

# Design and Implementation of a Radio Access Selection Algorithm for Multi-mode Mobile Terminals

Alexandros Kaloxylou<sup>1</sup>, Fotos Georgiadis<sup>2</sup>, Ioannis Modeas<sup>2</sup>, and Nikos Passas<sup>2</sup>

<sup>1</sup> Department of Telecommunications Science and Technology  
University of Peloponnese, Tripoli, Greece

<sup>2</sup> Department of Informatics and Telecommunications,  
University of Athens, Athens, Greece  
kaloxyl@uop.gr, {fotos, imodeas, passas}@di.uoa.gr

**Abstract.** Modern mobile terminals giving access to multiple radio access technologies at the same time allow users to select specific technologies and/or different operators for their services. This calls for an automated radio access selection mechanism. In this paper we propose such a mechanism with several novelties: i) it enables terminals to build prioritized lists of target access networks independently for each of their active connections; ii) it aims to satisfy user preferences instead of providing a mere load balancing between networks; iii) it is designed to operate with two decision-making points (mobile terminal and core network), splitting the complexity of the overall process. We discuss the main functionality of the proposed mechanism, its prototype design in SDL and specific details of our test-bed implementation using open source tools.

**Keywords:** 4G, radio access selection, multi-mode terminals.

## 1 Introduction

During the past years we are witnessing an increase of mobile phones that are equipped with more than one network interfaces and capabilities that characterize a small PC. The mobile operators deploy additional radio access technologies (e.g., WiFi, WiMAX), so that they can increase network utilization and provide for better services to the users. The idea is to use load-balancing mechanisms that will enable traffic sharing between different Radio Access Technologies (RATs). Thus, we end up with a heterogeneous environment, where users are flexible enough to select the most suitable RAT for their needs, while operators wish to enforce their policy and maximize utilization of their resources. In such heterogeneous environments, automatic RAT selection plays a crucial role in the functionality of the whole network. Most of the existing proposals focus on the selection of the “most suitable” target RAT during a handover (HO) execution or the establishment of a new call. Cost functions [1], fuzzy logic [2], neural networks [3] and policy-based schemes [4-6] are the candidates to tackle the issue.

Almost all of these algorithms have been designed with a single decision point, where all contextual information is evaluated. Their goal is to have either the user or

the operator to decide about the best RAT. Obviously, the interests and preferences of these two players are not always the same. Thus, the proposals either imply a future scenario where users will have total control over selecting operators and RATs, or a case that is closer to the current status where the operator has total control. Moreover, most of these proposals consider all a user's connections to be handled by the same RAT. This is a rather monolithic approach and by no means needs to be like this in the future. The terminals will be able to have active connections through different interfaces. For those proposals that suggest such a flexible management of connections, no implementation details are provided. Finally, most of the proposed mechanisms focus on the decision mechanism and do not provide information about how the additional functionality can be integrated into existing standards.

Having these issues in mind, we have designed and implemented a new RAT selection mechanism, where each connection of a Mobile Terminal (MT) is handled separately. This provides more flexibility on fine-tuning the available resources and at the same time increasing the user's satisfaction [7]. It is also split in two cooperative parts, at the MT and the network side. The MT part builds a prioritized list of associations between available RATs and active connections. The network part attempts to satisfy the top priorities of the user as long as there are adequate resources and the speed and direction of a user make the selection of a RAT meaningful (e.g., a WiFi should not serve a rapid moving user). This split of functionality allows alleviating the core network from certain processing load since some pre-processing is performed at the MT side. Moreover, several cases that can be evaluated and stopped in the MT will do so resulting in a significant reduction of unnecessary signaling exchanges. We expect that this does not place a serious burden in modern MTs that already have appreciable and all increasing processing power, memory storage capacity and battery longevity.

In the implementation of our mechanism we used Mobile IP (MIP), which is expected to be the de facto standard for mobility management. The mechanism can be used as-is in a loosed-coupled architecture but it can also be used in tight-coupled schemes as long as the appropriate extensions are provided to existing UMTS protocols [7]. The mechanism does not require complex calculations since initial processing is executed in the MT, while the final decision is taken inside the core network, allowing both sides (the user and the operator) to enforce their policies. The final decision is a trade-off between the user preferences, the terminal capabilities, the network load and the location and speed of the MT.

The remainder of the paper is organized as follows. Section 2 presents both parts of the proposed algorithm in detail. In section 3 we provide information about a prototype we have build using the Specification and Description Language SDL [8] and implementation details of our test-bed. Finally, section 5 concludes the paper.

## 2 Description of the Proposed Algorithm

The proposed distributed algorithm is split in two distinct and cooperating parts, the first running in the MT and the second in the core network. Its main output is the decision of the most suitable RAT during a new call establishment or upon a HO execution. The latter may be a horizontal (intra-RAT) HO, in which case the access

technology supporting a connection does not change, or a vertical (inter-RAT) HO, in which the supporting technology changes. The RAT selection is performed as a trade-off between the user preferences, the MT location and speed and the load of each RAT involved. The user preferences indicate which of the available RATs is preferable for each service provided. Although the algorithm has been designed to operate in environments where multiple heterogeneous radio access technologies exist, the subsequent section describes a case study with two access networks, namely UMTS and WLAN.

## 2.1 Algorithm Running in the Mobile Terminal (MT)

The MT part takes under consideration parameters located in the user profile and specifically the user's preferences related to the cost, the QoS and the battery duration. Also, the service requirements related to the Received Signal Strength (RSS) (error rates, delay, etc.) and the MT characteristics related to the power consumption of its radio interfaces. This algorithm is presented in Fig. 1. Its main purpose is to build prioritized lists of target RATs that are compatible with the user preferences. This is performed independently for each connection, either active or new, in order to provide for more flexibility and better user satisfaction. Five different stimuli may invoke the execution of the algorithm (cases (i)-(v) in Fig.1).

i) The MT has at least one active connection and a new RAT with strong enough radio signal is detected. This detection is based on the RAT dependent functionality of a new access point discovery. Next, the MT creates a two-dimensional priority list of  $N$  rows and  $M$  columns, where  $N$  is the number of active connections and  $M$  the number of alternative RATs. This priority list is filled in accordance with the user profile. As a next stage, the algorithm checks if all RATs involved in the previous step provide adequate radio link quality to support the requested services (e.g. through the radio signal strength – RSS – indicator). If a RAT does not fulfill these requirements, it is eliminated from the list. Next, the MT calculates the battery consumption for the simultaneous operation of all interfaces involved and modifies the priority list according to the importance the user gives to the battery duration. At this point, the list is sorted in descending order per line (i.e., connection) so that each line finally contains the alternative RATs for one particular connection, starting from the one that serves it best to the one that serves it worst. This list is sent to the core network along with a message requesting a HO.

ii) A forced HO command is received from the core network, concerning specific or all connections of the MT. This may be due to load balance purposes and is decided by the operator's network components. Thus, the time restrictions are not very tight, since the network operator should not initiate such a HO at the very last moment. Since the time restrictions are not very tight, this case is treated the same way as the previous one. As in the previous case, an effort is made to consider the user preferences, under more restrictive patterns this time, since some RATs may not be permitted if they are overloaded.

iii) A new call is initiated. The MT may be idle or have active connections. This time the algorithm considers the user preferences from the user profile and the battery condition and creates a list of prioritized RATs only for the new connection. Then it sends to the core network a message indicating the new call initiation along with this

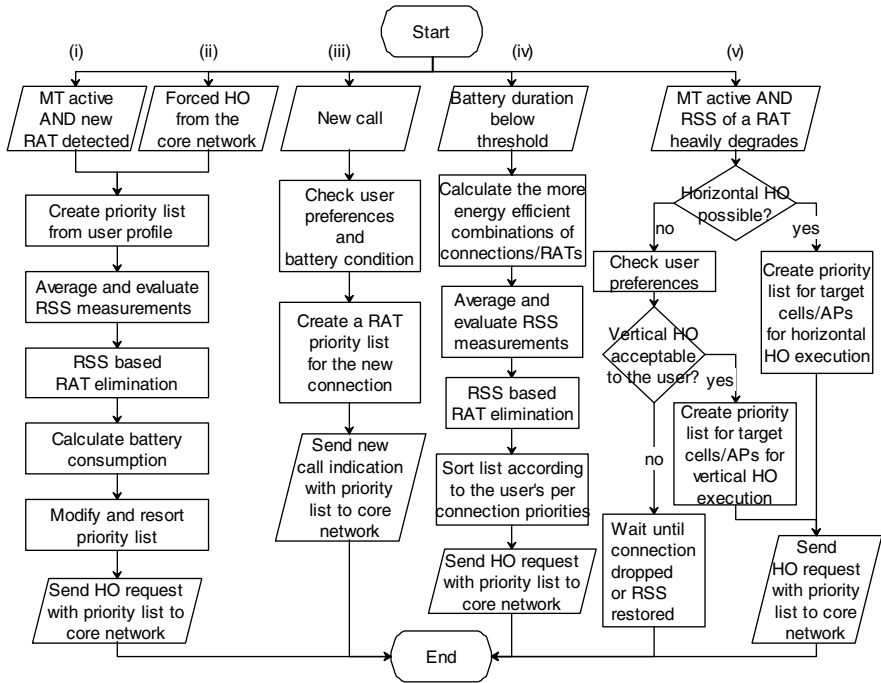


Fig. 1. Algorithm in the Mobile Terminal (MT)

priority list. The algorithm does not re-evaluate all active connections since this may only be necessary in case of low battery life. This case is taken care of from the next trigger.

iv) The remaining battery duration falls below a certain threshold, whose value is a user dependent factor and may be incorporated in the user profile. In this case, the MT finds which combinations of connections and RATs are the more energy efficient. This is feasible, since the MT is aware of the power consumption of each interface. Then, it eliminates all RATs with inadequate signal quality for each connection, so it rejects some of the combinations it just calculated. The remaining combinations are sorted according to the priority the user gives to each particular connection. This will allow the core network to reject, if required, the least important connections in case that the combinations sent cannot all be fulfilled.

v) The MT is active and the collected RSS measurements indicate a degrading signal from a RAT and an imminent HO. So, if a horizontal HO is feasible, a HO request is send to the core network along with a list of candidate cells/APs. The possibility of a vertical HO is considered only if a horizontal HO is not possible, i.e. there are no access points of this particular RAT in vicinity. Such a case could be when a WLAN connection has to be handed over, because the MT is moving out of the coverage area and there are no neighboring APs to be served by, and keep the connection within the WLAN. So, if it is acceptable, a similar procedure as before is followed. If not, there is no action and either the connection will be dropped as a result of radio

link degradation, or the signal may be restored (e.g. change of user direction movement) and the connection goes on.

## 2.2 Algorithm Running in the Core Network

The part of the algorithm running in the core network takes the final decision about the admittance of a new connection or the HO of an existing one ((i)-(iii) in Fig. 2). All these triggers are the messages received as the result of the corresponding algorithm at the MT, described in the previous subsection.

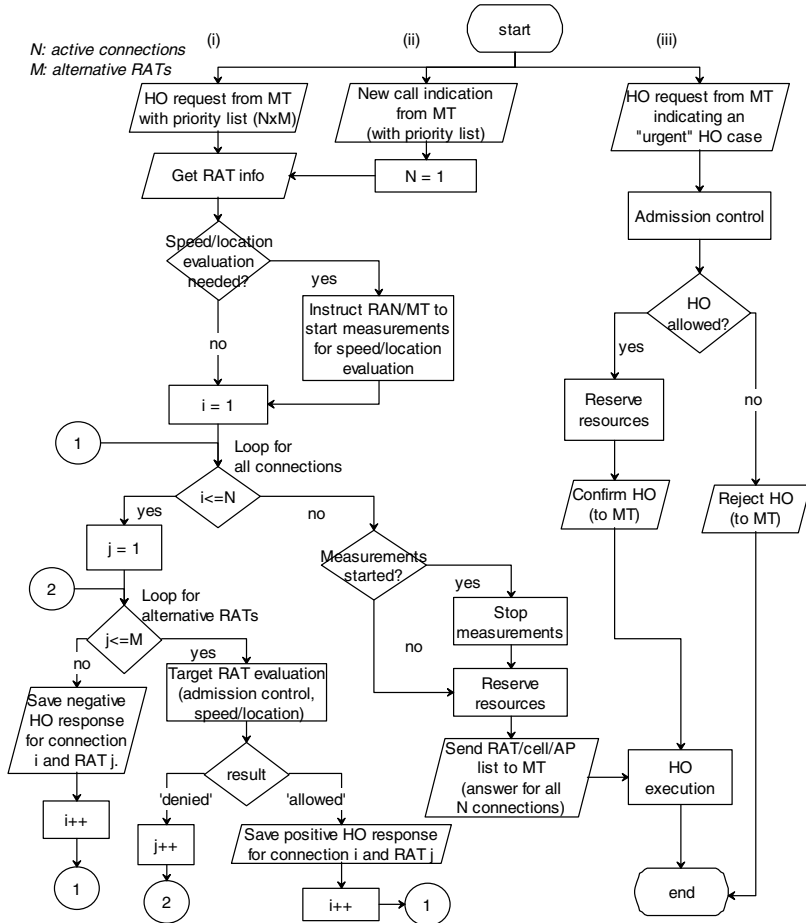


Fig. 2. Algorithm in the Core Network.

i) The first trigger is a HO request message sent by a MT to indicate the need to re-evaluate all its active connections. The priority list as formed by the MT algorithm is received as well. If the selection of a specific RAT implicates location and speed considerations, e.g. a limited range WLAN, then the procedure for their evaluation is

started. This is required since it is meaningless to HO to an AP if the MT's speed is very high and will leave the AP's coverage in few seconds, or if the MT is near the AP's border coverage and moving away from it [9].

At the next step a nested loop is started. The outer loop corresponds to the number of active connections. The inner loop is for the  $M$  alternative RATs of each connection. Thus, for each connection  $i$  ( $i = 1, \dots, N$ ) every alternative target RAT is evaluated, starting from the first one, since this was the decision taken by the MT after considering the user's preferences. This evaluation for the UMTS case means that the admission control algorithm is executed and the HO to this particular RAT is either 'allowed' or 'denied'. For the WLAN, apart from checking the load of the target AP, the speed and the location of the MT are estimated. When all connections are completed, the core network instructs the termination of the measurements for location and speed tracking. Then an answer to the HO request of the MT is sent back, along with the list of the final RAT choice for each connection.

ii) The second trigger is a new call indication from a MT. This is treated as the previous trigger, where all the connections are evaluated. The only difference, in this case, is that only the connection to be initiated is evaluated, therefore  $N=1$ .

iii) The last trigger is a HO request indicating an "urgent" HO. In this case, the admission control algorithm of the RAT selected by the algorithm in the MT is executed and decides upon the HO feasibility. We remind here that the MT has already checked the alternative RATs, and proposes only one of them according to coverage and user preferences. Then, if the HO is allowed, then the HO execution proceeds, else the MT is informed about the HO rejection.

### 3 SDL Prototype and Test-Bed Implementation

In Fig. 3 we present the top level of our SDL system. The purpose of using SDL prior to the implementation is to have a rapid prototype and check the validity of our design. The SDL system has been implemented in SDL using Telelogic's SDL-Suite tool [10]. After the specification completion the model was validated using SDT simulation tools, for different scenarios. The MT is connected to a UMTS and a WLAN for the exchange of data information. It is also connected directly with the module NDP (Network Decision Point) that is the network part of our mechanism. In this figure, the reader can also note that we use a block to model the Home Agent (HA) for the terminal, as well as a Correspondent Node (CN). The main target of this system is to demonstrate a functionality where the MT is communicating with the CN through UMTS or WLAN based on certain contextual information. This information (e.g., signal strength, speed and direction of the MT, level of congestion inside a network, level of battery in the MT) is provided during the guided simulation. What we have checked with this validation model is that our mechanism works smoothly during the establishment of a new connection or the execution of a vertical handover between networks in every test case scenario.

One main assumption we have made during the design and implementation of our mechanism is that the MT will always have access to the UMTS network. This is because a UMTS network is expected to be ubiquitous while other RAT technologies such as WLAN are expected to have discontinuous coverage areas. Thus, any

signaling exchange between the MT and the NDP is done through the UMTS network. Thus, in Fig. 3 there is only a channel between NDP and UMTS, but not between NDP and WLAN. Also, the MT is communicating directly with the NDP block for reasons of simplicity. During SDL simulation the MT link with the UMTS (and in this case with WLAN) is used just for the exchange of data between the MT and the CN. In our architecture, we have assumed the use of a loose coupling scheme between the UMTS and the WLAN networks. This means that the NDP entity can be placed on top of the GGSN (gateway GPRS support node) network component of UMTS. Although this implies a higher delay for the exchange of signaling between the MT Decision Point (MTDP) and NDP entities, it requires no modification at all on existing standards.

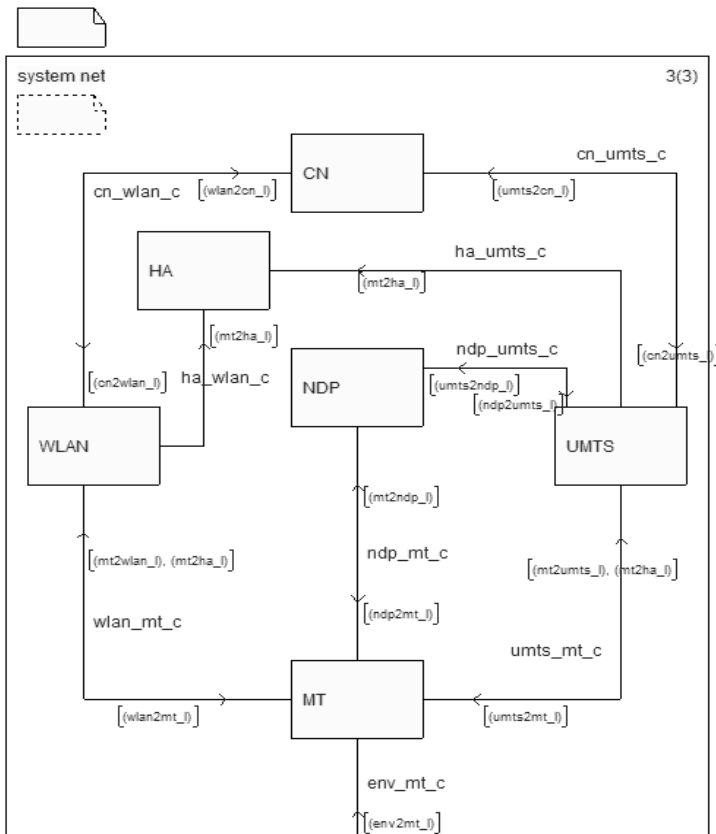


Fig. 3. Logical Architecture in SDL

Moreover, note that the HA has links with both the UMTS and the WLAN, since after the MT selects an interface for its connection and finalizes the association with the respective network it requires the appropriate message exchange for receiving a new IP address and to notify the HA about this.

Next, we discuss the implementation details of our mechanism in an appropriately configured test-bed and present the scenarios we have tested. More specifically, and due to the lack of real UMTS equipment, we have deployed two IEEE 802.11 Access Points (APs). One of them plays the role of the UMTS network, transferring all the signaling between the MT and the core network. Our mechanism comprises the two entities, one located in the terminal and the other in the network. The two entities exchange messages in order to decide the access network through which an active or new connection will be served. These entities take into consideration several pieces of information, such as the monetary cost, the network load, the signal strength, the battery level etc. Some of these characteristics are collected from the operating system of the devices or are hard-coded due to lack of appropriate equipment.

The whole test-bed is based on Linux. The distribution used in the APs is OpenWrt [11], a Linux distribution appropriately configured for use in embedded devices, enhanced with IPv6 functionality. This distribution was chosen because it supports the necessary flexibility for routing and an easy way to reach network statistics through the SNMP protocol. The Network Mobility / NEMO Platform for Linux (NEMO/NEPL) implementation [12] is used for mobility support, since at the time of the deployment of our test-bed it was the most stable open source implementation. The NEMO protocol is an extension of Mobile IP that supports mobility of entire networks. Network mobility arises when entire networks are changing their point of attachment with respect to the Internet topology. A mobile network is connected to the Internet via one or more mobile routers (MR). Nodes behind the MR are referred to as mobile network nodes. Thus, NEMO supports moving nodes inside moving networks (e.g., mobile users inside a train). NEMO/NEPL is based on the Mobile IPv6 for Linux (MIPL) stack that has been developed by the University of Helsinki [13]. Currently, the Universal Playground for IPv6 (USAGI project – [14]) task group provides support for MIPL.

In this implementation we have added the required functionality to support Multiple Care of Addresses (CoA) [15]. Multiple CoAs are required since the mobile terminal may have more than one open network interface and thus, requires more than one IP address in order to be reachable on every interface. The standard Mobile IP allows the registration of a single primary care of address with a HA. To have the assignment of multiple CoAs feasible there is a need for an additional identifier called Binding IDentification number (BID). For the needs of the test-bed we used a large IPv6 address space (/40) divided into several subnets. Appropriate routing rules have been setup between the nodes. The overall topology is depicted in Fig. 4. Two WiFi APs act as the different RATs. The direct Ethernet link with one of the APs acts as the ubiquitous UMTS connection transferring all messages between the MTDP and the NDP. The two APs are connected, through an access router, to the Internet and a corresponding node. In this router, the HA and the NDP are located. Moreover, the software for the NEMO mobile router was collocated with the mobile terminal since in our test-bed there was no need to demonstrate moving networks. In the mobile terminal we have also placed the MTDP entity.

As far as routing is concerned we have used the Quagga routing suite [16]. More specifically, we used the OSPF (Open Shortest Path First) protocol that supports routing in IPv6 networks [17]. The choice for dynamic routing, instead of a more static solution, enables the test-bed to be reconfigured easily with additional access technologies and networks.



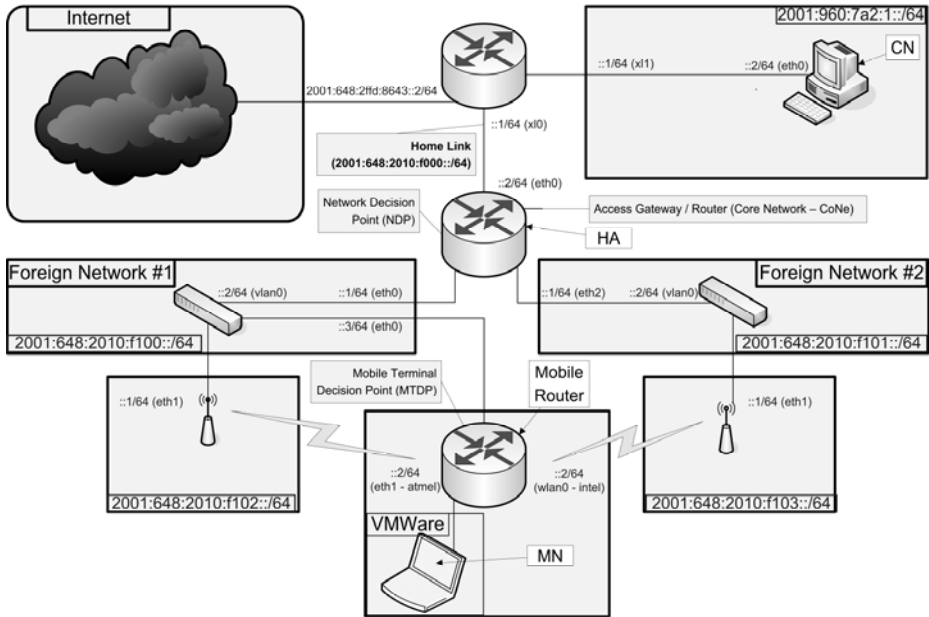


Fig. 4. Test-bed topology

For the automatic IPv6 addressing of mobile nodes, as they are connected to the APs, the Linux IPv6 Router Advertisement Daemon (radvd) was used [18]. The Router Advertisement Daemon advertises mobility options for the Home Link, such as HA availability, preference value and lifetime as well as Mobile Router support capabilities, at the NDP network and the wireless prefixes at the AP attachment points. Its operation is described in [19] and we have used the Linux implementation as described in [18].

Using the aforementioned software, we had to extend the basic NEMO functionality by:

1. defining new headers and messages for the exchange of information between the new entities (MTDP and NDP),
2. extending NEMO with support for multiple CoAs (note that NEMO is still under development and even now the MCoA feature is not fully supported),
3. defining a syntax and grammar used by the policy rules for selecting a RAT,
4. implementing both the MTDP and the NDP.

For the exchange of messages between MTDP and NDP there was a need to build a direct interface between them, however in practice this proved to be difficult. The reason was that before submitting policy messages, between the MTDP and the NDP, the network link must have been established. Delaying this necessary step, while trying to take a decision for a specific RAT and the added difficulty in controlling the whole NEMO stack with the Multiple CoAs extensions resulted in delays for the execution of handovers. The next best choice was to use existing messages and interfaces to piggyback the required information for our mechanism to work. This way

there is a guarantee that the required information will be exchanged safely between the MTDP and the NDP. Thus, the information needed by the new messages is being transferred as extensions to existing mobility messages (e.g., as Mobility Options in Binding Update and Acknowledgment messages exchanged between the HA and the MT).

The message used for this task is the binding update message that was exchanged every time there was a change on the operating conditions of the mobile terminal as for example when:

- i) discovering a new access network,
- ii) the signal strength of the current AP is very low,
- iii) battery level dropped under a certain threshold.

In the mobile terminal the user preferences are stored in a file together with additional information required for mobility management. The policies for each application can be in the form of a list. For example (WLAN, UMTS) means that for a specific application the first choice is to route the data through a WLAN and if this is not possible (e.g., no WLAN is available in the area, or the WLAN is congested) use a UMTS network as an alternative. In order to simplify this issue for the less experienced users, we can define generic pre-defined lists that will be easily understood by the users.

The test-bed was used successfully in a number of scenarios, which demonstrated that not only the key characteristics of our mechanism actually work, but also that it is feasible to treat each mobile terminal connection separately from the others. This of course is a very important characteristic that enables users to be quite flexible when selecting services from one or many RATs and/or operators. The program we used for recording messages and data traffic was tcpdump that is a well-known and established program [20]. Since tcpdump can be executed in the Linux platform we were able to collect information from several points of the network and even from the APs.

The analysis of the messages was performed using the graphical environment of Wireshark [21]. Using this program we were able to analyze the record files produced by tcpdump and monitor the exact times the messages were exchanged between the entities as well as their payloads. For example, in Fig. 5 we present the data collected by Wireshark when performing a handover while having a ping command active. As shown in the figure, we can monitor all the fields in all layers (e.g., see IPv6 addresses, port numbers), as well as the time these messages were exchanged.

Although in our cases we used real traffic through the Internet, the disruption time during a handover was minimal (in the order of magnitude of msecs). This was also the case when using video streams. The reason for this was that the HA in the testbed is located close to the MT and there were not many terminals requesting responses from the NDP. In a real situation it is expected that the system would present typical disruption times of Mobile IP.

Finally, our implementation has integrated the RAT selection mechanism with an authentication procedure. The authentication procedure includes the exchange of keys with the APs using the EAP-TTLS protocol [22]. These keys are transferred

No. -	Time	Source	Destination	Protocol	Info
11	5.005405	2001:648:2010:f101::1	2001:960:7a2::217:f2ff:fec5:d79a	ICMPv6	Echo reply
12	5.858554	2001:960:7a2::217:f2ff:	2001:648:2010:f101::1	ICMPv6	Echo request
14	6.005033	2001:648:2010:f101::1	2001:960:7a2::217:f2ff:fec5:d79a	ICMPv6	Echo reply
15	6.858639	2001:960:7a2::217:f2ff:	2001:648:2010:f101::1	ICMPv6	Echo request
16	7.005916	2001:648:2010:f101::1	2001:960:7a2::217:f2ff:fec5:d79a	ICMPv6	Echo reply
18	8.725319	2001:960:7a2::217:f2ff:	2001:960:7a2::1	ICMPv6	Neighbor solicitation
19	8.725741	2001:960:7a2::1	2001:960:7a2::217:f2ff:fec5:d79a	ICMPv6	Neighbor advertisement
20	9.795017	2001:960:7a2::217:f2ff:	2001:648:2010:f100::3	ICMPv6	Echo request
21	9.848835	2001:648:2010:f100::3	2001:960:7a2::217:f2ff:fec5:d79a	ICMPv6	Echo reply
23	10.795084	2001:960:7a2::217:f2ff:	2001:648:2010:f100::3	ICMPv6	Echo request
24	10.942110	2001:648:2010:f100::3	2001:960:7a2::217:f2ff:fec5:d79a	ICMPv6	Echo reply
25	11.795104	2001:960:7a2::217:f2ff:	2001:648:2010:f100::3	ICMPv6	Echo request
26	11.841437	2001:648:2010:f100::3	2001:960:7a2::217:f2ff:fec5:d79a	ICMPv6	Echo reply

```

▶ Frame 14 (70 bytes on wire, 70 bytes captured)
  ▶ Ethernet II, Src: 3com_ac:49:27 (00:01:02:ac:49:27), Dst: AppleCom_c5:d7:9a (00:17:f2:c5:d7:9a)
  ▶ Internet Protocol Version 6
  ▼ Internet Control Message Protocol v6
    Type: 129 (Echo reply)
    Code: 0
    Checksum: 0x7b79 [correct]
    ID: 0x02f5
    Sequence: 0x0003
    Data (8 bytes)

```

**Fig. 5.** Example of an ICMPv6 handover

encrypted to a RADIUS [23] server located in the access router. Using this information it will accept or reject the request for a connection.

## 4 Conclusions

In this paper we have presented a new mechanism for selecting RATs when moving in a heterogeneous environment. The mechanism takes into consideration a number of parameters such as user preferences, signal strength, battery level, network congestion, speed and direction of a terminal, etc. Its main goal is to satisfy the preferences of the users and not merely to load balance the traffic between the available networks. To achieve this, the mechanism has two decision points. The first one is on the mobile terminal and builds a prioritized list of RATs for each one of its connections, based on user preferences. The second decision point is located on the network and checks mainly if there are resources to satisfy user's requests.

The paper also describes an SDL prototype we have built to test the correctness of our mechanism, as well as the implementation of a test-bed that was based on open source software. This test-bed proved the feasibility of our proposal and additionally demonstrated a differentiated treatment of active connections (i.e., having some connections through one RAT while others can be served by a different RAT). It also proved that the ability to execute vertical handovers between RATs is a viable and easy to implement approach. We believe that such a capability will be the standard choice in future networks.

## References

1. McNair, J., Zhu, F.: Vertical Handoffs in Fourth-Generation Multi-network Environments. *IEEE Wireless Communications* 11, 8–15 (2004)
2. Hou, J., O'Brien, D.C.: Vertical Handover-Decision-Making Algorithm using Fuzzy Logic for the Integrated Radio-and-OW System. *IEEE Transactions on Wireless Communications* 5(1), 176–185 (2006)
3. Giupponi, L., Augusti, R., Pérez-Romero, J., Sallent, O.: A novel Joint Radio Resource Management Approach with Reinforcement Learning Mechanisms. In: 24th IEEE International Performance Computing & Communications Conference, IPCCC 2005, pp. 621–626 (2005)
4. Murray, K., Mathur, R., Pesch, D.: Intelligent Access and Mobility Management in Heterogeneous Wireless Networks using Policy. In: The 1st International Workshop in Information and Communication Technology, pp. 181–186 (2003)
5. Zhuang, W., Gan, Y.-S., Loh, K.-J., Chua, K.-C.: Policy-based QoS Management Architecture in an Integrated UMTS and WLAN Environment. *IEEE Communications Magazine* 41(11), 118–125 (2003)
6. Song, W., Zhuang, W., Cheng, Y.: Load Balancing for Cellular/WLAN Integrated Networks. *IEEE Network* 21(1), 27–33 (2007)
7. Lampropoulos, G., Passas, N., Kaloxylou, A., Merakos, L.: A Flexible UMTS/WLAN Architecture for Improved Network Performance. *Springer Wireless Personal Communications Journal*, special issue on Seamless Handover in Next Generation Wireless Mobile Networks 43(3), 889–906 (2007)
8. ITU-T. International Telecommunication Union, Specification and description language (SDL), Recommendation Z.100, ITU-T Study Group 17 (2009), <http://www.itu.int/ITU-T/studygroups/com17/languages/Z100.pdf>
9. 3GPP TS 25.305, Stage 2 functional specification of User Equipment (UE) positioning in UTRAN, Release 7 (2006)
10. Telelogic SDL Suite, <http://www.telelogic.com/products/sdl/index.cfm>
11. OpenWRT Linux distribution for embedded devices, <http://openwrt.org>
12. NEPL (NEMO Platform for Linux), <http://www.nautilus6.org/doc/nepl-howto>
13. MIPL – Mobile IPv6 for Linux, <http://www.mobile-ipv6.org>
14. USAGI Project - Linux IPv6 Development Project, <http://www.linux-ipv6.org>
15. Wakikawa, R., Ernst, T., Nagami, K., Devarapalli, V.: Multiple Care-of Address Registration, IETF Internet Draft Version 06 (August 2008), <http://www.ietf.org/internet-drafts/draft-ietf-monami6-multiplecoa-06.txt>
16. Quagga Routing Software Suite, GPL licensed IPv4/IPv6 routing software, <http://www.quagga.net>
17. Coltun, R., Ferguson, D., Moy, J.: OSPF for IPv6, IETF RFC 2740 (December 1999)
18. Linux IPv6 Router Advertisement Daemon (radvd), <http://www.litech.org/radvd>
19. Narten, T., Nordmark, E., Simpson, W.: Neighbour Discovery for IP Version 6 (IPv6), IETF RFC 2461 (December 1998)
20. tcpdump / libpcap, <http://www.tcpdump.org>
21. Wireshark network protocol analyzer, <http://www.wireshark.org>
22. Funk, P. and Blake-Wilson, S.: Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TLSv0), RFC 5281 (2008), <http://tools.ietf.org/html/rfc5281> (accessed 16/2/2009)
23. Rigney, C., Willens, S., Rubens, A., Simpson, W.: Remote Authentication Dial In User Service (RADIUS), RFC 2865 (2000), <http://www.ietf.org/rfc/rfc2865.txt>