Context-Aware Computing Support for Network-Assisted Seamless Vertical Handover in Remote Patient Monitoring

Pravin Pawar, Bert-Jan van Beijnum, Hermie Hermens
University of Twente
Enschede, The Netherlands
{p.pawar, beijnum@cs.utwente.nl}, h.hermens@rrd.nl

Katarzyna Wac, Dimitri Konstantas
University of Geneva
Geneva, Switzerland
{katarzyna.wac,dimitri.konstantas}@cui.unige.ch

Abstract— The advances in the area of mobile computing is likely to make it feasible to predict the availability of wireless networks and their application level Quality of Service (QoS) characteristics along the user mobility path. Such predictions are referred to as QoS predictions which are provided by the QoS Context Source (QoSCS) hosted in the fixed network. On the multi-homed mobile devices, the QoS Predictions could be used for the handover to the optimal wireless network which satisfies the QoS requirements of the mobile applications. However, to achieve this functionality, a middleware support is necessary to obtain and make use of QoS predictions in real-time. To this end, we present design, architecture and validation of the context-aware computing support for network-assisted seamless vertical handover that uses QoS predictions to take a handover decision. We evaluate the proposed solution, in a case where it is applied in the mobile health care applications. The obtained simulation results encourage us to conduct the real-time system validation by employing the proposed solution.

Keywords- multi-homing, vertical handover, M-health services, context-aware computing.

I. INTRODUCTION

Multi-homing is the capability of a mobile device to connect to multiple IP networks. On the multi-homed mobile devices, vertical handover technique could be used to select one of the available networks and use the selected network for data transfer to and from the mobile device. Generally, a vertical handover process is composed of the following three phases: 1) Handover information gathering phase collects information required for both; to identify the need for handover and subsequently initiate it; 2) Handover decision (network selection) phase determines the most suitable network for the handover execution; and 3) Handover execution phase performs the actual handover to the network selected in the second phase [1]. A seamless handover is the handover where the transition to the new network is transparent to the mobile application [1]. In the Mobile Controlled HandOver (MCHO), the mobile device must gather all the handover information locally and make the handover decision on its own; while in the Network-Assisted HandOver (NAHO), the information collected from the network could be used by the mobile device for the handover decision [1].

In this work, we apply NAHO scheme in the Mobile (M)-Health domain for the Remote Patient Monitoring Service [2]; where the mobile device associated with the patient (on the move) continuously acquires the vital signs data from sensors attached to the patient’s body, (pre-) processes the data locally at the mobile device, and sends the data in real-time using wireless connectivity to the back-end system in the fixed network. In line with the commercially available handheld mobile devices, we consider that a mobile device is equipped with one Wireless Wide Area Network (WWAN) interface (e.g. GPRS and/or UMTS) and one Wireless Local Area Network (WLAN) interface (e.g. WiFi). In the existing implementation [3] based on the principles of context-aware computing [4], the remote patient monitoring system has a provision for MCHO, where the mobile device gathers the handover information (list of currently available mobile networks, theoretical uplink bandwidth and delay, current state of the network interfaces on the mobile device) from the Communication Context Source [5] for a handover decision. The handover decision making phase executed by the context reasoner component always prefers WLAN network over the WWAN network. In case of the availability of multiple WLAN networks, the context reasoner selects the WLAN network randomly. The performance evaluation of MCHO scheme proposed in [3] concluded that the remote patient monitoring system requires accurate information about the application level QoS to take a handover decision because of the stringent requirements imposed by the medical professionals on the quality and the in-time reception of the vital signs data from the mobile devices. However, recently multi-homing capability of the mobile device combined with the auxiliary hardware (e.g. GPS receiver), smart software (e.g. NetPerf) and increasing user willingness to share the information available on and around mobile device (e.g. location, network information), is generating a vast amount of Quality of Service (QoS) information experienced by the mobile applications (for example, see http://www.wigle.net). One of the ongoing research works in the area of mobile computing is to use such QoS information collected over a period of time to predict the availability of wireless networks and their application level QoS characteristics (bandwidth and delay) along the location and time dimensions (so called QoS predictions) [6]. We refer to the service hosted in the fixed network which provides QoS predictions as QoS Predictions Context Source (QoSCS). The hypothesis addressed in this
paper is that the NAHO scheme which uses QoS predictions obtained from the fixed network combined with the other context information (e.g. location and time) obtained locally from the mobile device offers better QoS experience to the remote patient monitoring service. For this purpose, we present herewith the design and validation of the context-aware computing support aimed at obtaining and using QoS predictions and other context information in real-time on the mobile device.

More specifically, we elaborate on: 1) Interactions between the remote patient monitoring system and QoSCS to obtain QoS predictions; 2) Details of the Context Processor component which creates a set of complete context information necessary to make a handover decision; 3) Working of the Context Reasoner component which takes a handover decision based on the context information received from the context processor. Since the QoSCS is still in the development phase [6], to get a better understanding of how the proposed NAHO scheme could be viable in practice, we built the QoSCS simulator and a user trip simulator, which are also explained in details in this paper.

The reminder of this paper is as follows. Section 2 introduces the elements of the remote patient monitoring system and details on the context-aware computing components used in this work. Section 3 is on the design of the simulators and simulation setup. Section 4 presents the validation results of the NAHO scheme. Section 5 describes the related work. Section 6 concludes the paper and provides the future work areas.

II. REMOTE PATIENT MONITORING SYSTEM

The elements of the remote patient monitoring system are shown in the Figure 1, they are distributed over the mobile device and the fixed network. The description of these elements is as follows:

**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 the patient vital signs data.

**Remote Monitoring Service:** This service consists of two components: 1) Remote Monitoring Device Service on the mobile device; and 2) Remote Monitoring Surrogate hosted by the Surrogate Host in the fixed network. The monitoring device service consists of a service buffer which maintains the number of packets waiting to be processed by the Context-Aware Mobile Service Platform (explained in the following). This number is mapped to the fill level (0–100) of this buffer.

**Context-Aware Mobile Service Platform:** Mobile Service Platform (MSP) [7] is a central component responsible for the deployment and the lifecycle management of the remote monitoring service. The design of MSP is based on Jini technology. The communication between the device service and surrogate is specified by the Interconnect protocol in Jini Surrogate Architecture Specification [8]. We have developed an HTTP implementation (referred to as HTTPInterconnect) of the Interconnect protocol. MSP uses Surrogate Host HTTP Connection for communication between the mobile device and the back-end system.

The Context-Aware MSP (CA-MSP) uses context information obtained from a number of context sources (explained in the following) and subsequent context changes for the dynamic selection of and handover to the optimal wireless network which satisfies the following network selection objectives: 1) Maximize remote monitoring service’s bandwidth requirements; 2) Minimize remote monitoring service’s delay requirements. For this purpose, MSP interacts with the Context Sources (CS) shown in the Table 1.

The QoSCS aims at providing an efficient and accurate method that generates precise QoS-predictions for a broad range of mobile networks at the given geographic location and time. The core of the prediction method is based on the Dynamic Bayesian Networks (DBN). Currently we are working on calibration and (off-line) evaluation of the prediction method in an extensive set of trials with the remote patient monitoring service.

The predictions generated by the QoSCS are organized according to the hierarchical structure shown in the Figure 2. The first level of the structure consists of basic information such as network type, network name, operator name and authentication (e.g. WiFi, Guest WLAN, University of Geneva, and Open). The second level is the prediction information in the location dimension which consists of the geographic coordinates of the centre of the network and network radius (e.g. 46.179956, 6.13896, 55m). It is quite possible that one operator could have multiple wireless networks at different locations [9]. The QoS characteristics of the wireless networks vary according to the location and time [6]. Hence the third level in the prediction information is the QoS characteristics in the time dimension which consist of the predicted bandwidth, delay and the start and end time for which these predicted values are valid (e.g. 15149 bps, 650 ms, 24 APR 2008 13:55:30, 24 APR 2008 14:20:33). The bandwidth and delay values are averaged over the duration specified by the
start and end time, such that their standard deviation is within certain limits (e.g. 10% of value).

**Table 1: Description of the Context Sources**

<table>
<thead>
<tr>
<th>CS Name</th>
<th>Context Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location And Time Context Source</td>
<td>Coordinates of the device’s current geographic location (longitude, latitude) and time (Date, HH:MM:SS) as obtained from the GPS receiver.</td>
</tr>
<tr>
<td>Communication Context Source</td>
<td>A list of mobile networks along with provider names, technologies, theoretical uplink throughput and delay (Network Cross Layer Info. in XML) in the surroundings of a mobile device at a given time and location, current state of the NIs on the mobile device.</td>
</tr>
<tr>
<td>Device Service Context Source</td>
<td>Required bandwidth and delay of every running device service.</td>
</tr>
<tr>
<td>QoS-Predictions Context Source (QoSCS)</td>
<td>All the available mobile networks as specified by provider names, network names and technologies along with their coverage ranges and availability at a given location/time and predicted QoS information (bandwidth and delay).</td>
</tr>
</tbody>
</table>

**Figure 2: Hierarchical Structure of QoS Predictions**

Figure 3 shows the flowchart depicting the working of the Context Processor component. The role of the **Context Processor (CP)** component is to get/subscribe context information from the context sources to generate a current context snapshot and provide this context snapshot to the context reasoner to be able to make a network selection decision at a given location and time. The QoS predictions useful at a given location and time combined with the QoS requirements of the device service together form the current context snapshot.

The generation of the initial context snapshot is done as follows: Upon activation of the remote monitoring device service on a mobile device (referred to as **initialization** in the Figure 3), CP obtains the current location and time from the location and time CS and provides this information to the QoSCS to obtain QoS-predictions. CP uses the **Surrogate Host HTTP Connection** to request QoS-predictions using a message referred to as **GET_QOSPREDICTIONS**. The message parameters include location (latitude, longitude) and time at the mobile device. This message is received by the SH which initializes the client to forward this message to the QoSCS. Along with the QoS-predictions, the QoSCS specifies the number of networks, the distance between the farthest network from the current location of mobile device (predictions range) and the time for which the QoS-predictions are valid (predictions time). The obtained predictions, corresponding range and time values are later encoded in the String format by the surrogate host and sent in the reply to the CP. On receiving new predictions, CP updates the predictions stored locally on the mobile device. Then CP sets a QoS-predictions timer; when this timer expires new predictions are requested. CP also sets the prediction range on the mobile device, so that when the user is on the boundary (within certain distance) of the prediction range, new predictions are requested.

There have been some issues with obtaining QoS predictions. Initially, from the CP, we tried to request predictions based on a predefined range (e.g. 500m). However, in case of a densely populated area (e.g. Manhattan) with many wireless networks, the prediction information could consists of a huge number of networks (e.g. 120 networks) and transmitting such large amount data using MSP may cause processing problems. Therefore, QoSCS sends context information consisting of predictions about a pre-defined number of networks (e.g. 30 networks with maximum distance 1250m) available for a mobile device and closest to its current location. The validity period of the QoS predictions (prediction time) is about 30 minutes (average duration of the tele-monitoring session [3]).

**Figure 3: Flowchart Showing Working of the Context Processor**

Similar to the QoS-predictions, as shown in the Figure 2, there is also a range (value of radius in the location dimension) and QoS validity time (End-Time value in the time dimension) associated with the network to which the mobile device is connected. CP updates these values and initializes a network QoS timer when a mobile device handovers to the new network. CP also has provision for handling real-time context changes. As shown in the Figure 3, based on the the context changes such as location change,
update QoS requirements

Step 1: Decide the relative importance of the optimization objectives. The Analytic Hierarchy Process (AHP) method [10] for the optimizing the network selection objectives. AHP divides a problem into several sub-problems; the solutions to the sub-problems are aggregated into a conclusion. The AHP algorithm comprises the following three steps:

**Step 1: Decide the relative importance of the optimization objectives:** The importance of an objective is decided by the weight assigned to it. These objective weights are always assigned such that their combined sum is 1. For the proposed NAHO scheme, we assign equal weight (0.50) to each of the above objectives.

![Figure 4: Flowchart Showing Handover Decision and Execution Phases](image)

CR uses the Analytic Hierarchy Process (AHP) method [10] for the optimizing the network selection objectives. AHP divides a problem into several sub-problems; the solutions to the sub-problems are aggregated into a conclusion. The AHP algorithm comprises the following three steps:

**Step 1: Decide the relative importance of the optimization objectives:** The importance of an objective is decided by the weight assigned to it. These objective weights are always assigned such that their combined sum is 1. For the proposed NAHO scheme, we assign equal weight (0.50) to each of the above objectives.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>B</td>
<td>1/2</td>
<td>1</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>C</td>
<td>1/2</td>
<td>1/2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>1/9</td>
<td>1/6</td>
<td>1/4</td>
<td>1</td>
</tr>
</tbody>
</table>

**Step 2: Compute relative weight of each available network for each objective:** This step consists of the following sub-steps:

a). For each of the above objectives, assign score on the scale from 1 to 9 to each pair of the available networks for creating pair-wise comparison matrix $P_{ij}$. For a network pair $(N_1, N_2)$, the Score ($N_1, N_2$) of value 1 means that the network $N_1$ is equally important to network $N_2$. The score of value 5 (9) means that the network $N_1$ is strongly (absolutely) more important than network $N_2$. Score ($N_2, N_1$) is an inverse of Score ($N_1, N_2$). We assign these values to the pair of network depending on how they satisfy bandwidth and delay requirements of the remote monitoring service.

b). For each optimization objective $O_i$, normalize each $P_{ij}$ (divide by the sums of the columns) and average across rows to obtain the relative weights of the networks $W_{o_i}$. Considering that there are 4 networks (namely A, B, C and D) available, a possible pair-wise comparison matrix, normalized weight matrix and network weights are shown in the Figure 5.

**Step 3: Calculate the score for each network and select the network having the highest score:** The network score is the sum of relative network weights multiplied by the objective weight.

The handover execution phase is shown in the Figure 4. The context processor and handover execution algorithms collaborate /interact at the points labelled (B) and (C) shown in Figure 3 and 4 respectively. In case the selected network is different from the network which is in use by the mobile device, a Dwell Timer concept is applied to avoid unnecessary handovers [11]. Two of the possible vertical handover scenarios involve the handover from WWAN network (e.g. GPRS) to WLAN network (e.g. WiFi) and from one WLAN network to another WLAN network. In these scenarios, if the selected WLAN network serves a smaller geographic area and the user speed is sufficiently high, it is quite possible that user just passes the hotspots without benefiting from the network’s offered QoS. Previously, researchers have used dwell timers with fixed value, e.g. 60 seconds in [11], justified by the time required to travel 1 km at the speed of 60 km/h. However, in the network environment we consider that the radius of the WiFi network is smaller than 1 km. Moreover, we argue that the value of the dwell timer must adapt according to:

1. **User speed:** The user speed based dwell timer is assigned a value equal to the time required to move out of the WLAN coverage area based on the current user speed.

2. **Handover latency:** Handover latency is the minimum time elapsed between the network selection and start using the network for the data transfer. During this period, the remote monitoring service is still generating vital signs data. The benefits of handover can only be realized if after the handover, the data sent to the service buffer during handover latency could be transmitted by the connected network. Hence, the dwell timer value based on the handover latency is the time required by the network to transmit vital signs data written in the service buffer during the handover.

The selected value of the dwell timer herewith is the lower of the dwell timer value computed based on the user speed and the handover latency. When the dwell timer expires, and when the user is still in the coverage area of the selected network, then the handover is performed; otherwise a new context snapshot is calculated.
III. SIMULATOR DESIGN AND SIMULATION SETUP

For the evaluation of vertical handover mechanisms using simulations, previous research (e.g. [12]) have assumed areas as square planes with certain size and WLAN base stations distributed randomly or uniformly in the square plane. However, in real life scenarios, such situations rarely occur. To model the location of WLAN base stations as realistic as possible, we rely on the observation that nowadays most of the businesses and institutes are covered by WLAN and we observe that there is an increasing trend in the urban environments to deploy high capacity WLANs covering large geographic areas (e.g. city centre). Moreover, a large number of private WLAN deployments may allow others to use their access network (without requiring authentication). Thus, potentially these access networks can be used by any user carrying a device with suitable network interfaces. In designing the QoSCS simulator, we took advantage of this fact and of the availability of accurate location listings of businesses and institutes on the Internet (in our case, Google maps server). The QoSCS simulator employs a novel approach of using the locations of the businesses and institutes to assign WLAN base stations.

To model user movements, researchers have used a number of user mobility models (e.g. [13]). However, these models do not necessarily represent the real world situation. To address this problem, similar to the approach used by the QoSCS simulator, the user trip simulator also uses the route information available on the Internet (our case, Google maps server) to model user movements between a source and destination locations.

The components of the QoSCS simulator and interactions between them are shown in the Figure 6. On the implementation side, QoSCS simulator is a Jini service. As shown in the Figure 6, the initialization of the service contains the following steps:

A. The Activator component reads the configuration properties (step 1) of the simulator from the property file. These properties include Jini lookup service host name and port, longitude and latitude of the central location, Google maps server address, list of businesses (e.g. hotel, school, restaurant, market, hospital) and number of business of each category (e.g. 100) to be fetched from the Google maps server. minimum and maximum throughput as well as delay to be offered by the WLAN and WWAN networks, and minimum and maximum range of WLANs.

B. The Activator initializes Google maps client (step 2) and passes the relevant information so that the client sends request (step 3) to the Google maps server.

C. The Google maps server sends (step 4) the list of businesses including their names and location in the Keyhole Markup Language (KML) format to the client. The client writes this data to the KML data files in the local storage (step 5).

D. The data in the KML files is read by the KML Data Parser to extract (step 6) the names and locations of the businesses. This data combined with the bandwidth, delay and range values read from the property file is used to generate initial WLAN base stations data such as their initial bandwidth and delay characteristics (within the minimum and maximum value limits), ranges and authentication information. This data is written to the local storage (step 7) so that every time the simulator starts, it does not need to contact the Google maps server; it initializes the WLAN base stations information with the same set of values. This is cached in the simulator (step 8).

E. On receiving a request (step 9) at the Service End Point from the QoSCS client, the request is forwarded to the Predictions Generator module which adds prediction information (step 10) in time dimension (c.f. Figure 2) to the cached predictions (step 11). To model QoS fluctuations along the time dimension, the Predictions Generator randomly increases or decreases (up to 10%) the bandwidth and delay values (still within minimum and maximum values) in a certain time slot. The duration of the time slot is also assigned randomly. The cached predictions are always updated (step 12) to remember previously assigned QoS values. This data is then sent to the Predictions Selector (step 13).

F. The Predictions Selector uses the Geographic Calculation Library (step 14) to calculate the distance between a pair of coordinates and to select the closest networks to the current location of mobile device, it calculates the prediction range and sends it (step 15) to the Service End Point; this information is then relayed to the QoSCS client.
The block diagram of the User Trip Simulator is shown in Figure 7. Similar to the QoS simulator, the User Trip Simulator reads the trip configuration (source and destination coordinates, user speed, and Google maps server address) and requests route information from the Google maps server. The route information consists of a sequence of steps where each step has its own source and destination coordinates. Since no intermediate coordinates within a step are available, the Movement Generator assumes that the user moves in straight line between the source and destination coordinates of every step. This assumption approximately matches the real world situation, because the Google Maps server provides step information approximately on the direction change of the moving user. The movement generator uses a library to calculate the constant bearing between the source and destination coordinates represented by the corresponding step and later to calculate intermediate location coordinates based on the obtained bearing and user speed. Similar to the GPS receiver, the movement generator updates the simulation time and location changes every second.

Table 2: Simulation Parameters

<table>
<thead>
<tr>
<th>Simulation Parameter</th>
<th>WiFi</th>
<th>GPRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum bandwidth</td>
<td>55000 bps</td>
<td>25000 bps</td>
</tr>
<tr>
<td>Minimum bandwidth</td>
<td>10000 bps</td>
<td>10000 bps</td>
</tr>
<tr>
<td>Maximum delay</td>
<td>1000 ms</td>
<td>25000 ms</td>
</tr>
<tr>
<td>Minimum delay</td>
<td>35 ms</td>
<td>1000 ms</td>
</tr>
<tr>
<td>Maximum coverage radius</td>
<td>300 meters</td>
<td>Always available</td>
</tr>
<tr>
<td>Minimum coverage radius</td>
<td>10 meters</td>
<td>Always available</td>
</tr>
</tbody>
</table>

The simulation environment includes two types of wireless networks namely WiFi (WLAN) and GPRS (WWAN). The QoS simulator obtains the list of around 2000 businesses from the Google maps server covering an area with a radius of 70 km around Geneva (Switzerland) city centre. For simulating the user mobility, a user trip originates at Carouge and ends at Vernier (both are municipalities in the Canton of Geneva) via Geneva city centre covering a distance of 8.470 km and the corresponding route obtained from the Google maps server consists of 107 steps. The bandwidth and delay requirements of the remote patient monitoring device service are 25580 bps and 500 msec respectively [3]. The maximum radius of the WiFi base station is motivated by the availability of high range base stations as reported in [14]. As a reasonable assumption, GPRS networks are available wherever the user goes. The other parameters used for the simulation are provided in the Table 2.

IV. Validation Results

To investigate our hypothesis that a NAHO using QoS predictions is a better mechanism than a MCHO mechanism, the following performance metrics are evaluated:

- **Accumulated data goodput**: The amount of vital signs data transferred by the selected network over the simulation period. Goodput is widely defined as the application level throughput [15].
- **Accumulated data loss**: The amount of data lost over the simulation period. Note that the data loss occurs only when the service buffer is full and can not accommodate more data.
- **Average delay**: The average of delay experienced by the service by using the selected network over the simulation period. Herewith the delay represents Round Trip Time for the MSP Keep-alive message (this message ensures that the remote monitoring device service is still alive) originating from the mobile device to the surrogate host and receiving the corresponding acknowledgement.
- **Number of QoS-predictions requests**: The number of requests sent to the QoS simulator.

The graphs in the Figure 8, Figure 9 and Figure 10 show the accumulated goodput, accumulated data loss and average delay respectively during the simulation run where the user moves at the speed of 50 km/h. The graph shown in the Figure 10 shows higher average delay for the initial few numbers of seconds, because there is no WiFi network available at the location where the user trip starts; and hence the mobile device uses GPRS connection for the vital signs transfer. As can be observed from these graphs, the NAHO scheme provides higher throughput, lower data loss and lower delay as compared to the MCHO scheme. The graph in the Figure 11 shows the number of QoS prediction requests originating from the mobile device vs. speed. It shows that the number of requests increases as the speed increases. This is due to the special mechanism applied in the context processor. Namely, to avoid the situations where no predictions are available, the CP requests the predictions a few seconds (or simulations 3 seconds) before the user is likely to go out of the prediction range. In the area populated by higher concentration of WiFi base stations, we observed that this range is around 150 meters. The higher speed of the user frequently results in the condition that the user is likely to cross prediction range in 3 seconds. Consequently, more QoS prediction requests originate from the mobile device.
V. RELATED WORK

Vertical handovers has been studied extensively in the literature, we refer to [1] for a survey. Since in this paper, we focus on the context-aware NAHO strategy which uses QoS predictions, we summarize herewith the related work which uses context-awareness and some form of network QoS knowledge.

Balasubramanian and Indulska [16] propose a context-aware computing based vertical handover mechanism that matches QoS requirements of multimedia applications onto the QoS provided by the wireless networks. This work first determines if there is a need for the handover and if yes, then it evaluates which of the available networks will provide the required QoS. It describes a comprehensive context model for the handover information. Ahmed et. al. [17] also propose a context-aware vertical handover decision algorithm for the multi-homed devices. The work reported in [17] is similar to our work. However it assumes that the information about the offered QoS will be published by the corresponding wireless network; while we assume that it is provided by the QoS Predictions Context Source. Our use of AHP method for the optimization is motivated by the use of AHP in [16] and [17]. For the work reported herewith, we found the ability of AHP to vary the objective weights very useful for dealing with events considered by the context processor component. Moreover, AHP involves simple mathematical computations, which do not result in much processing overhead on the mobile device. Hong et al. [11] propose a vertical handover scheme for ubiquitous environments, where handovers to suitable wireless access networks are performed based on combined QoS requirements of all the applications running on the mobile device. It also considers the use of a dwelling timer for avoiding ping-pong effect. Chen et al. [12] propose to use a Location Service Server (LSS) in the fixed network which provides information such as coverage area, throughput and latency of available wireless networks around a mobile device. QoSCS is similar to LSS, except that QoSCS is designed to provide application level QoS predictions along a user trip path in addition to the above information. Han et. al. [18] propose a solution in which a mobile user of a streaming video application obtains information about the offered-QoS directly from the mobile operator network infrastructure. However, the proposed solution is mobile network operator specific.

Based on the above, the innovative contribution of our research to the existing state of the art as follows: 1) Design, architecture and validation of the context-aware computing support for the network-assisted seamless vertical handover which relies on the QoS predictions to make a handover decision; 2) Use of the geographic information available on the Internet to build QoSCS simulator and to realistically model user movements along the user trip path; 3) Application in the M-Health domain for real-time remote patient monitoring.
VI. CONCLUSION AND FUTURE WORK

The advances in the area of mobile computing will soon make it feasible to predict the availability of wireless networks and their application level Quality of Service (QoS) characteristics along the user’s mobility path. Such predictions are referred to as QoS predictions which are provided by the QoS Context Source (QoSCS) hosted in the fixed network. In this paper, we presented the design, architecture and evaluation results of the context-aware computing based infrastructure to use the QoS predictions to take Network-Assisted Handover (NAHO) decision. To evaluate the feasibility of such a mechanism, we built a QoSCS Simulator and User Trip Simulator that use the geographic information for a realistic modeling of the WLAN locations and user movements. The validation results for the remote patient monitoring service in the Mobile Health domain show that the NAHO scheme which uses QoS predictions outperforms the Mobile Controlled Handover (MCHO) mechanism used in this paper. In contrast to the proposed NAHO mechanism, the MCHO mechanism uses information available locally on the mobile device to take a vertical handover decision.

There are a number of research questions paving our future work, these include: 1) How to utilize QoS-predictions information to achieve power savings on the mobile device by selectively turning on the network interfaces only when required? 2) How to handle the situations where there is lack of QoS-predictions or the QoS-predictions are not accurate (i.e. service does not experience the QoS as predicted)? 3) What are the issues involved in deploying the developed vertical handover mechanism on the mobile device? 4) How to conduct validation of a system with such mechanisms in the real operational environment?

ACKNOWLEDGMENT

This work is supported by Dutch Freeband Awareness project (under grant BSIK5902390, http://awareness.freeband.nl/).

REFERENCES


