Handset Based Automatic Network Re-selection System for GSM/GPRS and WiFi

Andrés Navarro C. ¤, Andrés F Millán *, Alexander García D ¤,
Gabriel Tamura ¤, Oscar W Jiménez ¤

anavarro@icesi.edu.co ¤, afmillan@usc.edu.co *, agdavalos@uao.edu.co ¤,
gtamura@icesi.edu ¤, owjimenez@uao.edu.co ¤.

¤ Universidad Icesi - * Universidad Santiago de Cali - ¤ Universidad Autónoma de Occidente

ABSTRACT

This paper presents the results of the development of a handset based automatic network reselection software system, which allows keeping end-to-end connections through semi-transparent handoff on WiFi/GSM-GPRS networks.

A description of the developed system's software architecture is given, as well as of the components implemented for it, their functionalities and flow of control, all of which allow the permanent discovery of available networks for the mobile device, their evaluation, selection and switching to the network which offers the best ratings in its performance and operational profile, taking into account the preferences previously configured by the user. The system was developed in Java and C# for different operating systems (WinXP, WinMobile and Linux), and its interoperability has been tested in them. Some of the test scenarios are described and final comments about the tests results are given.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—Wireless communication.

C.2.0 [Computer-Communication Networks]: General— Data communications.

K.8.1 [Personal Computing]: Application Packages—Data communications.

General Terms

Algorithms, Experimentation, Performance

Keywords

Network re-selection, vertical handoff, wireless networks, user mobility.

MobiWac'08, October 30-31, 2008, Vancouver, BC, Canada.

1. INTRODUCTION

Nowadays mobile users continuously change their points of attachment to the networks as they move through wireless environments. In heterogeneous wireless network environments this situation presents some problems when talking about mobility, and for the solution of them, it is important to consider the issues that this entails, like choosing the best available network according to parameters such as efficiency, reliability and cost, as well as to other aspects, such as authentication and security.

Our proposal aims to solve these problems by developing a system that allows mobile devices to perform semitransparent handover between WiFi/GSM-GPRS networks, in different operating systems while trying to keep optimal conditions of network performance.

The first attempt was the development of a module which consisted of a WiFi/GPRS Monitor, which allows the detection of all available Access Points or GPRS networks in a zone, and a network switcher, which provides the service for changing the point of attachment amongst the discovered networks. The software system also has an optimizer that applies a policy-based algorithm for automatic re-selection of the most adequate network, based on pre-configured mobile user preferences.

The analysis, architecture and design of the system was based on the conceptualization presented by Dorfman [1] and Bass [2], using the UML notation for execution and deployment diagrams [3].

An initial phase of research and design was made, based on the analysis of the advantages and disadvantages of the OSI reference model, and its derived implementation technologies.

The rest of the paper is organized as follows. In section 2 some related works are presented. The considerations about context and system architecture are outlined in section 3. Section 4 identifies the main requirements for the system architecture. Sections 5 and 6 describe the system architecture and some of its implementation details. Then test scenario and results are presented and analyzed in section 7, which is followed by the conclusions and future work in sections 8 and 9.

2. RELATED WORKS

Some of the most known vertical handoff solutions based on different layers of the OSI Model are presented in Figure 1.

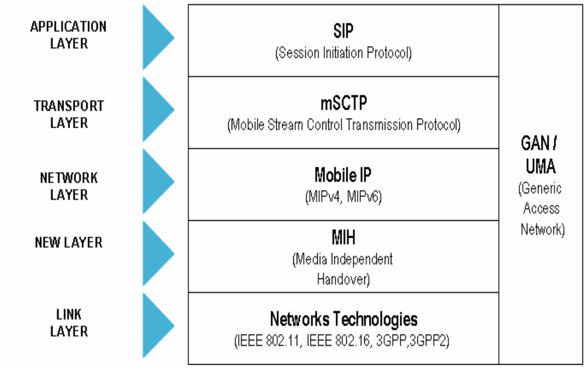

Figure 1. Vertical Handoff Solutions based on the OSI Model

GAN (Generic Access Network) is a 3GPP technical specification [5] [6] (commercially named UMA - Unlicensed Mobile Access) which enables the convergence between 3GPP and WLAN networks. GAN uses a tightly-coupled or integrated architecture, which requires GAN controllers in the network access. From the point of view of the OSI Model, the GAN model spans layers 2 to 4 (user plane) and layers 5 to 7 (control plane).

Another recent proposal is MIH [7] (Media-Independent Handover), which is an IEEE standard effort for vertical handoff between IEEE 802 and non-IEEE 802 networks. MIH has three components: MIES (Media-Independent Event Service), MICS (Media-Independent Command Service) and MIIS (Media-Independent Information Service). These components interact among themselves by sending alarms related to the mobility over nodes.

Mobile IP, in two versions, Mobile IPv4 [8] and Mobile IPv6 [9], are vertical handoff solutions more widely used. A number of network architectures proposals (3GPP and 3GPP2) have selected Mobile IPv6 as their mobility signaling protocol. However, this approach requires significant changes in the network architecture, for which service providers are still not willing to implement it.

SCTP (Stream Control Transmission Protocol) [10] is an IETF proposed standard protocol for the transport layer. Like TCP, SCTP is reliable but offers new features such as multi-streaming and multi-homing. The SCTP with the ADDIP extension is called mobile SCTP (mSCTP), and it provides a seamless handoff for mobile hosts that are roaming between IP networks.

A recent trend in this field is the IETF proposed Policy-based strategies, which has been adopted by IEEE P 1900 group and is currently in discussion in 3GPP[11].

SIP (Session Initiation Protocol) [12] was initially developed by IETF as an application layer multimedia signaling protocol, but it has the potential capabilities for Internet mobility. SIP theoretically supports mobility using the REINVITE message for maintaining the session active during handoff process.

Some solutions for 3G and WLAN network integration were implemented in experimental environments. Buddikot M. et al. proposed a solution that introduces two architectural choices for WLAN/3G networks [13]. The first one is tightly-coupled interworking and the second one is loosely-coupled interworking. However, they recommended the second approach, which consists of two key components: an additional network element called IOTA (Integration of Two Access Technologies) access gateway, deployed in the WLAN network, and a special client software for mobile devices. This kind of solution has some disadvantages because it is not so flexible and increases the network complexity as well as the cost for network service providers.

After reviewing different previous research projects, we decided to work on the application layer, mainly due to the following reasons:

- Implementation is more simple and faster because it is not necessary to recompile host kernels nor to install additional software such as servers or agents.
- Applications on this level are easily adaptable to changes on wireless network standards. It is possible to use them on different scenarios and they can use commercial products which operate in lower levels.

3. Considerations about Context and System Architecture

In order to achieve the main objective of the project, which is to maintain uninterrupted communication sessions on a mobile device while keeping its connectivity through different types of access and transport networks in a semi-transparent way, two aspects can be identified: (1) Automatic switching from a network to another when appropriate conditions are met; and (2) maintain the communication sessions at the application level.

The first aspect can be seen, from the applications and functional and structural point of view, as the network access and transport independence, whereas the second one can be seen as communication session independence. This second aspect was not worked in depth in this project, as it was beyond of its scope and, in fact, needs more development time. With the above considerations, the problem was divided in two phases, each one corresponding to each of the mentioned aspects.

The criteria for the architectural design are based on the following considerations and decisions:

- Network Assisted HO, Mobile Assisted HO or Mobile Controlled HO: we select Mobile Controlled because we cannot access available commercial networks and for that reason was not possible to implement network side elements.
- Tight Coupling or Loose Coupling: we choose Loose Coupling, for the lack of access to operator's network's exposed above
- Hard Handoff, Soft Handoff or Softer Handoff: we choose hard handoff, because the Windows Mobile device does not have enough computing power to manage both connections simultaneously. We are not quite sure if it was a hardware restriction or an operating system limitation.

In the following sections, the identified requirements for the system architecture are presented, followed by the description of the architecture designed to satisfy them.

4. Requirements for the System Architecture

The identified primary requirements for the project were defined as follows:

- Detect all available networks in all active network interfaces in the mobile device, with their corresponding levels of performance.
- Evaluate the detected networks and select the one amongst them which offers the best performance conditions, based on their individual levels of performance and scoring them according to the user predefined preferences.
- The network evaluation must consider the stability of the so selected network.
- The system could be configured for the user to decide if the system automatically selects the best available network or if it must ask the user every time to select the network. In the first case, the system can be configured to make the network discovery every certain time, for example every 10 seconds to every 3 minutes, depending on the user mobility variation.
- The system must register and modify user profiles and preferences; the variables are related to the weights of the network characteristics which are used to evaluate them. Such characteristics are, for example, bandwidth, cost of connection per time unit and battery power consumption.
- Execute the switching between networks, to the selected network, when appropriate conditions are met.
- The system must authenticate with the target network and support different authentication schemes and their variations, as for example in the WiFi case: different char codifications and combinations of different lengths for the passphrase.
- The system must manage the different types of network authentication systems and their corresponding parameters, associated with the network profiles, for each network the user has access rights, in such a way that the switching from a network to another can be executed completely without user intervention.
- In a network switching, the final connection will be effective over the target network once all the initialization processes have been completed, including the authentication process.
- The system must have an administrative user interface.

5. System Architecture and Component Description

The architecture diagram describing the whole system is shown in Figure 2. This UML deployment diagram shows the static structure of components and the dynamic relationships between them, with the most relevant interfaces for each component.

In the following sections each component is briefly described.

5.1 Hardware/Software interface

For all of the interactions between the software system and the hardware resources such as network adapters and drivers, measurement of physical attributes of available networks, network switching, etc., the OperatingSystemAgent component was designed.

This component groups together all the low-level functionalities, providing a common point of interaction with the hardware and localizing the operating system dependencies, in terms of

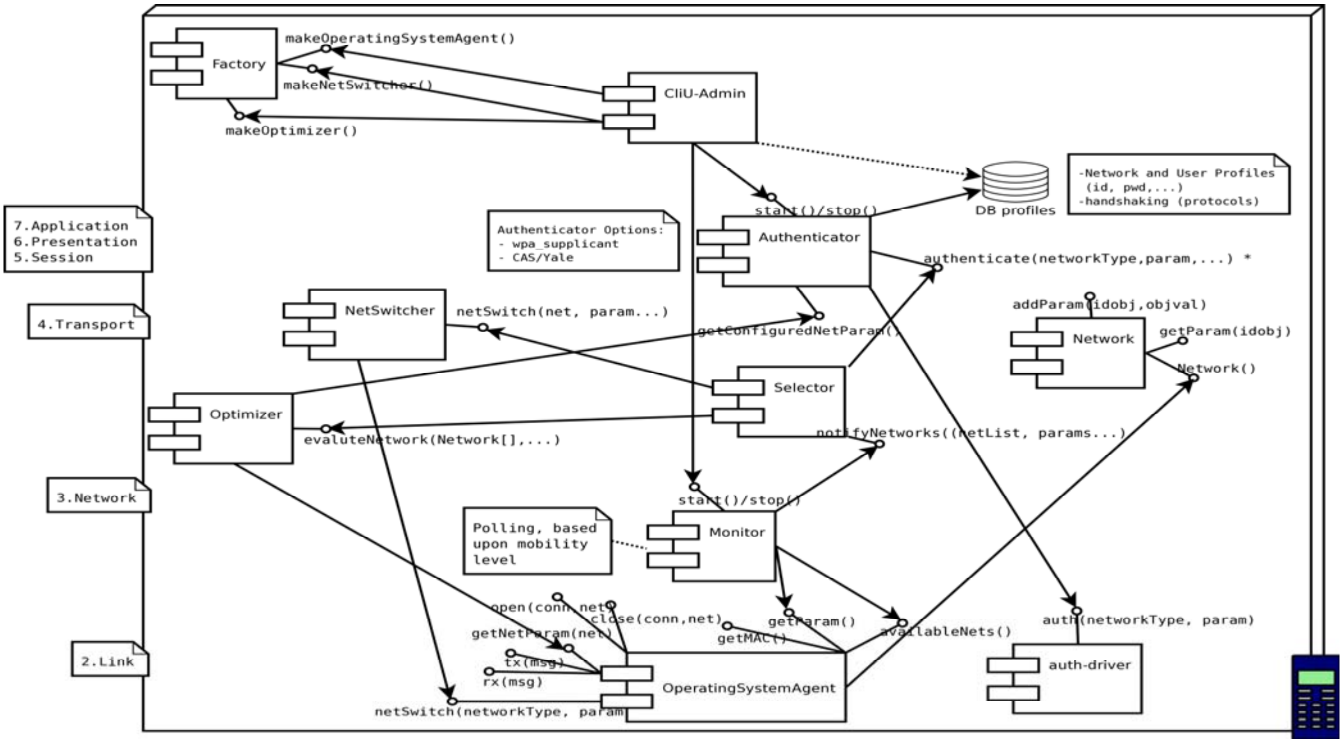

Figure 2. System Architecture: Deployment Diagram.

pplatform-dependent drivers and libraries function calls, in a single component. This is crucial for the maintainability and portability of the software system.

5.2 Networks discovery

This functionality is implemented in the monitor component. From the functional point of view, as it is illustrated in the architecture deployment diagram, this is perhaps the most important component of the system and therefore some details of its implementation will be described in more depth in section 6.

This component is responsible for the networks detection on all network interfaces available in the mobile equipment, and of gathering the levels of performance of the discovered networks attributes. Because there is no external identifiable trigger event to start the networks discovery process, the monitor was designed to be implemented as an independent thread that periodically polls for available networks, according to the user mobility level. For this, the system has a parameter which can be defined by the user.

5.3 Network Re-selection and decision

The selection and decision process is implemented throughout two different components: Optimizer and Selector. The first one is responsible for discovered networks evaluation; and for comparing them and suggesting the best one among them. The second component is responsible for the control of the process that begins when the monitor notifies it about available networks just discovered, and then it requests the NetworkSwitcher to execute the switch to the network suggested by the Optimizer, if the appropriate conditions are met. Some details about the implementation of this functionality will be given in section 6.

In order to achieve this task, the Optimizer receives from the Selector a set of available networks and evaluates them considering bandwidth, battery power consumption, monetary cost and user mobility level. These parameters have relative weights associated to a cost function of relative scale [14][15] which are set up by the user trough an administrative GUI. Additional elements like dwell-timer and hysteresis [14][15] are used for avoiding situations such as ping-pong effects in network connections, meaning that there could be oscillations and instability in the networks.

5.4 Network switching and packet re-routing

As it was previously mentioned, the network re-selection process can be partitioned by considering two aspects: (1) automatic network switching and (2) session reestablishment in a semi-transparent way. Put in a simple way, such aspects could be reflected in network switching and packet re-routing functionalities. But in order to fulfill the second aspect, non trivial problems have to be solved.

The network switching functionality was implemented in the NetworkSwitching component, which receives from the Selector component a request to make a switch to the network that was evaluated as the best available one. The NetworkSwitching component starts the switching process to the new network without affecting the active connections (multihoming). Before the execution of the network switch, the component also makes an authentication request (if necessary) to the Authenticator component. Once the authentication is confirmed and the network initialization is done, this component should re-route the traffic flow through the new connection.

This packet re-routing and session reestablishment was explored by using the SIP protocol and SIP public domain implementations like OpenSER [16] and OpenIMS [17]. However, the full implementation of SIP mobility is subject of further work, because the exploratory work allowed us to conclude that currently available implementations of SIP and related RFCs does not include mobility explicitly, and this implies a considerable additional development work, both in client application and in the server.

5.5 Authentication

The authentication functionality was implemented in the Authenticator component, which is responsible also for the management of user's authentication data (user profiles) and of network security information (network profiles) and configuration information.

The switching to the target network requires several steps, and involves the corresponding network authentication parameters. Once the authentication process is finished, the authenticator must inform the NetworkSwitch function if validation was successful or not. It is important to mention that some networks have an intrinsic authentication process, not dependent of the authenticator component.

5.6 System Administration and other aspects

The system administration was implemented in the CliUAdmin component, which is responsible for the general configuration parameters of the whole system, such as user mobility level, operating system, and general tasks such as registering and modifying user preferences, relative weights for the evaluation of different networks, amongst others.

Several other system parameters are managed in their corresponding components. For instance, the parameter MobilityOptimizer belongs to Optimizer and is independent from available networks, while the parameters Optimizer.paramWb, Optimizer.param.Wc and Optimizer.paramWp are particular for each configured network. In such a way, when the user modifies its preferences, the Optimizer must ask for corresponding values for each available network.

This component is also responsible for starting the Monitor and Authentication components

6. Implementation Details

6.1 Network Discovery

In this section, we will describe some implementation details of the monitor component, presenting the different approaches considered for its development in the Windows XP platform. Similar approaches were used for Windows Mobile and Linux platforms.

As was stated previously, this component is responsible for the networks detection on all network interfaces available in the mobile equipment, so its functionality is divided in two parts: firstly, the logic of the periodic process of network discovery, and secondly, the network discovery as such, i.e. its description, attributes and technical characteristics, this latter depending heavily on the OperatingSystemAgent. For their implementation several approaches were considered, as follow.

6.1.1 Java Sockets and WMI

In this approach, a sockets-based application on the Java programming language was developed, extending the Tadashi Okoshi's MobileSocket API [4], handling data and control channels of communication and considering four control channel states: closed, explicitly suspended, implicitly suspended and established. Although this implementation allowed continuity in the connection and mobility in the application level, besides handling consistency aspects, its main problem was its lack of transparency in its use, i.e. the mobility it implemented only benefited applications developed in Java, by changing in the source code the java.net.Socket class by the MobileSocket extended class.

In this approach, it was also considered the use of the Win32_NetworkAdapterConfiguration [20] parameter of the WMI API [21], to obtain information about the available network adapters. This forced the consideration of the Windows platform metrics, which turned the solution not viable, as far as there is no standard associated with metrics and routing tables in the Windows operating system family, and the API provides only limited information about networks.

6.1.2 Win32 Drivers and NDIS Protocol

This option considered the direct use of the Windows "Ws2_32.dll" library [22], having a stronger interaction with the operating system, by using fragments of C# unhandled code, which made it possible to obtain more information about the network adapters.

However, this scheme only worked in one direction, allowing to gather information about network adapters, but not being able to manage that information and modifying it back in the network adapters; it was then necessary to work at the device driver level, which was done through the NDIS [23]protocol driver. The problem with this implementation was the lack of control over the packet handling, making it necessary to develop packet disassembling/reassembling functions, besides intercepting them.

6.1.3 Integration of WMI, Drivers and Sockets

This approach used the Resilient Mobile Socket Library (RMS) [18] [19], which have been used to build real-time multimedia communication applications. This approach is similar to the used in the first described option, in which MobileSocket was used. RMS resulted of interest because it considers aspects that were identified as critical in the first option, like handling of routing tables and drivers, and because it uses the WRAPI (Wireless Research API) libraries [25] and IPHelper [26]. Nonetheless, these utilize functions that were not standardized and platform version dependent.

6.1.4 WiFi Monitor for Windows XP SP2

The last option was finally focused on developing the component using the NDISUIO driver [28] and the kernel32 libraries, complemented with the Robotics4Net library [29][30] developed by the University of Pisa.

For a proof-of-concept, an application which acts like a Wireless Network Monitor was developed. This application allows to select one NDIS device (network adapter) from a list of network adapters, and from that device, the user can select one Access Point, for example, from the ones that are available, to make the network connection.

The combination of libraries used in this option provided all of the information and handling required for the functionality assigned to this component.

6.2 Networks Evaluation and Selection

The process of evaluation and selection is achieved by using a vertical handoff algorithm (VHA) based on policy [24][27], which utilizes a cost function of relative scale [14][31], as proposed in (1).

$$f_i = W_b \ln\frac{1}{B_t} + W_p \ln P_t + W_c \ln C_t \tag{1}$$

The decision to implement this type of cost function is based on the qualities that it offers, since it allows to add new parameters in a very simple way.

This is done by evaluating the impact that has the parameter on the network performance, and its contribution is controlled by using the logarithm in normal form or evaluated on the inverse of the parameter, having as effect that the values of cost of the networks are determined in agreement to these criteria.

The parameters used for the evaluation of the networks associated with the cost function are the consumption power, connection cost, bandwidth and user mobility (level). The cost function is added with a parameter related to the degree of user mobility, in the following way:

- In the case of average or high mobility of the user, the cost function is:

$$f_i = W_b \ln\frac{1}{B_t} + W_p \ln P_t + W_c \ln C_t + \ln\frac{1}{V} \tag{2}$$

- In the case of low user mobility, the function will be executed in its original form, as in (1).

$$f_i = W_b \ln\frac{1}{B_t} + W_p \ln P_t + W_c \ln C_t \tag{3}$$

Where:

Wb: is the weight associated to the bandwidth parameter.

Wp: is the weight associated to the battery consumption.

Wc: is the weight associated to the connection cost.

B: Bandwidth.

P: consumption of the battery power.

C: Cost of connection to the network.

V: degree of speed of user movement.

The addition of the user mobility parameter to the cost function was done considering its importance for QoS [15], since this is dependent on the network coverage area, and which is also conditioned by the user mobility.

6.2.1 Stabilility of the Handoff Decision

In order to avoid unnecessary or erroneous switching in the network selection, elements of stability are used, with the following criteria [14][31]:

- Dwell-Timer: its function is to check that the conditions of the candidate network remain stable for enough time; for this purpose a lapse of time is defined to wait for checking if the conditions of the chosen network remain the same. If the conditions do not change, the network is selected and the switch to it is done; on the contrary, the current network is not changed. The lapse of time for checking the stability of the network is related to the mobility parameter.
- Hysteresis margin: its function is to add a small value to the cost of the current network that allows it to have a margin of comparison in its evaluation, when compared with the evaluation of the other available networks. For this purpose, the additional value that was added to the cost function is determined by the quality of the transmission power of the Access Points (AP).

7. Test Scenarios and Test Results

7.1 Network Environments and Hardware Devices Used in the Tests

The tests were performed in a wireless network environment with internet access and a DHCP and chat server. In this test environment, different models and types of access points (Linksys, Avaya, Cisco, NetGear) and GPRS card devices (Vodafone, Motorola), with different security access configurations (WiFi: WEP, WPA in its variants) were configured. As mobile devices were used Laptop PCs and smartphones, with different operating platforms (Windows XP, CE, GNU/Linux). The tests general network environment is illustrated in Figure 3. The test scenarios and some test results are presented in the following sections.

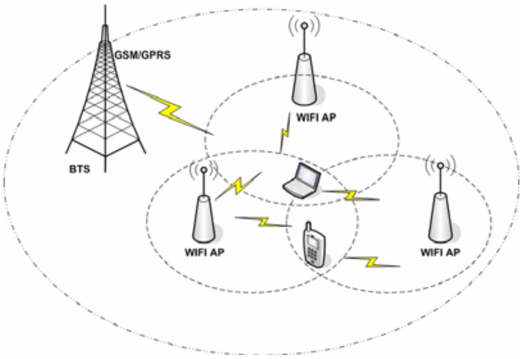

Figure 3. Tests Network Environment.

7.2 Analysis of VHA and Elements of Stability

Some variations of the stability parameters were made during the tests of the proposed algorithm, in order to improve its behavior in the network selection. Figure 4 shows the test results obtained:

- A: represents the evaluation of VHA considering all elements of stability in the algorithm (hysteresis, average value of the cost function, and dwell-timer).
- B: represents the evaluation performed by the algorithm considering only the hysteresis and dwell-timer as elements of stability.
- C: represents the evaluation performed by the algorithm considering only the elements of stability dwell-timer and average value of the cost function.
- D: for this test the algorithms use no stability elements.

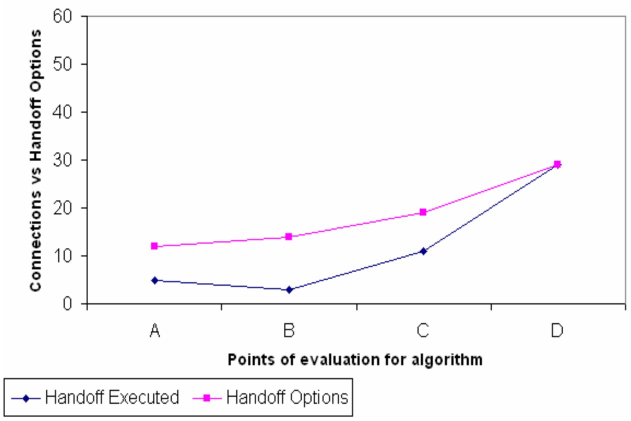

Figure 4. Analysis of the Behavior of VHA

For this test, 59 network scans for evaluation and switching were done. As is observed in the figure 4 for point A, there were presented 12 options for handoff, but only 5 of the total options were executed. For the case of point B, 14 options were presented but only 3 were executed. In C there were 19 options, but only 11 were executed and for the case of point D, 29 options there were presented and all of them were executed, which shows that stability elements used in the algorithm are effective in the network selection process.

7.3 Test Results

Considering that functionally, the tests of VHA were satisfactory, the time response of the switching between networks is not the best. While the mobile device is very slow for switching between networks (upward handoff) from WiFi network to GPRS cellular network, the process took about 8 to 12 seconds for establishing the network connection. In the other case switching from GPRS to WiFi network (downward handoff) took between 4 and 6 seconds in average. In the case of the tests performed with Laptops, the process of switching network connections (upward handoff) from WiFi to GPRS network, the elapsed time was about 1 to 1.5 seconds for establishing the network connection. In the other case switching from GPRS cellular network to WiFi network the time for network connection was of 0.5 to 0.8 seconds.

Figure 5 shows the average values of time for network selection and switching performed by a mobile device Pocket PC (Windows Mobile 5.0 operating system) and a laptop with Windows XP, respectively.

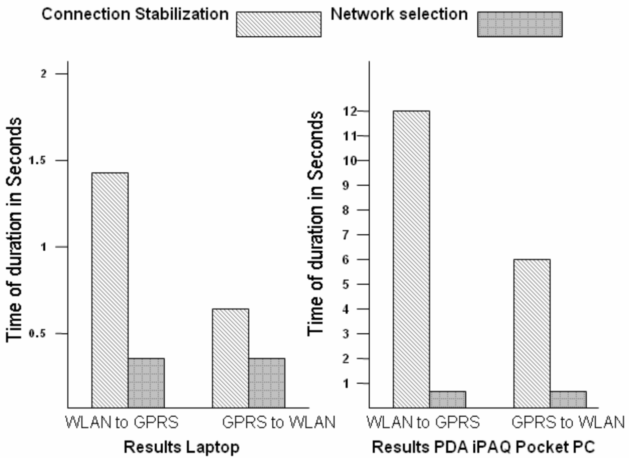

Figure 5. Time delay of network re-selection

There are three main reasons for the time delay of network re-selection process on the PDA: the first one is the time response of Win Mobile operating system, which presented some instability when the software system prototype was running; the second one is the less computing power of the PDA. And the third one is the time for the authentication process, which takes more time (twice as much than on the Laptop) when the software prototype is running on a PDA. The tests carried out using the Laptop and a cell phone as a GPRS modem showed that authentication process is executed faster under the cell phone operating system so the time response is better.

The first issue needs more research work and it was not considered in the scope of this project. The other two issues are more general, especially the processing power that is a big challenge to chip manufacturers.

8. CONCLUSIONS

In this paper we described a handset based automatic network re-selection software system, which allows keeping end-to-end connections over WiFi/GSM-GPRS networks in a semi-transparent way.

To achieve this objective, a rationale was chosen and a set of requirements were established. Based on this, a system architecture was designed. Its specification was presented in UML's deployment type of diagram. This specification was implemented in Java and C#, in the Windows XP, Windows Mobile and GNU/Linux platforms.

The stability elements (dwell- timer and hysteresis margin) used were an effective way for reducing unnecessary network switching due to instability of network behavior.

As a final comment, it is important to mention that the deployment of the system in the mobile devices had some problems with the development platform and virtual machines, especially the Microsoft's ones, exposing the high dependency that there still exists in the development of software applications for this kind of devices. These dependencies present important issues which must be evaluated, especially because they can cause serious waste of time in the development of a project.

9. FUTURE WORK

For improving the proposed VHA, it could be tested in a simulated environment (software simulation) that allows generating more complex testing scenarios, in order to do the adjustments that are necessary to guarantee the efficiency in the wireless network re-selection. Also it is necessary to make a comparison to other vertical handoff algorithms, which can be done in simulated environments.

Finally, the software system, which was developed as a proof-of-concept, could evolve into a driver, part of the operating system, or even burn in firmware, to obtain better response times.

10. ACKNOWLEDGMENTS

We are very thankful to the telecommunications research consortium I2ComM (www.i2comm.org), for having allowed us to develop this proposal, within the framework of the project "Development of a universal system for clients with mobile devices allowing portability between networks Wi-Fi/GSM-GPRS/CDMA2000/Wi-MAX".

This project was funded in part by "Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnologia Colciencias" (www.colciencias.gov.co), which is a Colombian Institute for Supporting Research and Innovation.

11. REFERENCES

[1] Dorfman Merlin. "Requirements Engineering" in "Software Requirements Engineering", 2nd ed. pp. 7-22. The Institute of Electrical and Electronics Engineers, USA, 1997.

[2] Bass Len, Clements Paul, Kazman Rick. Software Architecture in Practice. Addison Wesley, 1998.

[3] The International Organization for Standardization. ISO/IEC 19501:2005 Information technology -- Open Distributed Processing -- Unified Modeling Language (UML) Version 1.4.2. ISO, 2005.

[4] Okoshi T, Tobe Y and Tokuda H, "MobileSocket: Library based Continuous Operation Support for Java Applications," 1999.

[5] 3GPP, "Generic Access to the A/Gb interfaces", TS 43.318, January 2005.

[6] 3GPP, "Generic Access to the A/Gb interfaces. Mobile Generic Access Interface Layer 3 Specification", TS 44.318, January 2005.

[7] IEEE 802.21/D5.00, "Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services", April 2007.

[8] IETF, "IP Mobility Support for IPv4", RFC 3344, August 2002.

[9] IETF, "Mobility Support in IPv6", RFC 3775 , June 2004.

[10] IETF, "Stream Control Transmission Protocol", RFC 4960, September 2007.

[11] Horrich, Sana; Ben-Jamaa, Sana and Houzé, Paul. Policy Based RRM for Network Terminal Decision Sharing. In: IEEE Vehicular Technology Magazine, vol.2 No.3, sep. 2007, pp.35-40.

[12] Schulzrinne H. et al, “Application-layer mobility using SIP”, Service Portability and Virtual Customer Environments, 2000 IEEE, Page(s): 29 – 36. December 2000.

[13] Buddikot M., Chandranmenon G., Han A., Lee Y.W., Miller S., Salgarelli L., “Integration of 802.11 and Third-Generation Wireless Data Networks”, INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, Volume: 1, On page(s): 503- 512 vol.1, 2003.

[14] Nyberg G. “SEMO: A Policy-based Prototype for Handover in Heterogeneus Networks”, Master Thesis in Computer Science, Umea University (Sweden), 01-2006.

[15] Duda Adrian, Cormac J. Sreenan. “Challenges for Quality of Service in Next Generation Mobile Networks”. Department of Computer Science University College Cork (UCC), Ireland.

[16] The OpenSER Team. Openser, the Open Source SIP Server. http://www.openser.org/

[17] Fraunhofer FOKUS. OpenIMS Playground. http://www.fokus.fraunhofer.de/bereichsseiten/testbeds/ims_playground/index.php?lang=en

[18] Kristianson J and Parnes P, “Application-layer Mobility support for Streaming Real-time Media,” Lule°a University of Technology, Lule°a, Sweeden, 2004.

[19] Kristianson J, “Simulation Study of SRM”, Master thesis, Marratech, 2001.

[20] Win32_NetworkAdapterConfiguration [Online] URL: < http://msdn2.microsoft.com/en-us/library/aa394217.aspx >

[21] WMI - Windows Management Instrumentation [Online] URL:http://www.microsoft.com/whdc/system/pnppwr/wmi/default.mspx>

[22] DLL-files.com [Online] URL:< http://www.dll-files.com/dllindex/dll-files.shtml?ws2_32 >

[23] NDIS Protocol Driver (NDIS 5.1) [Online] URL: < http://msdn2.microsoft.com/en-us/library/ms801145.aspx >

[24] [R.Yavatkar, D. Pendarakis and R. Guerin, A, 2000 Framework for Policy-based Admission Control, RFC 2753.

[25] Wireless Research Api (WRAPI) [Online] URL: < http://sysnet.ucsd.edu/pawn/wrapi/ >

[26] Java IPHelper API [Online] URL: <http://media.csee.ltu.se/~johank/iphelper/ >

[27] Murray Ken, Mathur Rajiv & Pesch Dirk, “Intelligent Access and Mobility Management in Heterogeneous Wireless Networks using Policy”, 2003, P 1-6.

[28] Srinivasa, Sudhir and JAFAR, Syed Ali. “The Througput Potential of Cognitive Radio: A Theoretical Perspective”. En: IEEE Communications Magazine, Vol. 45, No.5, may 2007, pp.73-79.

[29] NDIS User-Mode I/O Driver [Online] URL: < http://msdn2.microsoft.com/en-us/library/ms892537.aspx >

[30] Robotics4.net – ManagedWiFi [Online] URL: < http://www.robotics4.net/Software/ManagedWiFi.aspx >

[31] H. J. Wang, R. H. Katz, and J. Giese, “Policy-enabled Handoffs Across Heterogeneous Wireless Networks”, Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, 1998, p. 1-10.