A Context-Aware Vertical Handover Decision Algorithm for Multimode Mobile Terminals and Its Performance

Tansir Ahmed[†], Kyandoghere Kyamakya[*], and Markus Ludwig[†]
[†]*BenQ Mobile, BMG CTM PIC UEMT, Haidenauplatz 1, 81667 Munich, Germany*
[*]*ISYS, University of Klagenfurt, Universitätstrasse 65, 9020 Klagenfurt, Austria*
[†]*{Tansir.Ahmed, Markus.Ludwig}@BenQ.com*
[*]*Kyandoghere.Kyamakya@uni-klu.ac.at*

Abstract

Traditional handover decision algorithms for mobile devices mainly depend on signal strength. In the very near future, these will be obsolete for wireless networks and mobile terminals that are rapidly evolving towards being heterogeneous and multimodal, respectively, in response to the huge market potential. In the given circumstances, more sophisticated and intelligent handover decision algorithm is needed so that terminals can select the best option available from diverse networks and services as per user requirements. The algorithm has to provide user applications the flexibility to switch automatically between active interfaces that best suit them based on application requirements and interface capabilities, as well as to use multiple radio interfaces simultaneously ensuring the optimum usage of the network resources available to the terminal. In order to realize the above requirements, this paper illustrates a novel context-aware vertical handover decision algorithm suitable for multimode mobile devices in heterogeneous wireless networks based on the Analytic Hierarchy Process (AHP) and evaluates its performance through simulation.

1. Introduction

Present day wireless communications networks and devices are experiencing a paradigm shift. Rapid emergence of diverse access technologies, e.g. WLAN, Bluetooth, 3GPP cellular networks (GSM, GPRS, UMTS), DVB-H (Digital Video Broadcasting-Handheld), etc would result in evolution of wireless networks towards heterogeneous all-IP infrastructure. In this heterogeneous *overlaying* infrastructure, users should be given the freedom to roam globally among multitude points of attachment of different access networks (*vertical handover*) as per their service requirements. Interworking heterogeneous wireless access technologies should ensure that users would always have the best available services.

Conventional single interface mobile terminals are also evolving into multimode terminals. Currently, these multimode terminals do not possess true multimode functionality. They are limited to use only one radio interface at a time. But in the given heterogeneous scenario, these terminals should have true multimode functionality that would enable user applications to switch automatically between active interfaces that best suit them based on application requirements and interface capabilities, as well as to use multiple radio interfaces simultaneously. This would definitely optimize the usage of the network resources available to the device. Traditional *horizontal handover (HO)* decision mechanisms that mainly depend on signal strength for decision making are unable to realize the above requirements.

In the given circumstances, we have developed and analyzed an intelligent HO decision algorithm including the session transfer, which takes into account, as much as possible, intelligence residing on the terminal side as well as on the network side, collectively known as *context information*. While designing the decision algorithm we have ensured that it is simple enough to be suitable for practical multimode mobile devices (e.g. PDA) having several capability constraints like processor speed, memory size, power consumption, etc. On the contrary, it is versatile enough to be easily configurable by users. The algorithm is based on the *Analytic Hierarchy Process (AHP)* [1].

The rest of the paper is organized as follows. Section 2 highlights the related work. Section 3 and 4, respectively, illustrate the design and the implementation of the context-aware vertical HO decision algorithm. Section 5 analyzes the performance of the algorithm through simulation results. Finally, section 6 concludes the paper.

2. Related work

Hongyan *et al.* [2] and Chan *et al.* [3] propose a *fuzzy* based multiple-criteria decision making process to perform access network selection and vertical HO based

on the cost constraints and application priorities specified by users. Stemm *et al.* [4] and Pahlavan *et al.* [5] describe intelligent HO procedures especially for hybrid networks considering the type of the radio access technology and the signal strength. In [6], different HO policies for heterogeneous networks are used considering as HO decision parameters mainly the type of air interface and the available bandwidth at the *access router (AR)*. Bing *et al.* [7] presents an analytical vertical HO initiation model based on the criteria of *received signal strength (RSS)* and distance. Ylianttila *et al.* [8] proposes some optimization schemes in the decision process while performing vertical HO between IEEE 802.11 WLAN and GPRS/EDGE. In [9], the vertical HO initiation is decided by the HO delay time and throughput according to traffic classes (real-time and non-real-time services).

The above mentioned methods either consider only a few context parameters or are too complex to be suitable for practical multimode mobile devices that possess limited resources.

Balasubramaniam *et al.* [10] also uses the AHP method in their decision making process. However, it lacks an elaborate model that would consider a wide variety of the most important contexts and their grouping, precise calculation methods for mapping relevant contexts in the chosen model, user interactions in the process, and lastly, the application management, i.e. session transfer based on the HO decision.

3. Vertical handover decision algorithm

The AHP model [1] is a well-known and proven mathematical process to identify the most suitable choice among multiple alternatives based on some predefined objectives. The task of our context-aware decision algorithm is to select the most suitable interface for a given application among multiple options that would satisfy some primary objectives based on the values of some context parameters. In this regard, the AHP model perfectly fits into our decision making process.

The hierarchical decision making approach of the decision model, adapted from the AHP model, is illustrated in Figure 1. In this approach, available options (wireless networks with network contexts) are compared in terms of each predefined objectives (defined by users and terminal contexts) in order to determine their relative suitability. Similarly, the objectives are compared with each other in order to determine their relative importance. Finally, the best option is determined on the basis of these two sets of data. We have considered *mobile-initiated and controlled* vertical HO for the decision algorithm.

At first, we need to define a context model, which would identify all the context information relevant to the decision making process.

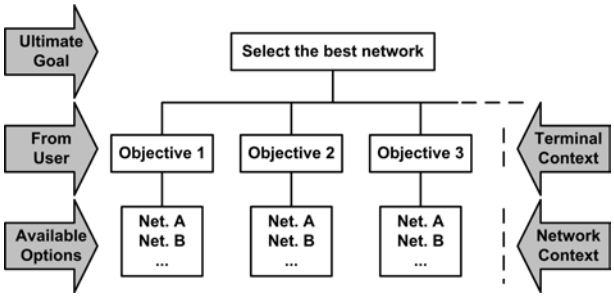

Figure 1. Decision hierarchy

3.1. Context model

The context model chosen for the decision algorithm is shown in Table 1. Context information may be classified based on their frequency of changes and based on their placement. In the former case, it is either *static* or *dynamic*, and in the latter case, it can be hosted either on the terminal side or on the network side. The contexts that do not change very often or at least not during runtime are static context information, whereas those that change quite frequently and may loose accuracy over time are dynamic context information.

Table 1. Context model for decision algorithm

Context Type	Terminal Side	Network Side
Static	Device capabilities, service types, QoS requirements of services, user preferences	Provider's profile
Dynamic	Running application type, reachable access points (APs)	Current QoS parameters of APs

On the terminal side, *device capabilities* include display size, resolution, battery life, memory, processor speed, and available interfaces. All services offered by a terminal are classified into three *service types*, namely conversational/real-time services, interactive services, and streaming services, where each of them has its own *QoS requirements*. *User preferences* are grouped as *interface preferences* for multimode terminal and *service preferences* (precedence of service types, expected QoS, and cost constraints). *Running application types* defines the service profile of currently running applications. *Reachable access points (APs)* identifies currently available networks and addresses of the APs.

On the network side, *Service provider's profiles* consist of provider's identity and charging models. *Current QoS parameters* define the current status of the available network QoS parameters.

3.2. Architecture of context-aware decision algorithm

The architecture of the decision algorithm is shown in Figure 2.

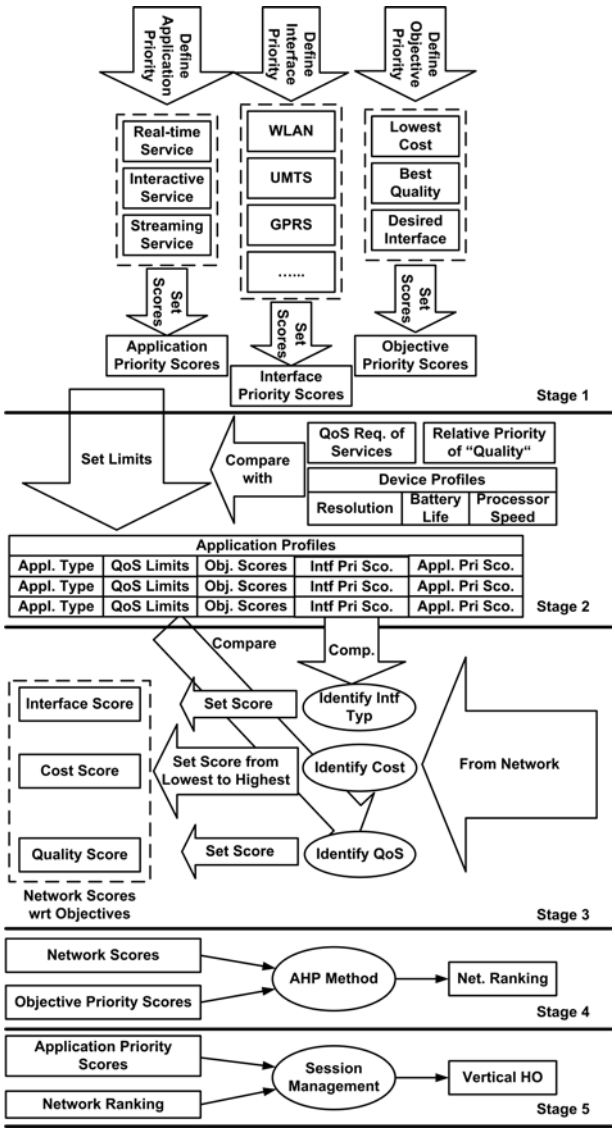

Figure 2. Architecture of the context-aware decision algorithm (for each type of application)

The working principle of the algorithm is simple. Here, a user defines his preferences in some categories that should meet both application requirements and device capabilities. Capabilities of available networks are discovered and compared with the defined preferences by employing the decision algorithm, and the most suitable network corresponding to the preferences is selected. Finally, applications that need to be transferred to the selected interface are switched. The decision algorithm is processed for each service type currently running in the device. Note that the AHP method only provides means of calculating the final decision when every parameter is already available. In our decision algorithm, we cannot use the method directly. We must, firstly, take inputs from user side, secondly, map the user inputs into some means of comparing network capabilities, and lastly, compare network capabilities using the means, before using the AHP method in the final stage of calculation.

In accordance with the AHP method, at first, we have to define some *primary objectives* for our decision algorithm that would cover almost all the preferences likely to be defined by users. For this reason, while defining the objectives, we have taken into account the preferences likely to be the most interesting to users and *3GPP* defined *Quality of Service (QoS)* parameters [11]. Obviously, the cost constraint is likely to be the most important user preference. Setting priorities among available interfaces in the multimode terminal should also be an important parameter from the user side as, for example, most users may like to give the cellular interface the highest priority especially for voice applications due to its almost ubiquitous coverage. QoS is another important parameter in order to ensure user satisfaction to the fullest extent. Consequently, we have chosen the following *six* primary objectives:

1) Consider interface priority
2) Minimize cost
3) Maximize mean throughput
4) Minimize delay
5) Minimize jitter
6) Minimize Bit/ Frame Error Rate (BER)/ (FER)

3.2.1. Pre-configuration.

Stage 1: Taking user inputs. For any of the three service types, a user needs to define *three* sets of relative priorities. These three sets are (i) relative priorities among primary objectives (*objective priorities*) (ii) relative priorities among available interfaces in a device (*interface priorities*) and (iii) relative priorities among three types of services (*application priorities*). User preferences are taken as *discrete* values or *scores*. However, in order to make the model more user-friendly available options, in each case, are labeled with suitable *literals*. The user only needs to arrange the literals in a descending order starting with the one with the highest priority. Based on the arrangement of the literals priority scores between 1 and 9 are assigned automatically at the backend, where 1 denotes the most preferred one and 9 denotes the least preferred one. Priority scores are equal-spaced integers whose space-gap is defined by (1), where N_p denotes the number of parameters, L_u and L_l denote the highest and lowest possible scores i.e. 9 and 1, respectively, and G

denotes the numeric space-gap between two subsequent scores, which is rounded off to the nearest integer.

$$G = \frac{L_u - L_l}{N_p} \quad (1)$$

As an example, among the primary objectives mentioned earlier objective 1 is labeled as "Desired Interface", objective 2 as "Lowest Cost", and objectives 3 to 6, in a group, as "Best Quality". Here, (1) results in $G = 3$ while using $L_u = 9$, $L_l = 1$, and $N_p = 3$. If a user arranges the literals as in the order "Lowest Cost", "Best Quality", and "Desired Interface" objective 2, objectives 3-6, and objective 1 have scores of 1, 4, and 7, respectively. Since "Best Quality" is the group of four parameters objectives 3-6 have the same score, 4. Similar measures are taken in case of interface and application priorities. For the former, N_p equals to the number of interfaces in the terminal and for the latter, types of services. It is worth mentioning that each of the sets of literals includes a "Default" option. The whole process is illustrated in Figure. 3.

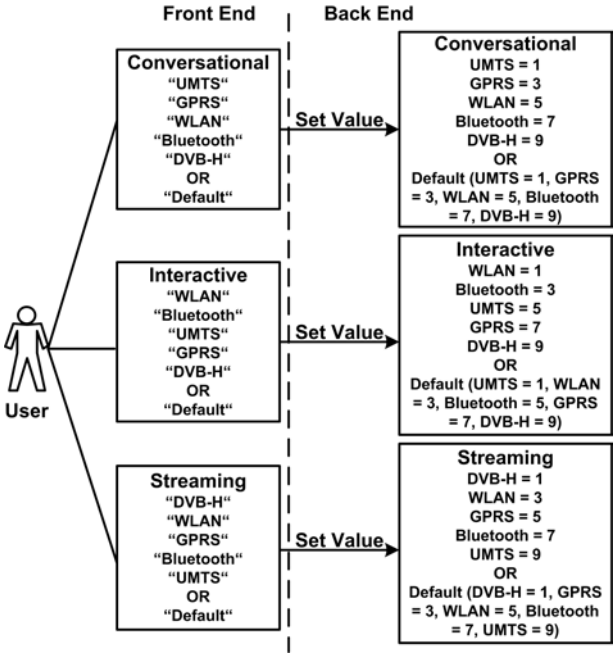

Figure 3. Taking user inputs on interface priority

Stage 2: Mapping limit values from discrete preferences. Some context information especially network QoS parameters are very dynamic. Therefore, it makes sense to express QoS preferences from users as *continuous* values or *limits* in order to provide better flexibility while comparing them with the network QoS parameters.

At this stage, suitable limit values (upper and lower) for the four QoS parameters related directly to objectives 3-6 are *mapped* at the backend for each of the three service types. While fixing the limit values, it is important to note that high values are not always better for all the four QoS parameters. It is always preferable to have values as high as possible for mean throughput, whereas as low as possible for delay, jitter, and BER/FER. In case of mean throughput, the lower limit is always a fixed value, i.e. *minimum requirement* (e.g. ≥ 4 kbps for conversational service [11]). This value is based on the contexts like QoS requirements of specific service type and device capabilities. The upper limit varies in accordance with the objective priority scores of the QoS based objectives (objectives 3-6) set earlier. For example, if the objectives 3-6 have the highest priority (priority score equals 1) the upper limit is set at the highest possible value (e.g. > 25 kbps for conversational service [11]), on the contrary, if they have the lowest priority (priority score equals 7) it is set much nearer to the lower limit (e.g. 8 kbps for conversational service). The limit values for the other three QoS parameters are fixed likewise, except that the upper limit is always a fixed value in this case, i.e. *maximum tolerance* (e.g. ≤ 400 ms one-way delay for conversational service [11]) and the lower limit varies according to the objective priority scores. The process of mapping limit values from discrete preferences is illustrated in Figure 4.

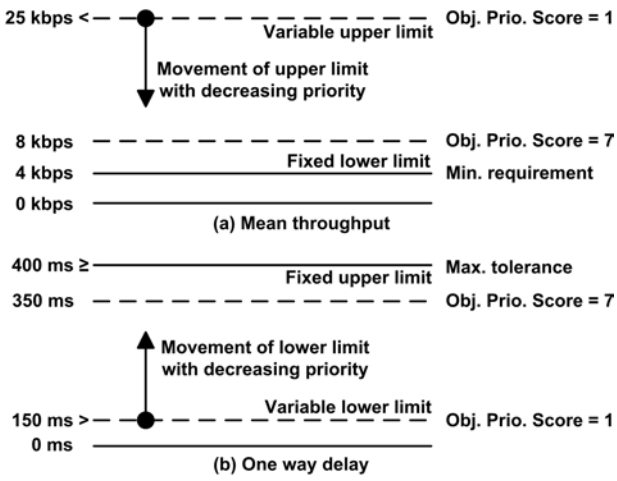

Figure 4. Mapping limit values for voice conversation

Note that in both the cases the difference between upper and lower limits (*quality sensitive window*) increases with the increasing priority (i.e. decreasing priority score) and vice versa. The quality sensitive window will be able to differentiate precisely any two equivalent network QoS parameters (e.g. mean throughput), even when their

values are nearly equal, only when their values fall inside the window. Anything outside the window will be coarsely identified either as the best (values that fall outside the variable limit) or the worst (values that fall outside the fixed limit). The wider the window is the more QoS sensitive it is. The idea is further illustrated in stage 3 where the limit values are applied to compare and score networks based on QoS parameters.

At the end, we have three sets of *preconfigured* data (scores and limits) for the three service types. They are grouped together and stored as the *application profiles* (see Figure 2) where individual service type is identified by the *application type*. All the running applications inside a mobile device would have predefined *application type*. Thus, any application, during runtime, could be paired with individual set of *application profiles* based on its *application type*.

3.2.2. Real-time calculations. The following stages perform real-time calculations for a particular type of running application.

Stage 3: Assigning scores to available networks. At this stage, capabilities of the reachable networks (including the current network, if any) are compared with the preconfigured user preferences (scores and limits based on the six objectives) and suitable scores are assigned to each of the networks. A multimode mobile device would always monitor (layer-2 or layer-3 monitoring, or both) each of its interfaces for reachable networks. It is assumed that shared contexts of networks like current QoS parameters and cost would always be advertised by the available networks, or the terminal may utilize layer-2 or layer-3 probing.

Assignment of scores to the available networks based on discrete preferences like interface priority and cost constraint is straightforward. The same interface priority score, already defined by the user in stage 1, is assigned to the available network depending on its type. In case of cost objective, all the available networks are compared with each other and assigned with appropriate equal-spaced scores between 1 and 9 based on (1) in a descending order, where the cheapest network has a score of 1. If a particular network does not advertise the cost information it is assigned with a score of 9 (costliest network) as a default value.

In case of continuous preferences (e.g. QoS preferences), QoS parameters of all available networks are compared with the individual parameter limit values defined in stage 2. If u_i and l_i denote the upper and lower limits of a particular continuous preference and n_i denotes the value offered by a network for that particular parameter the network score, S_i, based on that preference is calculated using (2) and (3). Eq. (2) is used for continuous preferences like mean throughput, where the target value is preferred to be as high as possible. On the contrary, (3) is used for continuous preferences like delay, jitter, and BER/FER, where the target value is preferred to be as low as possible. If there is any missing parameter i.e. not advertised by a particular network its default value is used. Values of l_i and u_i are the default values for (2) and (3), respectively.

$$\begin{aligned} S_i &= \left(1 - \frac{n_i - l_i}{u_i - l_i}\right) \times 10 && ; l_i < n_i < u_i \\ &= 1 && ; n_i \geq u_i \\ &= 9 && ; n_i \leq l_i \end{aligned} \tag{2}$$

$$\begin{aligned} S_i &= \left(\frac{n_i - l_i}{u_i - l_i}\right) \times 10 && ; l_i < n_i < u_i \\ &= 1 && ; n_i \leq l_i \\ &= 9 && ; n_i \geq u_i \end{aligned} \tag{3}$$

Stage 4: Calculating network ranking based on AHP method. At this stage, *ranking* of the available networks is performed based on the objective priority scores and network scores assigned at stage 1 and 3, respectively. The calculations use the AHP method, which is a three step process [1].

Step 1. At first, the relative scores among the objective priority scores set by the user at stage 1 are calculated. Relative scores are scaled linearly between 1-9 [1]. Relative scores between any two particular scores are calculated using (4), (5), and (6), where RS_{ab} is the relative score between parameters a and b, and S_a and S_b are their respective scores.

$$\frac{1}{RS_{ab}} = \left(1 - \frac{S_b}{S_a}\right) \times 10 \quad ; S_a > S_b \tag{4}$$

$$RS_{ab} = \left(1 - \frac{S_a}{S_b}\right) \times 10 \quad ; S_a < S_b \tag{5}$$

$$RS_{ab} = 1 \quad ; S_a = S_b \tag{6}$$

With the calculated relative scores the priorities (i.e. weights) for the six objectives in terms of the overall goal i.e. selecting a suitable network are calculated using *pairwise comparison matrix* [1] for objectives. It consists of the relative scores calculated in the previous step. The dimension of the pairwise comparison matrix A for the objectives, as shown in (7), is flexible and depends on the number of chosen objectives (6 × 6, in our case).

Matrix A is then normalized by dividing each element by individual sum of column. The normalized matrix A_{norm} is

shown in (8). At the end, the average values of each row for objective i are calculated to give the priorities for each objective (p_1, p_2, p_3, p_4, p_5, p_6) with respect to the overall goal using (9).

$$A = \begin{bmatrix} 1 & RS_{12} & RS_{13} & RS_{14} & RS_{15} & RS_{16} \\ \frac{1}{RS_{12}} & 1 & RS_{23} & RS_{24} & RS_{25} & RS_{26} \\ \frac{1}{RS_{13}} & \frac{1}{RS_{23}} & 1 & RS_{34} & RS_{35} & RS_{36} \\ \frac{1}{RS_{14}} & \frac{1}{RS_{24}} & \frac{1}{RS_{34}} & 1 & RS_{45} & RS_{46} \\ \frac{1}{RS_{15}} & \frac{1}{RS_{25}} & \frac{1}{RS_{35}} & \frac{1}{RS_{45}} & 1 & RS_{56} \\ \frac{1}{RS_{16}} & \frac{1}{RS_{26}} & \frac{1}{RS_{36}} & \frac{1}{RS_{46}} & \frac{1}{RS_{56}} & 1 \end{bmatrix} \quad (7)$$

$$A_{norm} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} & b_{46} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} & b_{56} \\ b_{61} & b_{62} & b_{63} & b_{64} & b_{65} & b_{66} \end{bmatrix} \quad (8)$$

$$p_i = \frac{b_{i1} + b_{i2} + b_{i3} + b_{i4} + b_{i5} + b_{i6}}{6} \quad (9)$$

Step 2. The relative scores among the scores of the available networks assigned at stage 3 in terms of individual objective are calculated using (4), (5), and (6). Then the network conformances (i.e. weights), c_{ij}, for i number of available networks in terms of each of j number objectives are calculated in similar fashion as described in step 1. For example, two available networks, WLAN and GPRS, score 1 and 8, respectively, in terms of cost. Their pairwise comparison matrix with respect to cost is shown below, where (4) is used for calculating the relative score RS_{12} between the two networks in terms of cost.

$$\begin{array}{c} \\ Net1 \\ Net2 \end{array} \begin{array}{c} \begin{array}{cc} Net1 & Net2 \end{array} \\ \begin{bmatrix} 1 & RS_{12} \\ \frac{1}{RS_{12}} & 1 \end{bmatrix} \end{array} \quad (10)$$

$$\begin{array}{c} \\ WLAN \\ GPRS \end{array} \begin{array}{c} \begin{array}{cc} WLAN & GPRS \end{array} \\ \begin{bmatrix} 1 & 8.75 \\ \frac{1}{8.75} & 1 \end{bmatrix} \end{array} \quad (11)$$

Normalizing (11) we get,

$$\begin{array}{c} \\ WLAN \\ GPRS \end{array} \begin{array}{c} \begin{array}{cc} WLAN & GPRS \end{array} \\ \begin{bmatrix} 0.9 & 0.9 \\ 0.1 & 0.1 \end{bmatrix} \end{array} \quad (12)$$

Now, the network conformances $c_{1,\ cost}$ and $c_{2,\ cost}$ for WLAN and GPRS, respectively, are calculated from (12) using (9). Thus, $c_{1,\ cost} = (0.9 + 0.9)/2 = 0.9$ and $c_{2,\ cost} = (0.1 + 0.1)/2 = 0.1$.

Step 3. The overall ranking of each available network is determined by calculating the sum of products of network conformances in terms of individual objective (obtained from step 2) and objective priorities for that particular objective (obtained from step 1). For i number of available networks and j number of objectives, the overall ranking R_i can be obtained from the following equation:

$$R_i = \sum_1^{ij} c_{ij}(p_j) \quad (13)$$

R_i is always in the range of 0-1. The network with the highest rank is finally selected.

Stage 5: Session management. At this final stage, an efficient session transfer scheduling algorithm is employed in order to switch applications to the selected network. The scheduling algorithm takes into account the application priority score set by the user at stage 1 and the rank of the selected network obtained from (13) at stage 4. For i number of running applications the overall score, O_i, is calculated using the following equation:

$$O_i = a_i'(R_i - R_d) \quad (14)$$

where, R_d and R_i, respectively, are the ranks of the current and the selected network for the i^{th} application, and a'_i is the normalized value of its application priority score, a_i, given by the following equation:

$$a_i' = \left(1 - \frac{a_i}{10}\right) \quad (15)$$

The value of O_i is always between -0.9 to +0.9. For a given application, $O_i = 0$ or $O_i < 0$ means that the application is already using the optimum interface and it needs not to be switched to an alternative one. For all $O_i > 0$, applications are switched in accordance with their O_is in a descending order starting with the one with the highest O_i.

4. Implementation

Over the past few years we have developed a *reference architecture* for use within multimode mobile terminals. This architecture supports a number of functions beyond those needed in a conventional *monomode* terminal. Basic modules of the reference architecture are shown in Figure 5. The vertical handover decision algorithm is implemented in the *decision module* of the architecture briefly described below. The target device for the implementation has been selected as *MDA III* from *T-Mobile* (Intel PXA-263 400 MHz processor, 128 MB RAM, 96 MB ROM, GSM quad-band, multimode PDA) on *Windows Mobile 2003 SE* platform, which should be updated to *Windows CE 5.0* at some later stage.

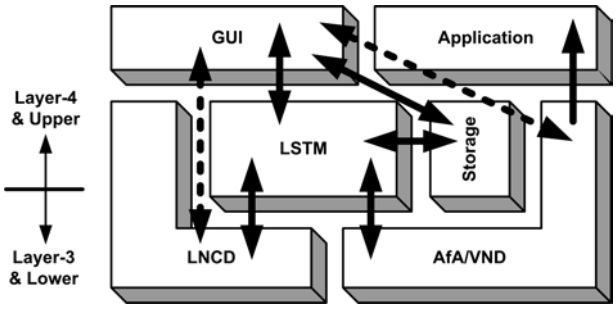

Figure 5. Stack diagram of the reference architecture

The *Light Network Capability Discovery (LNCD)* module periodically monitors all interfaces and discovers the capabilities of any active interface. Network profile is then forwarded to the *Light Session Transfer Management (LSTM)* module. The LSTM or the decision module is the heart of the whole architecture. It acts as a middleware between upper and lower layers, and receives network profiles and application profiles from the LNCD and the *Adaptation for Application (AfA)/Virtual Network Driver (VND)* modules, respectively. All information is stored in the *Storage*. The handover decision algorithm is processed in this module. After processing the decision algorithm LSTM notifies the AfA/VND module about the applications that should be moved to alternative interfaces. Components of the LSTM module are shown in Figure 6.

The AfA/VND module comprises of two separate software parts – the AfA part works on Linux platform (as a previous version of the reference architecture) and the VND part works on Windows Mobile 2003 SE platform. This module, thus, provides the reference architecture necessary flexibility to work on multiple platforms. This module takes care of the session mobility. In this architecture, no common mobility platform (e.g. Mobile IP) is present; instead, technology dependent individual and standard mobility features are used. The AfA/VND module receives notifications from the LSTM module and accordingly shifts applications to the alternative interface. In the process, it either generates a new IP address if the alternative interface does not have a configured one or retrieves the IP address already configured for the interface and then associates the address to applications. It also feeds the LSTM module with application profiles during initialization phase. The GUI provides means to follow the complete processing of the architecture. It also enables users to configure the decision algorithm, as shown in Figure 7.

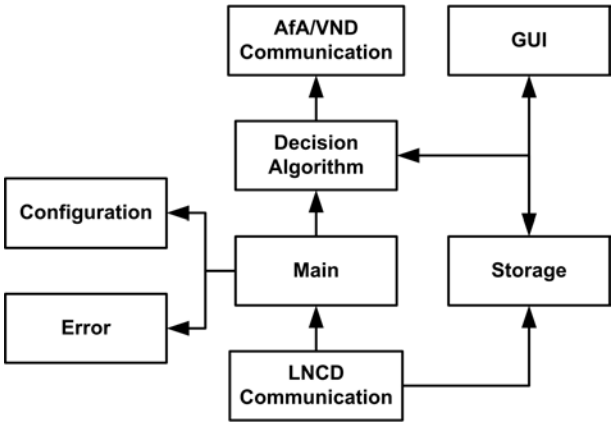

Figure 6. Components of the LSTM module

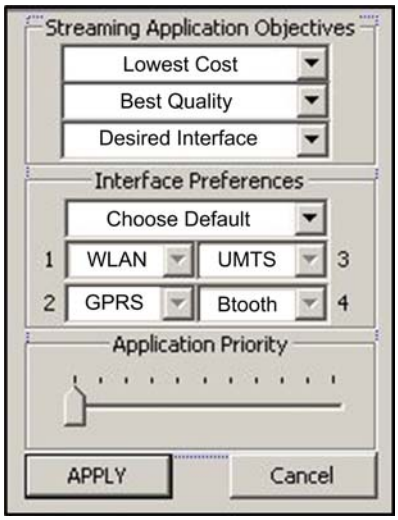

Figure 7. GUI window for configuring streaming application type

5. Simulation

The test setup for simulation is shown in Figure 8. A PC working on Linux environment acts as AR. Two WLAN APs with static IP addresses are employed. The APs are incorporated with layer-3 functionalities. A *dummy* web

browsing application (interactive service) called *AfAWeb* is developed for demonstration and testing. This simple web browsing application like *Internet Explorer* has the basic functionalities for browsing Internet and is used to verify session reestablishment in the simulation environment. Windows Mobile 2003 SE has a major limitation that it cannot deal with simultaneously active multiple interfaces of different kinds (e.g. WLAN and GPRS). It is widely expected that this drawback would be totally removed in future versions of Windows mobile platforms (Windows CE 5.0 or later). However, at the moment it does support simultaneously active multiple interface of the same kind (e.g. WLAN). Therefore, the MDA III is equipped with one internal and one external WLAN interfaces and WLAN (IEEE 802.11b) is used as the lone access network. However, additional techniques (e.g. for network capability discovery) are employed in order to simulate vertical handover.

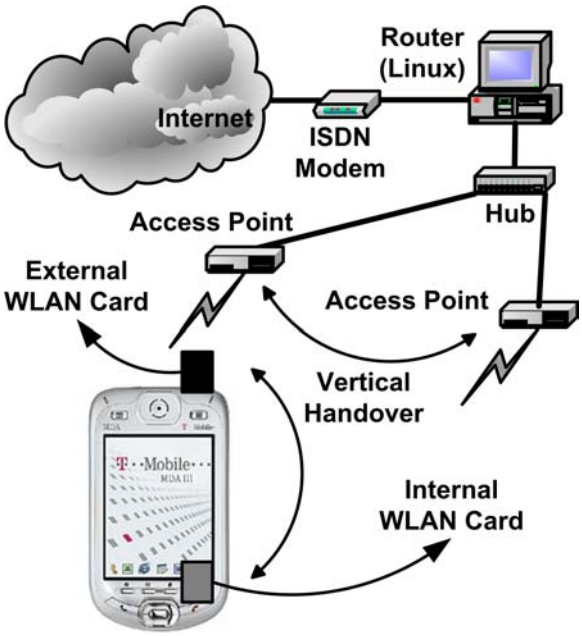

Figure 8. Test setup for simulation

Until today, no standard network capability discovery mechanism is available for WLAN. WLAN network advertises no additional information other than its identifiers (i) *Service Set Identifier (SSID)*, commonly known as the *network name* and (ii) *Basic Service Set Identifier (BSSID)*, a 48-bit identifier having the same format as IEEE 802 MAC address. In the simulation platform, two reconfigurable configuration files are used (inside the LNCD module in the reference architecture) in order to simulate the network capability discovery mechanism. Each file, identified by the BSSID of corresponding WLAN AP by the LNCD, represents individual network capabilities. In this way, it can be assumed that the simulation platform, which actually has Internet connection from only one provider, has two different wireless access networks. This assumption is fairly reasonable for our requirements as the simulation is performed in order to (i) verify whether the algorithm is processed in desired manner in necessary situations (ii) verify whether it gives accurate decision when processed and (iii) evaluate the performances of the algorithm and the software as a whole. Handover mechanisms e.g. processing of the decision module is performed automatically when more than one access networks are detected by the LNCD. For the ease of performing simulation we chose *three*, instead of six, primary objectives, namely (i) minimize cost (ii) consider interface priority and (iii) maximize quality.

We simulated two different test scenarios. In both the scenarios, AfAWeb was running in the device as the lone application. In *scenario 1*, the MDA III is connected to the Internet through a single AP and single WLAN interface. At some point, switching between APs is performed by shutting down the working AP and turning on the other one (horizontal handover). In *scenario 2*, the MDA III is connected to the Internet through a single AP and single WLAN interface. At some point, the other AP is also switched on and detected by the other WLAN interface. Using the configuration files a simulated vertical handover is performed, i.e. switching from the current interface to the alternative one having the better network capability.

In both the scenarios the algorithm worked precisely according to its working principle. It was not processed when only one wireless access network was available (scenario 1). In this case, the available network was selected as the default one. The algorithm was processed only while performing vertical handover (scenario 2) between two available networks and the handover decision precisely matched with that calculated theoretically. Also, the handover and session transfer mechanisms worked perfectly in this case.

Simulation results for scenario 2 are summarized in Table 2. The focus of the simulation was to evaluate, mainly, the time delays as a performance parameter experienced at different execution phases during vertical handover. Definitions of different time delays, as shown in Table 2, are illustrated in Figure 9. Average values are taken after performing several test runs. Average time taken for new interface detection (30-40 ms), although fairly acceptable in our case, is influenced by the technology specific detection mechanisms (WLAN in this case). The major strength of the decision algorithm is highlighted through the small average delay (5-10 ms) experienced during decision making. Since, the decision algorithm uses only basic mathematical calculations (addition, subtraction, multiplication, and division) the processing time is greatly reduced. As an additional but very important advantage,

the use of simple calculations in the algorithm guarantees that it is suitable for embedded hardware in practical mobile devices. Complex calculations like fuzzy logic or floating point exponential functions as proposed in some literatures ([2], [3], [6] and [7]) mentioned in section 2 are either not supported by embedded hardware of practical mobile devices or unfeasible to be implemented. During the session transfer with reestablishment phase the AfA/VND module retrieves already configured IP addresses from active interfaces and results in small average delay of 15 ms, which may go a bit higher if a new IP address needs to be generated. Note that delay for session reestablishment depends on the behavior of individual type of application and varies among different types of applications. The total average delay of only 50-65 ms implies that even with additional delays (e.g. the overall one way delay in the mobile network, from UE to PLMN border, is approximately 100 ms [11]) in real networks the algorithm would work perfectly for most delay sensitive applications like voice conversation or real-time video (one way delay: preferred < 150 ms, maximum tolerance < 400 ms [11]).

Table 2. Simulation results while both interfaces active (scenario 2)

Execution phases	Average delay (ms)
New interface detection	30-40
Decision making	5-10
Session transfer, including session reestablishment	15
Session reestablishment alone	10
Handover without session transfer	35-50
Handover, including session transfer	50-65

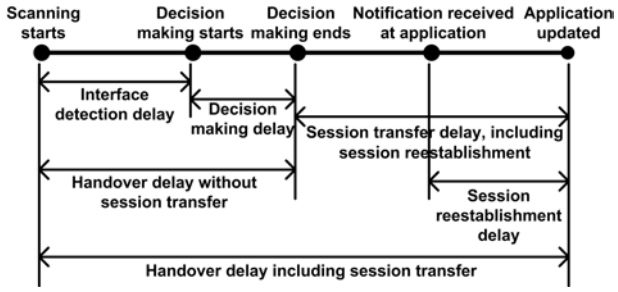

Figure 9. Time delays measured during simulation

6. Conclusions

In this paper, a context-aware decision algorithm based on the AHP method has been presented. The algorithm, which takes into account context information from both the terminal and network side, should be suitable for vertical HO decision making process in heterogeneous networks environment. The user-friendly approach of the algorithm ensures that it is easily configurable by average users. The algorithm is fully flexible and dependent on the number of chosen objectives that will determine the dimension of the pairwise comparison matrix for objectives as well as the number of pairwise comparison matrices for networks in terms of each objective. The decision algorithm uses basic mathematical calculations that could be particularly suitable for embedded hardware in practical mobile devices. It is a service type based algorithm which means that the whole process is executed once for each type of running application, not for every running application. Thus, even in the worst case the total number of execution of the whole process is restricted to only three times while applications of all three types are running. This is particularly useful in minimizing processing time, handover delay, and CPU and memory usage.

Simulation results with a market product give us the confidence that the decision algorithm would work perfectly and efficiently in heterogeneous environment once the capabilities of the available networks are known. It could make intelligent decision in accordance with user preferences. Thus, the algorithm should become a valuable aid for designers in designing smart software for future multimode terminals especially those having limited resources.

In future research, we intend to investigate the performance of the simulation software while working with conversational/real-time and streaming applications. We also intend to extend the algorithm further taking location information of users and the reachable APs into account, study and define a framework for context management including transfer methods and formats for context information, and database architecture for context information storage and maintenance, and possible context information measurement from available layer-2 or layer-3 signal. Furthermore, a framework for QoS-based HO activation and QoS support for multimode terminals would be investigated.

7. References

[1] T. L. Saaty, “How to make a decision: The Analytic Hierarchy Process”, *European Journal of Operational Research*, 1990, Vol. 48, pp. 9-26.

[2] B. Hongyan, H. Chen, J. Lingge, “Intelligent Signal Processing of Mobility Management for Heterogeneous Networks”, *Proc. of the 2003 IEEE Int. Conference on Neural Networks and Signal Processing*, Dec. 2003, Nanjing, China, Vol. 2, pp. 1578-1581.

[3] P. Chan et al., “Mobility Management incorporating Fuzzy Logic for a Heterogeneous IP Environment”, *IEEE*

Communications Magazine, Dec. 2001, Vol. 39, Issue 12, pp. 42-51.

[4] M. Stemm, R. H. Katz, "Vertical Handoffs in Wireless Overlay Networks", *ACM Journal on Mobile Networks and Applications*, Dec. 1998, Vol. 3, Issue 4, pp. 335-350.

[5] K. Pahlavan, et al., "Handoff in Hybrid Mobile Data Networks", *IEEE Personal Communications*, April 2000, Vol. 7, Issue 2, pp. 34-47.

[6] H. J. Wang, R. H. Katz, J. Giese, "Policy-Enabled Handoffs across Heterogeneous Wireless Networks", *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, New Orleans, USA, Feb. 1999, pp. 51-60.

[7] H. Bing, C. He, L. Jiang, "Performance Analysis of Vertical Handover in a UMTS-WLAN Integrated Network", *Proc. of the 14th IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2003)*, Sep. 2003, Vol. 1, pp. 187-191.

[8] M. Ylianttila, M. Pande, J. Mäkelä, P. Mähönen, "Optimization Scheme for Mobile Users Performing Vertical Handoffs between IEEE 802.11 and GPRS/EDGE Networks", *IEEE Global Communications Conference (GLOBECOM '01)*, Nov. 2001, San Antonio, USA, Vol. 6, pp. 3439-3443.

[9] H. Park et al., "Vertical Handoff Procedure and Algorithm between IEEE 802.11 WLAN and CDMA Cellular Network", *Proc. of the 7th CDMA International Conference on Mobile Communications*, Seoul, Korea 2002, pp. 103-112.

[10] S. Balasubramaniam, J. Indulska, "Handovers between Heterogeneous Networks in Pervasive Systems", *IEEE Proc. on Communication Technology (ICCT 2003)*, April 2003, Beijing, China, pp. 1056-1059.

[11] "Services and Service Capabilities", *3GPP TS 22.105 v.3.6.0*, 1999.