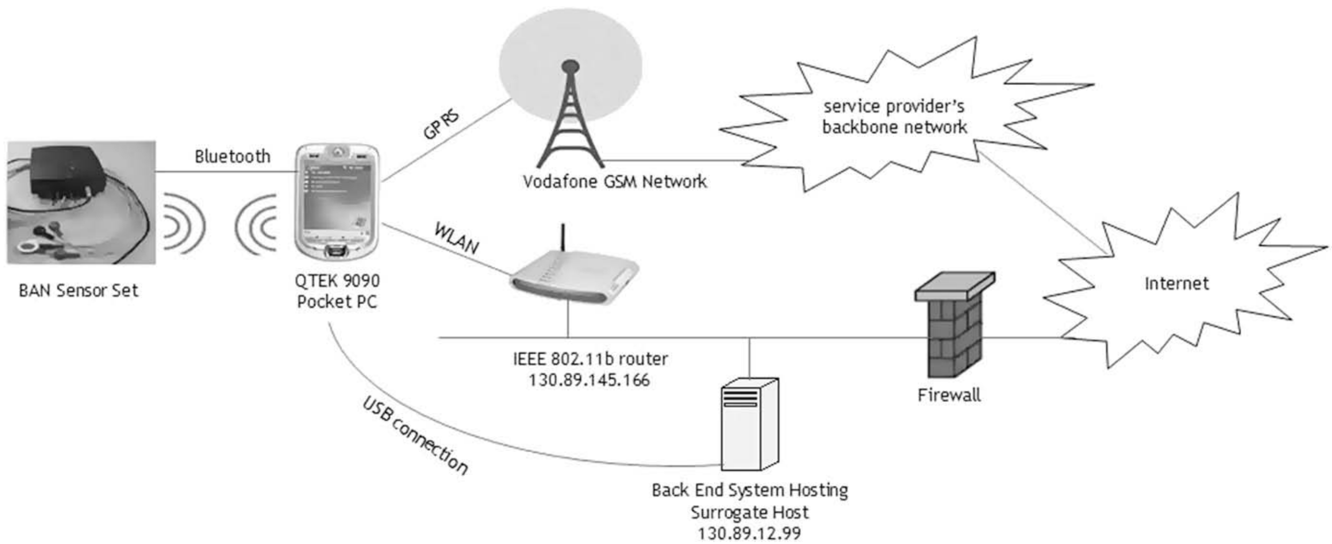

Fig. 3. Test-bed setup for the experimental evaluation.

5.2. Brief information about system implementation

The MSP is developed in Java. We have developed an *HTTP* implementation of the Interconnect protocol specified in [17] so that the device service is able to communicate with its surrogate. Context Processor, Context Reasoner, Message Worker and Stream Worker are a part of MSP-IO package (see Fig. 2). Context Processor is a thread, which interfaces with the CCS using *Java Native Interface* (JNI). The Context Reasoner uses the *KXML* library to parse the XML representation of the network state. The Message Worker and Stream Worker are also threads and use the *Apache HTTPClient* library to send messages and transmit streams to the surrogate host. The Context Reasoner converts the IP address of the best network interface to the *InetAddress* and changes the *hostConfiguration*, which is later used by the *HTTPClient* to open an HTTP connection. In case of no Internet connection is available; a device service is notified by means of a *Java exception*. The monitoring device service and surrogate are also implemented in Java.

The Communication Context Source (CCS) implementation is based on the *Network Abstraction Layer* (NAL) reference implementation for Windows CE [23] with the extensions to generate network resource descriptions in XML. The tool to log memory and CPU percentage on the QTEK 9090 Pocket PC is a variant of the *Task Manager 2.7* tool [24].

5.3. Data collection

Based on the performance evaluation objects, and the performance measures of interest identified previously, we instrumented nine data collection points. Based on the data collected from these data collection points, values for the performance measures are computed off-line. The collection points are shown in Fig. 4, the data collected at each point are:

1. Number of bytes of the vital signal data being written to the monitoring service buffer.
2. Monitoring service buffer fill level percentage.
3. Number of bytes of the vital sign data being sent by the message worker and stream worker to the MSP-IO.
4. Number of bytes of the vital signal data and control messages being sent over the selected communication network.
5. Delay D_t for configuring the new IP address.
6. Memory and CPU utilization.

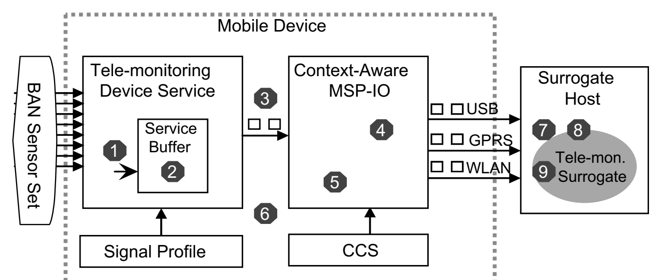

Fig. 4. Data collection points.

7. Handover executing delay D_e.
8. Number of bytes of the vital signal data and control messages received.
9. Keep-Alive messages RTT.

Regarding the time at which we ran the experiments as to collect the raw data special care has been taken for the following potential problem. From the test-bed set up it follows that the WLAN access point is dedicated to the PDA, whereas the GPRS network used is operated by a Public Network Operator. In particular, we deployed the WLAN access network without background traffic. The GPRS access network will in general have background traffic, of which we have no quantitative knowledge. Therefore, there is the potential problem that the circumstances under which the measurements are performed as such that the results become incomparable. However, the measurements have been carried out at the University of Twente and at night times, this has two main advantages. First of all, the university is located outside of the dense city area. Secondly, at night-times GPRS traffic may be expected to be low. The combination of these two makes the likelihood very high that the background traffic in the GPRS access network is close to zero. We therefore conclude that under these circumstances a fair comparison of the results can be made.

6. Experiment runs, results and their interpretation

In this section, we provide the description of the experiment runs, present the obtained performance evaluation results and interpret them. The results provided herewith are divided into three categories as identified in Section 4.

6.1. Description of the experiment runs

Similar to [14] we study the system behavior and performance for the following two different handover scenarios:

(1) *User handover (triggered manually)*: the WLAN interface in use is disabled/enabled using the WLAN connection settings on the mobile device. For the USB interface, we insert/remove the PDA in the USB cradle.
(2) *Forced handover (using unplugged base-station)*: the WLAN base-station is disabled by unplugging it from the power outlet.

With the three interfaces of the mobile devices, and the two different handover scenarios stated above we have conducted a number of experiment runs for the cardio signal profile and the generic monitoring signal profile, respectively. To analyze the network performance, the following is the sequence of actions for the Cardio profile:

1. The mobile device is connected to the GPRS network all the time.
2. The mobile device is connected to WLAN, which results in a handover to the WLAN network.
3. After some time, we switch off the WLAN base-station, which results in the forced handover to the GPRS network.
4. While connected to the GPRS network, when the monitoring service buffer fill level reaches its maximum value (because both the cardio and generic monitoring profile generate data at a higher rate than can be transferred over the GPRS interface), we switch on the WLAN base-station, resulting in the forced handover from the GPRS interface to the WLAN interface.
5. After a certain amount of time we disable the WLAN interface manually resulting in the user handover from the WLAN interface to the GPRS interface.
6. While connected to the GPRS network, after the buffer fill level reaches to its maximum, we connect the mobile device to the USB interface. After this step, MSP-IO uses USB connectivity.

For the generic monitoring profile, in addition to above steps, initially, we perform one extra handover from the GPRS to WLAN network.

To study the suitability of the network for the vital sign data transfer, we run the monitoring service over the duration of a general monitoring session (around 30 min) and observe the steady-state vital sign data transfer throughput and the Keep-Alive RTT over the selected network. Please, note that we did not perform any kind of handovers during this type of measurement. The results of the measured Buffer Fill Level, Keep-Alive RTT and vital signal delivery throughput are described in Section 6.2.

To analyze the performance of a vertical handover, we performed a set of experiments involving the user handover and forced handover. The forced handover could be performed only for the handover which involves the WLAN network. This is the case because it is possible to power off the WLAN base-station, thus forcing the MSP-IO to use the other available network. Hence, the results reported for the forced handover in Section 6.3 do not include handovers between USB and GPRS. The handover experiment is repeated for about 10 times for each possible handover in between the GPRS, WLAN and USB networks.

To obtain the resource utilization on the mobile device, the log of memory and CPU utilization of the monitoring device service and MSP-IO combined is recorded during one of the patient monitoring sessions.

6.2. Network performance

Fig. 5 shows *the monitoring service buffer fill level measured at the mobile device vs. time* for the cardio and generic monitoring signal profiles. Fig. 6 shows *the amount of vital sign data received at the surrogate (in bps) vs. time* for these profiles.

In Fig. 5a, for the Cardio signal profile, when the mobile device is connected to the WLAN network, the buffer fill level is almost zero. After the WLAN base-station is switched off, it takes a certain time to connect to the GPRS network (this is basically the time required to obtain an IP address from the GPRS network. We observed that this time varies from time to time), there is a steady rise in the buffer fill level. However, after the GPRS interface is selected, the buffer fill level drops and later increases gradually (As can be seen in Fig. 6a, the data throughput is variable, and remains below that of the WLAN network most of the time). On connecting later to the WLAN, this level drops rapidly because of the higher throughput (8678 bps) provided by the WLAN network. This throughput peak coincides with emptying the buffer as could be observed in Fig. 5a. Similar to the transition from GPRS to WLAN, a transition

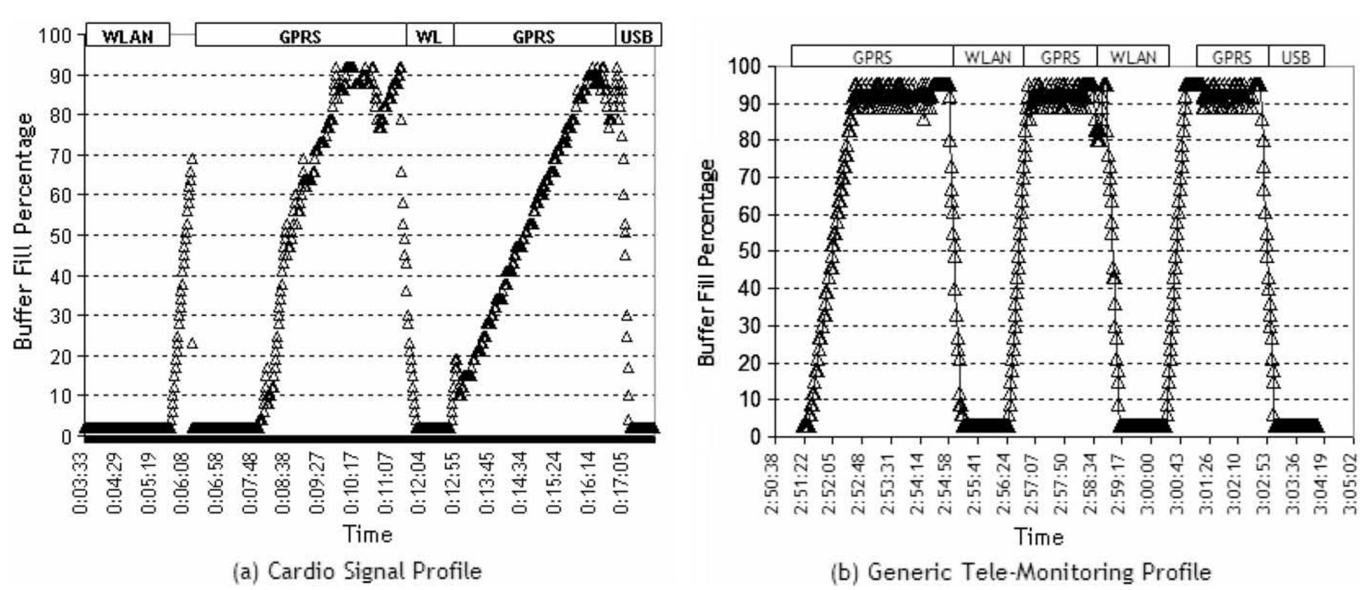

Fig. 5. Monitoring device service buffer fill percentage vs. time for the cardio and generic monitoring profiles (logged on the mobile device).

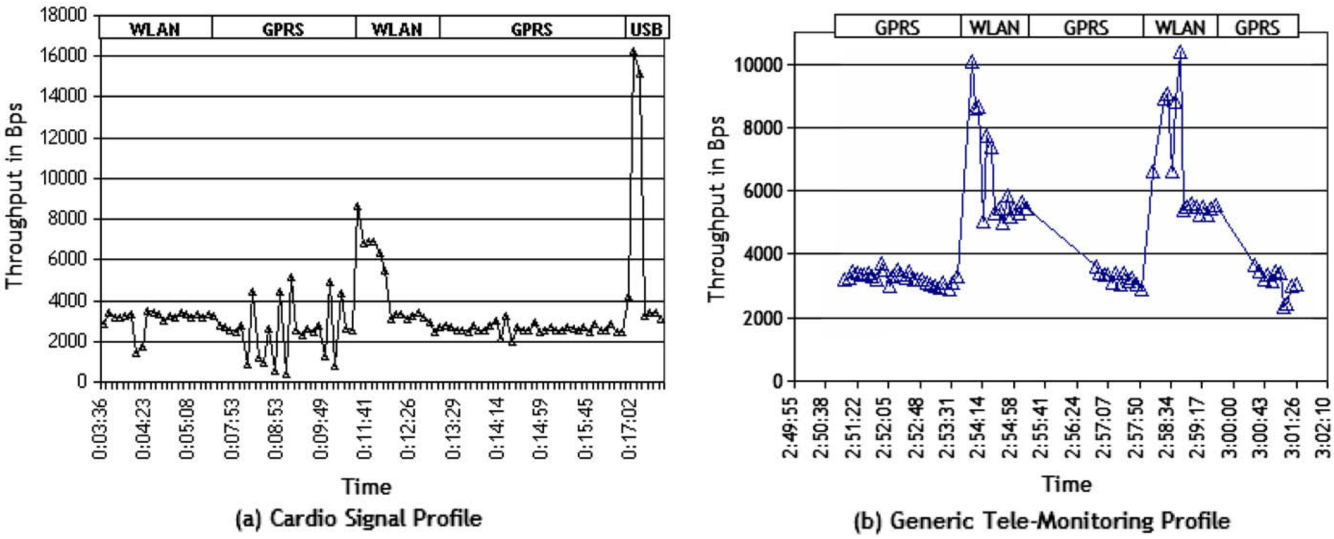

Fig. 6. Vital sign data transfer throughput vs. time for the cardio and generic monitoring profiles (logged at the surrogate).

from GPRS to USB also results in emptying the buffer, as can be seen in Fig. 5a. Fig. 6a shows that when the mobile device is connected to the USB there is also a data throughput peak (16,256 bps). For the generic monitoring profile, we observe the similar behavior for the service buffer fill level (Fig. 5b) and the vital signs data transfer throughput (Fig. 6b) as for the cardio signal profile. The highest throughput provided by the WLAN network is 10,396 bps.

Fig. 7 shows the RTT (in milliseconds) for the Keep-Alive messages logged at the surrogate vs. time. The Keep-Alive RTT is of the order of a few hundred ms for the WLAN and USB connections, however it is of the order of 10 s when connected to the GPRS network. The resulting graph also shows a lot of variance in the Keep-Alive RTT while the mobile device uses the GPRS network. One of the reasons behind this behavior is that the RTT depends (among others) on the load on the wireless link, this load in return depends on the buffer fill level. As it could be observed from the graph in Fig. 7b, before the transmission of the vital signs the Keep-Alive RTT is below 5000 ms. However, once the generation and transmission of the vital signs data start (at time 2:51:52), the Keep-Alive RTT increases substantially.

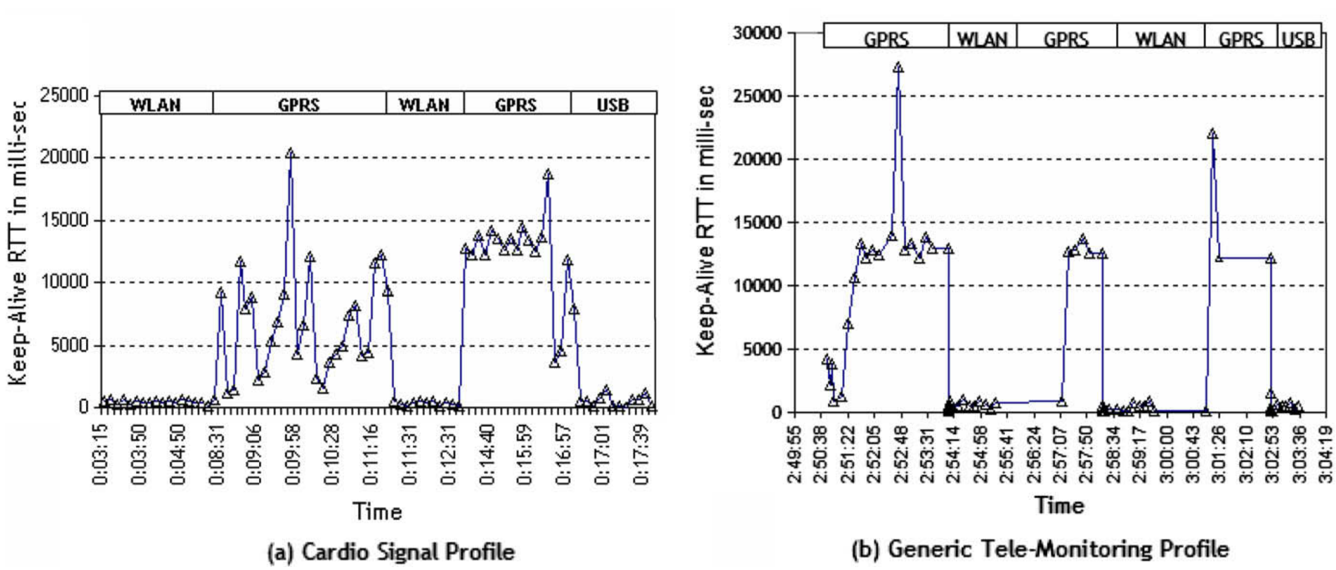

Fig. 7. Keep-Alive messages RTT vs. time for the cardio and generic monitoring profiles (logged at the surrogate).

Table 1
Vital sign data transfer throughput in bps (logged at the surrogate)

Network	Cardio profile				Generic tele-monitoring profile			
	Min.	Max.	Avg.	Std. Dev.	Min.	Max.	Avg.	Std. Dev.
GPRS	1104	29,072	20,616	2752	13,936	26,624	20,887	1873
WLAN	19,336	32,432	26,016	1417	26,720	52,648	36,997	2294
USB	18,152	46,464	25,903	3167	19,816	56,896	37,096	3304

Since we observed a significant variance in the throughput and RTT during the data transfer using the GPRS network, we conducted further experiments to observe these parameters for each network in the steady state (without any handovers). Table 1 shows the observed data transfer throughput in bps logged at the surrogate for the cardio and generic monitoring profiles. As can be observed from Table 1, the GPRS network provides around 20 kbps data transfer throughput for the vital signs. Hence it is suitable for the tele-monitoring profiles which produce vital sign data at the rate of 20 kbps or lower under the assumption that the buffers are big enough to cope with periods where the vital signs data rate requirement is higher than the available throughput. Compared to the GPRS network, the WLAN and USB connections offer better throughput for the Cardio and Generic tele-monitoring profiles. Fig. 8 shows the results in Table 1 graphically.

For the steady-state observations, Table 2 shows the Keep-Alive RTT in milliseconds for the Cardio and Generic tele-monitoring signal profiles. As can be observed from this data, the GPRS network has a much higher RTT as compared to the WLAN and USB connections. Fig. 9 shows the results in Table 2 graphically.

The results obtained for the network performance measurements show that the wireless access link capacity is not always the best indicators to choose the network. To check whether a particular network is suitable for the data transfer, the knowledge of the e2eQoS information is valuable information.

6.3. Vertical handover performance

This section describes the results of the experiments conducted according to the description earlier in Section 6 to measure the vertical handover performance. Tables 3 and 4 show the networks involved in the handover, IP address configuration delay D_t (minimum, maximum and average), handover execution delay D_e (minimum, maximum and average), total handover delay D

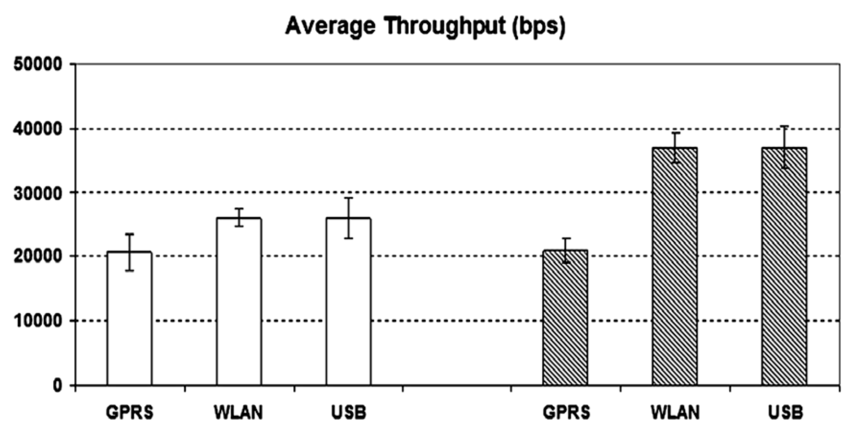

Fig. 8. The graph showing the average throughput during the steady measurement over the interval of 30 min. The white and shaded bars show the results for the Cardio and Generic tele-monitoring signal profiles. The error bars on the top of the white and shaded bars represent the standard deviation.

Table 2
Keep-Alive RTT in milliseconds (logged at the surrogate)

Network	Cardio profile				Generic tele-monitoring profile			
	Min.	Max.	Avg.	Std. Dev.	Min.	Max.	Avg.	Std. Dev.
GPRS	1066	33,379	11,408	5309	991	16,257	11,359	3677
WLAN	36	9503	563	1203	33	1052	362	220
USB	37	2833	563	323	49	2833	480	338

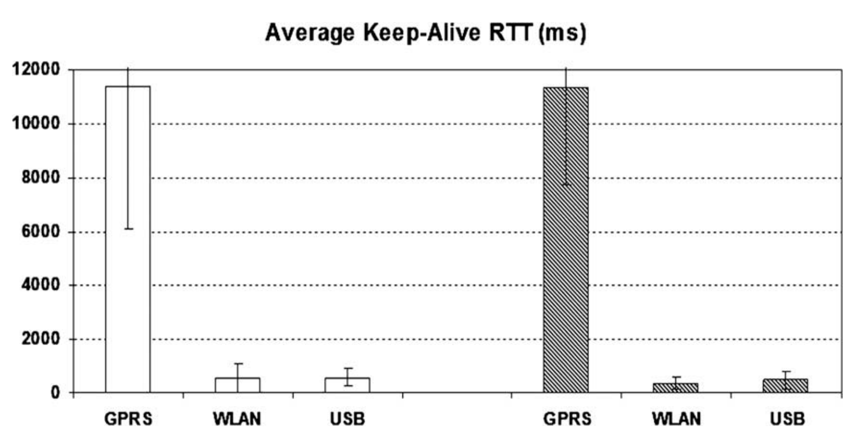

Fig. 9. The graph showing the results of Keep-Alive RTT during the steady measurement over the interval of 30 min. The white and shaded bars show the results for the Cardio and Generic tele-monitoring signal profiles. The error bars on the top of the white and shaded bars represent the standard deviation.

Table 3
IP address configuration, handover execution and total handoff delay results for the user handoff strategy

Handover	Min. D_t	Max. D_t	Avg. D_t	Min. D_e	Max. D_e	Avg. D_e	Min. D	Max. D	Avg. D	Std. Dev.
USB-WLAN	483	1301	740	1149	8042	1915	1676	14,783	2655	2238
WLAN-USB	363	870	604	333	17,503	3002	1075	18,028	3606	5466
GPRS-WLAN	584	3716	1469	328	20,634	3539	1300	21,254	5008	6172
WLAN-GPRS	520	1659	875	2060	35,538	9527	2580	36,283	10,402	11,562
USB-GPRS	648	965	747	19,627	29,713	23,388	20,592	30,355	24,135	3479
GPRS-USB	482	888	674	547	1205	819	1186	1856	1492	277

Table 4
IP address configuration, handoff execution and total handoff delay results for the forced handoff strategy

Handover	Min. D_t	Max. D_t	Avg. D_t	Min. D_e	Max. D_e	Avg. D_e	Min. D	Max. D	Avg. D	Std. Dev.
USB-WLAN	564	2868	1221	342	1309	921	1562	3210	2142	632
WLAN-USB	521	1506	933	725	6182	1954	1255	7285	2887	2224
GPRS-WLAN	449	1834	952	405	1349	796	832	3177	1747	720
WLAN-GPRS	423	853	668	5693	41,855	30,594	6386	42,673	31,262	12,143

(minimum, maximum, average and standard deviation) for the user handover and forced handover respectively. These readings are averaged over about 10 handovers in between the respective networks.

The data in Tables 3 and 4 show that the IP address configuration delay is an order of few hundred milliseconds. There is no significant difference in the IP address configuration delay between the user handover and forced handover. This could be due to the fact that we do not measure the delay for detecting the lower layer events. The handover execution delay is of the order of seconds. The handover to the USB or WLAN network results in the handover execution delay of 2–5 s. However, the handover to the GPRS network results in a large handover execution delay. Moreover, the standard deviation of the overall delay for the GPRS network is around 11.5 s, which shows the similar behavior observed for the Keep-Alive RTT in Fig. 7. However, these results are not conclusive enough because of the smaller sample size. A larger set of experimentations is needed to make useful conclusions regarding the handover delay over the GPRS network. Fig. 10 show the results in Tables 3 and 4 graphically.

6.4. System resource utilization

Fig. 11 shows the memory and CPU utilization of the remote patient tele-monitoring system during one of the tele-monitoring sessions. Initially, the memory utilization (indicated by the solid line) is low because the device service is not yet sending data. Once the device service connects to the BAN sensor set and starts sending data, the memory utilization increases and stabilizes at around 5 MB. The CPU utilization (indicated by the dashed line) is not significant at the later stage of the tele-monitoring session.

7. Summary and future work

This paper reports on the performance evaluation of the context-aware handover mechanism for the nomadic mobile services hosted on a multi-homed handheld mobile device. The performance evaluation objectives are motivated from the case study of the nomadic mobile services in the mobile health-care domain whereas a *remote patient tele-monitoring service* obtains the patient's vital signs (such as ECG) from the Body Area Network and transmit them to the health-care professionals in the real-time. We group the performance objectives into the following three types: (1) the network performance in terms of the application level End-to-End QoS (throughput and delay); (2) the vertical handover performance in terms of the handover triggering and handover execution delay; and (3) the resource utilization in terms of the memory and CPU usage on the mobile device. The results obtained for the network performance measurements show that the theoretical bandwidth and delay are not always the best indicators to choose the network. Among the networks used for the experi-

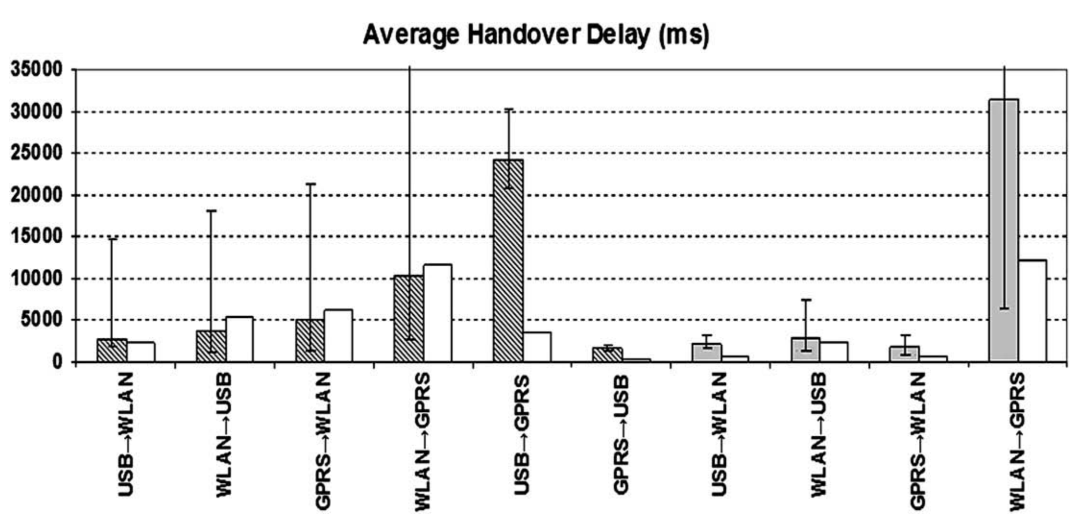

Fig. 10. The graph showing the average handover delay. The sample size for the user and forced handover is 10. Shaded, grey and white bars represent the user handover delay, forced handover delay and standard deviation respectively. The error bars on the top of these bars represent the minimum and maximum handover delay.

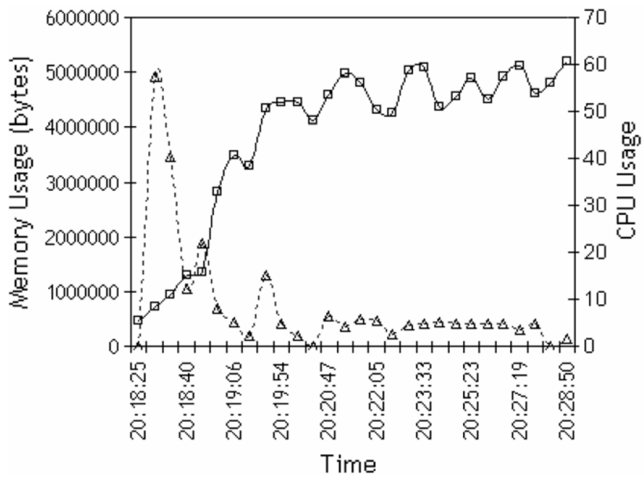

Fig. 11. Memory and CPU utilization during the remote patient tele-monitoring session.

mentation, for the given signal profiles, we observed that the GPRS network results in higher handover latency, provides lower bandwidth and higher delay as compared to the WLAN and USB connections. While the handover triggering delay is not significant, a handover execution delay is influenced by the delay characteristics of the underlying network. The resource utilization is well within the limits of the mobile device.

The unique aspects of the reported performance analysis include the following: (1) performance analysis in the m-Health domain for the remote patient tele-monitoring service where the role of a mobile device changes from the service consumer to the service provider; (2) application level measurements of the handover, network and mobile device performance; and (3) although powerful, the mobile devices (handhelds) we consider (in particular PDAs) have limited resources available compared to the laptops and routers.

During this exercise, we have the following learning: (1) along with the information about the available communication networks, it is also required to consider the end-to-end QoS (e2eQoS) requirements of the nomadic mobile services. This is particularly important because in the m-Health domain, the medical professionals have stringent requirements on the quality and the in-time reception of the vital signs data. (2) To check the suitability of a particular network for the data transfer, the knowledge of e2eQoS is valuable information. (3) One of the challenging tasks is to obtain the users preferences for the wireless networks. Specially, when the user is on the move and the available wireless networks vary as per the location and time. (4) Though the mobile devices have increased memory and processing capabilities, the power-consumption to keep all the interfaces *always powered on* to search for the communication networks in the vicinity is still a major issue. (5) Another important aspect regarding multi-homing is the handover functionality for a non-disruptive, i.e. continuous, service. However, so far, in our system, MSP-IO does not guarantee reliable vital sign data transfer (that is, due to handovers data may be lost). Data reliability (when needed) is realized by the device service (in particular the service buffer).

To handle the first four aspects outlined above, we are working on a number of context sources as following:

1. *Location and Time Context Source*: This context source provides the coordinates of the device's current geographic location and time as obtained from the Global Positioning System (GPS) receiver.
2. *Device Context Source*: For a given mobile device, this context source provides the information such as remaining battery level and the values of power-consumption per network.
3. *User Preferences Context Source*: The user preferences context source provides a ranked list of all the mobile network providers, network names and the network technology a user is subscribed to as well as a ranked list of all the device services ranked according to their importance to the user.
4. *Device Service Context Source*: This context source provides the required e2eQoS of every running device service.
5. *QoS Predictions Context Source*: This context source is based in the fixed network and will provide predictions of the expected offered-QoS in a reliable and timely manner using a multidimensional processing and history-based reasoning [18]. The information returned by the QoS predictions context source consists of all the available mobile networks along the user travel path as specified by the provider names, network names and technologies along with their coverage ranges and availability at a given location/time and predicted e2eQoS provisions (in a hierarchical structure similar to Network Cross Layer Information).

The network interface selection strategy which enhances the network interface selection mechanism proposed in [12] by considering the user, services and the communication context information available on the mobile device and the QoS context information available in the fixed network is proposed in [25]. The related work section described in [25] also describes the network interface selection strategies considered by peer researchers. To be able to take decision for the network selection using the context information obtained from the above context sources, we have extended the context reasoner component to apply an *Analytic Hierarchy Process* (AHP) [26] based optimization approach. Our future work also involves running a larger set of handover experiments so that we can identify the cause of high value of standard deviation observed for the handoff execution delay over the GPRS network.

Acknowledgements

This work is supported by Freeband Awareness project (under Grant BSIK5902390), Amigo project (IST-004182, partially funded by the European Commission) and Mobilité Internationale Rhône-Alpes (MIRA) research Grant, France. The authors thank to Aart van Halteren, Arjan Peddemors, Frank Thiele, Richard Bults, Daniel Knoppel, Peter van Tol and Katarzyana Wac, who have contributed to parts of the research described in this paper.

References

[1] C. Hesselman, et al., Discovery and composition of services for context-aware systems, in: 1st IEEE European Conference on Smart Sensing and Context (EuroSSC 2006), Enschede, The Netherlands, 2006.
[2] A.v. Halteren, P. Pawar, Mobile service platform: a middleware for nomadic mobile service provisioning, in: 2nd IEEE International Conference On Wireless and Mobile Computing, Networking and Communications (WiMob 2006), June, Montreal, Canada, 2006.
[3] S.N. Srirama, M. Jarke, W. Prinz, Mobile web service provisioning, in: International Conference on Internet and Web Applications and Services (ICIW'06), Guadeloupe, French Caribbean, 2006.
[4] D. Pratistha, N. Nicoloudis, S. Cuce, A micro-services framework on mobile devices, in: International Conference on Web Services, 2003, Nevada, USA.
[5] A. Alonso, Detailed Description of Trial Scenarios. D1.3, MobiHealth project: <http://www.mobihealth.org>.
[6] V.S. Kaulgud, S.A. Mondal, Exploiting multihoming for low latency handoff in heterogeneous networks, in: 8th International Conference on Telecommunications (ConTEL 2005), Zagreb, Croatia, 2005.
[7] R. Wakikawa, K. Uehara, J. Murai, Multiple network interfaces support by policy-based routing on mobile IPv6, in: International Conference on Wireless Networks (ICWN 2002), Las Vegas, USA, 2002.

[8] K. Mitsuya, K. Uehara, J. Murai, The in-vehicle router system to support network mobility, in: Information Networking, Networking Technologies for Enhanced Internet Services International Conference (ICOIN 2003), Cheju Island, Korea, 2003.
[9] J. Ylitalo, et al., Dynamic network interface selection in multihomed mobile hosts, in: 36th Annual Hawaii International Conference on System Sciences, Island of Hawaii, USA, 2003.
[10] N. Montavont, T. Noel, T. Ernst, Multihoming in nested mobile networking, in: Applications and the Internet Workshops Colocated with SAINT 2004, Tokyo, Japan, 2004.
[11] C.-M. Huang, C.-H. Tsai, P.-C. Su, A hybrid approach for Ipv6 multihoming providing fault tolerance and load balancing capabilities, in: 2nd IASTED International Conference on Communication and Computer Networks, Cambridge, MA, USA.
[12] P. Pawar, B.J.v. Beijnum, A. Peddemors, A.v. Halteren, context-aware middleware support for the nomadic mobile services on multi-homed handheld mobile devices, in: IEEE Symposium on Computers and Communications (ISCC'07), July, Aveiro, Portugal, 2007.
[13] H. Mei, I. Widya, T.H.F. Broens, P. Pawar, A.v. Halteren, B. Shishkov, M.J.v. Sinderen, A framework for smart distribution of bio-signal processing units in M-Health, in: Invited Paper in the 2nd International Conference on Software and Data Technologies (ICSOFT 2007), Barcelona, Spain, 22–25 July, 2007.
[14] M. Bernaschi, F. Cacace, G. Iannello, Vertical handoff performance in heterogeneous networks, in: International Conference on Parallel Processing Workshops (ICPPW'04), August, Montreal, Canada, 2004.
[15] R. Chakravorty, et al., Performance issues with vertical handovers – experiences from GPRS cellular and WLAN hot-spots integration, in: IEEE Pervasive Communications and Computing Conference (PerCOM 2004), March, Orlando, Florida, 2004.
[16] A. Dutta, S. Das, D. Famolari, et al., Seamless handoff across heterogeneous networks – an 802.21 centric approach, in: International Symposium on Wireless Personal Multimedia Communications (IEEE WPMC 2005), Aalborg, Denmark, September, 2005.
[17] Sun Microsystems, JINI Technology Surrogate Architecture Specification. Available from: <http://surrogate.JINI.org/sa.pdf>, October, 2003.
[18] K. Wac, A.v. Halteren, D. Konstantas, QoS-predictions service: infrastructural support for proactive QoS- and context-aware mobile services, in: International Workshop on Context-Aware Mobile Systems (CAMS) Colocated with On The Move 2006, October–November, Montpellier, France, 2006.
[19] R. Bults, et al., Goodput analysis of 3G wireless networks supporting m-health services, in: 8th International Conference on Telecommunications (ConTEL 2005), June, Zagreb, Croatia, 2005.
[20] A. Peddemors, H. Eertink, I. Niemegeers, Communication context for adaptive mobile applications, in: Proceedings of the 3rd Conference on Pervasive Computing – Middleware Support for Pervasive Computing Workshop (PerWare'05), March, New York, USA, 2005.
[21] Health Service 24 (eTEN-517352) project. Available from: <http://www.healthservice24.com/cms/index.asp?healthservice24>.
[22] M.O. Dimas, et al., Trial ready GPRS and UMTS networks (D 3.1), January 2003. Available from: <http://www.mobihealth.org/html/details/deliverables/pdf/new/MobiHealth_WP3_TME_D3.1_v1_01.01.03.pdf>.
[23] A. Peddemors, I. Niemegeers, H. Eertink, An extensible network resource abstraction for applications on mobile devices, in: 2nd IEEE/Create-Net/ICST International Conference on Communication System Software and Middleware (COMSWARE 2007), January, Bangalore, India, 2007.
[24] F.D. Clercq, Task Manager version 2.7 for Windows Mobile 2005 and Windows Mobile 2003, second ed., March 2006 [cited 8 February 2007]. Available from: <http://buzzdev.net/>.
[25] P. Pawar, van B.J. Beijnum, K. Wac, p. Maret, van A.T. Halteren, H. Hermens, Context-aware middleware architecture for vertical handover support to multi-homed nomadic mobile services, in: 23rd ACM Symposium on Applied Computing (ACM SAC 2008), Middleware Engineering Track, Brazil, March 2008.
[26] T.L. Saaty, How to make a decision: the analytic hierarchy process, European Journal Operation Research. 48 (1990) 9–26.