2009-IOT-4 (3)
2009 / 3 / 5

Linux環境へのPMIPv6の実装と評価

マシューポーラン[†,††]　寺岡　文男[††]

† Telecom Bretagne, Technopôle Brest-Iroise - CS 83818 - 29238 Brest Cedex 3, France
†† 慶應義塾大学理工学部. 〒 223–8522 横浜市港北区日吉 3-14-1
E-mail: †mathieu@tera.ics.keio.ac.jp, ††tera@ics.keio.ac.jp

Implementation and Evaluation of PMIPv6 on Linux Environment

Mathieu POULAIN[†,††] and Fumio TERAOKA[††]

† Telecom Bretange, Technopôle Brest-Iroise - CS 83818 - 29238 Brest Cedex 3, France
†† Faculty of Science and Technology, Keio University. Hiyoshi 3-14-1, Kohoku-ku, Yokohama-shi, 223-8522 Japan
E-mail: †mathieu@tera.ics.keio.ac.jp, ††tera@ics.keio.ac.jp

Abstract The mobility has been an active topic of research recently. In this context, the protocol Proxy Mobile IPv6 has been standardized last year. But no non-commercial implementation of this protocol currently exists. That's why we chose to develop our own implementation. Since this protocol is derived from the Mobile IPv6 protocol, we chose to take over the existing client of MIP for Linux, namely *umip*. Hence, in this paper we describe the evaluation of the PMIP client we realized. We describe first the principles of PMIPv6 and then how we chose to implement it. Finally, we show the results of our evaluation.

Key words Proxy Mobile IPv6, Performance, Implementation

1. Introduction

Mobile IP (MIP) was first designed for IPv4 [1] and then extended to IPv6 [2]. With MIP, all packets destined to the home address of a node (IP address of the node in its home network) will reach it whatever the node attachment to the Network is. Indeed, MIP introduces a new entity in the network, the Home Agent (HA), which will forward every packet to the current attachment of the node when it is different from its home link. For this, the HA will intercept every packets destined to the home address of the node and forward them to the node's current location through a tunnel. For handling the mobility, both the node and HA need to have a MIP client running. Thus, assuming the mobility requirements would be met in the current Internet, only nodes running the MIP daemon would have access to mobility. In order to expand the support of mobility to other nodes, the need for a network-based localized mobility management protocol appeared. As an answer to this need, the Proxy Mobile IPv6 (PMIP) protocol was specified [3] in August 2008. The protocol proposes to handle the mobility by the network only. As an advantage, it requires, for the host, only the usual IPv6 stack. Thus, the access to mobility for a host will be totally transparent, the user won't have to do any actions. The host will have no involvement in the mobility process, it won't even be aware of it. The PMIP protocol follows the MIP principles and performs the signaling process and mobility management on behalf of the node. This is realized by a new network entity, the Mobile Access Gateway (MAG). The concepts behind this protocol will be detailed in the next session.

Since no non-commercial implementation of PMIP currently exists, we decided to realize one. For this, knowing that PMIP is derived from MIP, we decided to take over the code of the Linux implementation of MIP, umip in its version 0.4 [4] and modify it for supporting PMIP. This paper describes the implementation and its evaluation. It must be noticed that a BSD implementation was designed at the same time.

We will first briefly detail the principles of PMIP, thus

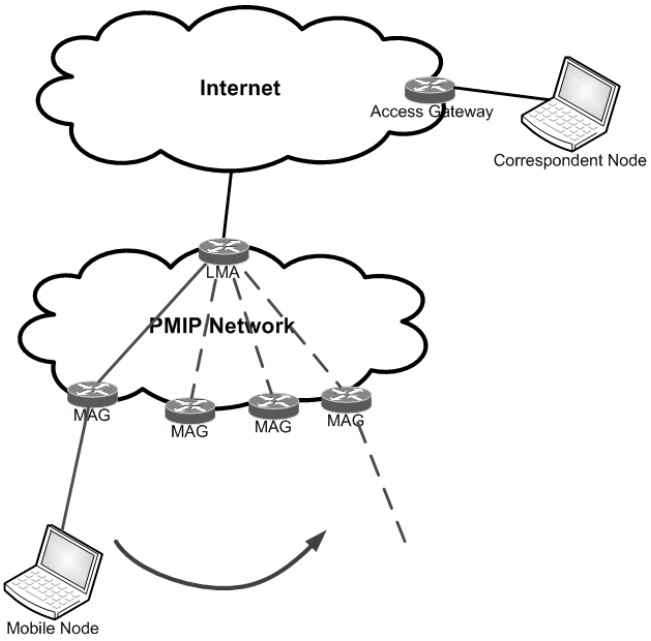

Figure 1 Architecture of PMIPv6

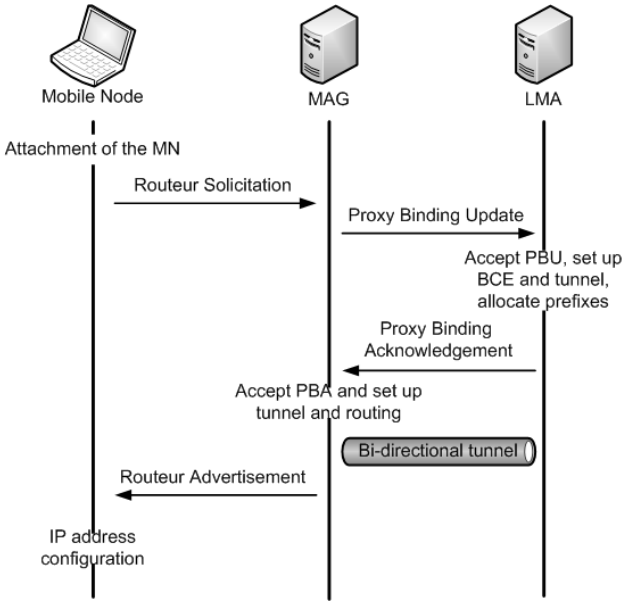

Figure 2 Registration Procedure of PMIPv6

providing a better understanding of what must be changed in MIP for implementing PMIP. Then, we will explain how the protocol was implemented, what limitations we assigned and the choices we made. Following this, we will describe the results of the evaluation of our implementation. Eventually we will conclude by stating the functionalities we could add in order to improve our work.

2. PMIP Principles

2.1 The architecture

Figure 1 shows the architecture of a typical PMIP network. We can see that PMIP introduces a new entity in the network, *the mobile access gateway (MAG)*. This entity will act as the access router of the network. It will also perform the mobility management on behalf of the mobile node. With PMIP, the HA is renamed *Local Mobility Anchor (LMA)* but its actions remain the same, it is responsible for maintaining the node's reachability state and the home network prefixes associated to the mobile node will be anchored at the LMA.

2.2 Main principles

The main principle of PMIP is to make the mobile node believe that in the PMIP network, it always has access to the same router, even when changing the physical attachment to the network. For this, the LMA and MAG must have access to the mobile node (MN) profile and policy, common to every PMIP entity. It means that a MN that wants to use the mobility in a PMIP network must be first registered. The profile of the MN will include the MN identifier and the home network prefixes allocated to the MN. So in the mobility registration process, the MAG will advertise to the MN its home network prefixes. Hence, even when using a different access to the PMIP network, the MN will receive the same advertised prefixes and will believe that it didn't change its network access.

2.3 Messages processing

2.3.1 First registration

The messages exchanged during the registration of the node for mobility support are described in figure 2. The registration process begins as soon as the MAG detects a new MN in the network. At the detection, it will get the MN identifier from the node information and get the node's profile from the identifier. If the node's profile states that the MN is allowed to use PMIP, the MAG will register the MN with the LMA. This registration is similar to the one used in MIP but adapted to PMIP. So the MAG will first send a *Proxy Binding Update (PBU)* to the LMA. The PBU is a Binding Update with the proxy flag set to 1. A PBU must carry the MN information such as its identifier, the home network prefixes if known by the MAG, the link local and link layer addresses of the MN, and other information concerning the access technology, the handoff status of the MN and a timestamp. The use of timestamp instead of sequence number in PMIP is necessary because there is no context transfer between MAGs and thus, a MAG that just detects the node on its link won't be able to get the previously used sequence number if the MN was previously registered for PMIP use. On reception of the PBU, the LMA will determine if it must accept or not the PBU. If it is accepted, the LMA will create a binding cache entry for the node's mobility session, allocate to the node its home network prefixes and then set up a tunnel to the MAG, if there isn't one previously existing, and the routes for forwarding the packets to and from the node's allocated prefixes. The LMA will then

generate a *Proxy Binding Acknowledgement (PBA)* that will echo the PBU information, indicate if the PBU was accepted or not, and carry the allocated prefixes with their associated lifetime. On reception of an accepted PBA, the MAG will set up its own tunnel and routes for forwarding the packets of and to the MN. It will also store the node's mobility session in its binding update list. Then, it will send to the MN a Router Advertisement message, as defined in Neighbor Discovery Protocol [5], advertising the mobile node allocated prefixes.

On reception of the router advertisement, the Mobile Node will configure its IP address itself (stateless autoconfiguration) or through DHCPv6 (stateful autoconfiguration), depending on the configuration of the network.

Then the MAG will regularly send router advertisement to the node, each time after successfully registered with the LMA, in order to extend the lifetime of the prefixes.

2. 3. 2 Mobile node de-registration

When the MAG detects the mobile node detachment, it sends to the LMA a PBU corresponding to the mobile node information but with an associated lifetime value of 0. The LMA on reception of this PBU will wait during a configured period of time for the reception of a PBU of another MAG in case of a handoff of the node between two MAGs. If it didn't receive anything during this time, it will end the mobility session by deleting the binding cache entry of the node, its associated routes and tunnel (if not in use by other nodes) and then send a PBA to the MAG, acknowledging the end of the mobility session. On reception of the PBA or after a period of time, the MAG will also delete the mobility session from its binding update list and delete the routes and tunnel associated.

2. 4 The support of mobility

With this registration process, once the node is registered, it will be able to configure its IP address and then send and receive packets. The packets sent by the MN will be routed to the MAG, because it acts as the default router in PMIP network, then they will be tunneled to the LMA and finally they will be forwarded to their destination. Since the home network prefixes allocated to the node are owned by the LMA, the packets sent to the MN will be routed first to the LMA which will forward them through the tunnel to the corresponding MAG, which will forward them to the mobile node.

Then, each time the node will change its attachment to the network within the PMIP network, it will receive the same prefixes. Plus, in PMIP, the MAGs must all have the same link local and link layer address. So the MN have the impression to have always the same default router and thus, doesn't detect a change of network.

3. Implementation

3. 1 The umip client

We realized the PMIP implementation based on the existing Linux MIP client named *umip* in its version 0.4 [4]. This client is the most used in the Linux environment and is a free software, thus we had a free access to its sources. The client is composed of several modules. We will now describe the main ones. One is related to the functionalities specific to the MN, one is related to those specific to the HA. Those functions will mainly concern sending and receiving mobility messages. In order to store the information of the binding update messages received by the HA, a module named *Binding Cache* was created. In this module every mobility session created will be stored and managed as a new entry in the binding cache. A similar module has been created for the MN. It is named *Binding Update List* and it stores, as indicated by the name, the binding update that have been sent by the node. Both modules are relying on another module to store the information, the hash tables. Another important module is the mobility one. This module handles the generation and parsing of mobility messages and options. There is also one module dedicated to the neighbor discovery protocol. Such module will allow the client to receive router advertisement or neighbor advertisement message and send neighbor solicitation messages. There is also a function to perform a duplicated address detection (DAD) [5]. Concerning the tunnel creation, a module is dedicated to the tunnel management. It is called by the modules HA and MN for adding and deleting tunnels. Finally, at the core of the client we have two modules. One is dedicated to the configuration management. It parses the configurations file of the client. The other one is the main module, it will initialize the configuration and then launch all the needed modules.

3. 2 The changes to make for supporting PMIP

Though PMIP is derived from MIP, it has a lot of differences with the former. The main changes are in the part of detection of the movements of the node because in MIP, since the daemon is running on the node it has access to all the desired information. That is not the case with PMIP since the actions realized on the node are transferred to the MAG. For implementing the detection of the node, we chose to not rely on the link layer events because they are specific to the link layer technology. Instead we used the IP layers events. For detecting the presence of the Mobile Node, we decided to use the Router Solicitation messages sent by a node when attaching to a new link. For detecting that a node is still present on the link, before sending a PBU for extending the lifetime, we perform a Duplicate Address Detection (DAD) [5] on the mobile node link local address. If

the DAD is a success, it means that the node is not on the link anymore. Else it means that the node is still there.

Apart from detecting the node, the other differences are that we need to implement all the new mobility options and status specific to PMIP that are defined in [3]. There is also the processing of proxy binding update and proxy binding acknowledgment, the timestamp management and the generation of router advertisement messages from PMIP information.

3.3 The developpement of the PMIP client

As said before, the MIP client is made of two distinct parts, one for the MN and one for the HA. Since the MAG actions were close to the ones from the MN in MIP, we took over the MN code for implementing the MAG. Similarly, the LMA's algorithms were coded over the HA. For implementing the changes described before, we began by defining the new mobility options and status used in mobility messages. We had a problem with the Mobile Node Identifier option because the RFC just specified it but didn't say how to actually implement it. So we eventually chose to use the Media Access Control (MAC) address of the mobile node as Mobile Node Identifier. Another possibility would have been to use a Network Access Identifier, as defined in [6], but it was harder to implement. Since the MAC address wasn't specified as a mobile node identifier, it didn't have a type defined for the type field of the mobile node identifier option. We chose to assign the type 2 for MAC address. All the other options and status were implemented as specified as in the RFC. Then, we implemented the options needed by PMIP in order to do the PMIP client configuration. For this task, we simply modified the modules configuration management and mobility. In the module configuration, we added the new client options to parse. In the module mobility, we added the parsing and creation of the new mobility options.

After having implemented the new options and status, we implemented the neighbor discovery messages used by PMIP. First, we added a function for the reception of router solicitation messages in the MAG module and then we added a function for sending the router advertisement messages in the neighbor discovery module. The router solicitation messages were quite easy to implement because of the existent neighbor discovery functions in the module. So we added a handler of router soliciation messages in the MAG code and coded the extraction of information such as the MAC address. Then, we implemented sending of router advertisement messages. Since with umip the router advertisement messages are sent via the *radvd* daemon and not with the MIP client itself, we had to code the whole feature. For simplicity's sake, we inspired ourselves from the radvd code which is also free. Thus, we adapted the code to our client and added it in the module neighbor discovery. After having coded this part, we tested it by sending messages through the PMIP client and analyzed them with the program *Wireshark* in order to check if they actually were as designed in [5].

Once the neighbor discovery messages were correctly handled, we focused on the mobility messages exchanged between the LMA and the MAG. We first added the generation of the new options and then the generation of PBU and PBA. For this, we modified in the module MAG the function for sending proxy binding aknowledgement. Similarly, we modified the function for sending proxy binding update in the module LMA. We tested it by sending a PBU to a non modified MIP client and saw how it reacted to the message. It behaved as we planned, meaning that it was recognized as a Binding Update but was rejected because it didn't carry the usual MIP options. So we then attached to develop the processing of those messages in the corresponding modules and implemented the forwarding of packets for the mobile node. The forwarding of packets is handled in the modules LMA and MAG, so we modified the related functions. PMIP, as specified in RFC 5213, allows for two possible ways of creating tunnels, statically (tunnels created permanently at the initialization of the MAG) or dynamically (tunnels created when they are needed and then deleted). Since in MIP they were created dynamically, we chose to stick with this method. In PMIP, the forwarding of packets for the mobile node is different (use of prefix instead of an address), particularly on the MAG, because it has to forward the packet to the MN, so we had to adapt it. We also implemented the timestamp management for the MAG and LMA. This was added to the modules LMA and MAG in the functions for processing the mobility messages. And of course the algorithms for processing the new options and for processing the proxy binding update and proxy binding acknowledgment were different too. So we had to further modify the functions for processing mobility messages.

Finally, when everything was implemented, we made some test and made sure that everything worked as planned. So we created a network composed of one LMA, two MAGs and one mobile node. We then configured it for the use of PMIP and realized some tests and performance evaluation. The results of those are described in the next session.

3.4 Limitations of the implementation

Because of the short time we had planned for the project, we limited ourselves. One first limitation is the handle of only one prefix for the mobile node. Although the RFC states that there may be more than one prefix assigned to the node, it was simpler to implement one. But one could easily add the possibility to handle more than one prefix. Since we used the MAC address as a mobile node identifier,

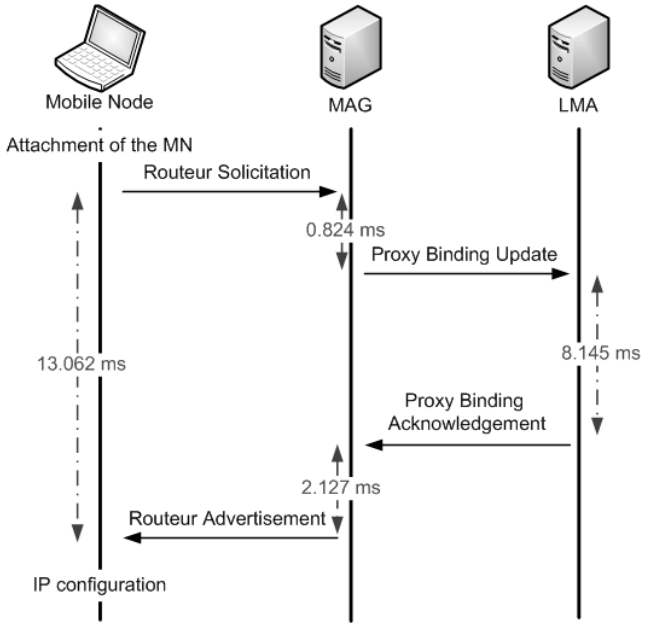

Figure 3 Performances of our implementation during the first registration

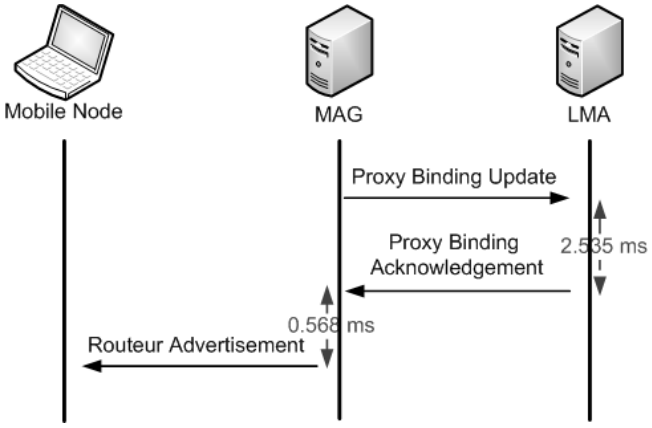

Figure 4 Performances of our implementation during the lifetime extension

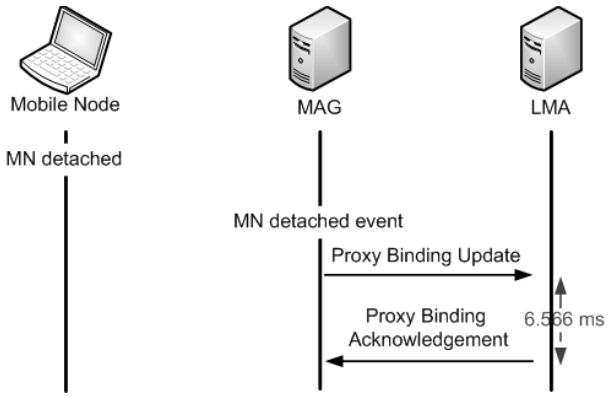

Figure 5 Performances of our implementation during the de-registration

we implicitly limited ourselves by assuming that the mobile node possesses only one interface (the MAC address is different for each interface). Getting over this limitation would more difficult because we would have to somehow find a way to link every interface to a common mobile node identifier. Due to a time limitation, we didn't either implement the use of IPSec for the mobility messages. Since this part was already implemented in the MIP client, one could easily add this possibility by taking over the MIP code. Lastly, since we chose to rely on the router solicitation messages instead of the link layer events, one could try to implement specific ways of detecting the MN, depending of the link layer type.

4. Evaluation

4.1 Time processing evaluation

For the performance evaluation of our client, we used for the MAG and LMA computers with as processor an Intel Celeron 1.5 GHz and for the random access memory (RAM) 2 GB of type DDR 266Hz. The results are given in the figures 3, 4, and 5. On those figures, we put the processing time for three distinct operations, the first registration, the lifetime extension and the de-registration.

For each action measured, we did 10 measurements. For the MAG and LMA, we used the network capture software *tcpdump* and for the MN we used *Wireshark*. Those software allowed us to visualize the packets that were sent and received on each interface.

From all the figures, we can see that the times are really different, depending on each action. The shortest times are seen with the generation of the Router Advertisement messages, in the case of a life extension. Indeed, the generation of those messages doesn't require much computing time because they are generated from the information given by the PBA just received. The node just needs to update the lifetime of the binding update list entry and then generate the message from the entry. Another message processing which is quick is the generation of a PBU from a Router Solicitation message. The computing time for this task is also short because the MAG only needs to read the information of the Router Solicitation message, create the entry in the binding update list, and then send the Proxy Binding Update to the LMA.

The other messages processing times are longer because they require more complex actions. In the case of the generation of the Proxy Binding Acknowledgment, it is because the Proxy Binding Update message must be processed first and there are a lot of verifications to do on the message and the number of mobility options carried by the Proxy Binding Update is important. In the case of the first time registration, for which the times are the longer, once the PBU has been processed, the LMA still needs to allocate the prefixes, add the new entry to its binding cache and finally set up the

tunnel and routes for forwarding the packets of the mobile node. We chose here to create the tunnels dynamically. So at each first registration or de-registration, a tunnel is added or deleted. We measured a time of 2 ms for the creation of the tunnel, hence with pre-established tunnels we could reduce the computing time for the generation of first registration and de-registration messages.

We can also note that the processing time is longer on the LMA than the MAG. This can be explained by the fact that the LMA creates a new thread for each proxy binding update received and because there are more checks in order to know if the PBU must be accepted or not, than for accepting the PBA.

Finally, the average time for registering the mobile node in our network for support of mobility is around 13 ms. In those 13 ms, we can distinguish around 11 ms for the processing of messages and 2 ms that corresponds to the round trip times between the LMA and the MAG and between the MAG and the MN. In general, the registration time can be represented as follows:

$$RTT_{MAG-LMA} + RTT_{MN-MAG} + 11 \text{ ms.}$$

where, $RTT_{MAG-LMA}$ is the round trip time between the MAG and the LMA and RTT_{MN-MAG} is the round trip time between the MN and the MAG. RTT_{MN-MAG} is negligible because the MN and the MAG are usually directly connected by a wireless link such as IEEE802.11. $RTT_{MAG-LMA}$ would be less than 10 ms because the MAG and the LMA are connected to the same PMIP domain. Thus, the registration time can be estimated as less than 20 ms in the actual environment.

4.2 Handover evaluation

Apart from those time measurements, we tested our final client by doing some handover between our two MAGs. Although packets were lost while we switched the access point to the network of the mobile node, the connections of the layers above the IP layer were not lost (UDP and TCP), meaning that the network correctly handled the mobility of the mobile node thanks to the LMA and both MAGs.

4.3 Overhead evaluation

Since with PMIP the packets are forwarded through a tunnel, we decided to measure the time processing related to the overhead. For this, we took some measures of the forwarding of packet by the LMA and the MAG through the PMIP protocol. Then, we compared them to the forwarding of the same packets without PMIP. So for forwarding an IPv6 packet through the tunnel, the MAG will need 0.039 ms. It is nearly the same for the LMA because it needs 0.040 ms. On the same computers, a simple forwarding of an IPv6 packet without PMIPv6 takes only 0.021 ms. So we can say that the time needed for processing the overhead is around 0.020 ms for both the LMA and the MAG, which is an acceptable value for most of the users.

5. Conclusion

We described in this paper the implementation of the PMIP protocol. We decided to develop it under the Linux environment, following the implementation of MIP, namely umip in its version 0.4. We chose to implement PMIP because we believe that offering to the users the possibility to access to mobility without requiring anything apart from the usual IPv6 stack, will help the use of mobility to spread.

The implementation we realized here is only a prototype, hence a lot of things are still needed. Among those we could state the need for IPSec, or adding the possibility to use more than one interface and prefix, or also change the node detection for using the link layer events instead of the neighbor discovery protocol. We could also offer the possibility to configure the method for tunnel creation with the use of a configuration variable for instance. This way we could easily switch from pre-established tunnels and dynamic creation. So a lot of work is still needed for improving the client and making it more mature for a larger distribution.

But even without those features, our client is already offering mobility support when a network is configured to use it. We managed to do some handover and keep the previous TCP and UDP connections open. The performances we evaluated are quite reassuring and are well acceptable for a general use of mobility.

So we hope that our prototype will be improved in the future in order to expand its use, but at the same time we are pleased to see that the mobility is already working well and that the performances of our client are good enough for a general use.

References

[1] C. Perkins. *IP Mobility Support for IPv4*, August 2002. RFC 3344.

[2] D. Johnson, C. Perkins, and J. Arkko. *Mobility Support in IPv6*, June 2004. RFC 3775.

[3] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. *Proxy Mobile IPv6*, August 2008. RFC 5213.

[4] Umip 0.4: Mobility client for linux. http://umip.linux-ipv6.org.

[5] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. *Neighbor Discovery for IP version 6 (IPv6)*, September 2007. RFC 4861.

[6] B. Aboba, M. Beadles, J. Arkko, and P. Eronen. *The Network Access Identifier*, December 2005. RFC 4282.