Context-Aware Middleware Architecture for Vertical Handover Support to Multi-homed Nomadic Mobile Services

Pravin Pawar
University of Twente
Enschede, The Netherlands
+31 53 489 3283
p.pawar@utwente.nl

Katarzyna Wac
University of Geneva
Geneva, Switzerland
+41 22 379 7663
katarzyna.wac@cui.unige.ch

Bert-Jan van Beijnum
University of Twente
Enschede, The Netherlands
+31 53 489 3283
beijnum@cs.utwente.nl

Pierre Maret
Institut National des Sciences
Appliquées de Lyon, Lyon, France
+33 472 437 096
pierre.maret@liris.cnrs.fr

Aart van Halteren
Philips Research
Eindhoven, The Netherlands
+31 40 274 9538
aart.van.halteren@philips.com

Hermie Hermens
University of Twente
Enschede, The Netherlands
+31 53 489 3283
h.hermens@rrd.nl

ABSTRACT

To accommodate the requirements such as high usability and personalization of 4G (mobile) networks, conventional handheld single network-interface mobile devices are evolving into multi-homed devices. Moreover, 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 (NMS) providers. For these devices, a vertical handover support is essential for the improved and reliable NMS delivery. Also, the fulfillment of the required QoS by the NMS is bounded by the *end-to-end QoS* (e2eQoS) provided by the underlying heterogeneous networks. To deal with these aspects, we propose a context-aware middleware architecture supporting vertical handover for the NMSs hosted on the handheld mobile devices. We emphasize the following features of the proposed middleware: 1) Context-aware computing based approach which uses an extensive set of context information collected from the mobile device and a fixed network; 2) Provisioning of and interaction with the end-to-end QoS (e2eQoS) predictions context source in the fixed network to obtain near-accurate estimation of the e2eQoS at a certain geographic location and to reduce unnecessary power usage in searching for available networks.

Categories and Subject Descriptors

C.2.6 [**Internetworking**];

General Terms

Algorithms, Performance, Design

Keywords

Nomadic mobile services, multi-homing, vertical handover, M-health services, context-aware computing, end-to-end QoS.

SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil.
Copyright 2008 ACM 978-1-59593-753-7/08/0003…$5.00.

1. INTRODUCTION

In the vision of 4G (mobile) networks, integrated and personalized services are envisaged at any desired time and any location. This vision is, partly, to be realized by making use of the mobile, often handheld or otherwise wearable devices, connected to the Internet using one or more of the network interfaces embedded in these devices [1]. Nowadays these mobile devices are evolving into *multi-homed* devices, which are able to connect to the Internet using multiple network interfaces e.g. WLAN, GPRS and UMTS. The forthcoming multi-homing enhancements include the device's ability to simultaneously use multiple networks of the same technology type (e.g. multiple WLANs) using a single network interface [2]. A *vertical handover* (handover between different network technologies) is an adaptation method for the multi-homed devices to dynamically redirect the data communication path of the mobile application to the networking interface different than the currently used one [3].

Until recently, the research in the mobile computing community has been focused on the use of mobile devices as *data consumers*. However, due to the advances in hardware (e.g. integrated camera, GPS) and software middleware technologies [4], and arising business needs for new services types, the mobile devices are turning also into *data producers*. In such a case, because of the inverted *producer-consumer* roles, the throughput achieved over the uplink by the application producing data is critical and the quality of this data is influenced by the characteristics of the available wireless networks as well as by the resources of a mobile device (e.g. CPU clock speed). Moreover, the data producer application has nomadic characteristics because it roams with the device on which it is hosted. In [5], van Halteren et al. extend the concept of Service Oriented Architecture (SOA) to the handheld mobile devices, to realize these applications as *Nomadic Mobile Services* (NMSs) and make these services available to the clients located anywhere on the Internet using the *Mobile Service Platform* (MSP) middleware. One of the applications of the nomadic mobile services is a *remote tele-monitoring service* [5] in an (mobile) m-Health domain. Using a *remote tele-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 an m-Health portal. At the portal, 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.

There exist a variety of architectures, algorithms and schemes for the vertical handover on multi-homed mobile devices. For example [3], [6] and [7] propose context-aware computing-based architectures, [8] and [9] propose to use policy-based approaches, while [10] introduces a generic vertical handover decision function. These approaches are mainly targeted towards providing vertical handover support for the *data consumer* mobile devices at the network layer level (e.g. using Mobile IP).

We consider herewith the problem of providing vertical handover support based on the principles of context-aware computing [11] for the NMS. The work reported in [12] illustrates design, validation and performance evaluation of the context-aware MSP which simply uses the information about the networks the mobile device is connected to, and always selects the network with the highest theoretical throughput to handover to (e.g. always prefers WLAN over GPRS). In [12], only one context source, particularly, the *Communication Context Source (CCS)* provides the knowledge on the current state of network resources, which is further processed by the *context processor* and used by the *context reasoner* for decision making. It is important to notice that the CCS does not distinguish between network's uplink and downlink direction. It provides information about the network's maximum downlink throughput, and not the uplink, which would be more appropriate for NMS.

This paper extends the work reported in [12] in the following three ways. Firstly, for a handover-decision, besides the theoretical throughput of the available networks, the MSP considers other relevant context information acquired directly from the context sources at the mobile device and provided by the context sources placed in the fixed network.

Secondly, we consider the NMS Quality of Service (QoS) requirements in terms of its required uplink throughput and delay. The fulfillment of the NMS's requirements is bounded by the *end-to-end QoS* (e2eQoS) provided by the underlying heterogeneous networks [13]. In the most of the cases, the first hop in that path is a wireless (mobile) network, which is a bottleneck in the end-to-end path. Hence the choice of a mobile network provider and network type (e.g. GPRS or UMTS), are critical for a NMS delivery. To deal with that, we propose the *QoS predictions context source*, which is based in the fixed network and provides the following information set to the NMS: a) information about the mobile networks available to a mobile device at a given geographical location and time and b) predicted e2eQoS provided by a particular mobile network. The existence of the QoS predictions context source empowers the mobile device in terms of the networks choice at a given location/time, but also along the device's mobility path, as we explain further in this paper. The provisioning of QoS predictions context source also helps to overcome the limits of the battery power of a mobile device. Usually, it is required to keep all the network interfaces on the mobile device *always switched on* in order to search for the mobile networks available in the vicinity. If, as we explained above, network availability is a priori known, then it suffices to switch on the required interface at a given location and time, and keep it off when not used.

Thirdly, once the *context processor* component obtains all the required context information alongwith the e2eQoS predictions, the *context reasoner* component applies a utility function to assign a score to each available network, and further decides whether the handover is necessary.

In summary, the distinguishing aspects of the proposed middleware architecture supporting the vertical handovers decision for multi-homed NMS providers are as follows: 1) The use of an extensive set of context information collected from the mobile device and from a fixed network; 2) Provisioning of and interaction with the QoS predictions context source in the fixed network to obtain a near-accurate estimation of the e2eQoS at a certain geographic location at a given time, reducing unnecessary power usage in searching for the available networks; and 3) A utility function-based vertical handover decision mechanism.

The reminder of this paper is organized as follows: Section 2 of the paper discusses the related work. Section 3 presents in details the proposed architecture, its elements and the interactions between them. Section 4 provides the information about the ongoing implementation. Section 5 concludes our finding and provides future work areas.

2. RELATED WORK

In the literature a variety of architectures, algorithms and schemes for vertical handover support for the multi-homed mobile devices have been reported. For example, [3], [6] and [7] use context-aware computing-based architectures with the analysis of the contributing factors, vertical handover decision models and the experimentation details. While the approaches proposed in [3] and [6] use vertical handover decision function based on *Analytical Hierarchy Process* (AHP), [7] proposes the use of *multi-network optimization protocol* after eliminating networks which do not satisfy certain constraints. An anticipated vertical handover scheme based on *Fast MIPv6* (FMIPv6) procedure for handover between the WLAN and UMTS networks has been proposed in ([14]). In this scheme, the mobile device initiates the registration request while still receiving packets on the old link and after receiving the *Care of Address* (CoA) it starts receiving the packets on the new link. Moreover, a generic vertical handover decision function which uses the weights assigned by the mobile device to different factors affecting handover decision has been proposed in [10]. [2] proposes a handover strategy for the video streaming based on the jitter experienced by the mobile device when it is close to the edge of an access network. [15] compares the performance of four vertical handover decision algorithms for the applications which fall into the following four classes: *conversational*, *streaming*, *interactive* and *background.* The comparison criteria include average bandwidth, delay, jitter and BER of the selected network by these algorithms.

There exist a number of vertical handover approaches based on the policies. [8] presents PROTON, a policy based architecture along with the context management components. The policy based solution proposed in [9] specifically emphasizes on choosing the correct time and selecting the correct network for the vertical handover. [16] studies in detail the effect of a vertical handover policy on the performance of Internet applications. The simulation results reported in [16] show that the throughput and RTT of the network have a low influence on the overall performance of short-lived TCP connections and user-interactive sessions (e.g. telnet), while delay of the selected network has an impact on the CBR traffic over UDP.

A few handover schemes specially emphasize on handling the resource limitations of the mobile device, user mobility and connectivity. For example, [17] proposes a prediction algorithm for vertical handover by using the speed of the mobile device and mobility patterns. The mobile device compares the predicted time to stay in the network with the handover delay and accordingly takes a handover decision. [18] presents a vertical handover policy based on the remaining battery status. If this value is higher than some specified threshold value, a greedy approach is chosen to obtain a higher throughput; otherwise a conservative approach is chosen to restrict the number of active network interfaces. [19] proposes to use a *Location Service Server* (LSS) in the fixed network which provides information about the coverage area, bandwidth and latency of wireless networks available to a mobile device. The information obtained from LSS is used to switch on the network interface only when the mobile device is in the network coverage area, thus resulting in the power savings. However, this research does not consider e2eQoS provisions.

[20] studies the required buffer size to achieve lossless upward (in terms of network coverage area, e.g. from WLAN to GPRS) vertical handover for the data traffic including multimedia as multimedia applications also generate data at the constant rate. The simulation concludes that with the reasonable buffer size (1-20 Kbps) the average packet delays are within acceptable QoS limits even for multimedia traffic. To overcome the problem of dealing with the multiple IP addresses during the vertical handover, [21] proposes to use an IP tunneling technique which uses two pairs of the virtual/fixed IP addresses, one pair for the mobile device and one pair for the handover server.

In order to make a vertical handover decision, researchers suggest using a variety of (context) information on the mobile device and in the network. The context information available on the mobile device could be generally classified as user preferences for the network and application, application QoS requirements, experienced network QoS, available network interfaces, reachable networks, *Received Signal Strength Indication* (RSSI), mobility information and device resource usage. On the other hand, the context information available in the (fixed) network includes a service provider's profile, network QoS, network coverage and the location of access points.

Since our approach specifically targets the data producing mobile devices hosting NMSs, we do not choose to use one particular approach as reviewed in the related work. However, the proposed architecture customizes the following concepts from the related work: 1) context requirements, 2) AHP based decision making function similar to [3] and [6], 3) a part of the e2eQoS prediction context source concept, as similar to LSS proposed in [19].

3. CONTEXT-AWARE VERTICAL HANDOVER ARCHITECTURE

In this section firstly we describe MSP in brief, secondly we elicit necessary context information contributing to the handover decision algorithm, and thirdly we present the proposed architecture and its components.

3.1 INTRODUCTION TO MOBILE SERVICE PLATFORM

A NMS realized using MSP consists of two components: 1) An application realizing a service running on the mobile device (referred to as a *device service*); and 2) a representation of the device service in the fixed network which is referred to as a *surrogate.* The surrogate functions as a proxy for the device service and participates in the service discovery network. A *Surrogate Host* is responsible for the management of surrogates. The main components of MSP include the following: 1) *MSP-Input/Output (MSP-IO)* resides on a mobile device and interacts with the device service. 2) *MSP-Interconnect* is located at the surrogate host and interacts with the surrogate. 3) *MSP-Messages* specifies the structure of messages exchanged between the device service and the surrogate. MSP uses HTTP as a data transfer protocol. The type of interactions between the device service and the surrogate are: 1) *One-Way messaging* for the un-acknowledged message delivery; 2) *Request-Response messaging* for the reliable message delivery; and 3) *Streaming* for the exchange of continuous data (streams). MSP uses dedicated control plane interactions for control, monitor and lifecycle management of the NMS.

On the instantiation of the first device service on a given mobile device, MSP-IO creates a *Device HTTP Connection* to handle all the interactions from the mobile device. After initialization of the *surrogate* in the surrogate host, MSP-IO creates a *surrogate connection* on top of the *Device HTTP Connection* and provides its handle to the corresponding device service. The *surrogate connection* is later used by the device service to exchange messages with its surrogate. The *message worker* and *stream worker* components in MSP-IO are responsible for the transmission of messages and streams respectively received from the device service. MSP design is based on *Jini* technology. The communication between the device service and surrogate is specified by the Interconnect protocol in *Jini Surrogate Architecture Specification* [22]. We have developed an *HTTP* implementation (referred to as *HTTPInterconnect*) of the Interconnect protocol. The device service is usually implemented using J2ME technology. For information on the architecture, design choices and implementation of MSP, we refer to [5].

3.2 ARCHITECTURE AND COMPONENTS DESCRIPTION

The proposed context-aware computing-based architecture consists of a number of *Context Sources* (CS) on the mobile device and in the fixed network, and *Context Processor* (CP) and *Context Reasoner* (CR) components on the mobile device. Traditionally, the handover decision could be taken entirely by the network, for the so-called Network Controlled Handoff (NCHO) or entirely by the mobile device, for the so-called Mobile Controlled Handoff (MCHO) [7]. There also exist some approaches which combine both of these strategies e.g. [23] and [24]. The later is more useful in case of Mobile IP where appropriate communication is required between the Home Agent and new Foreign Agent. We choose to take handover decision on the mobile device because most of the context sources are located on the mobile device and we do not use Mobile IP for handling mobility. Moreover, it supports autonomy of the mobile device such that it does not need to depend on the external entities for the handover decision.

3.2.1 CONTEXT SOURCES

Traditionally, the decision of a vertical handover to a new mobile network is based on the *Received Signal Strength Indication* (RSSI) of the (to be) connected mobile network. However, RSSI does not exhibit the network conditions adequately [19], and it does not reflect the important NMS-level objective: the *end-to-end QoS* (e2eQoS) experienced by a NMS when a given mobile network is chosen. Moreover, RSSI is a local information (on the wireless access link only), whereas we aim to select the mobile network that meets the e2eQoS requirements of the NMS. 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 its surrogate placed somewhere in the Internet. In order to consider this objective, we suggest using a variety of context information available on the mobile device and in the fixed network.

Table 1 shows the CSs on the mobile device, context information provided by them, context description and units, motivation behind its selection and the CS interfaces, over which the context could be obtained. Similarly, Table 2 presents the context information to be collected from the fixed network. In both tables, the interface methods starting with get provide context information only once upon the request, while those starting with subscribe continuously provide the context changes to the subscriber. For the sake of brevity, in this paper we do not provide the detailed design of the CS. However, since the architecture of the QoS prediction context source is not obvious from the information provided in Table 2, we choose to provide its brief overview in the further section. Our proposal of the chosen context information as presented in the Table 1, is based on the earlier experience with the remote tele-monitoring service using Context-Aware MSP [12], current experience with the design and development of the QoS context source and also the literature reviewed in the related work.

Table 1: Context sources on the mobile device

CONTEXT SRC.	*CONTEXT INFORMATION*	*MOTIVATION*	*INTERFACES*

Location And Time Context Source	Coordinates of the device's current geographic location (longitude, latitude) and time (Date, HH:MM:SS) as obtained from the GPS receiver.	It has been observed that the availability of and the e2eQoS provided by the mobile networks to NMS depends on the location and time.	getCurrentLocationAndTime();
			subscribeLocationAndTimeChanges();

Device Context Source	For a given mobile device, its model (String), CPU type (String), CPU clock speed (MHz), remaining battery level (%) could be obtained using the OS API calls.
	Standard values of power-consumption per network interface (milliAmps) obtained from the device specifications.
		For a given device, because of the processing power required for creating and transmitting NMS data, CPU type and clock speed influence the e2eQoS of the (currently used) network.
		Battery level could be combined with the user's power saving preferences. A decision about whether to actively search available networks or not, could also be taken based on the remaining battery power [19]. In [25] we observe that memory and CPU usage are well below limits for MSP.
			getDeviceType();
			getCPUInfo();
			subscribeBatteryLevel();
			getInterfacePowerRequirements()

User Preferences Context Source	A user's ranked list of all the mobile network providers, network names, and network technologies a user is subscribed to (List [String, String, String]).
	A list of all the device services (List [String]) ranked according to their importance to the user.
	User's power preference (Yes/No) indicating whether the middleware should consider/or not interface's power usage during selection.
	All the above information could be obtained from the user using the user interface.
		It may not be always the case that user's network preferences are based on only one factor such as usage cost. E.g. a businessman may rank the networks based on the security. Hence it is best to leave the ranking decision to the user.
		A ranked list of services could be used to provide preferential treatment to the services with high rank.
		It is very likely that the interface using more power provides higher e2eQoS. Also, some users always keep their mobile device charged while others do it when needed. Hence power preference context information is selected.
			subscribeNetworkPreferences();
			subscribeServicePreferences ();
			subscribePowerPreferences ()

Communication Context Source	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. We refer to [26] for details.	In case of i) unavailability of the predictions from the QoS Predictions CS or ii) an unpredicted loss of connectivity and lack of further QoS predictions, the information provided by a Communication CS could be used to make a (rough) handover decision to a new mobile network.	subscribeNetworkCrossLayerInfo()

Device Service Context Source	Required e2eQoS of every running device service.
	Device service's perception (%) representing its satisfaction for the provided e2eQoS by the currently selected network.
	Criticality alarm of the device service (Yes/No) representing its current importance. It depends on the situation in which the service is running.
		As observed in [25], each device service has different e2eQoS requirements.
		Device service's perception could be used to validate if the selected network satisfies device service requirements.
		Device Service criticality level can be assigned a higher value e.g. remote health tele-monitoring service in case of emergency health situations (e.g. high probability of seizure) and it could be changed to a lower value in case of non-emergency situation.
			subscribeDeviceServiceRequirements()
			subscribeDeviceServicePerception();
			subscribeDeviceServiceCriticality()

User Trip Information Context Source	User's trip information: in terms of source location (Location) and time (Date, HH:MM:SS), destination location (Location) and estimated time of arrival (Date, HH:MM:SS), and transportation mode (String) obtained using user i/f.	This information is useful for the QoS prediction CS to calculate the co-ordinates along the user travel path, estimated arrival time and use it further for the prediction of available mobile networks and associated predicted e2eQoS along the user mobility path with certain deviations.	subscribeTripLocations();
			subscribeTransportMode()

Table 2: Context sources in the fixed network

CONTEXT SOURCE	*CONTEXT INFORMATION DESCRIPTION*	*MOTIVATION*	*INTERFACES*

Surrogate Host Context Source	Location (longitude, latitude), time (Date, HH:MM:SS) at which the current mobile network is selected and mobile network provider, network name, and technology (String, String, String).
	A model (String), CPU type (String), CPU clock speed (MHz) of the mobile device on which the device service is running.
	Device service's e2eQoS requirements and observed e2eQoS as observed at its Surrogate.
		The predictions provided by the QoS context source are based on the historical data combined with learning mechanisms. The context information obtained from the surrogate host CS is an input to the learning of QoS prediction CS. However, for learning, the QoS predictions CS does not depend only on the context information provided by the surrogate object context source, but also use a variety of other context sources which are out of scope of the reported research. We refer to Section 3.2.1.1 for more details.	pushNetworkSelectionInformation();
			pushDeviceServiceRequirements();
			pushObservedE2EQoS()
			(In contrast to get and subscribe methods, the methods starting with push proactively sends context information to QoS predictions CS)

Qos Predictions Context Source	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 e2eQoS provisions (in an XML structure similar to Network Cross Layer Info.)	This information is useful to know the possible networks to handover to along the user's mobility path.	getNetworkPredictionInfo()

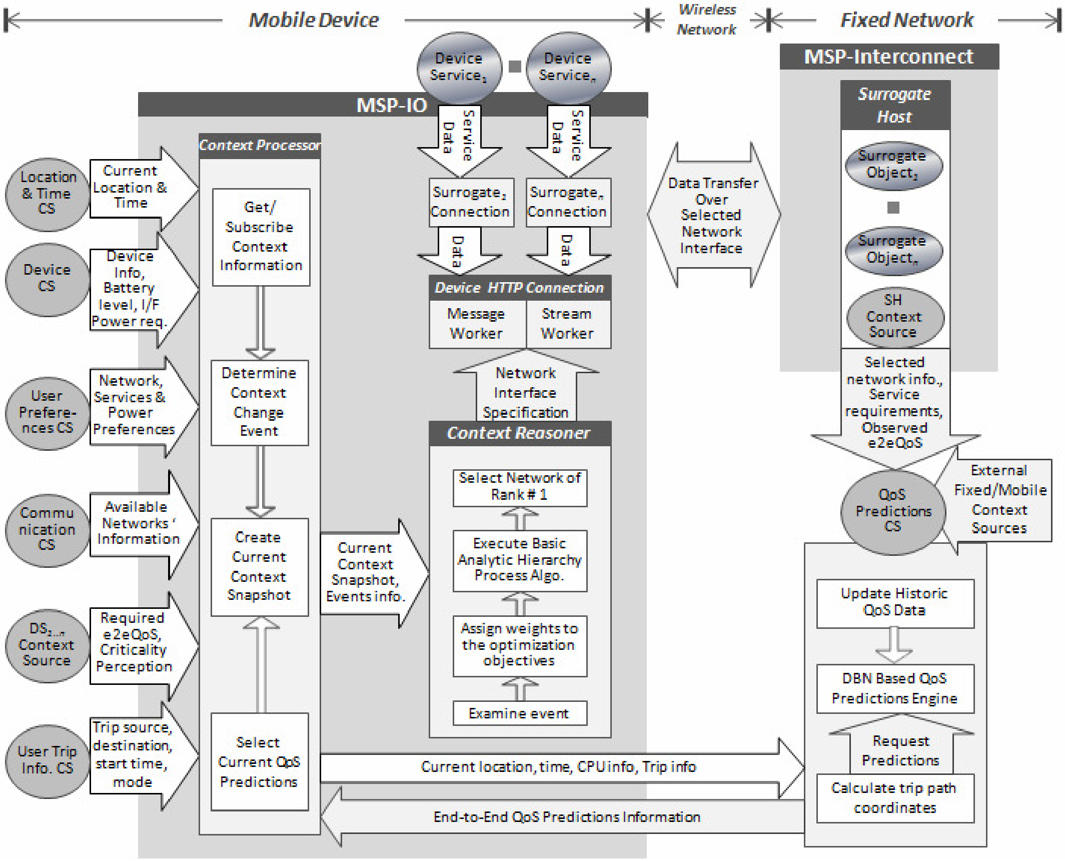

Mobile Device
Wireless Network
Fixed Network
Device Service$_1$
Device Service$_n$
MSP-IO
Context Processor
Service Data
Service Data
Location & Time CS
Current Location & Time
Get/ Subscribe Context Information
Surrogate$_1$ Connection
Surrogate$_n$ Connection
Data
Data
Device CS
Device Info, Battery level, I/F Power req.
Device HTTP Connection
Message Worker
Stream Worker
User Prefere-nces CS
Network, Services & Power Preferences
Determine Context Change Event
Network Interface Specification
Context Reasoner
Select Network of Rank # 1
Commu nication CS
Available Networks ' Information
Create Current Context Snapshot
Current Context Snapshot, Events info.
Execute Basic Analytic Hierarchy Process Algo.
DS$_{1...n}$ Context Source
Required e2eQoS, Criticality Perception
Assign weights to the optimization objectives
Examine event
User Trip Info. CS
Trip source, destination, start time, mode
Select Current QoS Predictions
Current location, time, CPU info, Trip info
End-to-End QoS Predictions Information
Data Transfer Over Selected Network Interface
MSP-Interconnect
Surrogate Host
Surrogate Object$_1$
Surrogate Object$_n$
SH Context Source
Selected network info., Service requirements, Observed e2eQoS
QoS Predictions CS
External Fixed/Mobile Context Sources
Update Historic QoS Data
DBN Based QoS Predictions Engine
Request Predictions
Calculate trip path coordinates

Figure 1: Elements and Interactions in the Context-Aware Middleware Architecture

3.2.1.1 QoS Predictions Context Source

The QoS predictions CS seeks to provide an efficient and accurate method that generates precise e2eQoS forecasts 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) [27] model. The choice of DBN has been motivated by the set of requirements for a machine learning task used in our prediction, such as: a) ability to predict, b) incremental learning, c) learning with the incomplete, uncertain, redundant and conflicting data, d) learning about causalities combining the domain knowledge and (historical) cases data set, e) prediction with missing values in queries, f) scalability and performance. The DBN based learning model employed in the QoS predictions CS uses valid historical data for learning. One of the real-time ways to obtain this data is to get it from the MSP via Surrogate Host CS. Towards this end, currently we work on calibration and (off-line) evaluation of the prediction method in an extensive set of trials with an m-health tele-monitoring service.

For providing the e2eQoS predictions, the QoS predictions CS needs the following set of data: a) current device location and time, b) device's CPU type, clock speed and battery level, c) user's mobility path information. The source and destination locations of a users's trip are mapped to the respective location coordinates and a route calculation technique (similar to the GPS navigation system) is to be used to calculate the coordinates along mobility path of the user. The transportation mode information is used to predict the expected time to reach the destination. Our preliminary results show that in the case of low mobility, the e2eQoS is predicted with 80% accuracy. However, this work is still ongoing and we are awaiting the conclusive results.

3.2.2 Context Processor

The role of the *Context Processor* (CP) component is to get/subscribe context information from the context sources and provide a necessary aggregated context information to the context reasoner to be able to make a network selection decision at a given time and location. Upon the activation of the first device service on a given mobile device, CP obtains the user's trip information from the user trip information CS, current location and time from the location and time CS, device context information from the device CS and provides this information to the QoS prediction CS to get a complete QoS prediction information along the user mobility path. We refer to a part of the QoS prediction information useful at a given location and time as *current QoS predictions*. On the activation of first device service, CP also subscribes to the user preferences CS, device service CS and communication CS. The information obtained from these context sources and the current network predictions together form the *current context information*. The current context information is updated in real-time to accommodate the context changes received from other context sources. CP sends the current context information tagged with the relevant *context change event* (refer to Section 3.3.3 for the event types) to CR for the further processing.

3.2.2.1 Handling Missing/Incomplete/Probabilistic Location and QoS Predictions Context Information

It is very likely that though the user provides the trip information, the actual path taken by the user (e.g. salesman) deviates from the mobility path predicted by the QoS Predictions CS. Moreover, it is also possible that the user is not on the move, but staying at the current location for a long time and just move around (e.g. home and office near to each other). To handle these problems, if the user trip information is available, then the QoS predictions CS also sends the e2eQoS predictions for the locations within a certain distance from the mobility path. If there is no user trip information available, then the e2eQoS predictions for all the locations within a certain radius of the current user location are sent. This behavior of the QoS predictions CS could be compared with the GPS navigation systems which cache the complete area map along the user mobility path. Herewith, the area map corresponds to the e2eQoS map.

The QoS prediction information is also probabilistic because it is derived from the historical data. Therefore, if the device services' perception of the currently selected network degrades, there is a mechanism in the CR for the selection of alternate network. Furthermore, in cases of: i) unavailability of the predictions or ii) an unpredicted loss of connectivity and lack of further QoS predictions, the information provided by a Communication CS is used to make a (rough) handover decision to a new mobile network.

3.2.3 Context Reasoner

The *Context Reasoner* (CR) is an event driven component and is responsible for the selection of one of the available networks to handover to, by considering user preferences, device service requirements, device services' perception and its criticality level. To be able to make use of the available context information, CR considers the following set *O* of objectives for optimizing the selection of the network:

Objective 1: Maximize user's network preferences.
Objective 2: Consider user preferences for power consumption.
Objective 3: Maximize device services' throughput requirements.
Objective 4: Minimize device services' delay requirements.

3.2.3.1 Basic Analytic Hierarchy Process Method

Since the problem of the selection of the desired network consists of satisfying a number of objectives and there are a limited number of networks to be chosen from, we follow similar approach to [3] and [6] of using the *Analytic Hierarchy Process* (AHP) method for the optimization. The ability of AHP to vary its weighting between each objective is useful for dealing with events as described further in the Section 3.2.3.2. AHP involves calculations using simple formulas and hence is expected to provide better computational performance than the alternative optimization techniques such as Genetic Algorithms. As described in [28], AHP is about dividing a problem into several sub-problems and later aggregating the solutions of these sub-problems into a conclusion. AHP method applied to our problem consists of the following three steps:

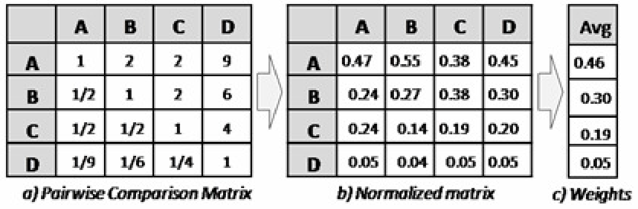
Step 1: *Decide the relative importance of the optimization objectives:* The importance of an objective is decided by the weight assigned to it. These objective weights are always assigned such that their combined sum is 1. For example, if the value of user's power preference is `Yes`, then every objective is assigned weight O_i = 0.25 and $1 \leq i \leq 4$; otherwise the second objective is assigned weight 0 and the other objectives' weights are 0.33. The weights of the objectives also vary as per the type of events described in the Section 3.2.3.2.

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

1. For each of the primary optimization objectives, assign an integer score between 1–9 for every available network depending on its position in the user preferences list, interface power requirements and predicted e2eQoS (throughput and delay available to the device service). The network with the best (worst) values in every category receives score 9 (1). E.g. four networks named A, B, C and D respectively may get score assigned 9, 6, 4 and 1 according to the user preferences list.
2. If *n* is the number of elements in a set of available networks *N*, for each objective, based on the network score S_i where $1 \leq i \leq n$, calculate the pair-wise comparison matrix with values P_{ij} = S_i/S_j for each *i* and *j* such that $1 \leq (i, j) \leq n$ and $i \leq j$. P_{ij} is rounded off to the nearest integer. For $i > j$, $P_{ij} = 1/P_{ji}$. E.g. Continuing example above, $P_{AB} \Rightarrow Int\ (S_A/S_B + 0.50) \Rightarrow Int\ (9/4 + 0.50) \Rightarrow Int\ (2.75) = 2$.

3. 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_{no}.

For the above example, the pair-wise comparison matrix, weight matrix and relative weights according to the user preferences are shown in Figure 2.



	A	B	C	D
A	1	2	2	9
B	1/2	1	2	6
C	1/2	1/2	1	4
D	1/9	1/6	1/4	1

a) Pairwise Comparison Matrix

	A	B	C	D
A	0.47	0.55	0.38	0.45
B	0.24	0.27	0.38	0.30
C	0.24	0.14	0.19	0.20
D	0.05	0.04	0.05	0.05

b) Normalized matrix

Avg
0.46
0.30
0.19
0.05

c) Weights

Figure 2: AHP Calculation Example

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.

After selecting the network with the highest score, CR instructs the *Message Worker* and *Stream Worker* components to use the IP address of this network for the data transfer (thus completing the handover procedure).

3.2.3.2 Dealing with Various Events

In principle, CR deals with different events received along with the current context information by changing the weights of the optimization objectives. Figure 3 shows the list of events and the corresponding option. The names of events 1–10 and assigned objective weights are self-explanatory. We explain *Event 11* herewith.

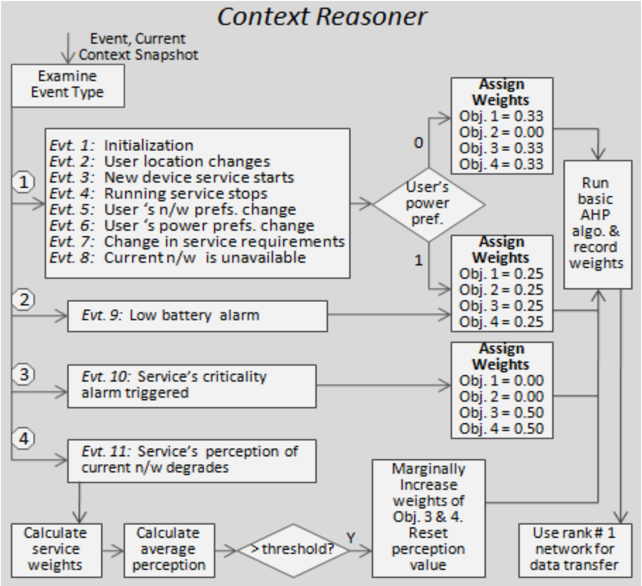

Context Reasoner
Event, Current Context Snapshot
Examine Event Type
Evt. 1: Initialization
Evt. 2: User location changes
Evt. 3: New device service starts
Evt. 4: Running service stops
Evt. 5: User 's n/w prefs. change
Evt. 6: User 's power prefs. change
Evt. 7: Change in service requirements
Evt. 8: Current n/w is unavailable
1
2
3
4
User's power pref.
0
1
Assign Weights
Obj. 1 = 0.33
Obj. 2 = 0.00
Obj. 3 = 0.33
Obj. 4 = 0.33
Assign Weights
Obj. 1 = 0.25
Obj. 2 = 0.25
Obj. 3 = 0.25
Obj. 4 = 0.25
Assign Weights
Obj. 1 = 0.00
Obj. 2 = 0.00
Obj. 3 = 0.50
Obj. 4 = 0.50
Run basic AHP algo. & record weights
Evt. 9: Low battery alarm
Evt. 10: Service's criticality alarm triggered
Evt. 11: Service's perception of current n/w degrades
Calculate service weights
Calculate average perception
> threshold?
Y
Marginally Increase weights of Obj. 3 & 4. Reset perception value
Use rank # 1 network for data transfer

Figure 3: Context Reasoner Working

Event 11: *Device service's perception of the network currently in use degrades.* To consider the user's device service priorities, on the receipt of this event, CR calculates the relative weight of each running device service according to the user's service preferences (Similar to *Step 2* of the basic AHP algorithm). The overall perception is averaged over the sum of multiplication of the relative weight of each device service and the corresponding perception level. If the overall perception is beyond a certain threshold, then the weights of objectives 3 and 4 are increased marginally (and objectives 1 and 2 are decreased accordingly), the device service perception level is reset and the basic AHP algorithm is executed. This strategy ensures that the device services which are important to the user are given a fair treatment.

4. TECHNOLOGY CHOICES FOR THE IMPLEMENTATION

As described in the Section 3.1, MSP implementation is based on Jini surrogate architecture specification. The Message Worker and Stream Worker are java threads and use *Apache HTTPClient library* to send messages and transmit data to the surrogate host. CR converts the IP address of the selected network interface to the *InetAddress* and changes the *hostConfiguration* which is later used by the *HTTPClient* to open an HTTP connection. The communication CS implementation is based on the *Network Abstraction Layer* (NAL) reference implementation for Windows CE with the extensions to generate the network resource descriptions in XML [26]. The current implementation of the QoS prediction engine using DBN is programmed in a Matlab environment; however we also work on deployment of the trained DBN model on the mobile device.

We are currently evaluating technical choices for the implementation of other elements. The CP, QoS prediction CS and surrogate host CS will use the *Context Distribution Framework* (CDF) for context exchange. CDF proposed in [29] takes a service oriented and *Quality of Context* driven approach for the distribution of context information within the mobile/fixed environment. The development of location and time CS could be based on GPS library developed in J2ME and available at [30] to interface with the GPS device. For developing the device CS, device service CS, user preferences CS, user trip information CS and surrogate host CS, MSP will provides necessary base classes developed in Java which could be extended to implement the desired functionality. For calculating the coordinates of the user mobility path, we are expecting to use a modified version of an open source street navigation solution named *Roadnav* [31] in the QoS predictions CS. The validation setup for the proposed architecture will an extension of the *System Under Test* (SUT) described in [25]. The SUT used in [25] consists of a Body Area Network (BAN), Qtek9090 PDA running Windows Mobile 2003 and a J2ME compliant JVM, MSP, communication CS and the GPRS, WLAN and USB network connections.

5. CONCLUSION AND FUTURE WORK

This paper presents a context-aware middleware architecture for providing the vertical handover support to nomadic mobile services which are hosted on a multi-homed mobile device and provide data to the clients located in the Internet. The proposed context information set for the mobile network selection includes service requirements, user preferences, device capabilities, interface power consumption, user mobility, service criticality and end-to-end QoS (e2eQoS) prediction information. Particularly, for providing the near-accurate estimate of the e2eQoS and minimizing power required at the mobile device to search all the available wireless networks, the QoS prediction context source located in the fixed network provides predictions on the availability of mobile networks and their estimated e2eQoS along the user's travel path. Our middleware also provisions the mechanisms to handle missing and incomplete and probabilistic location (or other context) information and still be able to provide e2eQoS predictions. The context reasoner is an event-driven component which deals with different context change events by changing the weights of optimization objectives and further using the Analytic Hierarchy Process method for handover-decision making. A few of the concepts in the proposed architecture are derived from the related work.

While some of the elements of the proposed architecture are already implemented, currently other few are under implementation and we are evaluating technical choices for the rest of them. We propose the validation of the proposed architecture for the remote patient tele-monitoring service in the (mobile) m-health domain aiming to achieve a better perceived performance for healthcare professionals, while optimizing battery usage.

6. ACKNOWLEDGMENTS

This work is supported by Dutch Freeband Awareness project (under grant BSIK5902390), Amigo project (IST-004182, partially funded by the European Commission) and Mobility International region of Rhone Alps (MIRA), France.

7. REFERENCES

1. Hui, S.Y. and K.H. Yeung, *Challenges in the Migration to 4G Mobile Systems.* IEEE Communications Magazine, December 2003. **41**(12).
2. Cunnigham, G., P. Perry, and L. Murphy, *Soft, Vertical Handover of Streamed Video*, in *Fifth IEE International Conference on 3G Mobile Communication Technologies, 2004*, October 2004: London, UK.
3. Balasubramaniam, S. and J. Indulska, *Vertical handover supporting pervasive computing in future wireless networks.* Elsevier Computer Communications, March 2004. **27**(8): p. 708-719.
4. Pawar, P., et al., *A Comparative Study of Nomadic Mobile Service Provisioning Approaches*, in *International Conference and Exhibition on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007)*. September, 2007: Cardiff, UK.
5. Halteren, A.v. and 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 2006. Montreal, Canada.
6. Ahmed, T., K. Kyamakya, and M. Ludwig, *Architecture of a Context-Aware Vertical Handover Decision Model and Its Performance Analysis for GPRS – WiFi Handove*, in *11th IEEE Symposium on Computers and Communications (ISCC'06)*. 2006: Sardinia, Italy.
7. McNair, J. and F. Zhu, *Vertical handoffs in fourth-generation multinetwork environments.* IEEE Wireless Communications, June 2004. **11**(3): p. 8-15.
8. Vidales, P., R. Chakravorty, and C. Policroniades, *PROTON: A Policy-based Solution for Future 4G devices*, in *Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'04)*. June 2004: New York, US.
9. Murray, K., R. Mathur, and D. Pesch, *Intelligent Access and Mobility Management in Heterogeneous Wireless Networks using Policy*, in *ACM 1st International Workshop on Information and Communication technologies*. 2003: Dublin, Ireland. p. 181-186.
10. Hasswa, A., N. Nasser, and H. Hassanein, *Generic Vertical Handoff Decision Function for Heterogeneous Wireless Networks*, in *2nd IEEE/IFIP International Conference on Wireless and Optical Communications Networks (WOCN '05)*. March 2005: Dubai, UAE.
11. Sinderen, M.J.v., et al., *Supporting context-aware mobile applications: an infrastructure approach.* IEEE Communications Magazine, 2006. **44**(9): p. 96-104.
12. Pawar, P., et al. *Context-Aware Middleware Support for the Nomadic Mobile Services on Multi-homed Handheld Mobile Devices*. in *12th IEEE Symposium on Computers and Communications (ISCC 2007)*. 2007. Aveiro, Portugal.
13. Wac, K., A.v. Halteren, and 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 2006. Montpellier, France.
14. Akkari, N., S. Tohmé, and M. Doughan, *Applying Anticipated vertical handover (AVHO) in Next Generation Networks*, in *Fourth European Conference on Universal Multiservice Networks (ECUMN'07)*. February 2007: Toulouse, France. p. 311-319.
15. Stevens-Navarro, E. and V.W.S. Wong, *Comparison between Vertical Handoff Decision Algorithms for Heterogeneous Wireless Networks*, in *IEEE 63rd Vehicular Technology Conference, 2006 (VTC 2006-Spring)*. May 2006: Melbourne, Australia. p. 947-951.
16. Calvagna, A. and G.D. Modica, *Effects of the Vertical Handover Policy on the Performance of Internet Applications*, in *3rd European Conference on Universal Multiservice Networks (ECUMN'2004)*. October 2004: Porto, Portugal. p. 358-366.
17. Joe, I. and S. Hong, *A Mobility-based Prediction Algorithm for Vertical Handover in Hybrid Wireless Networks*, in *2nd IEEE/IFIP International Workshop on Broadband Convergence Networks, 2007 (BcN '07)*. May 2007: Munich, Germany p. 1-5.
18. Hui-ling, J., Z. Zhao-yang, and L. Shi-ju, *A Power Threshold Based Policy for Vertical Handoff in Heterogeneous Networks*, in *International Conference on Wireless Communications, Networking and Mobile Computing, 2005 (WCNM 2005)*. September 2005: Wuhan, China. p. 1052-1055.
19. Chen, W.-T. and Y.-Y. Shu, *Active Application Oriented Vertical Handoff in Next-Generation Wireless Networks*, in *IEEE Wireless Communications and Networking Conference 2005 (WCNC 2005)*. 2005: New Orleans, LA, USA. p. 1383-1388.
20. Salamah, M., F. Tansu, and N. Khalil, *Buffering Requirements for Lossless Vertical Handoffs in Wireless Overlay Networks*, in *The 57th IEEE Semiannual Vehicular Technology Conference, 2003 (VTC 2003-Spring)*. April 2003: Jeju Island, Korea. p. 1984-1987.
21. Chen, L.-J., et al., *USHA: a simple and practical seamless vertical handoff solution*, in *IEEE Consumer Communications and Networking Conference (CCNC'06)*. 2006: Las Vegas, USA.
22. *Jini Technology Surrogate Architecture Specification*. October 2003 [cited; Available from: https://surrogate.dev.java.net/specs.html.
23. Wang, Q., et al., *Hybrid User- and Network-Initiated Flow Handoff Support for Multihomed Mobile Hosts*, in *65th IEEE Vehicular Technology Conference, 2007 (VTC2007-Spring)*. April, 2007: Dublin, Ireland. p. 748-752.
24. Jesus, V., et al., *Mobility with QoS Support for Multi-Interface Terminals: Combined User and Network Approach*, in *12th IEEE Symposium on Computers and Communications (ISCC 2007)*. July, 2007: Aveiro, Portugal. p. 325-332.
25. Pawar, P., et al. *Performance Analysis of Nomadic Mobile Services on Multi-homed Handheld Devices*. in *2007 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2007)*. July 2007. San Diego, CA.
26. Peddemors, A., I. Niemegeers, and 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 2007. Bangalore, India.
27. Murphy, K., *The Bayes Net Toolbox for Matlab.* Computing Science and Statistics, 2001. **33**.
28. Saaty, T.L., *How to make a decision: The analytic hierarchy process.* European Journal Operation Research, 1990. **48**: p. 9-26.
29. Pawar, P., A.v. Halteren, and K. Sheikh. *Enabling Context-Aware Computing for the Nomadic Mobile User: A Service Oriented and Quality Driven Approach*. in *IEEE Wireless Communications & Networking Conference (WCNC 2007)*. 2007. Hong Kong.
30. Deriaz, M. *GPS Library in J2ME*. 2006 27 April 2006 [cited 2007 21 August 2007]; Available from: http://www.universal-locator.com/.
31. Lynch, R. *Roadnav: Open Source Street Navigation Solution*. 2004 - 2007 30 June 2007 [cited 01 September 2007]; Available from: http://roadnav.sourceforge.net/.