Architecture of a Context-Aware Vertical Handover Decision Model and Its Performance Analysis for GPRS – WiFi Handover

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

Abstract

The common desire to be connected "anytime, anywhere, anyway, and to anything" leads to explosive growth of mobile computing and speedy emergence of new wireless technologies, applications, and devices in recent years. In this array of heterogeneous systems, intelligent handover (HO) decision, beyond traditional ones that are based on only signal strength, is needed so that terminals can select the best option available from diverse networks and services as per user requirements. In the process, it would enable user applications to switch automatically between active interfaces that best suit them based on application requirements and interface capabilities, and to use multiple radio interfaces simultaneously. To fulfill the above requirements, this paper describes the architecture of a context-aware vertical HO decision model suitable for multimode mobile devices in heterogeneous networks and evaluates its performance for vertical HO between GPRS and WiFi.

1. Introduction

Recent year have witnesses a paradigm shift of wireless communications networks and devices. Rapid growth of mobile computing and emergence of new wireless technologies like GPRS, UMTS, WiFi, WiMAX, Bluetooth, DVB-H, etc would result in evolution of wireless networks towards heterogeneous all-IP infrastructure. 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 (*vertical HO*) that best suit them based on user preferences, application requirements and interface capabilities, and to use multiple radio interfaces simultaneously. Traditional *horizontal 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 an intelligent HO decision model including the session transfer, which takes into account intelligence residing on the terminal side as well as on the network side, collectively known as *context information*. Emphasis is given to make the design and working principle of the model simple enough to be suitable for practical multimode mobile devices (e.g. pocket PC phone) that have several capability constraints like processor speed, memory size, power consumption, etc. On the contrary, it is versatile enough to be configurable by users. The model 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 illustrates the design of the decision model. Section 4 and 5, respectively, describe the implementation of the model and simulation results. Finally, section 6 concludes the paper.

2. Related work

Pandolfi *et al.* [2] propose an algorithm for optimized simultaneous interface usage in a converged multimode terminal based on user preferences and network capabilities, however, its use of exponential functions (sigmoid) makes it unsuitable for implementation in existing mobile devices whose embedded hardware does not support these types of complex calculations. On the same ground, Hongyan *et al.* [3] and Chan *et al.* [4] proposed *fuzzy* based multiple-criteria decision making processes to perform access network selection and vertical HO based on the cost constraints and application priorities specified by users are also unsuitable for practical mobile devices.

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)*. These 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.* [7] also use 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 session transfer based on the HO decision.

3. Vertical handover decision model

In the hierarchical decision making approach of the decision model, adapted from the AHP model [1], 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. Based on these two sets of data, the best option is determined. Finally, applications that need to be transferred to the selected interface are switched based on their relative priority. We have considered *mobile-initiated and controlled* vertical HO for the decision model.

3.1. Context model

The context model chosen for the decision model is shown in Table 1. 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 five *service types*, where applications of a specific service type have similar Quality of Service (QoS) requirements. These are – (i) voice applications (ii) real-time streaming (iii) interactive (iv) non-real-time streaming and (v) background, 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* define the service profile of currently running applications. *Reachable access points (APs)* identify 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.

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

3.2. Architecture of decision model

The architecture of the decision model is shown in Figure 1, which is processed for a particular service type. In accordance with the AHP method, at first, we have to define some *primary objectives* for our decision model taking into account the preferences likely to be the most interesting to users (e.g. cost, expected QoS, interface priority based on coverage, etc) and 3GPP defined QoS parameters [8]. 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 Error Rate (BER).

3.2.1. Pre-configuration.

Stage 1: Taking user inputs. For any of the five 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 five 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 the following equation:

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

In (1), 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.

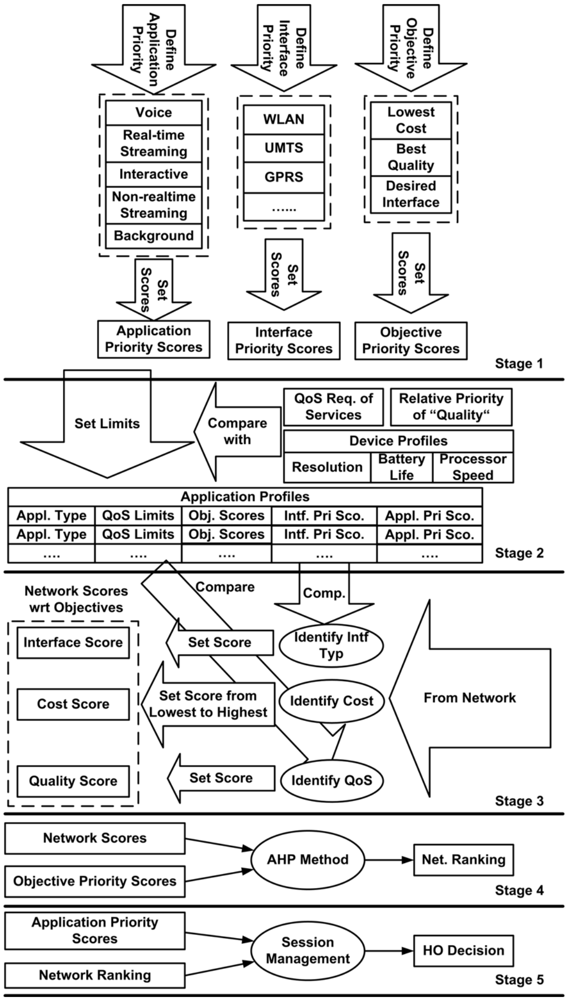

Figure 1. Architecture of decision model

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 five 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. In case of mean throughput, the lower limit is always a fixed value, i.e. *minimum requirement* (e.g. ≥ 4 kbps for voice applications [8]). 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 voice applications [8]), 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 voice applications). 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 voice applications [8]).

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. It is assumed that a multimode mobile device would learn about network capability contexts from the reachable networks.

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, 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 i and n_i denotes the value offered by a network for that particular parameter, the network score, S_i, based on the 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, 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.

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

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

Stage 4: Calculating network ranking. At this stage, *ranking* of the available networks are 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].

Step1. 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 and 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.

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

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

$$RS_{ab} = 1 \qquad ;S_a = S_b \qquad (6)$$

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 dependent 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).

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

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

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

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.

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

$$R_i = \sum\nolimits_1^{ij} c_{ij}(p_j) \qquad (10)$$

Stage 5: Session management. At this final stage, an efficient session transfer scheduling algorithm is employed in order to switch applications to the selected interface. 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 (10) at stage 4. For i number of running applications the overall score, O_i, is calculated using (11), 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 (12).

$$O_i = a_i'\left(R_i - R_d\right) \tag{11}$$

$$a_i' = \left(1 - \frac{a_i}{10}\right) \tag{12}$$

The value of O_i is always between -0.9 and +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

Basic modules of the *Reference Architecture* are shown in Figure 2. 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 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)* modules, respectively. All information is stored in the *Storage*. The HO decision algorithm is processed in this module. After processing the decision algorithm LSTM notifies the AfA module about the applications that should be moved to alternative interfaces.

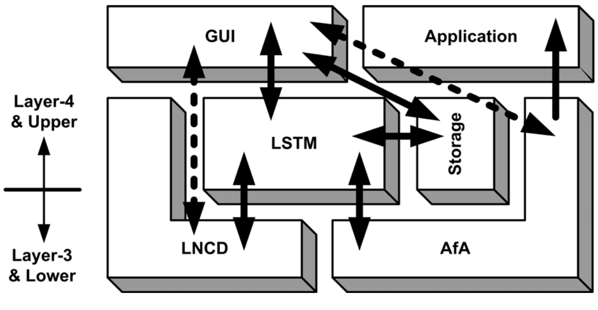

Figure 2. Reference architecture

The AfA module takes care of the session mobility. In this architecture no common mobility platform (e.g. Mobile IP) is present; instead, interface technology dependent individual and standard mobility features are used. It receives notifications from the LSTM module and accordingly shifts applications to alternative interfaces. In the process, it takes care of the generation of new IP addresses and assigns them to applications. It also feeds the LSTM module with application profiles during the initialization phase. The GUI enables users to configure the decision algorithm.

5. Simulation

The test setup for simulation is shown in Figure 3. The WiFi AP also acts as an AR and has DHCP functionality. A *dummy* application (interactive service type) 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. The pocket PC phone (PPC-P) runs on *Windows Mobile 5.0* platform, which has Bluetooth and WiFi stacks. The device has both GPRS and WiFi interfaces.

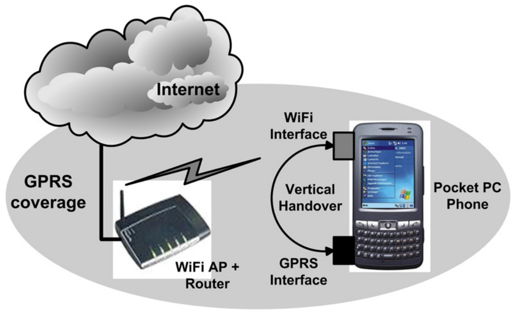

Figure 3. Simulation setup

Access networks like GPRS and WiFi advertise no additional information (e.g. current QoS) other than merely the coverage indication or IDs (SSID and BSSID for WiFi). 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 corresponding access network type by the LNCD, represents individual network capabilities. WiFi always offers better capabilities according to the settings of the configuration files. 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, where AfAWeb was running in the device as the lone application. In both the scenarios, the availability of GPRS and WiFi networks were arranged in such a way so that it represented their typical attributes (i.e. almost ubiquitous coverage for GPRS and *hotspots* for WiFi). In *scenario 1*, the PPC-P is connected to the Internet through GPRS. At some point, the WiFi AP is turned on and vertical handoff is performed between GPRS and WiFi. In *scenario 2*, the PPC-P is connected to the Internet through the WiFi AP while GPRS interface is also active. At some point, the WiFi AP is switched off and the terminal handovers to GPRS (default network).

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 2). In this case, the available network was selected as the default one. The algorithm was processed only while performing vertical HO (scenario 1) between two available networks and the HO decision precisely matched with that calculated theoretically. Also, the HO and session transfer mechanisms worked perfectly in this case.

Average time delays experienced at different execution phases during vertical HO, whose definitions are illustrated in Figure 4, are summarized in Table 2. Average time taken for interface detection is much higher for WiFi than for GPRS because the former is influenced by the technology specific detection mechanism, whereas, the interface is always active in the latter case. The major strength of the decision algorithm is highlighted through the small average delay (10 ms) experienced during decision making. Note that delay for session reestablishment depends on the behavior of individual type of application and varies among different types of applications. The small total average delay implies that even with additional delays in real networks the algorithm would work perfectly for most delay sensitive applications like voice conversation or real-time video.

Table 2. Simulation results

Execution Phases	Average delay (ms)	
	GPRS→WiFi	WiFi→GPRS
Interface detection	35	5
Decision making	10	-
Sess. trans. with re-estbl.	15	15
Sess. re-estbl. alone	10	10
HO w/o sess. trans.	45	5
HO with sess. trans.	60	20

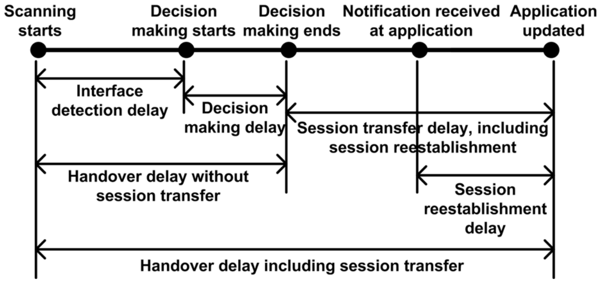
Scanning starts
Decision making starts
Decision making ends
Notification received at application
Application updated
Interface detection delay
Decision making delay
Session transfer delay, including session reestablishment
Handover delay without session transfer
Session reestablishment delay
Handover delay including session transfer

Figure 4. Definitions of time delays

6. Conclusions

The decision model presented in the paper, 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 model is fully flexible and dependent on the number of chosen objectives that will determine – (i) dimension of the pairwise comparison matrix for objectives and (ii) 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 a mobile device. It is a service type based model 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 five times while applications of all five types are running. This is particularly useful in minimizing processing time, HO delay, and CPU and memory usage.

Simulation results with a market product give us the confidence that the decision model would work perfectly and efficiently in heterogeneous environment once the capabilities of the available networks are known.

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] A. Pandolfi *et al.*, “Optimising Simultaneous Interface Usage In A Converged Multimode Terminal”, *Proc. of the 14th IST Mobile & Wireless Communications Summit*, Dresden, Germany, June 2005.

[3] 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*, 14-17 Dec. 2003, Vol. 2, pp. 1578-1581.

[4] 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.

[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)*, 25-26 Feb. 1999, pp. 51-60.

[7] S. Balasubramaniam, J. Indulska, “Handovers between Heterogeneous Networks in Pervasive Systems”, *IEEE Proc. on Communication Technology (ICCT 2003)*, April 2003.

[8] “Services and Service Capabilities”, *3GPP TS 22.105 v. 3.6.0*, 1999.