Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2009, Vol. 1, No. 2

Network Selection Algorithm for Heterogeneous Wireless Networks: from Design to Implementation

Alexandros Kaloxylos

Dept. of Telecommunications Science and Technology, University of Peloponnese
Karaiskaki Area - St George Park, Tripoli 22100, Greece
E-mail: kaloxyl@uop.gr

Ioannis Modeas (corresponding author)

Dept. of Informatics and Telecommunications, University of Athens
Panepistimiopolis, Ilissia, Athens 15784, Greece
E-mail: imodeas@di.uoa.gr

Fotos Georgiadis

Dept. of Informatics and Telecommunications, University of Athens
Panepistimiopolis, Ilissia, Athens 15784, Greece
E-mail: fotos@di.uoa.gr

Nikos Passas

Dept. of Informatics and Telecommunications, University of Athens
Panepistimiopolis, Ilissia, Athens 15784, Greece
E-mail: passas@di.uoa.gr

Abstract

Heterogeneous networks allow mobile terminals to take advantage of complementary radio technologies for their concurrent connections. In this paper we propose a mechanism for automated radio access network selection with several novelties: It enables terminals to build prioritized lists of target access networks independently for each of their active connections. It aims to satisfy user preferences. Lastly, it operates with two decision-making points (mobile terminal and core network), splitting the complexity of the overall process. After discussing the functionality of the proposed mechanism, we present its formal specification in SDL. Finally, a test-bed implementation comprising two access networks is presented. It uses open source tools and it is based on Mobile IP, showing the feasibility of our proposal.

Keywords: access network selection, heterogeneous networks, test-bed implementation

1. Introduction

In our days, an all increasing number of mobile terminals are equipped with more than one radio interface and abilities similar to a small PC. Also, their users are becoming familiarized with the capabilities of their equipment and can select the best means of communication in respect of charging, supported Quality of Service (QoS), etc. Currently, this selection is performed manually, and a user may choose between different access networks available, e.g., a UMTS network for the phone calls and a WiFi hot-spot for web browsing and email. At the same time, mobile operators are attempting to increase their overall resources by deploying additional radio access technologies (e.g., WiFi, WiMAX). 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, fuzzy logic, neural networks and policy-based schemes are all candidates to tackle the issue.

Cost functions suggest the use of numerical functions to calculate the cost of alternative RATs, based on several weighted metrics and parameters, e.g., [1]. Such functions provide a simple way to make a HO decision and a comparative measure on the suitability of one RAT against others. It should be noted that there is an issue to consider when mixing different parameters of different measurement units in a single equation (e.g., monetary cost with signal strength and QoS type). Furthermore, there are parameters that cannot be directly measured by a certain unit (e.g., user preferences, security) and it is not always clear in which way they can be incorporated in such mechanisms.

Fuzzy logic, e.g., [2] and neural networks, e.g., [3], can be used as tools in a RAT selection algorithm. They can both minimize the number of unnecessary HOs and maximize the percentage of satisfied users. These solutions take into consideration both user preferences as well as operators’ policies. Their disadvantage is that they increase the overall system complexity and in the case of neural networks a pre-training session of the system is required.

Policy-based schemes, e.g., [4]-[6], are based in a set of rules that manage the execution of appropriate actions based on specific events. When these rules are simple, they provide a fast and easily implemented solution at the expense of non-optimal resource utilization. To cope with the later, more sophisticated policies can be introduced, but they increase the system complexity. Special care is needed in order to avoid conflicts between different policies, especially when these policies reside in different network nodes. These schemes may be combined with one of the previous mechanisms in order to make the final decision. A point to consider when using policy based schemes is that strict rules do not provide for scalability and flexibility to cope with all the contradicting parameters involved.

Almost all of the aforementioned 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, their interests and preferences 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 the aforementioned proposals usually consider all the 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 de facto 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. The first runs at the MT side and the second at 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 WLAN covering a small area 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 can be evaluated and stopped in the MT, resulting in a significant reduction of unnecessary signalling exchange. 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.

The first step towards the implementation was to formally specify a prototype using Specification and Description Language (SDL) [8]. Based to this prototype, we proceed to the implementation of our mechanism using 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 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.

This paper proposes a network access selection mechanism and its implementation. In order to take the final decision, both the MT and the core network need to have certain information, such as signal strength measurements, network load, user preferences, etc. The underlying mechanism to provide this information, even though a very interesting topic, it is out of the scope of this paper. A recent and very promising effort in this latest field is IEEE 802.21 Media Independent Handover (MIH) concept. It aims to help the decision making process, by using standard functions to gather necessary network parameters. Nevertheless, it does not define any network selection procedure [9]. From this point of view it is a complementary framework to the work presented here. It is in our interests to investigate how such a merge is feasible.

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 implementation we have build using the Specification and Description Language (SDL) [8]. Finally, section 4 presents implementation details of our test-bed, while 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.

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 Figure 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. These are indicated as (i), (ii), (iii), (iv) and (v) in Figure 1 and are described hereafter.

(i) The MT has at least one active connection and a new RAT with strong enough radio signal is detected. Then, 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. After that, the algorithm checks if all RATs involved in the previous step provide adequate radio link quality to support the requested services. This is accomplished by evaluating all available RSS measurements collected by the MT. If a RAT does not fulfil 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. All the processing required here needs some time to be accomplished. We assume that there is enough time for this, since the triggering event is the discovery of a new RAT and not the degradation of the RSS, the algorithm tries to see if it can better satisfy the user by providing access to a more favourite RAT.

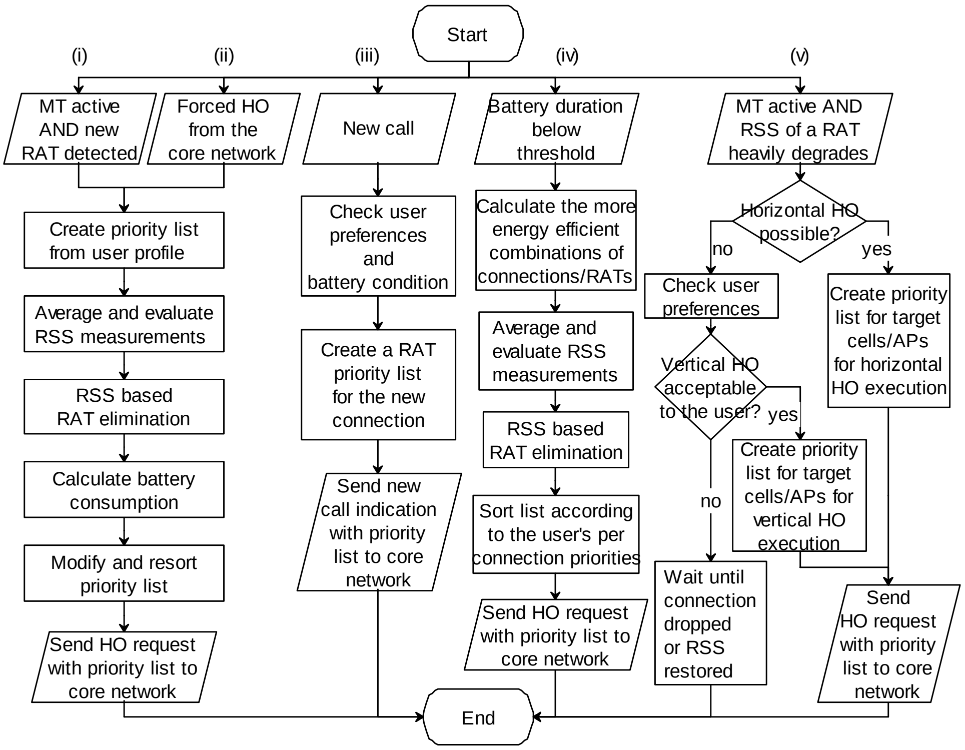

Start
(i)
(ii)
(iii)
(iv)
(v)
MT active AND new RAT detected
Forced HO from the core network
New call
Battery duration below threshold
MT active AND RSS of a RAT heavily degrades
Create priority list from user profile
Average and evaluate RSS measurements
RSS based RAT elimination
Calculate battery consumption
Modify and resort priority list
Send HO request with priority list to core network
Check user preferences and battery condition
Create a RAT priority list for the new connection
Send new call indication with priority list to core network
Calculate the more energy efficient combinations of connections/RATs
Average and evaluate RSS measurements
RSS based RAT elimination
Sort list according to the user's per connection priorities
Send HO request with priority list to core network
Horizontal HO possible?
no
yes
Check user preferences
Vertical HO acceptable to the user?
yes
no
Create priority list for target cells/APs for vertical HO execution
Create priority list for target cells/APs for horizontal HO execution
Wait until connection dropped or RSS restored
Send HO request with priority list to core network
End

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

(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. This case is treated the same way as the previous one, since the time restrictions are not very tight again. The network operator is responsible not to trigger such HOs at the very last moment. 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 priority list. The algorithm does not re-evaluate all active connections since this may only be necessary in case of low battery life. But 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. When this duration falls below a time interval, a user may be willing to sacrifice some QoS or certain connections in order to remain reachable. Another user not interested in such proactive behaviour may have this threshold value equal to zero. This is clearly a per-user setting. The remaining battery life is heavily dependent on the active radio interfaces of the multi-mode MT. When this falls under certain time duration, a rearrangement of the active connections is performed, in order to maximize battery life. First 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. In this case, the time constraints are tight. The highest priority is on the continuation of the call rather on fine-tuning the decision according to all parameters. Here, we consider that a horizontal HO is preferable to a vertical one. This is due to the fact that the used RAT is already known to be acceptable to the user and a horizontal HO has usually less latency. This is especially true in a loose coupling scenario. 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. 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 neighbouring APs to be served by. A vertical HO will be allowed, only if it is acceptable by the user preferences. For example, the user may not be willing to pay a higher price for UMTS connectivity for an FTP download. 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.

2.1 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. It is invoked under three triggers, (i), (ii) and (iii), shown in Figure 2. All these triggers are the messages received as the result of the corresponding algorithm at the MT, described in the previous subsection.

(i) The first trigger is a HO request message from a MT indicating the need to re-evaluate all its active connections. It comes along the priority list formed by the MT algorithm. Then, information related to all target RATs, such as coverage and load, is taken into account. 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. There are a lot of methods for speed and location evaluation. Some are proposed by 3GPP [10], where the MT's location and optionally velocity estimation is achieved by measuring the radio signals and using methods such as OTDOA (Observed Time Difference Of Arrival). The explanation of the particular method of velocity and location calculation is out of the scope of this paper. At the next step a nested loop is started. The outer one corresponds to the number N of the active connections. So, each connection is evaluated separately. The inner loop is for the M alternative RATs of each connection. Thus, for each connection i (i = 1,…,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. This is feasible after collecting all measurements made up to now for this specific reason. If the MT's speed and location are acceptable, the HO is 'allowed' for this specific RAT and the decision taken is positive. Then, the algorithm goes on with the next connection and the same procedure is repeated for every RAT j. 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.

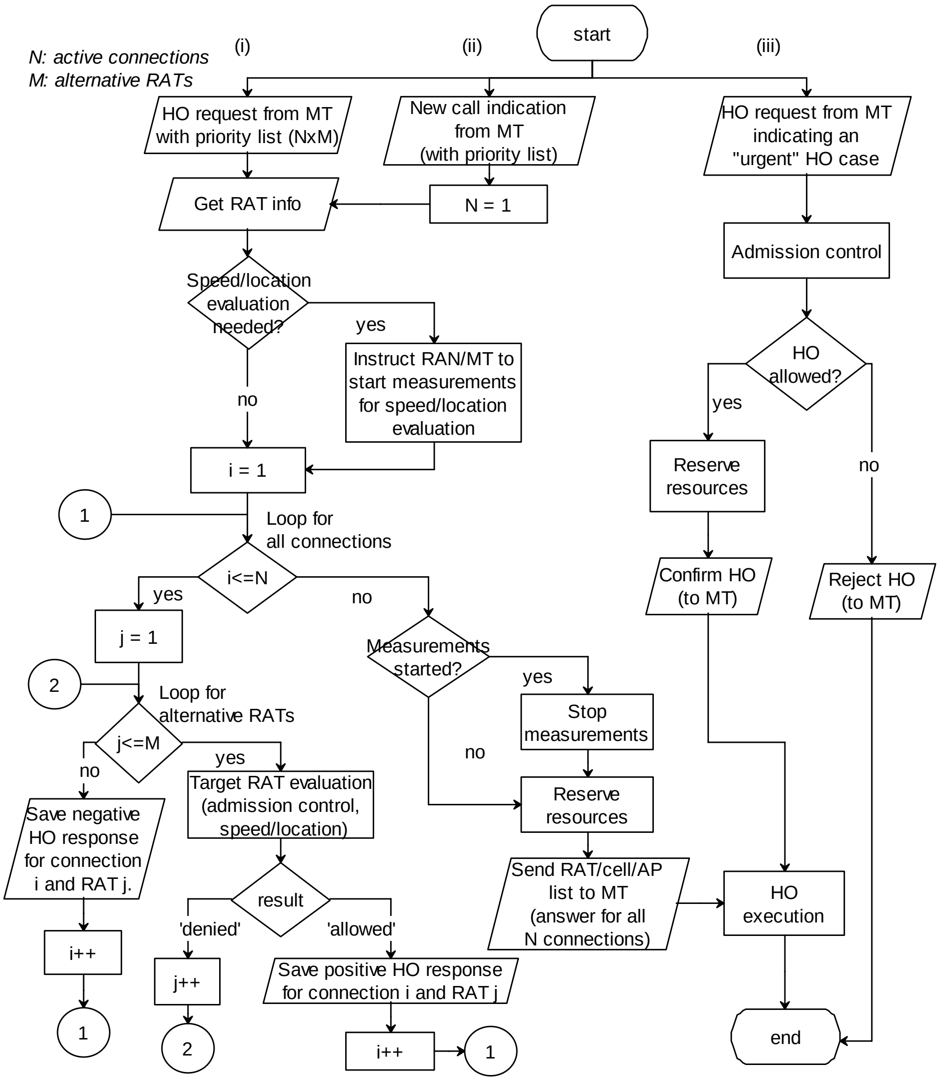

start
N: active connections
M: alternative RATs
(i)
(ii)
(iii)
HO request from MT with priority list (NxM)
New call indication from MT (with priority list)
HO request from MT indicating an "urgent" HO case
Get RAT info
N = 1
Admission control
Speed/location evaluation needed?
yes
Instruct RAN/MT to start measurements for speed/location evaluation
no
HO allowed?
yes
no
Reserve resources
i = 1
1
Loop for all connections
i<=N
yes
no
Confirm HO (to MT)
Reject HO (to MT)
j = 1
Measurements started?
2
Loop for alternative RATs
yes
Stop measurements
j<=M
no
yes
no
Target RAT evaluation (admission control, speed/location)
Save negative HO response for connection i and RAT j.
Reserve resources
Send RAT/cell/AP list to MT (answer for all N connections)
result
'denied'
'allowed'
HO execution
i++
j++
Save positive HO response for connection i and RAT j
1
2
end
i++
1

Figure 2. Algorithm in the core network.

(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

In this section we describe the mechanism’s specification in SDL. The purpose of using SDL prior to the implementation is to have a prototype and check the validity of our design. The SDL system has been built using Telelogic's SDL-Suite tool [11]. As a test case we considered a heterogeneous network comprising two alternative RATs: UMTS having ubiquitous coverage and WLAN having a partial coverage area. Loose coupling and Mobile IP are also considered for both the formal specification and the implementation.

In Figure 3 we present the top level of our SDL system. The MT is connected to UMTS and 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 signalling exchange between the MT and the NDP is done through the UMTS network. In Figure 3, the MT is communicating directly with the NDP block for reasons of simplicity. During SDL simulation the 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 signalling between the MT Decision Point (MTDP) and NDP entities, it requires no modification at all on existing standards.

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.

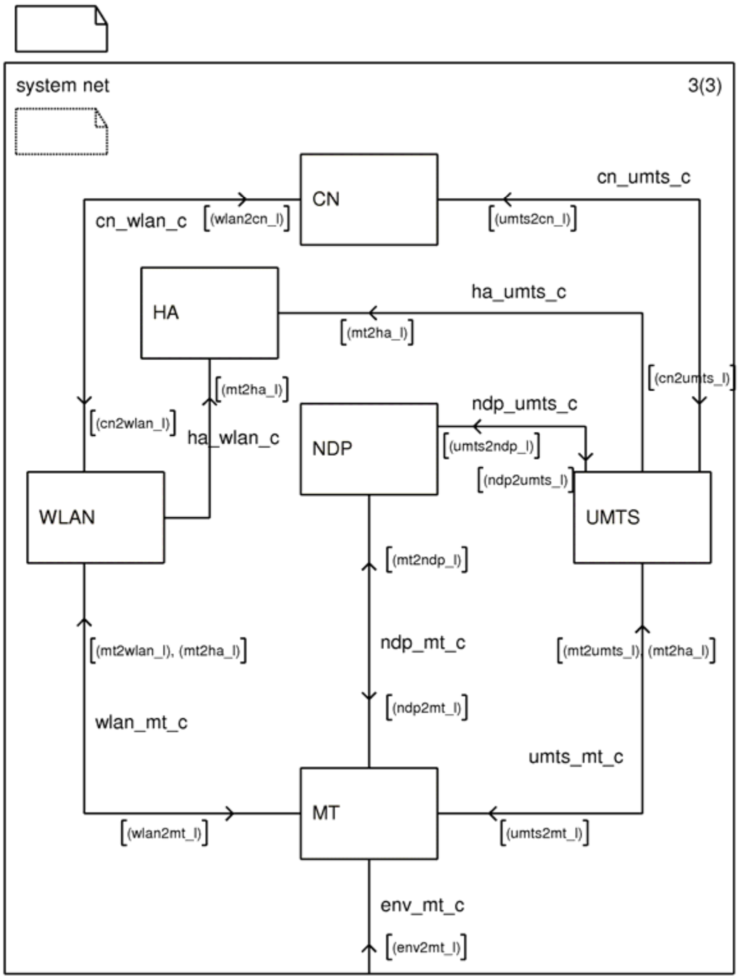


Figure 3. Logical Architecture in SDL.

To have a complete picture of our SDL validation model we illustrate in Figure 4 the analysis of the MT SDL block as well. As can be seen, it has been separated in four different processes which are:

i) appl: This process generates data packets at regular intervals after the establishment of a new connection between the MT and the CN and vice versa (i.e., incoming and outgoing communication). It also receives data originated from the CN.

ii) MTDP: This process is the MT part of our proposed mechanism and communicates with its counterpart entity in the network (i.e., NDP). This process is provided with new events that affect the RAT selection process for serving a specific application. For example, such events can be the deterioration of the signal strength, the fall of the battery duration below a certain threshold, the identification of a new network in the area (in this case the WLAN). All of these potential events were described in the previous section.

iii) mip: This process corresponds to the IP/MIP protocols where packets originated/targeted from/to the application layer are encapsulated/decapsulated accordingly to the network used for a specific application. The process also contains some of the primitives to support the execution of the MIP protocol, such as the binding update process.

iv) IWL (Inter-Working Layer): This is an intermediate layer that acts as the multiplexing and de-multiplexing unit between the IP layer and the networking interfaces. In other words it splits outgoing traffic through the appropriate network interface (i.e., WLAN or UMTS) and performs the opposite task for the incoming traffic.

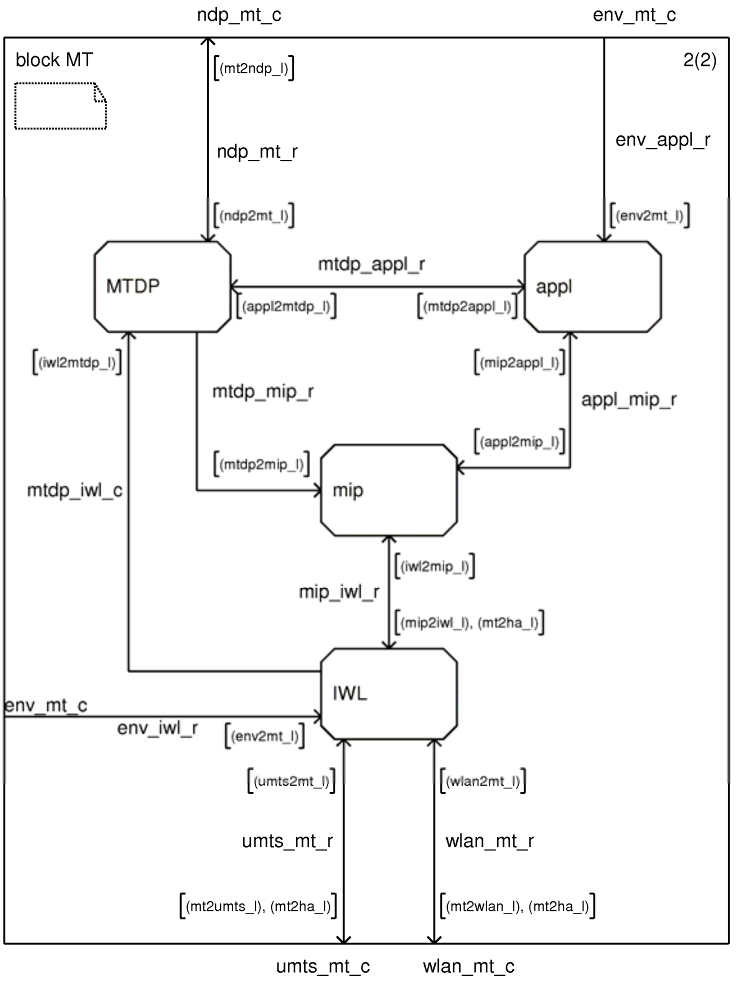


Figure 4. Analysis of the MT Block.

The SDL implementation as well as the extensive guided simulations assisted us not only in improving the algorithm and correcting logical bugs, but also in making specific implementation choices on how to represent the profile of the user, the prioritized list of associated connections to RATs etc. Also, after successfully compiled and built the SDL prototype, several Message Sequence Charts (MSCs) have been generated for a multitude of different scenarios. These MSCs show all messages' exchange between processes, along with all related parameters. This paper does not incorporate any MSCs from a full scenario, because of their significant length. The interested reader can download such a full MSC in [12]. In the following paragraphs we present a part of this scenario, using partial MSCs generated from SDT Simulator (Figures 5 and 6). For readability purposes, in these figures all parameters' numerical values are replaced with some textual info.

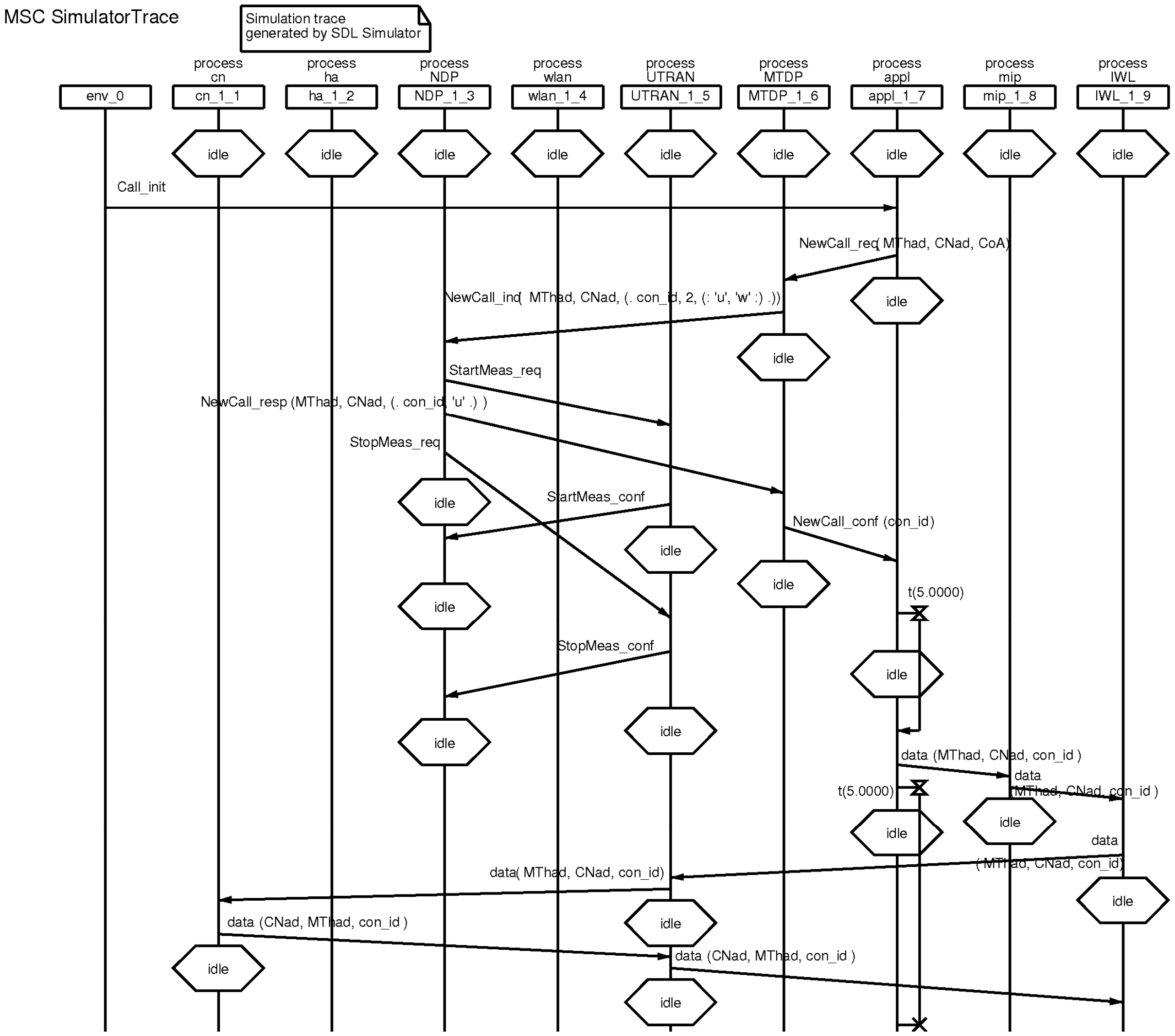


Figure 5. Message Sequence Chart for a new connection.

Figure 5 is the initiation of a new call from the MT to the CN, through UMTS, until the establishment of the connection and the data flow. To initiate a new call, a Call_init message is sent from the environment process "env", to the MT block (process appl). The latter reacts with a NewCall_req message towards MTDP process, indicating in the parameters that the call is from the MT (with home address MThad) towards the CN (with address CNad). The algorithm running at the MT has already started at this point. Then, MTDP sends to NDP a NewCall_ind message, along with the priority list for that particular connection (con_id), that is the outcome of the algorithm running at the MT block. In this case, the user's first preference indicates two target RATs, UMTS ('u') as a first preferable option and WLAN ('w') as a second one. The algorithm running at the MT terminates here. The reception of the NewCall_ind message from the NDP is the trigger that activates the algorithm running in the core network. Then, NDP instructs the UTRAN to start the procedure of the measurements to support speed estimation of the user. After evaluating all required parameters, NDP responds to MTDP with a NewCall_resp message indicating that UMTS ('u') is the selected RAT for connection con_id. This is the point where the algorithm running at the core network terminates. Then, MTDP responds with a NewCall_conf message to the appl process and then data (data messages) start flowing between the MT block and the CN, through UTRAN process, since the selected RAT was UMTS.

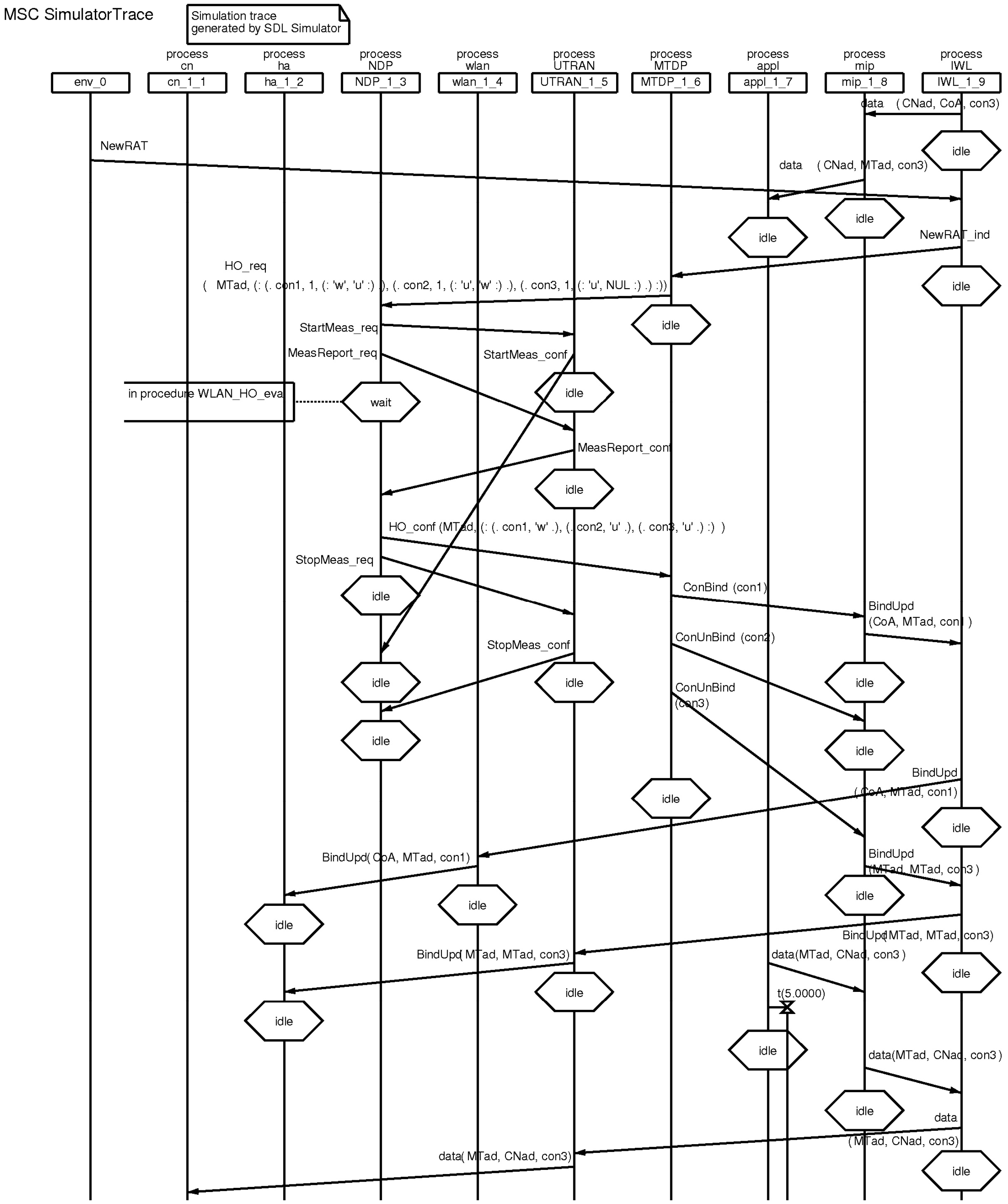


Figure 6. Message Sequence Chart for a vertical HO.

Figure 6 is a part of a SDT Simulator MSC showing a vertical HO case. Before the “NewRAT” message from the env process, there are three active connections (con1, con2, con3), all through UMTS (process UTRAN). The NewRAT message emulates a lower layer trigger indicating that a new RAT is available. This message is received by the MT block and initiates the algorithm running there and more specifically at the process MTDP. The latter sends a HO_req message to NDP, with the priority list as formed by the algorithm. Since there are three active connections involved, the priority list comprises the user preferences for all these three connections. For the first one 'con1', ('w','u') indicates the user preference for WLAN over UMTS. For the second one 'con2', ('u','w') indicates the opposite. For the last one 'con3', ('u',NUL) indicates only UMTS as acceptable. This HO_req message initiates the part of the algorithm running at the core network (process NDP). Then, StartMeas_req from NDP initiates the procedure to support user's velocity estimation, since the limited coverage WLAN is involved. The part of the algorithm at the core network terminates after sending the final decision at the form of a HO_conf message to the MTDP, indicating for each one of three active connections, the destination RAT (con1-'w', con2-'u', con3-'u'). After that, the HO execution starts, with the necessary binding updates messages and finally the data exchange over the new selected RATs.

In this section we presented several steps taken related to the specification phase. All these were necessary in order to demonstrate the right functionality of the algorithm in all test cases, so as to proceed to the implementation phase. The details of the test-bed implementation are presented in the following section.

4. The test-bed implementation

In this section, we discuss the implementation details of our mechanism in an appropriately configured test-bed and present the scenarios we have tested. More specifically, 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 signalling between the MT and the core network. Our mechanism comprises the two entities, one located in the terminal and the other in the network. In the SDL specification these are the NDP and MTDP processes. 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 (e.g., speed and location determination using OTDOA techniques require a number of UMTS Node B's, monetary cost is fluctuant and based on providers' economic policies).

The whole test-bed is based on Linux. The distribution used in the APs is OpenWrt [13], 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 [14] 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 [15]. Currently, the Universal Playground for IPv6 (USAGI project – [16]) task group provides support for MIPL.

In this implementation we have added the required functionality to support Multiple Care of Addresses (CoA) [17]. 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).

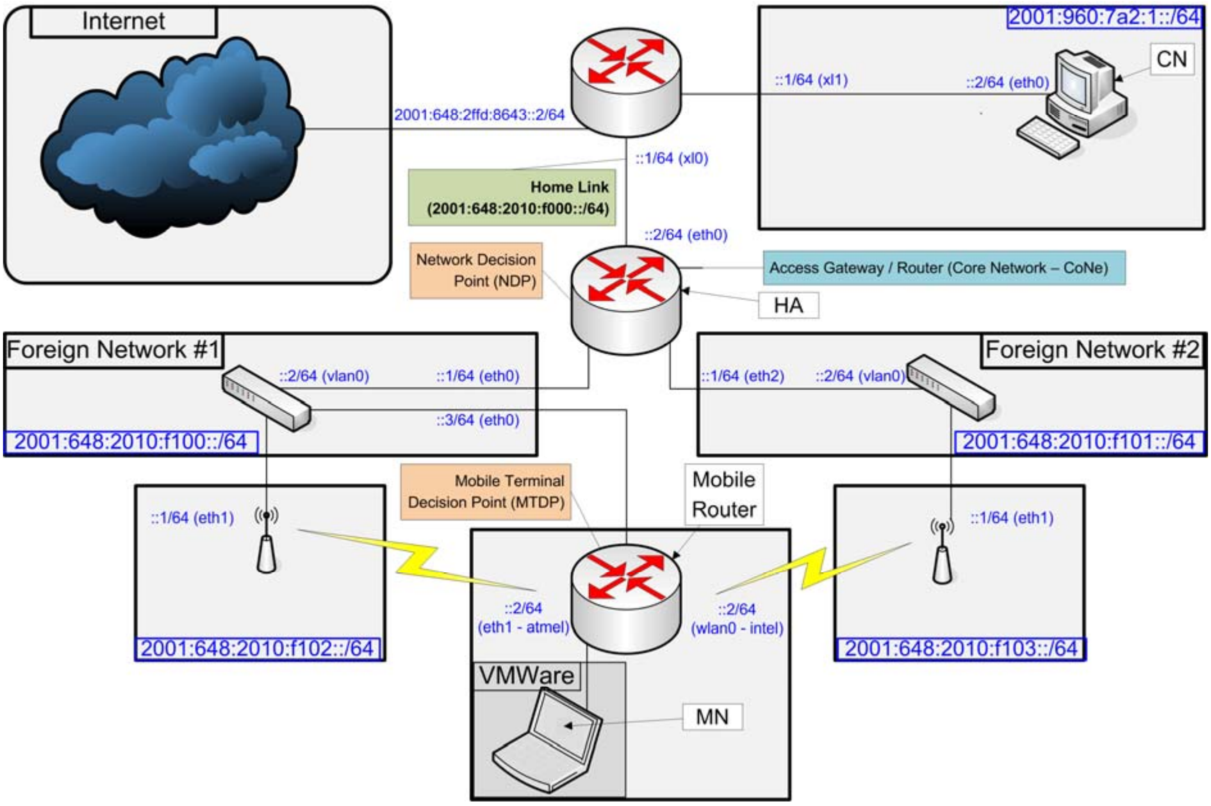


Figure 7. Test-bed topology.

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 Figure 7. Two WiFi APs act as the different RATs. Foreign Network #1 at the left part of the figure corresponds to the UMTS. The direct Ethernet link with one of the APs acts as the ubiquitous UMTS connection transferring all messages between the MTDP and the NDP. Foreign Network #2 in the right part of Figure 7 corresponds to the

WLAN. 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 [18]. More specifically, we used the OSPF (Open Shortest Path First) protocol that supports routing in IPv6 networks [19]. 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.

For the automatic IPv6 addressing of mobile nodes, as they are connected to the APs, the Linux IPv6 Router Advertisement Daemon (radvd) was used [20]. 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 [21] and we have used the Linux implementation as described in [20].

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

- The definition of new headers and messages for the exchange of information between the new entities (MTDP and NDP).
- NEMO extension with support for multiple CoAs. Note that NEMO is still under development and even now the multiple CoAs feature is not fully supported.
- The definition of syntax and grammar used by the policy rules for selecting a RAT.
- The implementation of 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 the network link between these two entities has to be established before submitting any policy messages. If we delay this necessary step while trying to select a specific RAT, we get delays in the handover execution, since we have the added difficulty to control the whole NEMO stack with the Multiple CoAs extensions. The next best choice was to use existing messages and interfaces to piggyback the information required for our mechanism to work. Thus, extensions to existing mobility messages transfer all information needed by the new messages (e.g., as Mobility Options in Binding Update and Acknowledgment messages exchanged between the HA and the MT). This way, MTDP and NDP exchange all this information safely.

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,

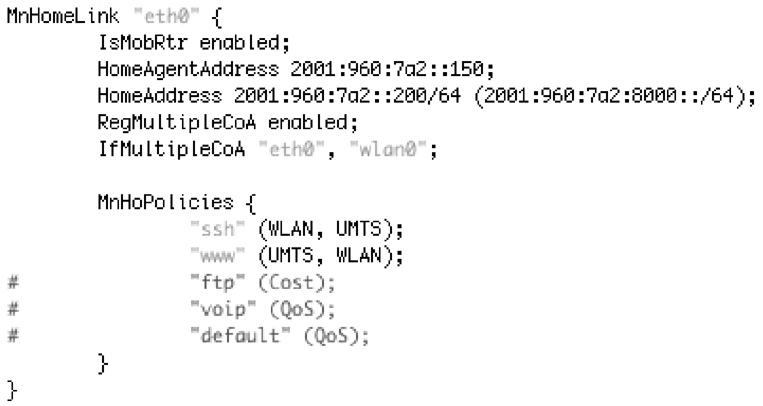
ii) the signal strength of the current AP is very low,

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. In Figure 8 we provide an example of such a file. In the file we define the required IPv6 address for the Ethernet interface and we enable multiple CoA support. Moreover, the way we have selected to define policies is per application. This means that every application (e.g., ssh, www, voip) has predefined preferences. Hence, policies are based on well defined TCP and UDP port numbers and therefore each application is monitored and routed through the desired interface.

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. For example, for ftp traffic a user may require to have the cheapest network (e.g., WLAN) only. This has been marked with the keyword “Cost” in Figure 8. As another example, VoIP can be configured to select the network that offers the best QoS etc.

As discussed in the previous sections, the MTDP will construct the priority list based on this profile and will also check other parameters such as the signal strength and the battery level before sending the prioritized list to the NDP. The latter will check if there are available resources and if it is possible it will calculate the speed and direction of a terminal to avoid unnecessary handovers (e.g., for example to WiFi islands for fast moving users). As was the case for MTDP, the NDP sends its reply piggybacked in a Binding Acknowledgment message and expects that the mobile terminal will respect this decision.



```
MnHomeLink "eth0" {
        IsMobRtr enabled;
        HomeAgentAddress 2001:960:7a2::150;
        HomeAddress 2001:960:7a2::200/64 (2001:960:7a2:8000::/64);
        RegMultipleCoA enabled;
        IfMultipleCoA "eth0", "wlan0";

        MnHoPolicies {
                "ssh" (WLAN, UMTS);
                "www" (UMTS, WLAN);
#               "ftp" (Cost);
#               "voip" (QoS);
#               "default" (QoS);
        }
}
```

Figure 8. Handover policy and mobility management information file.

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 following process was followed in order to setup the test-bed before each test case:

1. The test-bed is configured in an initial state where the NDP is expecting the registration of a mobile terminal.

2. The terminal is not connected to any wireless access network and does not transfer any information.

3. The profile of the user is parameterized according to the scenario we will showcase.

4. We initialize, in specific nodes (e.g., APs, routers) and based on the scenario, programs that monitor and register the network traffic in order to prove the operation of the mechanism. These programs have been configured to filter and store part of the exchanged information since otherwise the total amount of stored information is very large and it is very difficult to trace specific messages and their parameters that are essential in our mechanism.

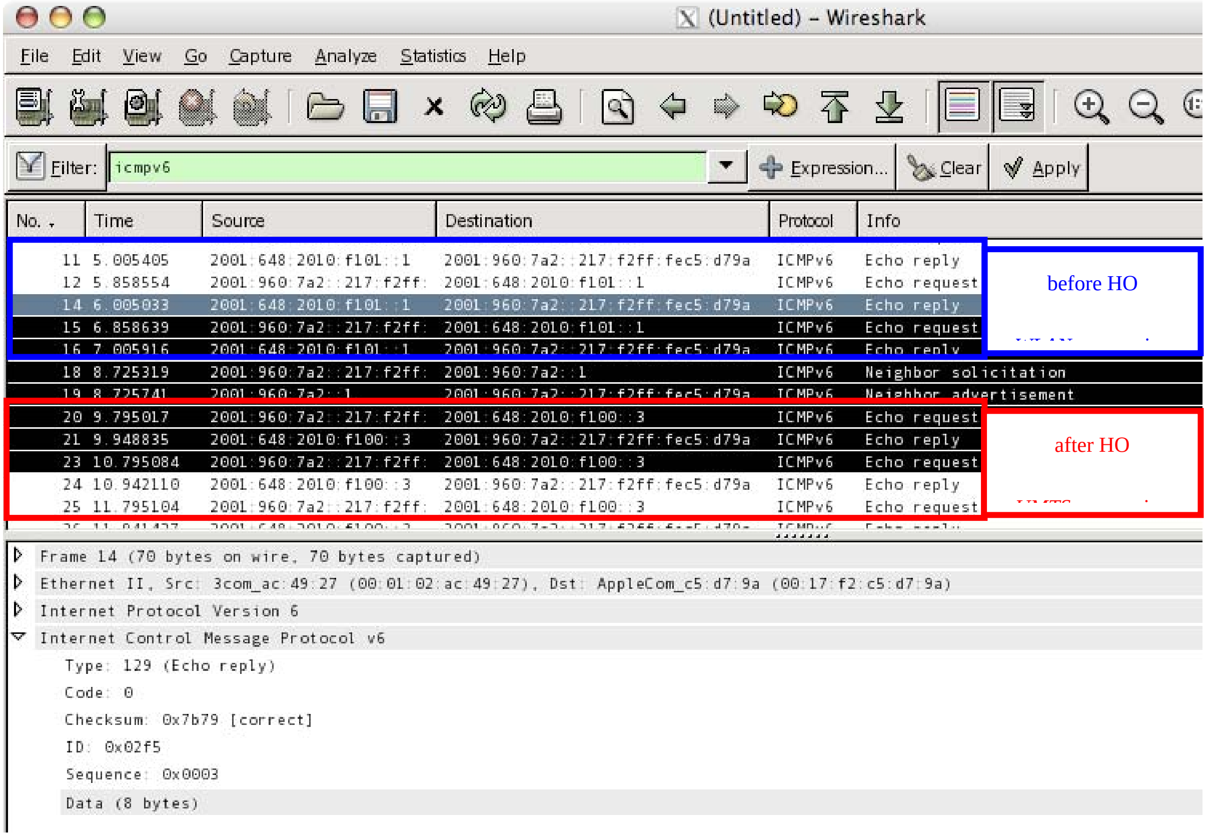
5. The MTDP is initialized and the scenario is demonstrated.

The program used for recording messages and data traffic was tcpdump, a well-known and established program [22]. 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 [23]. 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 Figure 9 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. This scenario demonstrates a vertical HO from WLAN to UMTS, i.e. from the Foreign Network #2 to the Foreign Network #1 in Figure 7. At the first five lines of Figure 9, we see the exchange of echo request and reply messages between the CN (2001:960:7a2::… in source/destination column) and Foreign Network #2 , i.e. WLAN (2001:648:2010:f101::…). After the network advertisement message, we see that the same echo messages are exchanged via the Foreign Network #1, i.e. UMTS. This indicates a successful vertical HO from WLAN to UMTS.

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 test-bed 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 [24]. These keys are transferred encrypted to a RADIUS [25] server located in the access router. Using this information it will accept or reject the request for a connection.



```
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  9.948835    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

▷ 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)
```

Figure 9. Example of an ICMPv6 vertical handover from WLAN to UMTS.

5. Conclusions and future work

In this paper we presented a new mechanism for selecting among several RATs 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. 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. Both the SDL specification and the 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.

Although both the SDL specification and the test-bed provided useful conclusions, they do not aim to deeply evaluate the performance of the algorithm. Thus, it is in our future plans to build simulation models comprising more than two candidate RATs. This will help us to evaluate the performance of the algorithm and compare it with other schemes for access network selection. Finally, we investigate the possibility of complementing the proposed algorithm with IEEE 802.21 Media Independent Handover (MIH) for the parameters' exchange between the involved entities.

References

[1] McNair J., Zhu F. (2004). 'Vertical Handoffs in Fourth-Generation Multi-network Environments' IEEE Wireless Communications, Vol. 11, pp. 8-15

[2] Hou J., O'Brien D. C. (2006). 'Vertical Handover-Decision-Making Algorithm using Fuzzy Logic for the Integrated Radio-and-OW System', IEEE Transactions on Wireless Communications, vol. 5, no. 1, pp. 176-185

[3] Giupponi L., Augusti R., Pérez-Romero J., Sallent O. (2005). 'A novel Joint Radio Resource Management Approach with Reinforcement Learning Mechanisms', 24th IEEE International Performance Computing & Communications Conference, IPCCC 2005, 2005, pp. 621-626

[4] Murray K., Mathur R., Pesch D. (2003). 'Intelligent Access and Mobility Management in Heterogeneous Wireless Networks using Policy', The 1st International Workshop in Information and Communication Technology, 2003, pp. 181-186

[5] Zhuang W., Gan Y.-S., Loh K.-J., Chua K.-C. (2003). 'Policy-based QoS Management Architecture in an Integrated UMTS and WLAN Environment', IEEE Communications Magazine, vol. 41, no. 11, pp. 118-125

[6] Song W., Zhuang W., Cheng Y. (2007). 'Load Balancing for Cellular/WLAN Integrated Networks', IEEE Network, vol. 21, no. 1, pp. 27-33

[7] Lampropoulos G., Passas N., Kaloxylos A., Merakos L. (2007). '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", Vol. 43, No. 3, pp. 889-906

[8] ITU-T. International Telecommunication Union, 'Specification and description language (SDL)', Recommendation Z.100, ITU-T Study Group 17, http://www.itu.int/ITU-T/studygroups/com17/languages/Z100.pdf

[9] Lampropoulos G., Salkintzis A., Passas N., 'Media Independent Handover for Seamless Service Provision in Heterogeneous Networks', IEEE Communication Magazine, vol. 46, no. 1, Jan. 2008.

[10] 3GPP TS 25.305 (2006). 'Stage 2 functional specification of User Equipment (UE) positioning in UTRAN (Release 7)'

[11] Telelogic SDL Suite, http://www.telelogic.com/products/sdl/index.cfm.

[12] An example SDL Simulator MSC with 3 calls, http://cgi.di.uoa.gr/~imodeas/msc.pdf

[13] OpenWRT Linux distribution for embedded devices, http://openwrt.org.

[14] NEPL (NEMO Platform for Linux), http://www.nautilus6.org/doc/nepl-howto

[15] MIPL – Mobile IPv6 for Linux, http://www.mobile-ipv6.org.

[16] USAGI Project - Linux IPv6 Development Project, http://www.linux-ipv6.org

[17] Wakikawa R., Ernst T., Nagami K., Devarapalli V. (2008). 'Multiple Care-of Address Registration'. IETF Internet Draft Version 06, August 2008, http://www.ietf.org/internet-drafts/draft-ietf-monami6-multiplecoa-06.txt.

[18] Quagga Routing Software Suite, GPL licensed IPv4/IPv6 routing software, http://www.quagga.net

[19] Coltun R., Ferguson D., Moy J. (1999). 'OSPF for IPv6', IETF RFC 2740, December 1999.

[20] Linux IPv6 Router Advertisement Daemon (radvd), http://www.litech.org/radvd

[21] Narten T., Nordmark E., Simpson W. (1998). 'Neighbour Discovery for IP Version 6 (IPv6)', IETF RFC 2461, December 1998

[22] tcpdump / libpcap, http://www.tcpdump.org

[23] Wireshark network protocol analyzer, http://www.wireshark.org

[24] Funk P., Blake-Wilson S. (2008). 'Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)', RFC 5281, http://tools.ietf.org/html/rfc5281

[25] Rigney C., Willens S., Rubens A., Simpson W. (2000). 'Remote Authentication Dial In User Service (RADIUS)', RFC 2865, http://www.ietf.org/rfc/rfc2865.txt