

Analytical and Experimental Evaluation of Handovers in IPv6 Mobility Management Protocols

Javier Carmona-Murillo¹, David Cortés-Polo², Jesús Calle-Cancho², José-Luis González-Sánchez², Francisco-Javier Rodríguez-Pérez¹

¹ Department of Computing and Telematics System Engineering,
University of Extremadura, Spain

{jcarmur, fjrodri}@unex.es

² Research, Technological Innovation and Supercomputing Center of Extremadura
(CénitS), Cáceres, Spain.

{david.cortes, jesus.calle, joseluis.gonzalez}@cenits.es

Received: Dec. 14, 2015

Accepted: March 14, 2016

Published: March 31, 2016

DOI: 10.5296/npa.v8i1.8725

URL: <http://dx.doi.org/10.5296/npa.v8i1.8725>

Abstract

Mobile data traffic in the Internet has experienced an exponential growth due to the widespread presence of multimedia capable mobile devices and the deployment of multiple wireless networks. With this continuous development of mobile communications, the achievement of an efficient IP mobility management protocol has revealed as one of the major challenges in next-generation wireless networks. Mobility management solutions are responsible for maintaining the ongoing communications while the user roams among distinct networks. Mobile IPv6 and Proxy Mobile IPv6 are the most representative solutions standardized by the IETF. Recently, the IPv6 mobility support has been newly integrated into the kernel sources and Linux mobility ready kernels are available from versions 3.8.1. In this paper, we conduct an analytic and experimental evaluation of Mobile IPv6 and Proxy Mobile IPv6. We develop an analytic model of the signaling and handover latency. Moreover, we present an experimental study of these protocols based on their open source implementations. We provide numerical results based on experiments made in real scenarios under different network conditions.

Keywords: IPv6 Mobility Management, Mobile IPv6, Proxy Mobile IPv6, Experimental Testbed, Multimedia Communications.

1. Introduction

Over the last few years, Internet data communications have experienced a paradigm shift from the traditional fixed cable access to the wireless and mobile world. From the beginning of the century, wireless technologies have evolved up to approximately a 1000-fold increase in data rate. This evolution, together with the enormous proliferation of powerful mobile devices, is showing a high demand of mobile data traffic that grows year by year [1].

Along this time, mobility support in the Internet has been an active research topic and numerous protocols have been proposed. The main purpose of these mobility management protocols is to provide continuous service to mobile users, even if they change its point of attachment to the network. To support continuous service, mobility protocols should maintain connections during handover and thus, provide seamlessness while the user moves through different wireless networks. From a standard perspective, the most relevant organizations in the field of mobile communications, have designed different mobility management solutions to be adopted in the mobile network architectures.

In order to provide this continuous communication for mobile devices, the IETF (Internet Engineering Task Force) has standardized a fair number of mobility management protocols such as Mobile IPv6 (MIPv6) [2] or Proxy Mobile IPv6 (PMIPv6) [3]. These protocols are widely accepted as the most appropriate mechanisms for addressing seamless IP mobility in future wireless networks. In fact, the continuous development of these protocols has made the 3GPP (Third Generation Partnership Project) EPS (Evolved Packet System), commonly referred to as the 4G LTE (Long Term Evolution), to adopt PMIPv6 and Dual Stack MIPv6 for network based and host based mobility management respectively [4]. Mobile IPv6 is probably the most widely known IP mobility support protocol and it provides the mobility functionalities for a Mobile Node (MN) away from its home network by updating its TCP/IP stack. This means that the MN is responsible for all mobility related signaling. Unlike the MIPv6 host based approach, in a network based mobility management approach such as PMIPv6, the serving network handles the mobility management on behalf of the MN; thus, the MN is not required to participate in any mobility related signaling.

The design of these mobility management protocols, similar to almost all protocols in use on the Internet, has followed an open, all inclusive process in which all documents are freely available over the Internet. Particularly, the IETF's open document process is a case study in the potential of the Open Source movement. The design process of an Internet protocol at IETF is an open call in which researchers, companies and individuals can participate [5]. Before its publication, the documents follow an extensive review process and only the standards that meet specific real world requirements and operate as expected, become true standards. In this process, open source implementation plays an important role due to its availability to the community. The experimentation in real environments allows the evaluation of the behavior and performance of the implementations under certain conditions. Thus, both the protocol design and implementation process share the same open philosophy and serve as a solid base for discussion and experimentation.

Moreover, the Linux based mobile platforms market is increasing rapidly and various

open source solutions that offer seamless IP mobility in a device equipped with a mobility ready Linux kernel have been developed.

This work is an extended version of our previous work [6]. In this case, we give a more exhaustive description of the network protocols. Additionally we also include an analytical evaluation and a performance analysis that allows to compare both MIPv6 and PMPv6 in terms of handover latency and signaling cost. With the experimental study of the handover latency, based on the open source implementations of these protocols (UMIP [7] and Open Air Interface PMIPv6 [8] respectively) in the Linux TCP/IPv6 stack, we complete a comprehensive evaluation of both mobility management solutions.

The rest of the manuscript is organized as follows. In Section 2, we provide a background about mobility management and its most recent open source implementations. Then we describe the real infrastructure over which the experimental study has been done. The analytical evaluation is developed in Section 3. The experimental results of handover latency and multimedia results are shown in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

In this section we describe briefly the most representative approaches such as MIPv6 and PMIPv6, and give a review of their more recent open source implementations for the Linux IPv6 stack.

2.1 Mobile IPv6

Until now, Mobile IPv6 is the most representative mobile management scheme developed by the IETF on the way towards next generation mobile networks. It allows nodes to remain reachable while moving around in IPv6 networks. Without specific support for mobility, packets destined to a mobile node would not be able to reach it while the mobile node is away from its home link. In order to continue communication in spite of its movement, a mobile node could change its IP address each time it moves to a new link, but the mobile node would then not be able to maintain transport and higher-layer connections when it changes location.

MIPv6 supports mobility for the MN by providing it with at least two addresses: a Home Address (HoA) which is a fixed address provided by the Home Agent (HA) and Care-of Address (CoA), which is obtained in the foreign access network and changes when MN moves to a new subnet. Fig. 1 illustrates an overview of Mobile IPv6 and its basic terminology. The operation of the protocol is explained next.

When a mobile node stays in the home domain, it is able to receive packets destined to its Home Address and being forwarded by means of conventional IP routing mechanisms. Periodically, or whenever the user attaches to another Access Router (AR), movement detection is performed in order to identify its new point of attachment and a new CoA is acquired.

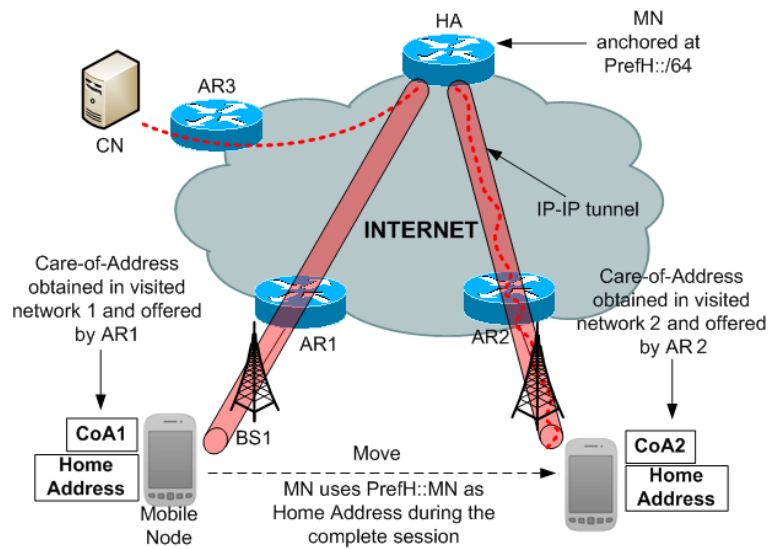


Figure 1. Overview of Mobile IPv6

Once configured with a new CoA, the MN registers with the HA through Binding Update (BU) messages, informing of the user's current location and establishing a tunnel (IP-in-IP or Generic Routing Encapsulation, GRE) between the HA and the MN located in a visited network. The message flow of this registration procedure is shown in Fig 2.

Thus, when the MN is away from home, the HA has a legal mobility binding and it will act as MN's proxy entity. This means that any packet addressed to the MN will end up at the HA because the HA will respond to all Neighbor Solicitation (NS) request for the MN. Once the HA has intercepted a packet, it will encapsulate the packet destination address of the MN's CoA. The MN decapsulates the packet upon its arrival to reveal the original packet, as if the Correspondent Node (CN) had sent it directly to the MN. When the MN has not established a connection with its CN, it should send the packets destined to the CN via the HA using the reverse tunnelling procedure. In this operation, the Home Agent is the critical part of the system since it is on the path of both signaling and data for mobile users.

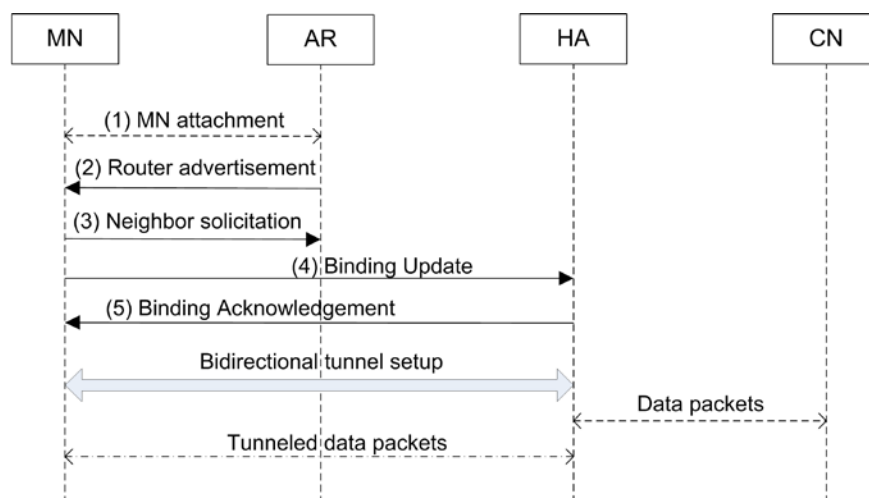


Figure 2. Message flow in Mobile IPv6

2.2 Proxy Mobile IPv6

PMIPv6, the main network-based protocol, is based on MIPv6 in the sense that it extends MIPv6 signaling and reuses many concepts such as HA functionality. The new principal functional entities of PMIPv6 are the Mobile Access Gateway (MAG) and the Local Mobility Anchor (LMA). The MAG typically runs on the AR. Its main role is to detect the MN's movements and initiate mobility-related signaling with the LMA on behalf of the MN. In addition, the MAG establishes a tunnel with the LMA to enable the MN to use an address from its home network prefix and emulates the MN's home network on the access network for each MN. On the other hand, the LMA is similar to the HA in MIPv6. As in the case of MIPv6, the location update and packet delivery procedures in PMIPv6 are described next.

Starting from a generic architecture of PMIPv6 as is shown in Fig. 3, in PMIPv6 the mobility support is offered in a portion of the network called Local Mobility Domain (LMD). When a MN moves into the LMD, it attaches to MAG1, which sends a Proxy Binding Update (PBU) message to LMA to establish a bidirectional tunnel between MAG1 and LMA. This bidirectional tunnel is used for routing the packets to and from the MN. On receiving the PBU message from MAG1, LMA recognizes that the MN is now under MAG1 so that the LMA can use its binding cache entry of the MN for managing the session and routing information. Then the MN receives a Router Advertisement message from MAG1 which includes the Home Network Prefix (HNP) allocated by LMA. The MN creates its address based on the prefix information. The message flow of this registration procedure is shown in Fig. 4.

If the MN moves from MAG1 to MAG2 (see Fig. 3), MAG2 also sends a PBU message to LMA and then a bi-directional tunnel between MAG2 and LMA is created for the MN. Because MAG2 also sends the same HNP to the MN, the MN does not observe any IP level mobility, i.e., its IP address remains unchanged. Thus, the MN can move within LMD without participating in any mobility-related signaling.

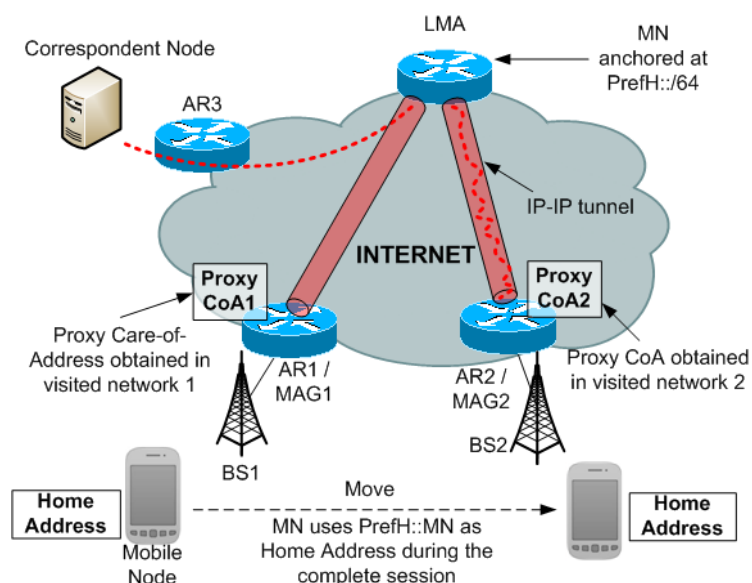


Figure 3. Overview of Proxy Mobile IPv6

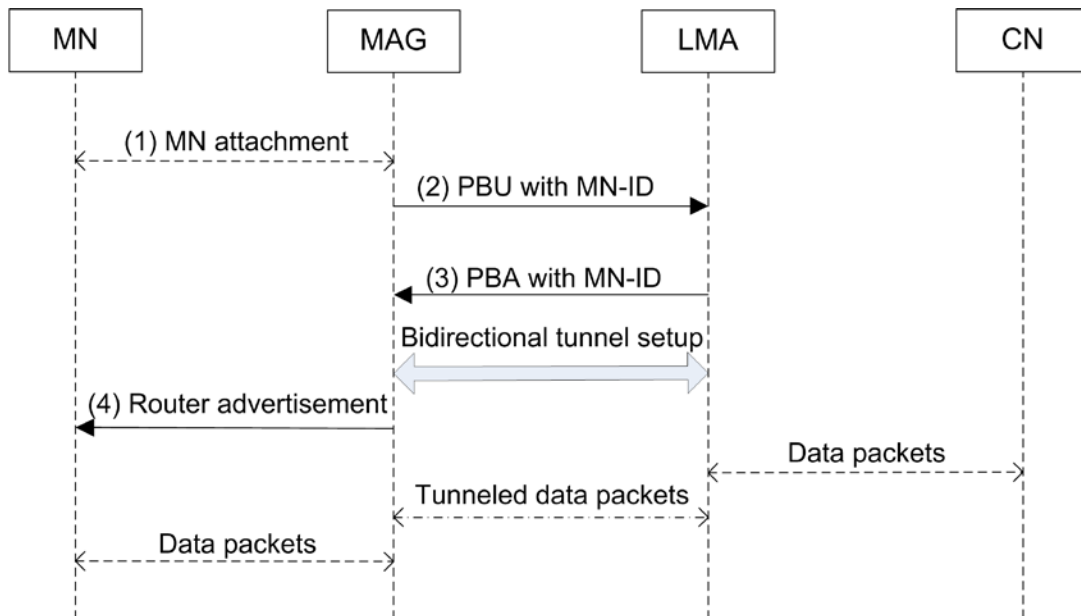


Figure 4. Message flow of the registration procedure in PMIPv6

As it has been described, both MIPv6 and PMIPv6 have similar agents and messages. The main difference between them is that in MIPv6, the tunnel is established between the HA and the MN. So, the IP layer of the MN must be updated in order to operate with the MIPv6 protocol. However, in PMIPv6, the tunnel is created between the MAG and the LMA. This means that the mobile node is not aware of the mobility process and it is not necessary to update its TCP/IP stack. Table 1 shows a comparison of both mobility management protocols.

Previous work related to the handover analysis in develop a framework to analyze the behavior of MIPv6-based [9] and PMIPv6-based protocols [10]. Other works focus on the comparison of MIPv6 and PMIPv6 highlighting the differences in the handover process of both protocols [11]. A study of handover performance is also done in [12] comparing MIPv6, PMIPv6, FMIPv6 (MIPv6 Fast Handovers) and FPMIPv6 (Fast PMIPv6). Both FMIPv6 and FPMIPv6 are combination of MIPv6 and PMIPv6 respectively and link layer triggers to anticipate handover. The numerical analysis confirms that the use of link layer information in FMIPv6 and FPMIPv6 help considerably to reduce the handover latency. Next, we provide an extensive analysis and comparison of these handover latency in MIPv6 and PMIPv6, including an experimental evaluation using their open source implementations.

2.3 Open source implementation

As well as the development of the RFC standards of IP mobility management protocols, where PMIPv6 was designed based on MIPv6, the implementation of both protocols has followed the same timeline and PMIPv6 implementation has been developed based on the MIPv6 code.

Table 1. Comparison of MIPv6 and PMIPv6

	Mobile IPv6	Proxy Mobile IPv6
Mobility management type	Host-based	Network-based
Mobility scope	Global mobility	Localized mobility
Mobility anchor	Home Agent	Local Mobility Anchor
First MN's IP-capable router	Access Router	Mobile Access Gateway
Location update message	Binding Update (BU) message	Proxy Binding Update (PBU) message
MN IPv6 address(es)	HoA and CoA	HoA
TCP/IP stack update	Required	Not required

Linux implementation of mobility is divided into both kernel space and user space. This implementation strategy has been made in order to allow easily the extension of other IP mobility protocols with a minimal kernel side and a user space where different protocols can be implemented. In this way, the changes in the kernel can be kept to a minimum, which means more robust implementations. The kernel side support consists of a single module (*mip6.ko*) that has been newly integrated into the kernel sources since version 3.8.2 and most mobility protocol functionality such as MIPv6 or PMIPv6 are implemented in the user space.

In this work, MIPv6 support is provided by UMIP under a Linux mobility ready kernel. UMIP is an open source implementation of the MIPv6 protocol for the Linux operating system. To support PMIPv6, the Open Air Interface (OAI) PMIPv6 was developed extending UMIP functionality in order to support all necessary PMIPv6 messages and events. In both cases, the user space implements a daemon that takes care of the MIPv6 or PMIPv6 logic, such as tunnelling, binding signaling, security associations and IPv6 extensions, if needed. Some of these features are configured in each agent by means of configuration files that are read by the daemon.

3. Analytic evaluation

In this section we investigate analytically the performance of the most representative mobility management protocols described in the previous section (MIPv6 and PMIPv6). This analysis is based on various evaluation criteria such as the cost functions of handover related metrics, that is, signaling cost of registration update and handover latency.

We analyze the mobility behavior of the MN, keeping in mind a topology where a terminal could move to every neighbor network with the same probability. The parameters to be used are the following:

- t_s Average connection time for a session;
- t_r Average stay time at a visited network;
- N_h Average number of level 3 handover in a session ($N_h = t_s/t_r$);

- s_u Average size of a signaling message for record update;
- h_{x-y} Average number of hops between x and y in the wired network;
- B_w Bandwidth of the wired link;
- B_{wl} Bandwidth of the wireless link;
- L_w Latency of the wired link (propagation delay);
- L_{wl} Latency of the wireless link (propagation delay);
- P_t Routing and processing delay;

On the other hand, $t(s, h_{x-y})$ is the time spent for a packet with size s to be sent from x towards y across wired and wireless links. $t(s, h_{x-y})$ can be expressed in the form:

$$t(s, h_{x-y}) = c + h_{x-y} \cdot \left(\frac{s}{B_w} + L_w \right) + (h_{x-y} + 1) \cdot P_t \quad (1)$$

$$\text{where } c = \begin{cases} \frac{s}{B_{wl}} + L_{wl} & \text{if } x = MN \\ 0 & \text{if } x \neq MN \end{cases} \quad (2)$$

3.1 Signaling Cost

As we have described previously, one of the main functionalities for any IP mobility management protocol is the process of maintaining the MN's mobility session up to date while a MN moves among subnets. Such tasks require control messages that needs to be sent among the mobility agents in the network. Therefore, an important performance metric is the cost associated with it.

In general, a mobility management protocol requires that an MN sends a location update to its mobility anchor whenever it moves from one subnet to another one. This location registration is required even though the MN does not communicate with others while moving. This signaling cost associated with location updates may become very significant as the number of MNs increases. Moreover, this cost depends on the size of the signaling messages and the number of hops in every level 3 handover process during the time interval that the MN communication remains active.

In Mobile IPv6, the registration update with the HA is needed, whereas in PMIPv6, the update is local with the root of the domain. Therefore, the signaling cost (C_s) of MIPv6 and PMIPv6 can be expressed as,

$$C_s^{MIP} = 2 \cdot s_u \cdot h_{MN-HA} \cdot N_h \quad (3)$$

$$C_s^{PMIP} = 2 \cdot s_u \cdot h_{MAG-LMA} \cdot N_h \quad (4)$$

3.2 Handover latency

Other critical metric that has a huge impact in the performance of the system is the

handover latency, T_H , which can be defined as the time interval in which an MN does not have global IP connectivity as a result of a handover. This handover process is caused by the nature of the mobility when an MN changes its point of attachment to the network and a disruption time exists.

The handover delay of mobility protocols can be expressed as a combination of independent factors such as the layer-2 handover time (t_{L2}), the movement detection time (t_{MD}), the IP configuration time (t_{IP}), and the specific mobility signaling delay (t_{sig}) [13]. Thus, the handover latency can be divided into these phases as shown in Fig. 5 and can be expressed as

$$T_H = t_{L2} + t_{MD} + t_{IP} + t_{sig} \quad (5)$$

The operations leading to t_{L2} are heavily dependent on the wireless technology deployed and do not actually depend on the layer-3 mobility protocol. t_{MD} is composed of exchanging the router solicitation (RS) and router advertisement (RA) messages between the MN and the new AR over the wireless link. For fairness, we assume that t_{MD} is the same for all protocols. Moreover, t_{IP} is the time required by the IP stack to configure a new IP address and update the forwarding table.

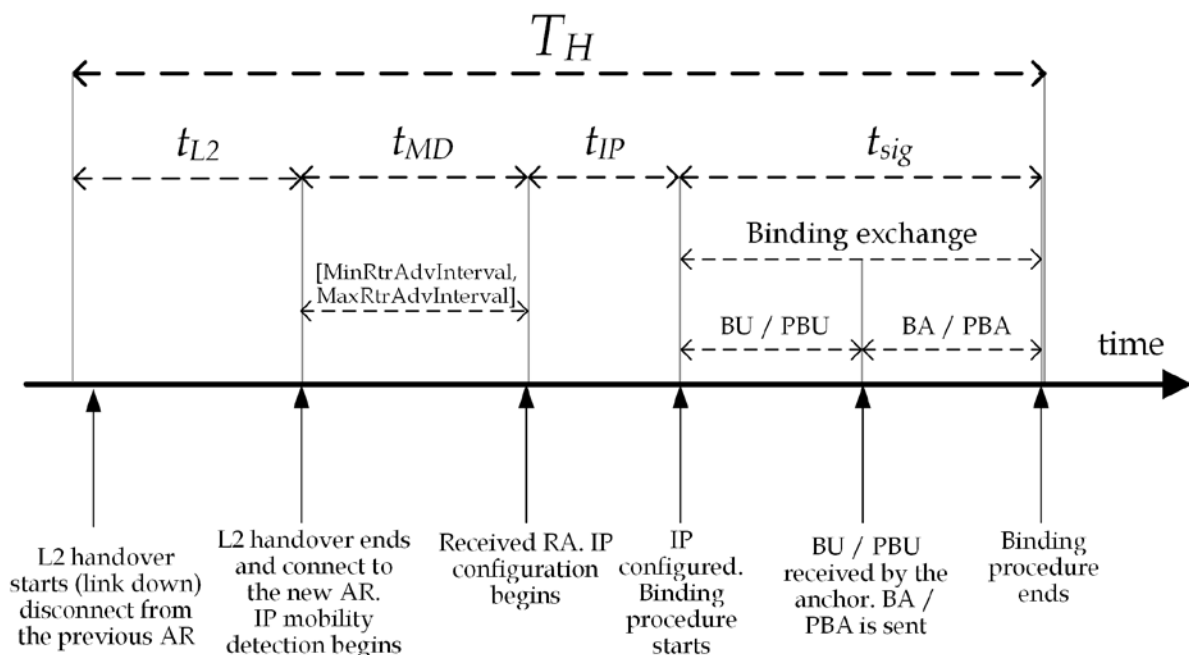


Figure 5. Timing diagram for handover latency

This time depends on the hardware and operating system since this operation is generally performed by the kernel. In network-based protocols, this time is not required as the protocol ensure that the mobile node remain the same IP during its movement. In MIPv6, we assume that this configuration time is very short (negligible), since the operations of configuring a CoA and updating the routing in the IP stack should not require a long time in modern computers and mobile terminals. Thus, for handover latency, only the signaling for registration delay t_{sig} depends on the mobility protocol and its specific procedures.

Considering that T_H^{MIP} represents the handover delay due to the mobility management mechanisms of Mobile IPv6, this metric can be written as

$$T_H^{MIP} = 2 \cdot t(s_u, h_{MN-HA}) \quad (7)$$

With respect to the handover latency in PMIPv6, the behavior is very similar to MIPv6, with the difference that the mobility bindings, needed to configure the correct routing with the mobility anchor, are sent from the MAG to the LMA instead of from the MN. Hence, the T_H^{PMIP} can be expressed as follows

$$T_H^{PMIP} = 2 \cdot t(s_{pu}, h_{MAG-LMA}) \quad (8)$$

3.3 Performance analysis

This section discusses the performance evaluation of MIPv6 and PMIPv6. Fig. 6 shows the comparison of signaling cost of registration update as a function of the cell residence time, which varies from 20 to 140 seconds.

As could be expected, the value of this metric achieves the highest values when the cell residence time is low. In conditions of very high mobility (the cell residence time takes low values), both MIPv6 and PMIPv6 achieve very high values.

With respect to the handover latency, the numerical results for this metric is shown in Fig. 7. In this case, the main difference between the handover latency of the mobility management protocols is the signaling exchanged during handovers, necessary to maintain active ongoing IP flows when the MN changes its point of attachment to the network. Both MIPv6 and PMIPv6, depend on the time needed for establishing a new binding with the HA/LMA agent. Considering that all AR are at the same distance from the centralized anchor, its handover latency is constant.

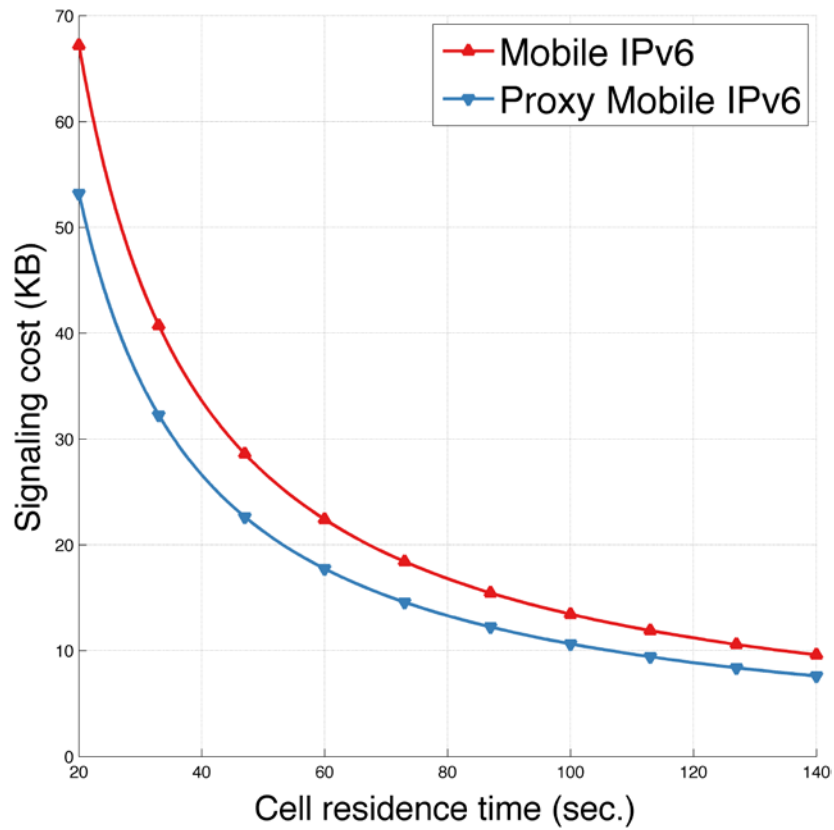


Figure 6. Signaling cost of registration updates

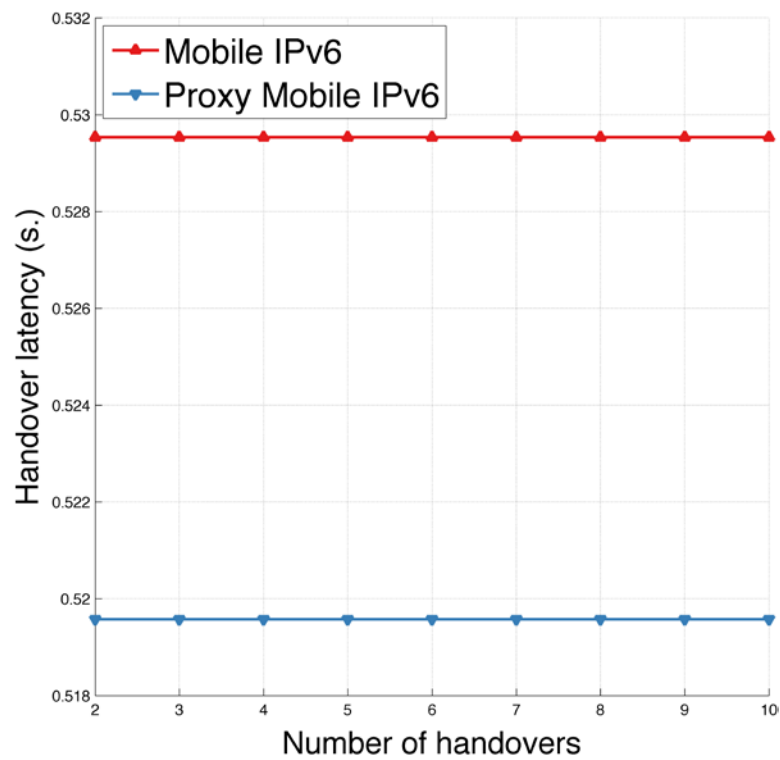


Figure 7. Handover latency

4. Experimental evaluation

In the previous section, we have introduced an analytic evaluation of the main IPv6 mobility management protocols developed by the IETF. With the aim to offer a complete performance and to understand the behavior of real implementations of both MIPv6 and PMIPv6, in this section we report on the experimental evaluation of these protocols through the evaluation of handover latency in a real scenario.

Fig. 8 illustrates the modules of the mobility agents involved in both approaches from an architectural point of view. The bottom layer represents the hardware and the wireless infrastructure; the Operative System (OS) layer includes the kernel space and the module to provide the mobility mentioned previously. In these experimental configurations, Ubuntu 12.04 OS is running in the mobility agents. The Linux 3.8.2 kernel with the mobility support has been compiled and installed in the OS to provide the basis to execute the mobility management daemons.

As can be observed, the two open source approaches are implemented in the Linux user space. MIPv6 approach implements only UMIP as well as the MIPv6 daemon to provide the mobility management in the MN and the HA. The HA also requires the *radvd* daemon to send the Router Advertisement in the Home Network of the MN. The HA is integrated as a router in the access network, so it must exchange the necessary IPv6 routing information with other routers in the network. This function is implemented with Quagga routing software suite. This suite provides the implementation of Routing Information Protocol (RIP) for IPv6 using *zebra* and *ripngd* daemons.

On the other hand, PMIPv6 approach requires the OAI Proxy Mobile IPv6 implementation, in which the PMIPv6 functionality is included. Similarly to UMIP, OAI PMIPv6 provides a daemon that controls the protocol operation in the network entities (LMA and MAG). This daemon is implemented over UMIP and takes advantage of the similarities of both protocols.

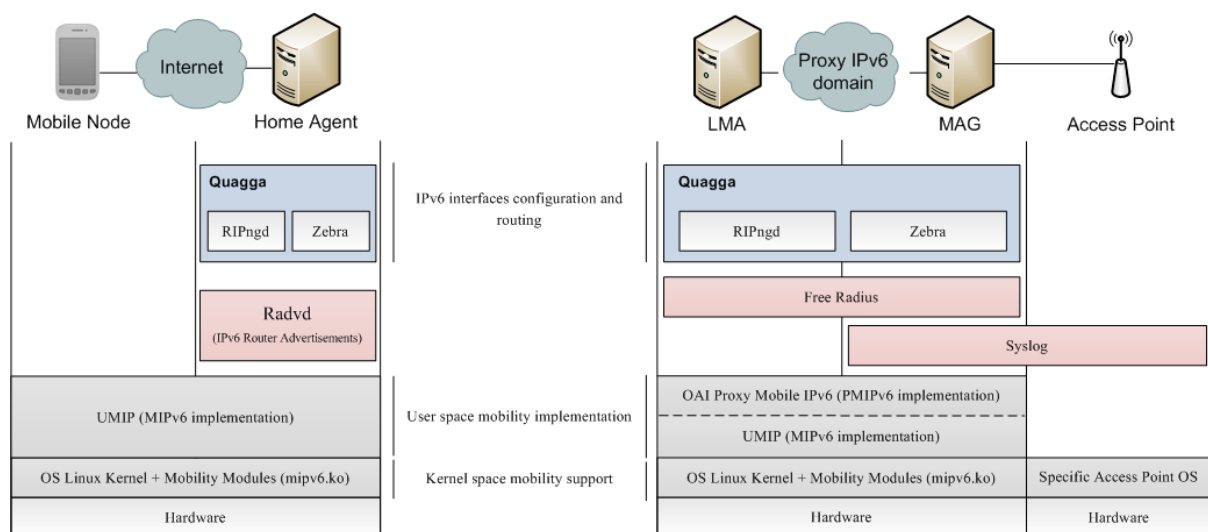


Figure 8. Architecture of the mobility agents in MIPv6 and PMIPv6 approaches

The LMA and MAG agents must also implement Free Radius server and client respectively to manage the IPv6 addresses and also implement the security mechanism to register the users in the access network.

Both LMA and MAG entities are located at the edges of the Proxy IPv6 domain and must implement the Quagga routing software suite in order to exchange the routing information with other access network routers. Finally, The MAG and the access point must implement a *Syslog* server and client respectively to track the movement of the MN in the wireless network and to signal the movements to the LMA using the PMIPv6 Binding Update/ACK messages.

4.1 Testbed scenarios

Fig. 9 depicts the Mobile IPv6 scenario used for the tests made in the experimental study. The MIPv6 scenario consists of several Cisco 1921/K9 routers which support IPv6 and RIPng that interconnect the HA, and the access points which provide the wireless access to the MN. These APs are Cisco Aironet 1130 AG series that support IEEE 802.11b/g specifications.

The PMIPv6 scenario (see Fig. 10) is similar to the previous MIPv6 one with the difference that in PMIPv6 two new agents (LMA and MAGs) must be configured with the OAI PMIPv6 implementation. LMA functionality is similar to HA whereas MAGs are introduced instead of the access routers that serve the MN. In PMIPv6 the MN does not need any additional configuration to its default IPv6 stack.

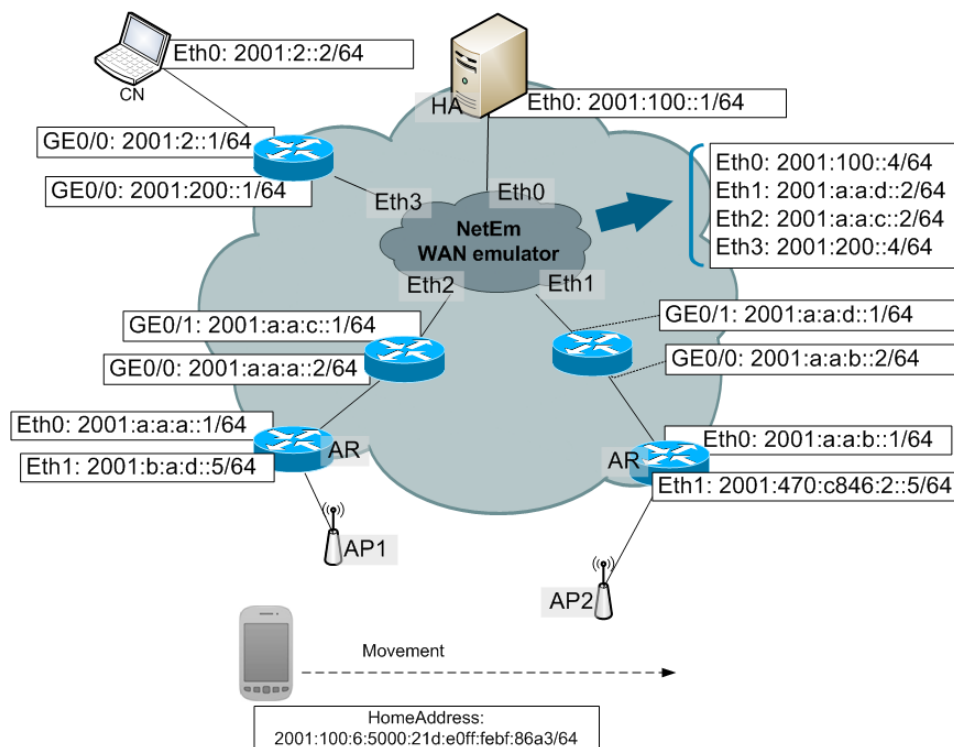


Figure 9. Mobile IPv6 testbed topology

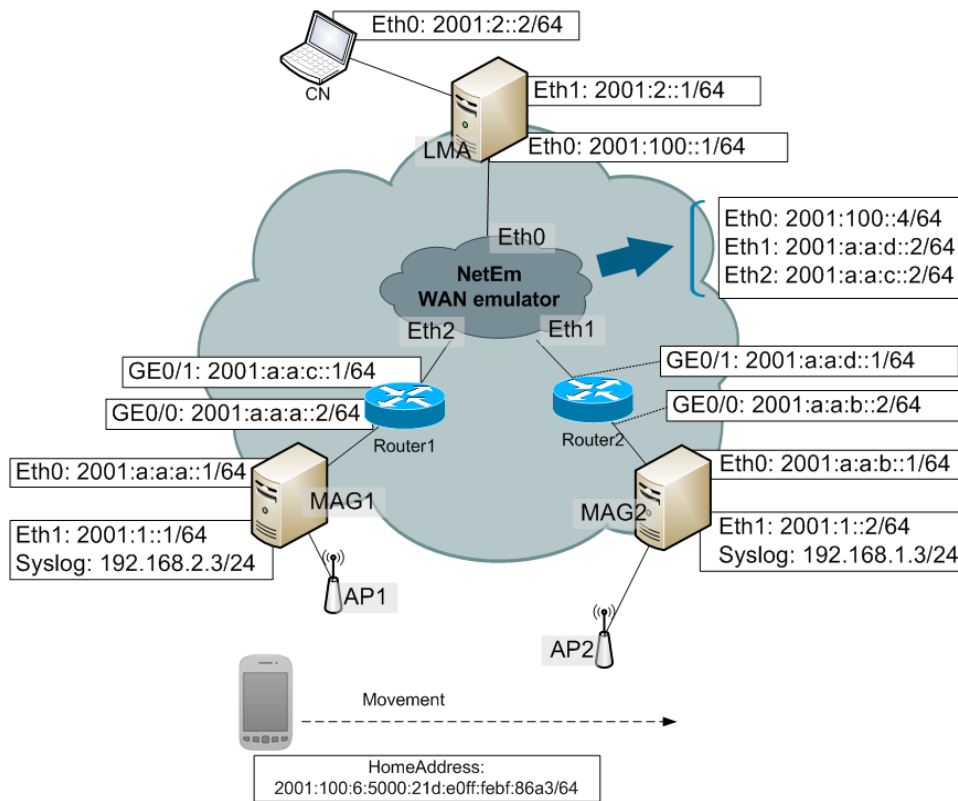


Figure 10. Proxy Mobile IPv6 testbed topology

One important point is the requirement to emulate the conditions of the access network in order to evaluate how the access network characteristics affect mobile communication. For this purpose, the Netem network emulator is used. Netem allows a single Linux PC, set up as a router, to emulate a wide variety of network conditions (e.g. latency, jitter, packet loss,...). Both scenarios are similar so that the experiments obtain comparable results. The MN's handover is performed between two WLAN cells. These cells have enough overlapping surface so there is no possibility of the MN being unable to communicate with either of them. Notice that cell overlap is a requirement for seamless handover. Each WLAN cell belongs to a different IPv6 subnet.

4.2 Quantitative tests

This section provides a description of the test performed, as well as the open source tools used and the measurements performed. The goal of these quantitative experimentations is to evaluate the performance of both MIPv6 and PMIPv6 Linux based open source implementations in terms of handover delay and the behavior of multimedia traffic under different circumstances such as packet delay or packet loss using the topologies observed in Fig. 9 and Fig. 10.

The handover latency (disruption time) is defined as the time elapsed between the last data frame being transmitted/received through the old interface and the first data frame being transmitted/received through the new interface. This time is a common parameter evaluated in a mobility scenario because the handover is one of the most critical processes. There are

several comparative, analytical or simulation based studies of the behavior of handover latency in different mobility management protocols, however we have evaluated and compared the performance of both open source implementations using real infrastructure. The data given in this work allows to measure the gap between analytical or simulated data and real implementations. We have evaluated this parameter by modifying the routing advertisement interval from 0.5 to 4 seconds.

The measurement of the handover latency consisted in sending packets at a high Constant Bit Rate (CBR), so that we could measure the time between the last packet received by the MN before the handover and the first packet received after the movement. The open source tool used to generate CBR traffic was Ostinato, whereas Wireshark was used to analyze the time stamped CBR traffic received by the MN during its movement.

The handover delay for each of the RA intervals in MIPv6 is shown in Fig. 11, while Table 2 represents the numerical values of the handover delay parameter for the MIPv6 and PMIPv6 scenarios with a confidence interval of 95 % for the different router advertisement values. In order to assure the confidence of the results, for each RA value 100 repetitions of the test have been made.

From the results shown in Fig. 11 we can quantify the dependency of the MIPv6 latency with the different Router Advertisement intervals, from 0.5 to 4 seconds. The minimum handover latency has been obtained with a RA interval of 0.5 and is 2.02 sec. The confidence interval for the mean at 95 % limits is short as we can observe in Table 2. As the RA interval increase, the confidence interval also increases due to the variability of the arrival to the MN of the unsolicited RA from the access router. As can be seen, PMIPv6 latency has not been included in Fig. 11. It is necessary to note that PMIPv6 is a network based approach and the home network is responsible for detecting that a new MN has been attached. In PMIPv6, the movement detection mechanism is not dependent on the RA messages. Table 2 demonstrates that the open source implementation of the protocol follows this behavior. The result obtained of the mean handover delay value for the 800 repetitions is, 2.788 seconds. The 95 % confidence interval for this mean ranges from 2.788 to 2.790. It is worthwhile mentioning that the results obtained in our experiments for PMIPv6 are significantly higher from the considered in our theoretical analysis. The reason behind this behavior is that the software implementation requires the LMA to remove the tunnel that was being used before doing any further processing of a received PBU.

The difference between these results and the numerical values shown in the Analytic evaluation section, reflect that in real environments, more mechanisms need to be done during the movement of the mobile nodes. Additionally, the movement detection used in the real scenario is performed by the IP layer. Fig. 12 and Fig 13 show the signaling exchange in the MIPv6 and PMIPv6 scenarios respectively. This means that IPv6 messages such as Routing Advertisement (RA) are used instead of layer-2 information. In the testbed, the access router periodically multicasts unsolicited RA messages.

Table 2. Values of the handover latency intervals with a confidence interval of 95 %

RA interval (s.)	Hand. Interval in MIPv6 (s.)		Hand. Interval in PMIPv6 (s.)	
	<i>Min.</i>	<i>Max.</i>	<i>Min.</i>	<i>Max.</i>
RA interval = 0.5	2.020	2.149	2.788	2.788
RA interval = 1	2.287	2.477	2.789	2.790
RA interval = 1.5	2.449	2.697	2.779	2.780
RA interval = 2	2.810	3.055	2.783	2.784
RA interval = 2.5	2.944	3.218	2.781	2.784
RA interval = 3	3.252	3.624	2.794	2.797
RA interval = 3.5	3.485	3.864	2.807	2.811
RA interval = 4	3.540	4.078	2.790	2.790

To improve this problem, new optimized mechanisms to detect the connection are being developed. That is the case of a mechanism known as IEEE 802.21 Media Independent Handover. The inclusion of this protocol is a future work in this scenario to achieve smaller delays in the movement detection and, therefore, in the handover latency.

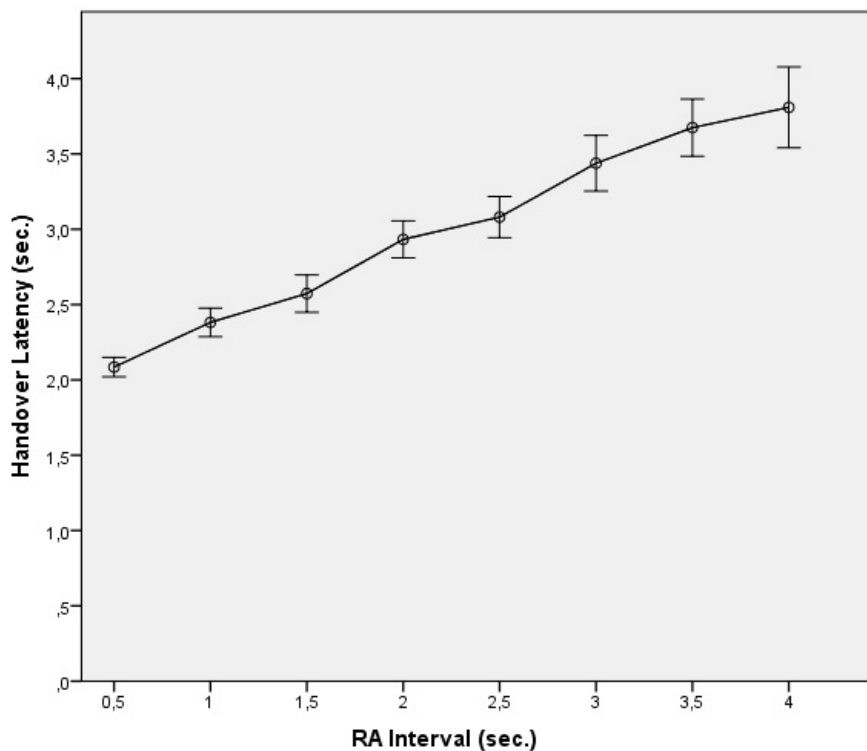


Figure 11. Handover latency in MIPv6 experiments

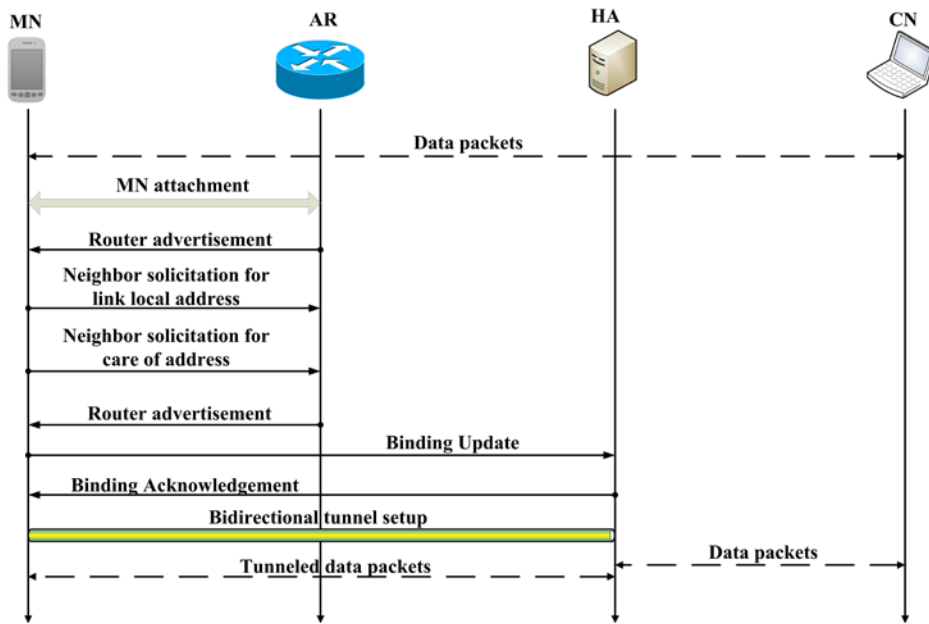


Figure 12. Signaling exchange during a handover in a real MIPv6 scenario

As we previously mentioned, we have also evaluated the effect caused by IP mobility in multimedia communications. In this case, two types of multimedia transmissions have been considered and compared under different conditions. On the one hand, UDP real time communications based on RTP (Real Time Protocol) and, on the other hand, TCP multimedia streaming. Some network situations can affect the UDP multimedia stream such as network congestion, packet loss during handover and RTP packets arriving out of the playout time. In the TCP streaming case, the multimedia server sends the stream flow content to the receiver that is buffered in the client side. This mechanism avoids packet loss and minimizes the delay and jitter effects in the stream flow.

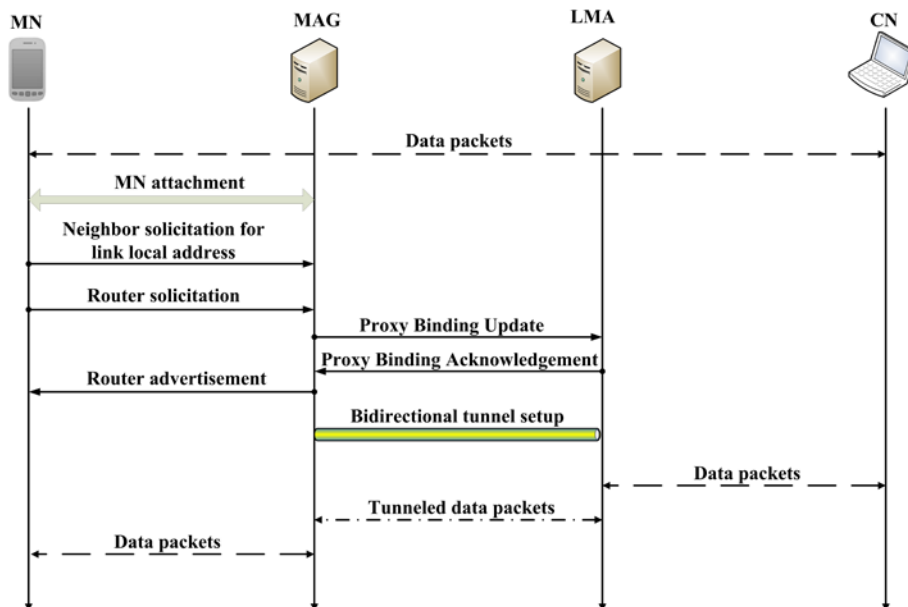


Figure 13. Signaling exchange during a handover in a real PMIPv6 scenario

For each type of multimedia traffic, several parameters have been modified in the MIPv6 access network or in the PMIPv6 domain in order to evaluate its effect on the overall performance of the communication. These parameters are delay and packet loss, which were introduced by Netem emulator.

The experiments consist of the same movement of a terminal as in our testbed (causing a handover) during a real time or a streaming multimedia transmission under the different aforementioned conditions. The multimedia transmission is the Open Source Film Big Buck Bunny (duration, 120 s., bitrate approximately: 1626 kbps, resolution: 640x360). A handover is performed in each test at second 53 approximately. In order to measure the transmission and to evaluate the received video quality by the MN, the Peak Signal to Noise Ratio (PSNR) indicator has been used. PSNR is generally considered to be a reference benchmark for developing objective perceptual video quality assessment models [14]. In this case, the reference model required by PSNR is the video transmitted in the testbed without the influence of the mobility management protocol and without any access network parameter modification. This reference is compared with the video received by the MN during the test using one of the evaluated mobility management protocols and changing the access network parameters. The MSU Video Quality Measurement Tool was used for the PSNR evaluation.

Fig. 14, and Fig. 15 show the behavior of both real time and streaming multimedia communications respectively in MIPv6. In this case, just three representative tests are present in the graph. The first one is a communication without any access network parameter modification, the access network in the second one introduces a delay of 100 ms, and finally, a test with a packet loss of 10 %. The result of the PSNR evaluation of all the experiments is presented in Fig. 16.

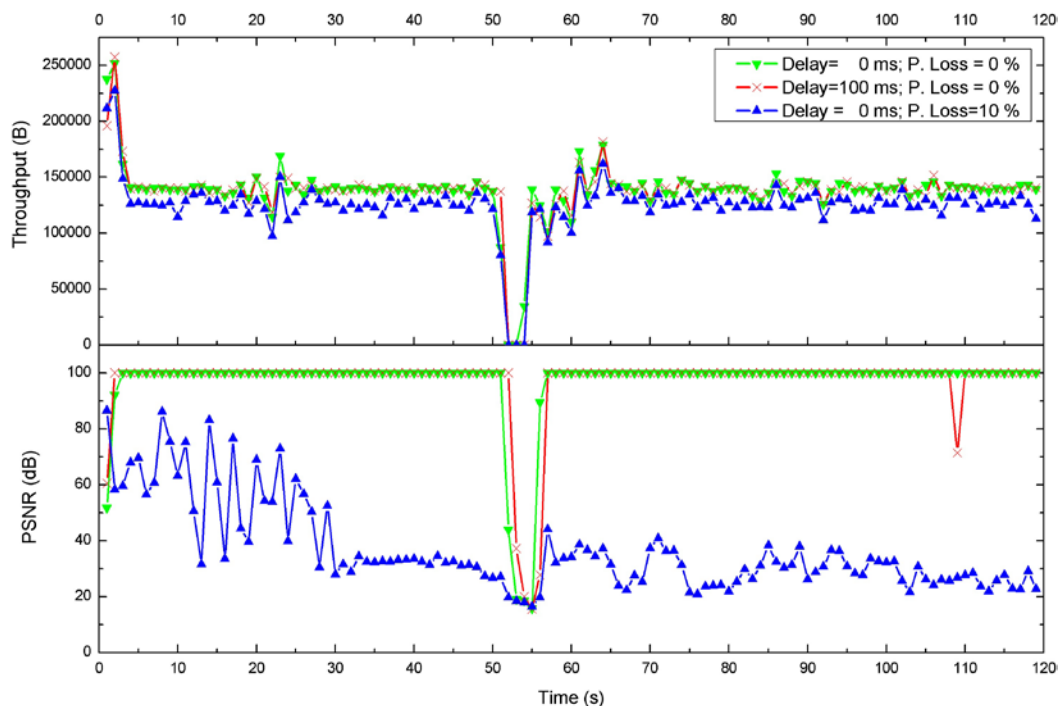


Figure 14. Throughput and PSNR in real time experiments

In Fig. 15 we can observe a different behavior by the PSNR. In this case, three TCP streaming communications are compared in the same conditions as stated before. The streaming application provides a buffer that avoids packet loss and reduces the delay and the jitter of the packets. As can be observed in Fig. 15, when the delay is high the application must wait to recompose the video transmission and the PSNR is penalized due to long waiting times. In contrast, when the packet loss is high, the video frames are retransmitted by the transmitter and the PSNR value is higher compared with the real time transmission. In this case there are no waiting times and the PSNR is not penalized. This implies that PSNR parameter in streaming transmission decreases when the delay of the access network increases. As could be observed, although the handover is produced, the PSNR is not affected because of the buffer. Only when the delay is increased, the PSNR decreases.

Fig. 16 shows the different experiments made in both MIPv6 and PMIPv6 testbed using UDP real time and TCP streaming communications. In each of these four categories, eight tests have been made and each one represents a PSNR value in the plot (5 repetitions have been made for each PSNR value). Regular case refers to tests made with the configuration of Netem at Delay = 0 and without packet loss. The behavior of MIPv6 corresponds to the one explained previously in Fig. 14 and Fig. 15 whereas in PMIPv6 we can observe some differences. Real time transmission in the PMIPv6 testbed demonstrates the same behavior as in MIPv6 when the delay of the access network increases. When delay exceeds 75 ms., the PSNR decreases because the packets arrive too late to the MN and the video frames are discarded by the player as the time limit to be reproduced has been exceeded. In our experiments, when the handover occurs, the serving MAG of the visited network introduces long delays to forward packets to the MN.

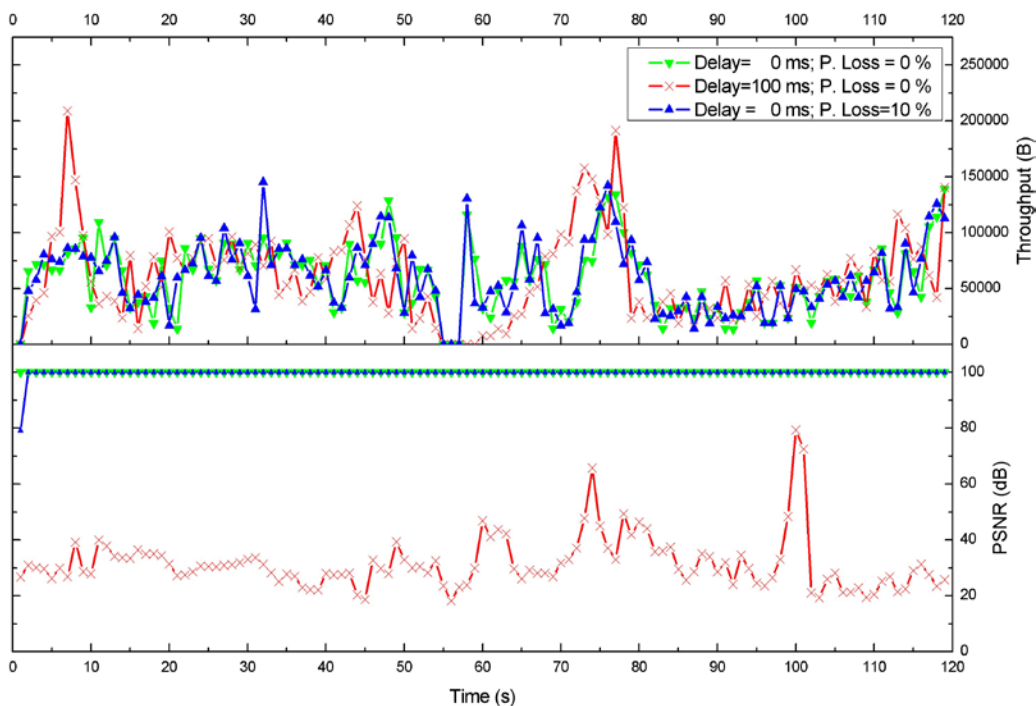


Figure 15. Throughput and PSNR in TCP streaming experiments

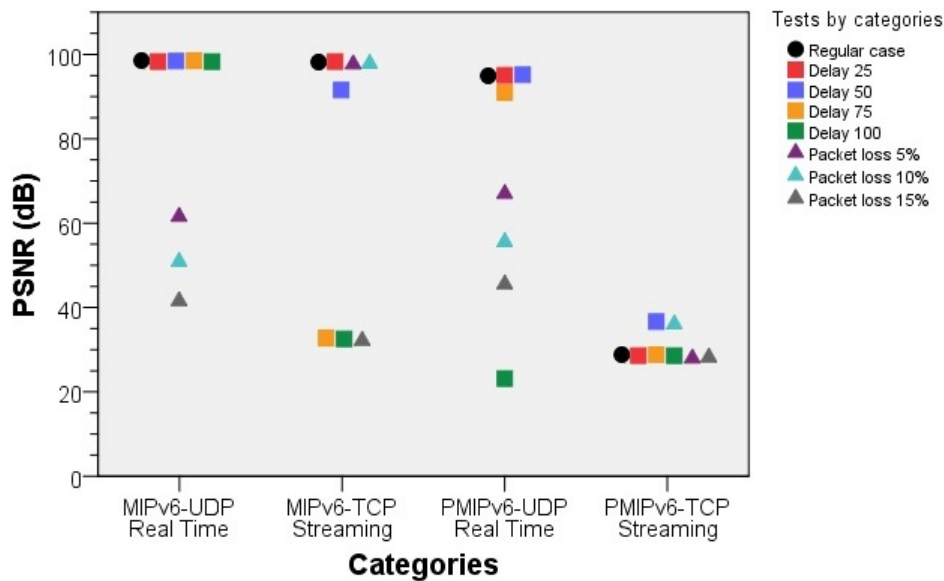


Figure 16. Summarize of the obtained PSNR results in the experiments conducted

These latencies penalize the communication and the PSNR values. The same effect is produced in PMIPv6 streaming, where the PSNR achieves low values because the reproduction pauses until it has enough information to continue the reproduction due to buffer starvation. Despite of the PSNR values, that reflect the comparison between the original video and the received one, the videos seems to be reproduced at good quality.

5. Conclusion and Future Work

Testbed are, by far, the most realistic method for evaluating performance, as they practically use the protocol implementations and the hardware that is the same, or very similar, to the one used for the production networks. However, large testbed are expensive to build and manage and for very large (or highly mobile) networks, practically impossible to implement. Furthermore, the degrees of freedom in a testbed is significantly reduced in comparison with a mathematical analysis or network simulation. Assuming these limitations, in this paper we have conducted a study related to the performance of real IPv6 mobility management implementations in the IPv6 Linux stack, evaluating the current state of these protocols by means of an experimental testbed.

We have focused on the most representative IETF solutions: MIPv6 and PMIPv6. Numerical results have been obtained from both analytical and real implementations under different network conditions. Quantitative results have shown the handover latency produced by both open source implementations. We quantify the dependency of the MIPv6 handover delay in the Routing Advertisement interval, whereas the PMIPv6 implementation is not dependent of RA messages, as expected. Moreover, the PSNR video quality indicator has been used to evaluate the multimedia transmission. It should be noted that streaming traffic in the PMIPv6 testbed gives low PSNR values due to the short additional delays produced by

the MAG which penalizes the PSNR value because some wait times are introduced due to buffer starvation.

Although the open source implementations of MIPv6 and PMIPv6 are operational, the results presented in this work raise the need to improve the software implementations of both protocols, especially the tunnel management in PMIPv6 and also the need of link-layer mechanisms that can assist upper layers during the movement detection phase.

Thus, future work focuses on handover optimizations enabled by the use of the IEEE 802.21 Media Independent Handover Services. Additionally, future mobility protocols will require a more flexible and dynamic behavior. New mechanisms will require the combination of both MIPv6 and PMIPv6. This hybrid solution could improve the handover delay, by alleviating the drawbacks of each protocol while keeping their advantages.

Acknowledgement

This work is partly financed by the European Regional Development Fund Programme (2014-2020) through Computing and Advanced Technologies Foundation of Extremadura (COMPUTAEX).

References

- [1] Cisco, Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019, White Paper, Cisco, February 2015.
- [2] C. Perkins, D. Johnson, and J. Arkko, "Mobility Support in IPv6," RFC 6275, IETF, July 2011. <http://dx.doi.org/10.17487/RFC6275>
- [3] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6," RFC 5213, IETF, August 2008, updated by RFC 6543. <http://dx.doi.org/10.17487/RFC5213>
- [4] J. C. Zuniga, C. J. Bernardos, A. De La Oliva, T. Melia, R. Costa, and A. Reznik, "Distributed mobility management: a standards landscape," IEEE Communications Magazine, 51(3):80–87, March 2013. <http://dx.doi.org/10.1109/MCOM.2013.6476870>
- [5] C. DiBona and S. Ockman. "Open sources: Voices from the open source revolution," O'Reilly, 1999.
- [6] J. Carmona-Murillo, D. Cortés-Polo, J. Calle-Cancho, J. L. González-Sánchez, F. J. Rodríguez-Pérez, "Experimental evaluation of mobility management protocols for multimedia communications," XII Jornadas de Ingeniería Telemática, pp. 293-299, 14-16 October 2015.
- [7] UMIP. "Mobile IPv6 and NEMO for Linux," <http://umip.org>, May 2015.
- [8] EURECOM, "Open Air Interface Proxy Mobile IPv6 (OAI PMIPv6)," <http://www.openairinterface.org/openairinterface-proxy-mobile-ipv6-oai-pmipv6>, Aug. 2013.
- [9] C. Makaya, S. Pierre. "An Analytical Framework for Performance Evaluation of IPv6-Based Mobility Management Protocols," IEEE Transactions on Wireless

- Communications. 7(3): 972-983, 2008. <http://dx.doi.org/10.1109/TWC.2008.060725>
- [10]H. Ali-Ahmad, M. Ouzzif, P. Bertin, and Xavier Lagrange. “Performance Analysis on Network-Based Distributed Mobility Management,” *Wireless Personal Communications*, 74(4): 1245-1263, 02/01, 2014. <http://dx.doi.org/10.1007/s11277-013-1575-0>
- [11]Ki-Sik Kong, W. Lee, Y-H. Han, M-K Shin and H. You, “Mobility management for all-IP mobile networks: Mobile IPv6 vs. Proxy Mobile IPv6,” *IEEE Wireless Communications*, vol. 15, no. 2, pp. 36-45, April 2008. <http://dx.doi.org/10.1109/MWC.2008.4492976>
- [12]J.-H. Lee, J.-M. Bonnin, I. You, and T.-M. Chung, “Comparative Handover Performance Analysis of IPv6 Mobility Management Protocols,” *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 3, pp. 1077-1088, March 2013. <http://dx.doi.org/10.1109/TIE.2012.2198035>.
- [13]A. de la Oliva, I. Soto, M. Calderón, C. J. Bernardos, M. I. Sánchez, “The costs and benefits of combining different IP mobility standards,” *Computer Standards & Interfaces*, vol. 35, no. 2, February 2013. <http://dx.doi.org/10.1016/j.csi.2012.08.003>
- [14]Q. Huynh-Thu and M. Ghanbari, “The accuracy of PSNR in predicting video quality for different video scenes and frame rates,” *Telecommunication systems*, vol. 49, no. 1, pp 35-48, January 2012. <http://dx.doi.org/10.1007/s11235-010-9351-x>

Copyright Disclaimer

Copyright reserved by the author(s).

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).