Experimental Evaluation of Proxy Mobile IPv6: an Implementation Perspective

Giuliana Iapichino and Christian Bonnet
Mobile Communications Department
Eurecom
Sophia Antipolis, France
{Giuliana.Iapichino, Christian.Bonnet}@eurecom.fr

Abstract—**Proxy Mobile IPv6 (PMIPv6) is a network-based mobility management protocol for localized domains, which has been actively standardized by the IETF NETLMM Working Group. PMIPv6 is starting to attract much attention among telecommunication and internet communities due to its salient features and it is expected to expedite the real deployment of IP-based mobility management. However, an experimental evaluation of PMIPv6, which analyzes the impact of its practical constraints, is still missing. From their analysis it is possible to highlight the requirements for PMIPv6's real implementation and to accelerate the practical deployment of IP-based network mobility management. In this paper, we present an experimental evaluation of PMIPv6 under different implementation configurations and we evaluate their impact on the performances of PMIPv6 in a real test-bed.**

Keywords: Proxy Mobile IPv6, Network-based Mobility Management, Implementation, Experimentation, Evaluation.

I. INTRODUCTION

In the past years, there have been two approaches for managing localized mobility [1]: the host-based approach used by most IETF protocols, such as Hierarchical Mobile IPv6 (HMIPv6) [2], and the proprietary Wireless LAN (WLAN) switch approach used between WLAN switches in different subnets. None of the two has a complete solution to the problem. While the WLAN switch approach is most convenient for network operators and users because it requires no software on the Mobile Node (MN) other than the standard drivers for WiFi, the proprietary nature limits the interoperability, and the restriction to a single last-hop link type restricts scalability. The IETF host-based protocols require host software stack changes that may not be compatible with all global mobility protocols. They also require specialized and complex security transactions with the network that may limit deployability. The conclusion is that a localized mobility management protocol that is network-based and requires no software on the host for localized mobility management is desirable.

Having these requirements in mind, IETF NETLMM WG has proposed Proxy Mobile (PMIPv6) [3] as a new network-based mobility protocol for IPv6 nodes which does not require host involvements. It extends Mobile IPv6 (MIPv6) [4] signaling and reuses many concepts such as the Home Agent (HA) functionalities. As PMIPv6 is designed to provide network-based mobility management support to a MN in a topologically localized domain, its innovative point is that it exempts the MN from participating in any mobility-related signaling and proxy mobility agents in the serving network perform mobility-related signaling on behalf of the MN. Once the MN enters a PMIPv6 domain and performs access authentication, the serving network ensures that the MN believes it is always on its home network and can obtain its Home Address (HoA) on any access network. The serving network assigns a unique Home Network Prefix (HNP) to each MN, and this prefix always follows the MN whenever it moves within the PMIPv6 domain. Thus, for the MN the entire PMIPv6 domain appears as its home network.

To speed up PMIPv6's adoption by mobile network operators, we provide in this paper an implementation analysis of PMIPv6, which takes into account all the important recommendations for respecting the standard and, at the same time, for reducing handover delays. To the best of our knowledge, this paper is the first attempt to study PMIPv6's implementation issues, such as Layer 2 attachment and detachment, unicasted Router Advertisement (RA) messages, default router detection and tunneling, and to evaluate their impact on protocol's performances. Our PMIPv6's implementation is developed under Linux vanilla kernel 2.6.20 reusing Mobile IPv6 for Linux (MIPL) v 2.0.2 [5] on a real test-bed for an experimental evaluation of PMIPv6. Analysis of each implementation configuration and evaluation of different performance metrics are provided.

The rest of this paper is organized as follows. Section II describes PMIPv6's architecture and signaling. Section III illustrates our PMIPv6 implementation in the real test-bed. Section IV presents an experimental evaluation of PMIPv6 under different implementation configurations with an analysis of their impact on PMIPv6's performances. Finally Section V concludes the paper.

II. PROXY MOBILE IPv6

Figure 1 illustrates the PMIPv6 architecture with the two core functional entities:

- *Local Mobility Anchor (LMA)*: it is similar to HA in MIPv6. LMA is responsible for maintaining the MN's reachability state and it is the topological anchor point for the MN's HNP. LMA includes a Binding Cache Entry (BCE) for each currently registered MN with the MN-Identifier, the MN's HNP, a flag indicating the proxy registration and the interface identifier of the bi-directional tunnel between the LMA and the MAG.

This work is supported by the ESA Networking Partnering Initiative (NPI) program. It is co-funded by the European Space Agency and Thales Alenia Space.

- *Mobile Access Gateway (MAG)*: it is the entity that performs the mobility management on behalf of the MN and it resides on the access link where the MN is anchored. The MAG is responsible for detecting the MN's movements to and from the access link and for initiating binding registrations to the MN's LMA. Moreover, the MAG establishes a tunnel with the LMA for enabling the MN to use the address from its HNP and emulates the MN's home network on the access network for each MN.

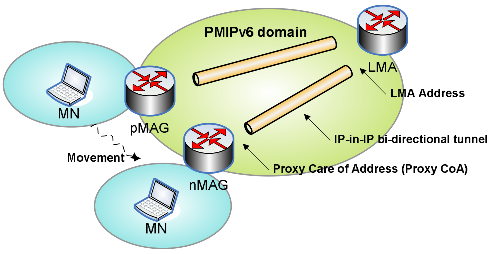

Figure 1. Overview of PMIPv6 architecture

The main steps in the PMIPv6 mobility management scheme are described hereafter and shown in Fig. 2:

- MN attachment: once a MN enters a PMIPv6 domain and attaches to an access link, the MAG on that access link performs the access authentication procedure with a policy server using the MN's profile, which contains the MN-Identifier, the LMA's address and other related configuration parameters;
- Proxy Binding exchange: the MAG sends to the LMA a Proxy Binding Update (PBU) message on behalf of the MN including the MN-Identifier. Upon accepting the message, the LMA replies with a Proxy Binding Acknowledgment (PBA) message including the MN's HNP. With this procedure the LMA creates a BCE for the MN and a bi-directional tunnel between the LMA and the MAG is set up;
- Address Configuration procedure: at this point the MAG has all the required information for emulating the MN's home link. It sends RA message to the MN on the access link advertising the MN's HNP as the hosted on-link-prefix. On receiving this message, the MN configures its interface either using stateful or stateless address configuration modes. Finally the MN ends up with an address from its HNP, which it can use while moving in the PMIPv6 domain.

The LMA, being the topological anchor point for the MN's HNP, receives all packets sent to the MN by any Correspondent Node (CN) and forwards them to the serving MAG through the bi-directional tunnel. The MAG on other end of the tunnel, after receiving the packet, removes the outer header and forwards the packet on the access link to the MN.

The MAG typically acts as a default router on the access link. It intercepts any packet that the MN sends to any CN and sends them to its LMA through the bi-directional tunnel. The LMA on the other end of the tunnel, after receiving the packet, removes the outer header and routes them to the destination.

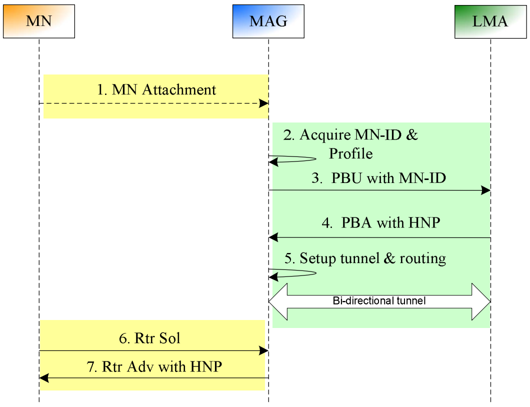

Figure 2. Message flows in PMIPv6

III. REAL IMPLEMENTATION OF PMIPV6

We have implemented PMIPv6 under Linux vanilla kernel 2.6.20 reusing Mobile IPv6 for Linux (MIPL) v 2.0.2. All the basic bricks of MIPL are used in an efficient way as shown in Figure 3.

In MIPL, Mobile IPv6 is implemented using multi threads: one for handling the ICMPv6 messages, one for handling Mobility Header messages, and another one for handling tasks and time events. To support PMIPv6, we have extended these elements and implemented handlers for all necessary messages and events. ICMPv6 messages and Mobility Header messages are parsed by the Handler as inputs to the Finite State Machine, which is the heart of the system. Two different Finite State Machines are defined for LMA and MAG. They are in charge of making appropriate decisions and controlling all the other elements to provide a correct predefined protocol behavior. The PMIPv6 Binding Cache stores all information about MNs' points of attachment and it is kept up-to-date with the mobility of MNs.

As PMIPv6 implementation is built on top of MIPL version 2.0.2, it could be, in the future, easily integrated in MIPL, growing in line with the standards as well as with MIPL source code.

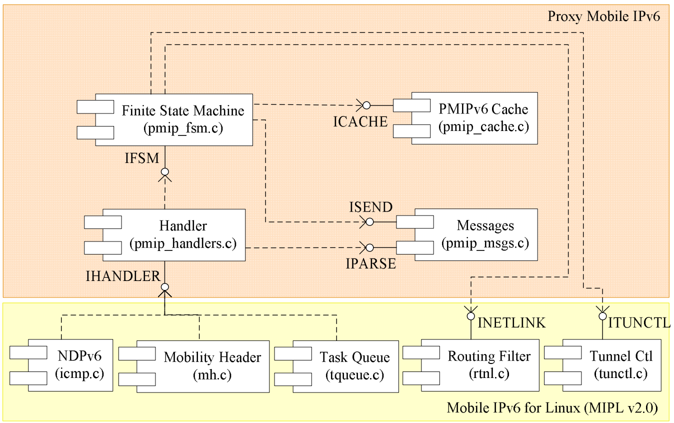

Figure 3. PMIPv6 software architecture

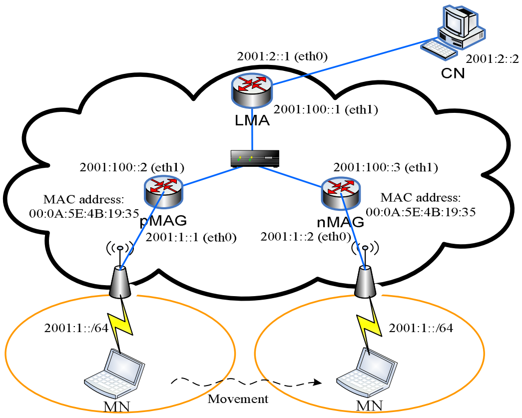

Figure 4. Test-bed topology

Figure 4 shows the experimental topology of our test-bed. An *unmodified* MN, which does not have any specific software for mobility, uses its Netgear wireless card to attach to one of the two Cisco Aironet 1100 series Access Points (APs), which support IEEE 802.11a/g specifications. Each AP is directly connected with a MAG. The implementation of MAG functionalities contains additional features and modifications of MIPL to handle PBU and PBA messages and mobility options, and a modified Router Advertisement daemon (RADVD), which *unicasts* RAs with a specific HNP per MN. Each MAG is connected to the LMA. The LMA is configured as a modified HA in MIPL which stores a unique HNP in the BCE for each MN and it is able to handle PBU and PBA messages. Finally, an unmodified CN is connected to the LMA. All the entities in the test-bed are running Ubuntu 7.10 with Linux vanilla kernel 2.6.20 in LMA and MAGs and with 2.6.22-15-generic Linux kernel in MN and CN. More detailed specifications of each device are presented in Table I.

TABLE I. HARDWARE CONFIGURATION OF DEVICES

Name	**Hardware configuration**
LMA	CPU Pentium 4 2 GHz, RAM 512 MB, NIC 3com 3C905C-TX
MAG	CPU Pentium 4 2,66 GHz, RAM 1 GB, NIC 3com 3C905C-TX
AP	Cisco Aironet 1100 series AIR-AP 1120 B SERIES
MN	CPU Pentium M 1,6 GHz, RAM 512 MB, NIC Netgear WAG511 v2
CN	CPU Core 2 Duo 2,6 GHz, RAM 4 GB, NIC Broadcom BCM5755M
Hub	Dell Power Connect 2716 1G Ethernet

Our implementation of PMIPv6 protocol is not the first tentative to provide experimental results on PMIPv6, but it is the only one that analyses the implementation issues of the protocol and gives an implementation perspective. In [6] authors compared PMIPv6 with other local mobility management protocols, but the implementation is using IPv6-in-IPv4 tunnel to emulate the IPv6 network and a network emulator to emulate the network environment. Moreover, no specifications are given to operators for PMIPv6 implementation. [7] focuses its analysis on the empirical comparison between MIPv6 and PMIPv6, demonstrating the superiority of PMIPv6, but no details are provided for important aspects in the PMIPv6 implementation. Also [8] compares MIPv6 with PMIPV6, with the difference that the measurements have been made over two different access networks, WLAN and HSDPA. Signaling and processing overheads have been analyzed, but no mobility handover analysis is provided. In this paper we consider the most important practical constraints that we have faced when implementing the standard PMIPv6. They can be summarized as follow:

- <u>Attachment and detachment phases:</u> standard PMIPv6 [3] does not specify any functionality for these two phases as its main purpose is to define only the elements and the signaling messages inside the PMIPv6 domain. As point of reference we have considered [9], in which suggestions on the MN-MAG interface are provided. One possibility is to use an IP layer-based solution, the second one is to develop a specific link-layer mechanism. We have chosen the latest as the use of triggers at layer 2 allows faster movement detection. We have used the Syslog messages sent by the Cisco APs to the MAGs containing ‘‘associate’’, ‘‘disassociate’’ and ‘‘reassociate’’ information to detect attachments and detachments of the MN from the PMIPv6 domain. As future work, we will integrate our PMIPv6 implementation with the IEEE 802.21 Media Independent Handover (MIH) protocol [10] in order to benefit of a mechanism to gather information from various link types and associated networks in a timely and consistent manner, and deliver it to network layer entities.
- <u>Unicast RA</u>: as the HNP is unique per MN, it needs to be sent in a unicast RA message by the MAG to the specific MN. We have developed and integrated a functionality in the PMIPv6 daemon for MAGs based on RADVD daemon to unicast RAs. MN’s address is auto-configured through IPv6 Stateless Address Auto Configuration.
- <u>MAG’s link-local address configuration</u>: as specified in [3] the MAG is the IPv6 default-router for the mobile node on the access link. However, as the MN moves from one access link to another, the serving MAG on those respective links will send the RA messages. If these RAs are sent using a different link-local address or a different link-layer address, the MN will always detect a new default-router after every handoff. For solving this problem, standard PMIPv6 requires all the MAGs in the domain to use the same link-local and link-layer address on any of the access links wherever the MN attaches. In order to follow this important specification we have configured all MAGs with the same link-local address using the command

```
Macchanger –m newMAC@ interface
```

- Tunneling: bi-directional tunnel is used for routing data traffic to and from the MN between the MAG and the LMA. A tunnel hides the topology and enables a MN to use the address from its HNP from any access link in the PMIPv6 domain. A tunnel may be created dynamically when needed and removed when not needed. However, implementations may choose to use static pre-established tunnels instead of dynamically creating and tearing them down on a need basis. We have implemented a static and shared tunnel between each MAG and the LMA in order to serve all the MNs attached to the same MAG with the same tunnel.

The impact of these implementation configurations on PMIPv6 performances are analyzed in the following section.

IV. EXPERIMENTAL RESULTS

We have tested the handover performances of our PMIPv6 implementation under the previously described configuration setup and with the test-bed configuration illustrated in Fig. 4. Iperf v 2.0.2 is used to generate TCP/UDP traffic. Through Wireshark v 1.0.1we have analyzed the test runs.

First of all, we have analyzed the different behavior of PMIPv6 implementation under different MAG's link local address configurations. In the first scenario we do not use the Macchanger function and we leave the two MAGs with their own MAC addresses, while in the second scenario we apply the modification as shown in Fig. 4. Figures 5 and 6 illustrate the UDP throughput when the MN performs handover from AP1 to AP2 in the respectively two scenarios. We can see that the UDP performances for the second scenario are slightly better than the ones for the first scenario.

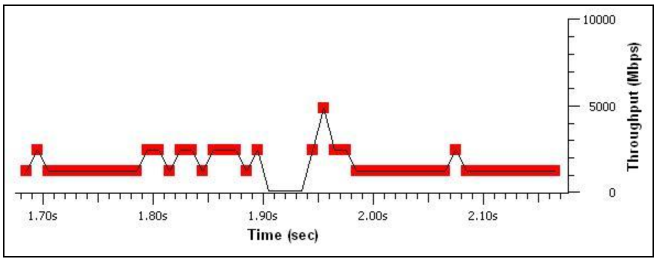

Figure 5. UDP throughput during handover in first scenario

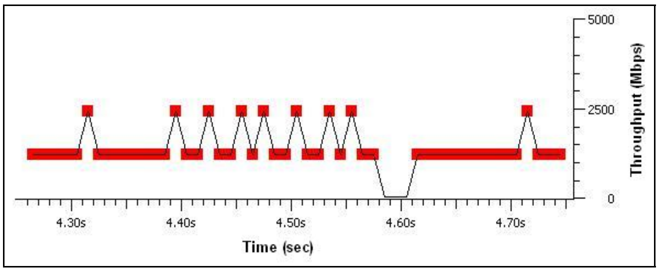

Figure 6. UDP throughput during handover in second scenario

To better evaluate the handover latency we have repeated the test 10 times for each scenario. As shown in Fig. 7, in the case of different MAC address configuration the handover latency is in average higher than 40 ms, while if we configure the same MAC address in both MAGs the handover latency does never exceed that value.

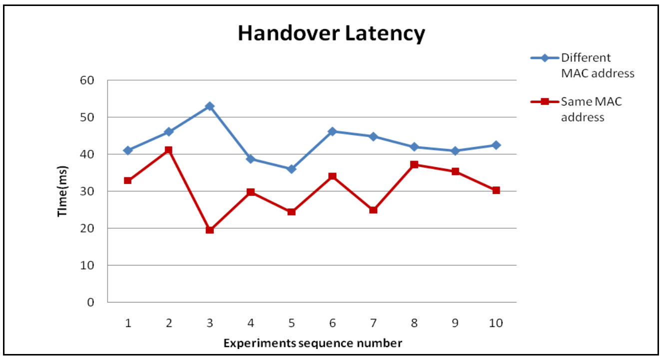

Figure 7. Handover latency of UDP traffic in scenarios 1 and 2

Different are the considerations when we analyze the performances of TCP during handover for the two scenarios. Figures 8 and 9 show, in the time-sequence graphs of TCP, a handover delay higher than 100 s for scenario 1, while around 0.5 s for scenario 2. This result shows the importance of configuring the same link-local address for all the MAGs in order to give the possibility to the MN of using it for routing in the mean-time the default-router is configured.

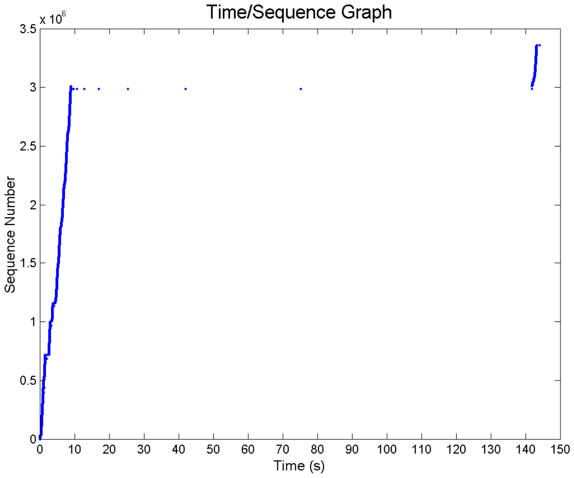

Figure 8. Handover performance for TCP in scenario1

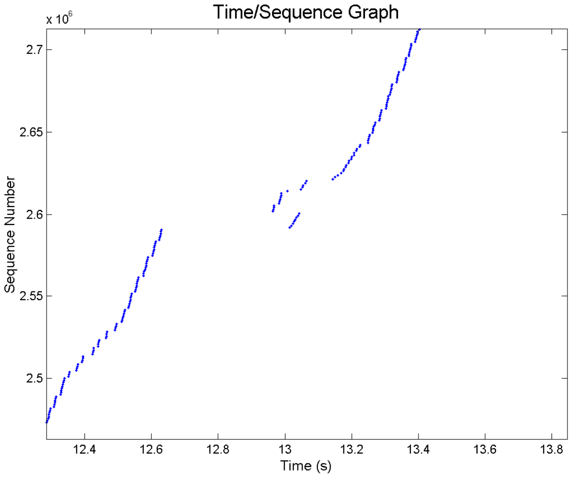

Figure 9. Handover performance for TCP in scenario 2

TABLE II. HANDOVER LATENCY OF PMIPV6 PHASES

Phases (ms)	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	No. 7	No. 8	No. 9	No. 10	Average
L2 Attachment – PBU	9.2	4	3.2	4.2	4.2	5.2	6.3	5.2	5	3	4.95
PBU – PBA	0.7	4.1	4.1	0.7	0.7	0.7	4.1	4.2	4.1	0.8	2.42
PBA – RA	4.6	4.7	10.4	4.6	4.7	4.8	10.2	9.9	10.9	4.7	6.95
Total PMIPv6 Latency	14.5	12.8	17.7	9.5	9.6	10.7	20.6	19.3	20	8.5	14.32

Then we have considered a third scenario in which the bi-directional tunnel between MAG and LMA is dynamically created. We want to specify that the previously defined scenario 2 has static tunnel. We have compared the handover latency for UDP traffic between scenarios 2 and 3. As we can see from Fig. 10 the performances are mainly the same, thus the delay for tunnel creation can be considered irrelevant.

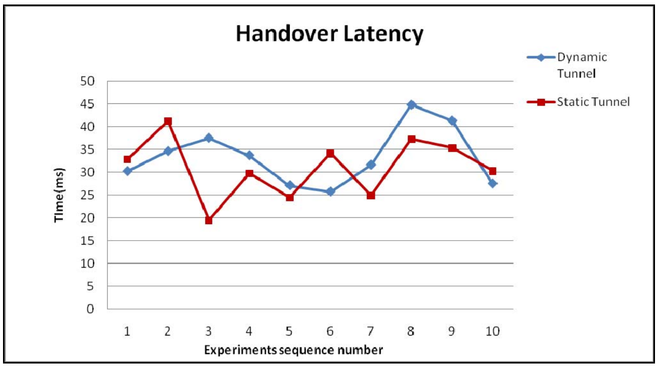

Figure 10. Handover latency of UDP traffic in scenarios 2 and 3

Finally we have considered the different phases of PMIPv6 protocol and analyzed the handover latency of each phase as reported in Table II. It is important to notice that the handover breakdown in Table II represents the handover latency of PMIPv6 procedure and not the total handover MNs are facing. This table refers to scenario 2 and shows that there is no significant difference between the delays that each phase brings.

V. CONCLUSIONS

In this paper we have completed an entirely empirical study based on real experiments of PMIPv6. To the best of our knowledge, this work is the first to provide an implementation perspective on the standard PMIPv6 under different implementation configurations. Our implementation is fully complaint with RFC 5213 and in line with the directives provided in the standard. The per-MN-prefix allocation scheme and unicast RAs have been implemented, as well as the BCE at LMA and dynamic bi-directional tunnel between MAG and LMA. Moreover, the important feature of allocating the same link-local address to all MAGs has been respected. The experimental results show that the latest aspect cannot be omitted in the implementation, while the fact of implementing a dynamic or permanent tunnel between MAG and LMA can be freely decided as it does not impact the handover performances. In the future we plan to integrate MIH protocol into our real test-bed in order to benefit of multi-technology layer 2 triggers for attachment and detachment. We have released our PMIPv6 implementation as open source in [11]. It does not require any modification in the IPv6 standard kernel.

ACKNOWLEDGEMENTS

Authors would like to thank Huu Nghia Nguyen, Lionel Gauthier and Philippe Foubert for their help in implementing PMIPv6 software.

REFERENCES

[1] J. Kempf, “Goals for Network-Based Localized Mobility Management (NETLMM)”, IETF RFC 4831, April 2007.

[2] H. Soliman, C. Castelluccia, K. El Malki and L. Bellier, “Hierarchical Mobile IPv6 Mobility Management”, IETF RFC 4140, August 2005.

[3] S. Gundavelli et al., “Proxy Mobile IPv6”, IETF RFC 5213, August 2008.

[4] D. Johnson, C. Perkins, and J. Arkko, “Mobility Support in IPv6”, IETF RFC 3775, June 2004.

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

[6] J. Guan, et al, “Implementation and Analysis of Network-based Mobility Management Protocol in WLAN Environments”, MobiWorld 08, in conjunction with Mobility Conference 2008, Ylan, Taiwan, September 2008.

[7] S. Hyeon, Y.-H. Han, H.-B. Lee, and H.-Y. Choi, ‘‘Empirical Performance Evaluation of IETF Mobile IPv6 and Proxy Mobile IPv6’’, Mobility Conference 2008, Ylan, Taiwan, September 2008.

[8] A. Udugama, et al., ‘‘Evaluation of a Network based Mobility Management Protocol: PMIPv6’’, IEEE VTC 2009, Barcelona, Spain, April 2009.

[9] J. Laganier, S. Narayanan, and P. McCann, ‘‘Interface between a Proxy MIPv6 Mobility Access Gateway and a Mobile Node’’, draft-ietf-netlmm-mn-ar-if-03, IETF Internet draft, February 2008.

[10] IEEE Std 802.21-2008, “IEEE Standard for Local and Metropolitan Area Networks, Part 21: Media Indipendent Handover Services”, January 2009.

[11] PMIPv6 Open Source, http://www.openairinterface.org/