UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria
Corso di Laurea in Ingegneria delle Telecomunicazioni

Tesi di Laurea

IP Flow Mobility support for Proxy Mobile IPv6 based networks

Relatori:
Prof. Michele Zorzi
Dott. Telemaco Melia

Candidato:
Fabio Giust

Anno accademico 2010-2011

Summary

The ability of offloading selected IP data traffic from 3G to WLAN access networks is considered a key feature in the upcoming 3GPP specifications, being the main goal to alleviate data congestion in cellular networks while delivering a positive user experience. Lately, the 3GPP has adopted solutions that enable mobility of IP-based wireless devices relocating mobility functions from the terminal to the network. To this end, the IETF has standardized Proxy Mobile IPv6 (PMIPv6), a protocol capable to hide often complex mobility procedures from the mobile devices.

This thesis, in line with the mentioned offload requirement, further extends PMIPv6 to support dynamic IP flow mobility management across access wireless networks according to operator policies. In this work, we assess the feasibility of the proposed solution and provide an experimental analysis based on a prototype network setup, implementing the PMIPv6 protocol and the related enhancements for flow mobility support.

La capacità di spostare flussi IP da una rete di accesso 3G ad una di tipo WLAN è considerata una caratteristica chiave nelle specifiche future di 3GPP, essendo il principale metodo per alleviare la congestione nelle reti cellulari mantenendo al contempo una ragionevole qualità percepita dall'utente. Recentemente, 3GPP ha adottato soluzioni di mobilità per dispositivi con accesso radio basato su IP, traslando le funzioni di supporto dal terminale alla rete, e, a questo scopo, IETF ha standardizzato Proxy Mobile IPv6 (PMIPv6), un protocollo studiato per nascondere le procedure di mobilità ai sistemi mobili.

Questa tesi, in linea con la citata esigenza di spostare flussi IP, estende ulteriormente PMIPv6 per consentire il supporto alla mobilità di flussi tra diverse reti di accesso wireless, assecondando le regole e/o politiche definite da un operatore. In questo lavoro, ci proponiamo di asserire la fattibilità della soluzione proposta, fornendo un'analisi sperimentale di essa sulla base di un prototipo di rete che implementa il protocollo PMIPv6 e le relative migliorie per il supporto alla mobilità di flussi.

List of Abbreviations

3GPP	3rd Generation Partnership Project
AA	Agent Advertisement
AAA	Authentication, Authorization and Accounting
AP	Access Point
AS	Agent Solicitation
ATT	Access Technology Type
BA	Binding Acknowledgment
BC	Binding Cache
BCE	Binding Cache Entry
BU	Binding Update
BUL	Binding Update List
BULE	Binding Update List Entry
CM	Connection Manager
CN	Correspondent Node
CoA	Care-of Address
DAD	Duplicate Address Detection
DL	Downlink
DPI	Deep Packet Inspection

DSMIPv6	Dual-Stack Mobile IPv6
FA	Foreign Agent
FHRP	Flow Handover Reply
FHRQ	Flow Handover Request
FM	Flow Manager
GGSN	Gateway GPRS Support Node
HA	Home Agent
HI	Handoff Indicator
HMIPv6	Hierarchical Mobile IPv6
HNP	Home Network Prefix
HO	Handover or Handoff
HoA	Home Address
IANA	The Internet Assigned Numbers Authority
IEEE	The Institute of Electrical and Electronics Engineers
IETF	The Internet Engineering Task Force
IF	interface
ISATAP	Intra-Site Automatic Tunnel Addressing Protocol
ISP	Internet Service Provider
LMA	Local Mobility Anchor
LMD	Localized Mobility Domain
MAC	Medium Access Control
MAG	Mobile Access Gateway
MH	Mobility Header

MIPv4	Mobile IPv4
MIPv6	Mobile IPv6
MN	Mobile Node
MN-ID	Mobile Node Identifier
MuHo	Multi-homing
NAT	Network Address Translation
ND	Neighbor Discovery
NetLMM	Network-based Localized Mobility Management
PBA	Proxy Binding Acknowledgment
PBU	Proxy Binding Update
P-CoA	Proxy Care-of Address
PDP	Packet Data Protocol
PMIPv6	Proxy Mobile IPv6
PPP	Point-to-Point Protocol
QoE	Quality of Experience
QoS	Quality of Service
RA	Router Advertisement
RD	Request Dispatcher
RO	Route Optimization
RR	Return Routability
RS	Router Solicitation
SLAAC	StateLess Address Auto-Configuration
UL	Uplink

Contents

List of Figures

Chapter 1

Introduction

The exponential growth in mobile data applications and the resultant increase of traffic volume in 3G data networks has placed mobile operators in the challenging position – particularly when licensed spectrum is limited – of supporting large amounts of traffic chunks. With much of this increased IP data traffic directly attributable to the availability of affordable smart-phones featuring both 3G and WLAN access, mobile operators are now looking at WLAN networks as a low cost alternative to offload data from their 3G infrastructure. Offloading alleviates data congestion in cellular networks while delivering a positive user experience.

A first approach to the problem could be to perform an inter-technology handoff whenever WLAN connectivity becomes available, with all the traffic routed through the WLAN access. However, having the capability to move selected IP traffic (i.e., HTTP, video, etc.) while supporting simultaneous 3G and WLAN access seems a more appealing solution. In this environment, mobile operators can develop policies for IP flow mobility, and control which traffic is routed over the WLAN and which one is kept on the 3G. For example, it seems reasonable that some IP flows (e.g., related to VoIP) are sent over 3G to benefit from its QoS capabilities, while IP flows related to "best-effort" Internet traffic can be moved to the WLAN access. Inter-working between 3G and WLAN access networks is not a new topic by itself, however the availability of smart-phones, netbooks and tablet PCs to the mass market and the proliferation of new Internet-based applications running on these terminals renewed the interest by mobile operators in the subject.

Lately, we have been assisting to the development of new solutions that enable IP mobility of wireless devices within a local domain by means of special purpose functions installed in network components. We refer to these solution as network-based mobility management, as opposed to host-based mobility management – e.g., Mobile IPv6 (MIPv6) [1], Dual-Stack Mobile IPv6 (DSMIPv6) [2].

Network-based Localized Mobility Management (NetLMM) [3] allows conventional IP devices to roam across wireless access networks without the support of mobility clients. This is an appealing feature from the service provider's viewpoint, since it enables mobility support without strong dependence on software and complex mobility related configuration in the user terminals. To this end, the Internet Engineering Task Force (IETF) has standardized Proxy Mobile IPv6 (PMIPv6) [4]. However, current specifications only provide mobility management at the granularity of interfaces, meaning that the network is only able to move all the communications associated with a particular interface (IF) of a mobile node, but they do not consider more granular management strategies.

This work focuses on the design and implementation of flow mobility extensions for PMIPv6, driven by the indications specified in [5]. The thesis describes the functional components required in the network to support smart traffic steering while minimizing the impact on the mobile devices and augmenting user Quality of Experience (QoE). In our proposal, the network (in particular the mobility anchor) is the decision control entity. It performs flow mobility based on network operator policies, which may dynamically react upon the network load.

Flow mobility can be enabled when the terminal is connected to the network through more interfaces at a time, thus multiple links are available (we refer to this situation as *multi-attachment*). We consider two different types of mobile devices:

1. terminals with a single interface visible from the IP stack, where the link-layer hides the use of multiple physical interfaces as in [6, 7];

2. terminals with multiple IP interfaces visible to the upper layers where the IP stack behaves according to the *weak host* model [8, 9].

Our customized PMIPv6 protocol stack has been extended to support both types of terminals and an experimental evaluation has been carried out. The experimental results demonstrate the viability of performing flow mobility in network-based mobility management scenarios.

This work is mainly based on an article by the same author in collaboration with Dr. Telemaco Melia, Prof. Carlos J. Bernardos, Dr. Antonio de la Oliva and Prof. Maria Calderon, entitled "IP flow Mobility in PMIPv6 Based Networks: Solution Design and Experimental Evaluation", accepted for publication in the 2011 special issue of the Springer journal *Wireless Personal Communication* [10]. In the article, authors reported the same solution and results presented here, but they omitted the implementation details. Alternatively, they investigated an interesting point, not covered in the present work, concerning whether the simultaneous use of two or more wireless interfaces can be a blocking factor to the wide adoption of seamless IP flow mobility management, due to the additional battery consumption. To show its feasibility, authors have analyzed the energy consumption of a simultaneous use of multiple network interfaces, focusing on WLAN and 3G access. The tests, conducted on an experimental platform, successfully demonstrate the feasibility of the approach.

The solution designed to allow flow mobility in a PMIPv6 architecture proposed in this thesis has been validated through the implementation of a prototype testbed in the Alcatel-Lucent Bell Labs[1] laboratories in the site of Villarceaux near Paris. The prototype has been improved and enhanced with an interactive graphical interface by a joint research team accounting InterDigital[2] and Alcatel-Lucent as partners, and then presented in a demo stand at the *World Mobile Congress* held in Barcelona in February 2011[3]. Moreover, the original prototype has been shown in a demo stand at the *Future Networks & Mobile Summit* held in Warsaw in June 2011[4], where the multimode features of a mobile terminal were exploited to deliver the different layers of an SVC[5] video file through all terminal's interfaces.

The remainder of this work is structured as follows:

- Chapter 2 presents some IP mobility protocols currently standardized by the IETF, divided in two main categories reflecting the approach followed in the development. Thus *host-based* and *network-based* solutions are given, with a particular focus on the latter, as the protocol chosen for our study, i.e., Proxy Mobile IPv6, belongs to this category;

- Chapter 3 first introduces the concepts of *multihoming* and *flow mobility* in a general way, and then elaborates them in the PMIPv6 context, providing some state-of-art solutions proposed so far;

- Chapter 4 is devoted to the description of our flow mobility solution in a PMIPv6 domain, going through the necessary extensions applied to the legacy PMIPv6 protocol;

- Chapter 5 reports the testbed setup built to show the feasibility of the solution, giving an insight of all the functional blocks implemented in all the nodes comprised in the overall architecture;

- Chapter 6 is dedicated to the description of the tests conducted over the platform and the evaluation of the results obtained after the execution of such experiments;

[1] http://www.alcatel-lucent.com

[2] http://www.interdigital.com

[3] http://www.mobileworldcongress.com/
Additional information can be found in the following article:
http://www.businesswire.com/news/home/20110209006805/en/InterDigital-Mobile-World-Congress-2011-Showcasing-Suite

[4] http://www.futurenetworksummit.eu

[5] Scalable Video Coding, Annex G extension of the H.264/MPEG-4 AVC standard, available at http://www.itu.int/rec/T-REC-H.264-201003-I/en

- Chapter 7 concludes the thesis, highlighting the most relevant part of the extended design, the results obtained and the contributions of the work in international conferences and expositions.

Chapter 2

IETF Mobility Protocols

The scientific progress during the last 10 years has lead to a massive penetration in our daily lives of two particular technologies, that nowadays are deeply linked each other. On the one hand, we have the huge proliferation of light devices with high computing power and (often multiple) radio connections, as, for instance, laptops, netbooks and smartphones. One the other hand, the last decade has witnessed an amazing growth of the Internet in our society, accessed through cable or wireless, and becoming a main actor for the provision of a vast amount of different services.

These two technologies in conjunction have risen the need for many users to be "on-line" regardless their physical location, or despite the fact that they are moving, "anywhere, anytime". Unfortunately, a change of point of access might have as a consequence the non-reachability of the IP address configured by the terminal, because it becomes not topologically correct in the new access network. It is obvious that the packets carrying that address are lost, as they are still delivered through the old path towards the former access network. Ongoing sessions would be recovered if packets carried the new address configured by the terminal, but, however, the sender might not be aware of the new recipient's address.

With this simple description we have just introduced the dual role of the IP address: it is an *identifier*, as it names a node in a network, and it is also a *locator*, as it allows the routing infrastructure to deliver packets to that node. Thus, for a moving terminal, it would be necessary to change the location and to keep the same name, with a clear conflict between the two requirements. If simply the IP address is changed, without any expedient for the host name, we solve what is known as *portability*: the connectivity is maintained, but ongoing sessions need to be refreshed or restarted, usually by manual intervention. Nevertheless, some applications, like web browsing or e-mails, do not suffer excessively such a disruption, but some other applications, like VoIP communications or real time gaming, cannot survive an IP address change without producing a considerably disturbing the service quality perceived by the user. This issue is known as *mobility*, that refers to the possibility of keeping active ongoing sessions in a seamless manner for the user (either human or an application).

The mobility support can be offered at different layers of the TCP/IP stack, each approach has its advantages and disadvantages. The following list provides a rationale for applying mobility at the IP layer.

- **Physical/Link layer.** It provides fast and seamless handover, but a dedicated solution needs to be designed for each technology (e.g., cellular networks, IEEE 802.11, etc.). Moreover, form the IP layer point of view, such a solution can be applied only when the terminal is roaming within the same subnet.
- **Network layer.** This is the only layer providing a common framework to what resides at upper or lower layers, thus, in principle, only a single protocol is required. However it is not straightforward to design optimally such protocol.
- **Transport layer.** It would require a solution per each transport protocol (e.g., mSCTP [11]). Moreover, the most common transport protocols, as UDP and TCP, are not designed with this requirement in mind and thus they would suffer consistent changes.
- **Application layer.** Some applications are developed with mobility features, but most of them are not. Thus, this approach would require to write new applications (or upgrade the old ones) with this extra functionality.

The remainder of this chapter describes the solutions standardized by the IETF community for IP mobility, stressing the two mainstream approaches being investigated so far, i.e., the host-based set of solutions (we focus on Mobile IPv4 and Mobile IPv6 only) and the network-based approach. The latter, in the Proxy Mobile IPv6 flavor, is the architecture chosen for the extensions and enhancements proposed by the whole thesis.

2.1 Mobile IPv4

The solutions developed in the late '90 by Charlie Perkins and other researchers to address the mobility issue in IPv4 networks, resulted in the first mobility protocol standardized in 2002 by the IETF community, with the name of *IP Mobility support for IPv4*, (see RFC 3344, [12]). This solution is rather a theoretic exercise, as it has not seen a real commercial deployment so far, but, still, it is a very important milestone, as it was taken as starting point for the extensions and changes that came later with other protocols and ideas.

The principle is that a moving terminal configures a permanent globally reachable address when it is in its home network, where a special node is in charge of mapping this permanent address with a temporary one, configured by the the Mobile Node (MN) when it is away. The fundamental concepts of the protocol are already there: next subsections detail the entities involved and the operations. Note that most of the terms introduced in the following paragraphs are common to more mobility protocols, and thus are used throughout the work with that same meaning, except when stated otherwise.

2.1.1 Entities

Here is the list of nodes comprised in the MIPv4 architecture.

- **Mobile Node (MN)**. It is the moving host, usually referred as a terminal that changes point of attachment; the notation of **Mobile Terminal (MT)** is also used in literature[1]. The MN can configure two types of addresses:
  - **Home Address (HoA)**. It is the permanent and globally reachable IP address configured by the MN when at home. The *Home Network* is thus defined as the network where the HoA is topologically valid.
  - **Care-of Address (CoA)**. It is the temporary IP address configured by the MN to maintain connectivity when in a foreign network.
- **Home Agent (HA)**. This node is in charge of storing an association between the HoA and CoA (a *binding*) per each MN. If the MN is present in the Home Network, the HA is not necessarily involved in the packets delivery to the MN, otherwise, if the MN registered a CoA at the HA, it intercepts the packets destined to the HoA, and it encapsulates them to the CoA to properly route them to the final recipient. This procedure is known also as *IP tunneling* [13].
- **Foreign Agent (FA)**. It is the router set as default gateway for the MN when visiting a foreign network. It intercepts the registration messages sent by the MN to the HA and advertises one of its addresses as endpoint for the mobility tunnel (i.e., as CoA). The FA is introduced mainly to overcome the limitations of IPv4 address space, so that the actual CoA associated to the MN's HoA is the FA's address, while the MN is allowed, for instance, to configure an address in the private range. Given these considerations, if the IPv4 addresses availability is not imposing a strict limitation, the FA functionalities can be removed from the access router and implemented by the MN itslef. This is the so-called *co-located CoA* mode, as, in this case, the CoA belongs to the MN.

[1] For the sake of completeness, in the documents and specifications produced by the 3rd Generation Partnership Project (3GPP) the moving host is called User Equipment (UE). This convention is typical of cellular networks and it refers to the user terminal at all layers of the 3G network stack, while, in the IETF, the term Mobile Node is intended to be technology agnostic, referring in particular at the IP layer.

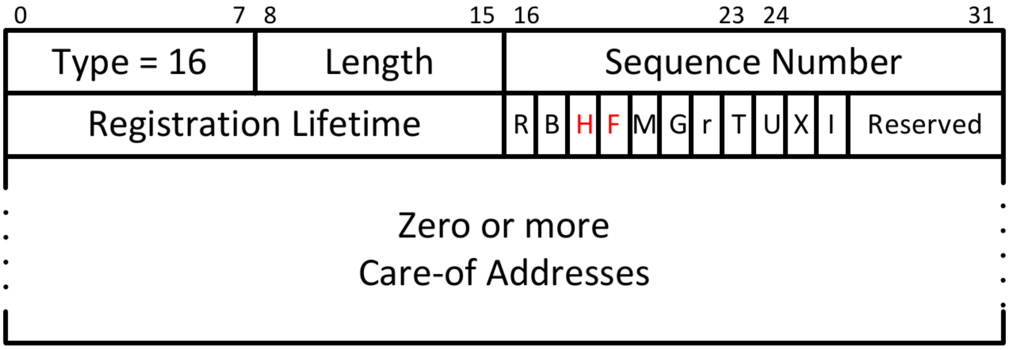


Figure 2.1. Agent Advertisement message format.

- **Correspondent Node (CN)**. It is the other endpoint of a communication with the MN. It can be a fixed or moving node.

2.1.2 Operations

Mobile IPv4 operations can be divided into three groups

- Agents Discovery
- Registration
- Data forwarding

Agents Discovery

The mobility agents periodically broadcast a particular message to indicate if they are an HA or an FA. This message is called *Agent Advertisement (AA)* and is built adding a mobility extension to the ICMPv4 Router Advertisement (RA) [14]; the flags contained in the message reveal the nature of the agent: “H” and “F” stand respectively for home and foreign agents, as shown in Figure 2.1. A mobile node, upon joining a network, can explicitly request such message by sending an *Agent Solicitation (AS)* message, equivalent to a Router Solicitation (RS).

An MN realizes if it is in his home network or a foreign one by inspecting the AA, and then it takes the appropriate actions. Also, the message informs the MN when it moves from a foreign network to another. When away from home, the MN configures the temporal address, or CoA, that, as introduced in Section 2.1.1, can be chosen alternatively between *i)* the FA’s IP address, more efficient in terms of addresses usage, or *ii)* the MN’s new address, configured, for instance, using DHCP, and leading to an overall faster registration phase and lower delay in the

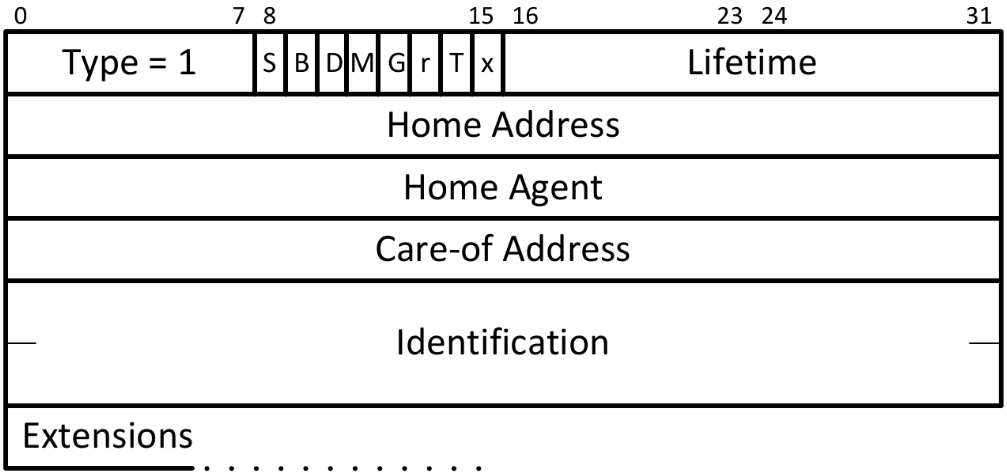


Figure 2.2. Registration Request message format.

communication (next subsections show the advantages and disadvantages of the two solutions).

Registration

The registration phase is a flexible method used by mobile nodes to *i)* request forwarding services when visiting a foreign network, *ii)* inform their home agent of their current care-of address, *iii)* renew a registration which is due to expire, and/or *iv)* de-register when they return home. A registration message is usually exchanged between the MN and the HA, but also between the MN and the FA.

The messages defined for this procedure are the *Registration Request*, sent by the MN to the HA (or FA) to request the service, and the *Registration Reply*, sent by the HA (or FA) to grant/deny the service (Figure 2.2 shows the Registration Request message format). The two messages are carried within a UDP datagram with destination port 434. The registration process is depicted in Figure 2.3, and it consists on 4 messages: *1)* the MN, upon detecting to be in a foreign network, sends a Registration Request to the FA, which processes the message and *2)* relays it to the HA. The message contains the HoA, the CoA, and a flag indicating whether the CoA belongs to the FA, or is a co-located CoA. Additionally, the message incorporates a lifetime, used to negotiate the service duration. The home agent receives the message and processes it. If the service can be provided, it stores an entry with the association between the HoA and CoA announced by the MN, and *3)* it sends back a Registration Reply with the affirmative response and the lifetime. Moreover, it sets a tunnel configuring as endpoints its own address on the link towards the MN and the MN's CoA. When the foreign agent receives the affirmative Registration

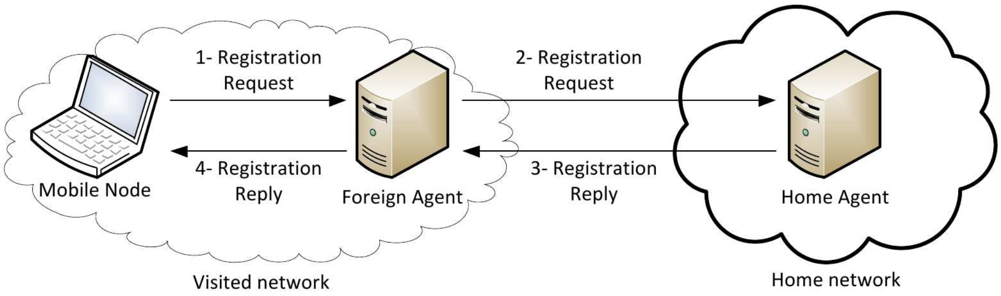


Figure 2.3. Registration in Mobile IPv4.

Reply, it adds a new entry in its visitor list, and, in case the Registration Request was sent with the reverse tunnel flag set, then it establish a tunnel with the HA in the reverse direction, so that the tunnel created between them is bidirectional. This feature is recommended for security reasons, as access routers should drop packets that are generated with a non topologically valid address (in this case the HoA, see next paragraph). Finally, *4)* the Registration Reply is delivered to the MN, which can now benefit of the mobility support.

Data forwarding

As mentioned in the previous paragraph, a tunnel is created between the HA and the CoA. It should be noted again that the CoA can be either the FA's address, either co-located. The two options show dual advantages and drawbacks. Indeed, the FA CoA is recommendable when the lack of available addresses is crucial. Such an option permits the MN to configure a private address, and the FA to behave as a NAT router. Conversely, this solution introduces extra processing at the FA, that might be overloaded while handling sessions for several MNs. Thus, it would be faster to encapsulate and desencapsulate packets at the MN, but this requires a globally reachable address. This latter approach is the one adopted in MIPv6, see Section 2.2, as the address space in IPv6 does not suffer this hard limitation.

However, a packet destined by the MN is first intercepted by the HA and then tunneled to the FA (or MN). Indeed, the packets in downlink carry the HoA as destination address, and the HA sends proxy ARP messages to the previous hop on the MN's behalf, to attract those packets and process them. As the HA is storing a binding for the HoA, it encapsulates the packets in another IPv4 header, filling the source address with its own, and the destination address with the CoA. If the tunnel endpoint is the FA, then the router desencapsulates the packets and delivers them to the final recipient, else, it the endpoint is the MN itself, the desencapsulation takes place internally to the MN. Figure 2.4 shows this procedure with the FA in

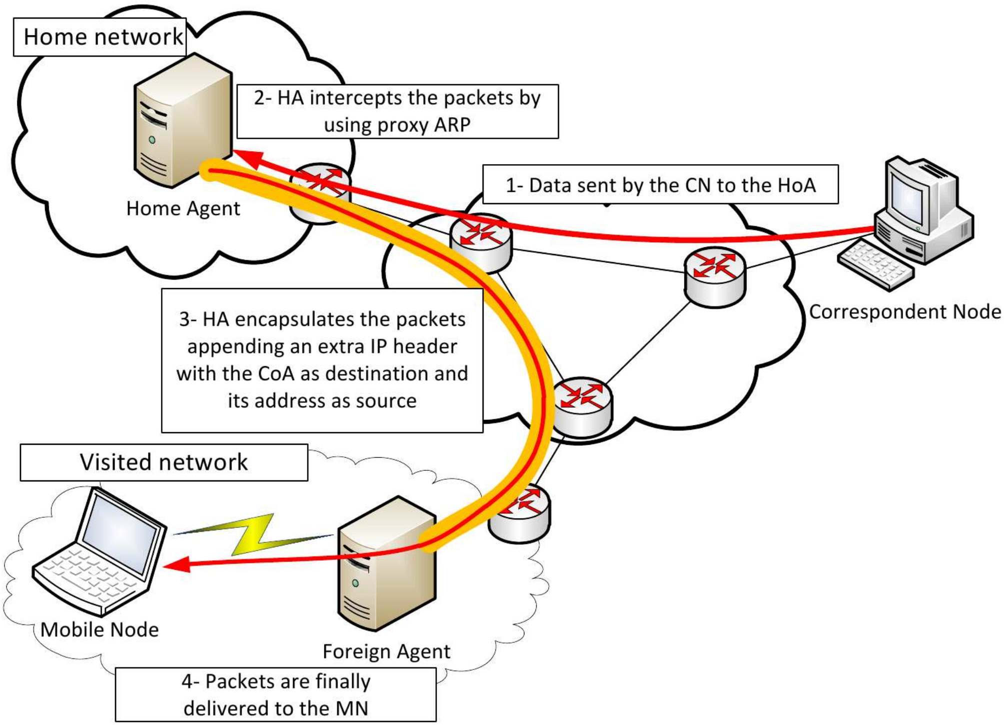

Home network
2- HA intercepts the packets by using proxy ARP
Home Agent
1- Data sent by the CN to the HoA
Correspondent Node
3- HA encapsulates the packets appending an extra IP header with the CoA as destination and its address as source
Visited network
Mobile Node
Foreign Agent
4- Packets are finally delivered to the MN

Figure 2.4. Downlink data forwarding in MIPv4.

the MN's access router.

For the uplink path, depicted in Figure 2.5, the packets' source address is the HoA, so they might follow the reverse path, going through the tunnel to the HA and then forwarded to the CN (reverse tunnels mode), or simply being transmitted via a direct path (reverse direction mode). However, the reverse direction mode is not recommended for security issues, as, for instance, those tackled by the *ingress filtering*[2] [15] practice. More on this topic can be found in [16].

2.1.3 Security

Even if security is out of scope in this work, it is worth noting that security mechanisms are crucial in a mobility protocol design. Indeed, besides the vulnerability intrinsic in the radio channel nature, some malicious attacks are possible when hosts are allowed to change and announce IP addresses. The most common is known as

[2] Ingress filter can be briefly described here as a security mechanism by which an Internet Service Provider (ISP) should not accept packets generated within its network containing as source a prefix that was not announced by the ISP itself. For this reason, a foreign agent in MIPv4 should not route packets sent by an MN using the HoA, but it should encapsulate them using the reverse tunnels mode

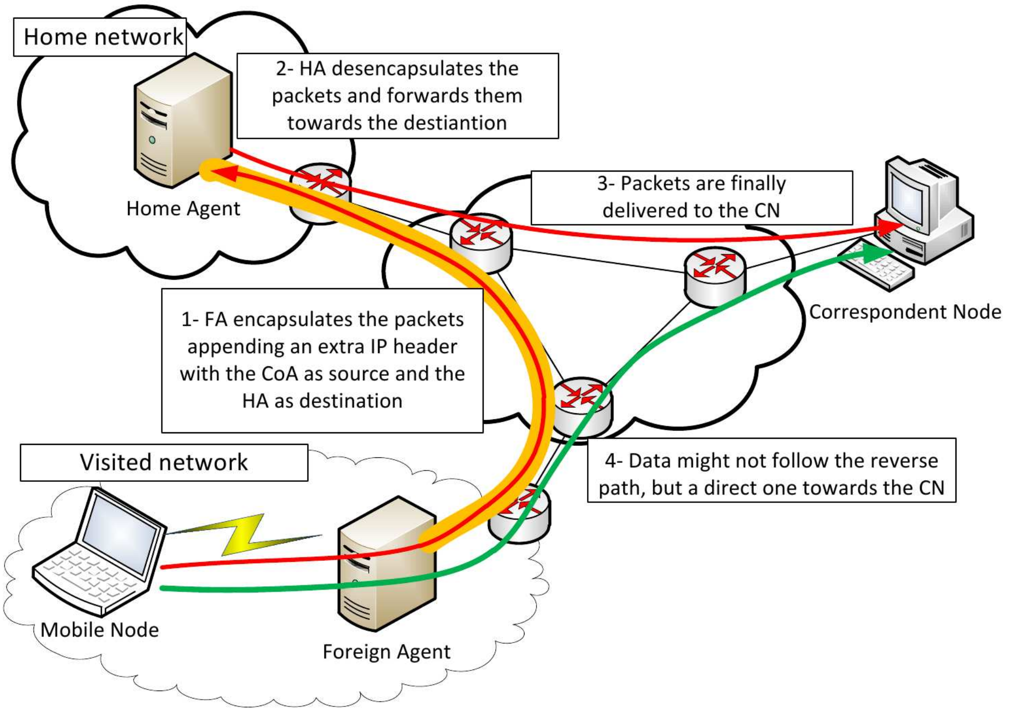

Home network
2- HA desencapsulates the packets and forwards them towards the destiantion
Home Agent
3- Packets are finally delivered to the CN
Correspondent Node
1- FA encapsulates the packets appending an extra IP header with the CoA as source and the HA as destination
Visited network
4- Data might not follow the reverse path, but a direct one towards the CN
Mobile Node
Foreign Agent

Figure 2.5. Uplink data forwarding in MIPv4.

redirection attack. A bogus node registers his address as CoA for a HoA that belongs to another MN. If the attempt is successful, the attacker is able to steal traffic from other users. Conversely, the misbehaving node registers its HoA with a third party's CoA and starts several IP sessions with the purpose of flooding the victim's links and/or draining its resources.

Mobile IPv4 recommends the use of the HMAC-MD5 algorithm [17] with a 128 bits key length to build secure associations between the MN and the HA, and/or between MN–FA, and FA–HA.

2.2 Mobile IPv6

A step forward in the design of an efficient mobility protocol was achieved with the solution for IPv6 networks, after the release of RFC 3775 [1] by David Johnson, Charlie Perkins and Jari Arkko in 2004, named *Mobility Support in IPv6*[3], or simply *Mobile IPv6 (MIPv6)*. Mobile IPv6 inherits most of the mechanisms introduced with

[3] At the time of writing this report, a new version of the RFC is being released that obsoletes the mentioned one [18]. However, the new release is not relevant for the purposes of the work, as MIPv6 is introduced as reference, whilst the research focuses on PMIPv6

its predecessor, but, also, it brings several enhancements due to the wise exploitation of IPv6 features. Next subsection presents the improvements achieved with the new protocol.

2.2.1 Comparison with MIPv4

This is the list of differences between the two mobility protocols for IPv4 and IPv6 taken from the RFC 3775:

- There is no need to deploy special routers as foreign agents, as in Mobile IPv4. Mobile IPv6 operates in any location without any special support required from the local router – the rationale behind this statement is that there is no lack of addresses in IPv6, thus a FA can be incorporated as a functional block of the MN. This feature of MIPv6 enables what is known as *global mobility*.
- Support for route optimization is a fundamental part of the protocol, rather than a nonstandard set of extensions – in MIPv4 we have the twofold option reverse tunnels and reverse direction. Route optimization refers to the latter, but MIPv6 specifies how to securely and effectively use both of them in a standard way.
- MIPv6 route optimization can operate securely even without pre-arranged security associations. It is expected that route optimization can be deployed on a global scale between all mobile nodes and correspondent nodes.
- Support is also integrated into Mobile IPv6 for allowing route optimization to coexist efficiently with routers that perform ingress filtering [15].
- The IPv6 Neighbor Unreachability Detection assures symmetric reachability between the mobile node and its default router in the current location.
- Most packets sent to a mobile node while away from home in Mobile IPv6 are sent using an IPv6 routing header rather than IP encapsulation, reducing the amount of resulting overhead compared to Mobile IPv4.
- Mobile IPv6 is decoupled from any particular link layer, as it uses IPv6 Neighbor Discovery [19] instead of ARP. This also improves the robustness of the protocol.
- The use of IPv6 encapsulation (and the routing header) removes the need in Mobile IPv6 to manage *tunnel soft state*.
- The dynamic home agent address discovery mechanism in Mobile IPv6 returns a single reply to the mobile node. The directed broadcast approach used in IPv4 returns separate replies from each home agent.

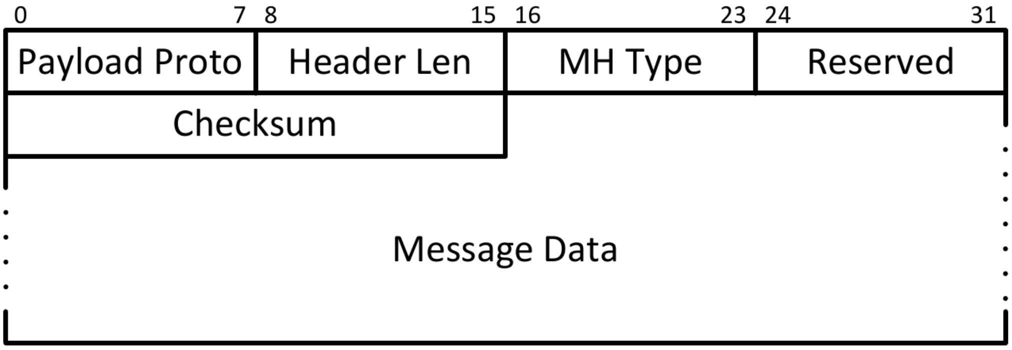


Figure 2.6. Mobility Header message format.

2.2.2 Protocol Overview

Mobile IPv6 enables global reachability and session continuity by using the same entities defined in the IPv4 version, except for the foreign agent. Hence, in MIPv6 it is defined the Home Agent, an entity located at the home network of the mobile node which anchors the permanent IP address used by the MN, the (HoA). The HA is in charge of defending the HoA's reachability when the MN is not at home, and redirecting received traffic to the MN's current location. When away from its home network, the MN acquires a temporal IP address from the visited network – the (CoA) – and informs the HA about its current location by sending a Binding Update (BU) message. An IP bi-directional tunnel between the MN and the HA is then used to redirect traffic from and to the MN. There is also extra support to avoid this suboptimal routing and enable the MN to directly exchange traffic with its communication peers - the CNs - without traversing the HA. This additional support is called Route Optimization (RO), and allows the MN to also inform a CN about its current location.

Differently from a Registration Request, a Binding Update message is not sent over UDP, but it is generated in a modular way, appending a *Mobility Header (MH)* to the IP packet. The mobility header's presence in the packet is notified by setting the Next Header field in the IPv6 header with the value of 135. The mobility header format is defined in MIPv6 and shown in Figure 2.6: the MH field indicates the nature of the message carried in the Message Data field, and, in case of a BU message, the MH value is 5. Nevertheless, the mobility header is used to bear also the Binding Acknowledgment (BA) message (with MH value 6), sent by the HA back to the MN as a Registration Reply, and some other control messages useful in the registration phase and for the RO process. The Binding Update and Acknowledgment message format is illustrated in Figure 2.7, in which it can be noted that it is aligned in order to be tacked as a trailer to the mobility header. Similarly, the binding messages are

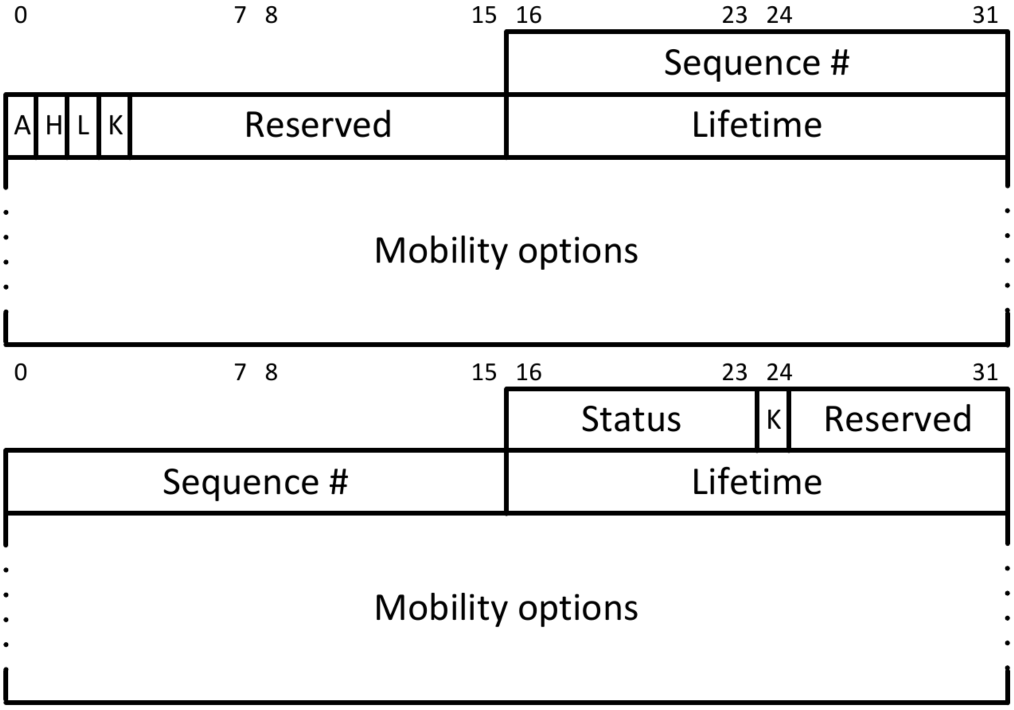


Figure 2.7. Binding Update (above) and Binding Acknowledgment (below) message format.

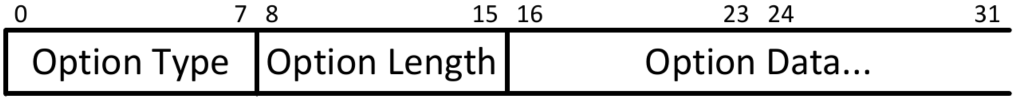


Figure 2.8. Mobility Options message format.

structured in order to convey several mobility options, each of the them with a specific purpose, as, for instance, to indicate the CoA, the MN's link layer address, the timestamp, etc (see Figure 2.8). This represents a great improvement compared to the IPv4 counterpart, as control messages are built dynamically by appending the desired mobility options to the mobility header. Moreover, when packets are destined to the MN, the use of encapsulation can be replaced by the Type 2 routing header, which contains the real MN's address, i.e., its location. In uplink, the Home Address destination option can be used to include the real (i.e., logical) source of packets. Basically, this mechanism allows to stick an extra IPv6 address to the packet without the need of a 40 bytes overhead due to encapsulation.

The working scheme of Mobile IPv6 mostly reflects that for IPv4, with minor variations due to the flexibility gained with IPv6. For instance, when a MN is away, it can configure its address in the foreign network using standard Neighbor Discovery (ND) [19] operations and *StateLess Address Auto-Configuration (SLAAC)* [19]. There is no need for a foreign agent to take part of the procedure, also for the subsequent signaling. Indeed, once the CoA is set up, the MN updates its location sending a BU message to the HA, which, in turn, updates the session information related to the MN stored in a database called *Binding Cache (BC)* and sends back the response in a Binding Acknowledgment (the format of a Binding Cache Entry is depicted in Figure 4.2). A bi-directional tunnel is established between the two nodes and the data traffic is transmitted through it. The necessity for data packets to traverse the HA lays down a suboptimal path between MN and CN called triangular routing, thus an additional operation has been defined to overcome this limitation. Route optimization is the procedure standardized to register the MN's CoA at the CN as well, so that the communication can be moved (i.e., packets are forwarded) through a shorter path to destination. Unfortunately, in order to perform RO, the CN needs to be provided with the MIPv6 module in its IPv6 stack implementation, otherwise the control messages cannot be processed correctly. Usually, the devices currently available in the market implement the IPv6 stack without the mobility component. The fact that both MN and CN need to run an extra module in addition to the IPv6 stack is one of the reasons that led to the development of network based mobility solutions (see Section 2.3).

2.2.3 Security

The security mechanisms adopted in MIPv6 leverage the IPsec suite [20, 21] in Encrypted Security Payload mode [22, 23]. The present work does not deal with the security issues, thus the scope is limited to mentioning only the names of the protocols, while further details can be found in the cited references.

However, some lines are worthy to describe briefly the principle followed to design the security mechanism for the route optimization procedure. Again, a comprehensive study is carried out in [1, 24]. It was mentioned in Section 2.1.3 that a mobility protocol is intrinsically prone to some kind of attacks as fake redirection or IP spoofing. This is especially true for the RO procedure, in which the MN informs the CN about its actual location. After this procedure, a CN basically stores a binding between the MN's HoA and CoA as an HA does, but BU and BA messages are secured whilst RO messages are not. Hence, a CN must make sure that *i)* the MN is really identified by the HoA it claims, and *ii)* it is actually reachable at the CoA. The security mechanism designed for this scope is called *Return Routability (RR)*, and, for the sake of simplicity, all the details are skipped to highlight the elegance of the basic idea. The MN starts the procedure sending a request through both paths to the CN, i.e., through the direct one and that via the HA. The CN replies transmitting two tokens through the distinct paths, i.e., one with destination the HoA and the other with destination the CoA. The two tokens are necessary to generate a key used to encrypt the subsequent registration message (a BU) sent by the MN to the CN. If the key is correct, i.e., if the CN is able to decrypt the message, then it means that the MN has successfully received both tokens, hence it is the legitimate owner of the HoA and it is actually located at the CoA claimed. Hence a BA is sent back to conclude the procedure and enable the optimized path.

2.3 Proxy Mobile IPv6

This section is devoted to the description of another mobility approach known as *network-based localized mobility management*. The IETF protocol that implements the results on this field is called Proxy Mobile IPv6 or PMIPv6 [4].

The motivation behind this new research area comes with the idea of bringing the mobility management closer to the MN, as developed in Hierarchical Mobile IPv6 (HMIPv6) [25]. In HMIPv6, the architecture designed for MIPv6 is broken into a two level hierarchy by introducing an additional entity between the HA and the MN, with the task of managing mobility within a smaller area, whereas the HA handles the mobility when a MN moves from one of these areas to another. Hence the name *hierarchical*, as the HA is aware of the area where the MN is, but not the exact location, while the new entity knows where the MN is currently attached. In this way, updating the location after small movements does not suffer of long latencies due to the distance between HA and MN, as only the local manager needs to be updated.

HMIPv6, hence, brought the concept of localized mobility management, in the sense that the deployed architecture defines a part of the network in which the mobility support can be provided to the users, but, upon crossing the boundaries of such network, the service cannot be offered anymore. The set of nodes running such a support form the so-called *Localized Mobility Domain (LMD)*. MIPv6, instead, is intended for global mobility, as the HA is always reachable by the MN's Binding Updates.

The improvement achieved with PMIPv6 is provisioning the mobility service within an LMD without involving the MNs. Hence the name *network-based*. MIPv6 requires the MN to implement the mobility module in addition to the IPv6 stack, while PMIPv6 only requires the latter; moreover, it is clear that relieving the terminal from mobility operations represents a saving in the terms of over-the-air signaling, and, therefore, battery consumption.

The network based scheme is achieved by relocating relevant functionality for mobility management from the MN to the network. In a Network-based Localized Mobility Domain, the network learns through standard terminal operation, such as ND [19] or by means of link-layer support [26], about an MN's movement and coordinates routing state updates without any mobility specific support from the terminal. While moving inside the LMD, the MN keeps its IP address, and the network is in charge of updating its location in an efficient manner [27]. From now on, the terminology LMD will be used only to denote the PMIPv6 domain, i.e., a domain in which the mobility management is network-based and localized. The following subsections give an insight of the entities and operations defined in PMIPv6.

2.3.1 Entities

The core functional entities in the PMIPv6 infrastructure are (see Fig. 2.9):

- **Mobile Node (MN)**. It is the moving host.
- **Mobile Access Gateway (MAG)**. This entity performs the mobility related signalling on behalf of an MN that it is attached to one of its access links. The MAG is usually the access router for the MN, i.e., the first hop router in the localized mobility management infrastructure. It is responsible for tracking the MN's movements on the access network. There are multiple MAGs in an LMD.
- **Local Mobility Anchor (LMA)**. This is an entity within the backbone network that maintains a collection of routes for individual MNs within the LMD (i.e., it is the entity that manages the MN's binding state). The routes point to MAGs managing the links in which the MNs are currently located. Packets for an MN are routed to and from the MN through tunnels between the LMA and the corresponding MAG. The LMA is also responsible for assigning IPv6 prefixes to MNs (e.g., it is the topological anchor point for the prefixes assigned to the MN). There may be more than one LMA in an LMD.

2.3.2 Operations

The sequence of operations in PMIPv6 is quite similar to that drawn in MIPv6, except that those actions performed by the MN in MIPv6 are now responsibility of the MAG.

Once an MN enters an LMD and attaches to an access link, the MAG in that access link, upon identifying the MN, receives a RS from the MN and performs the mobility signaling on behalf of it. The MAG hence sends to the LMA a Proxy Binding Update (PBU), associating its own address with the MN's identity (e.g., the MN's MAC address or an ID related with the MN's authentication in the network). Upon receiving this request, the LMA assigns a prefix – called Home Network Prefix

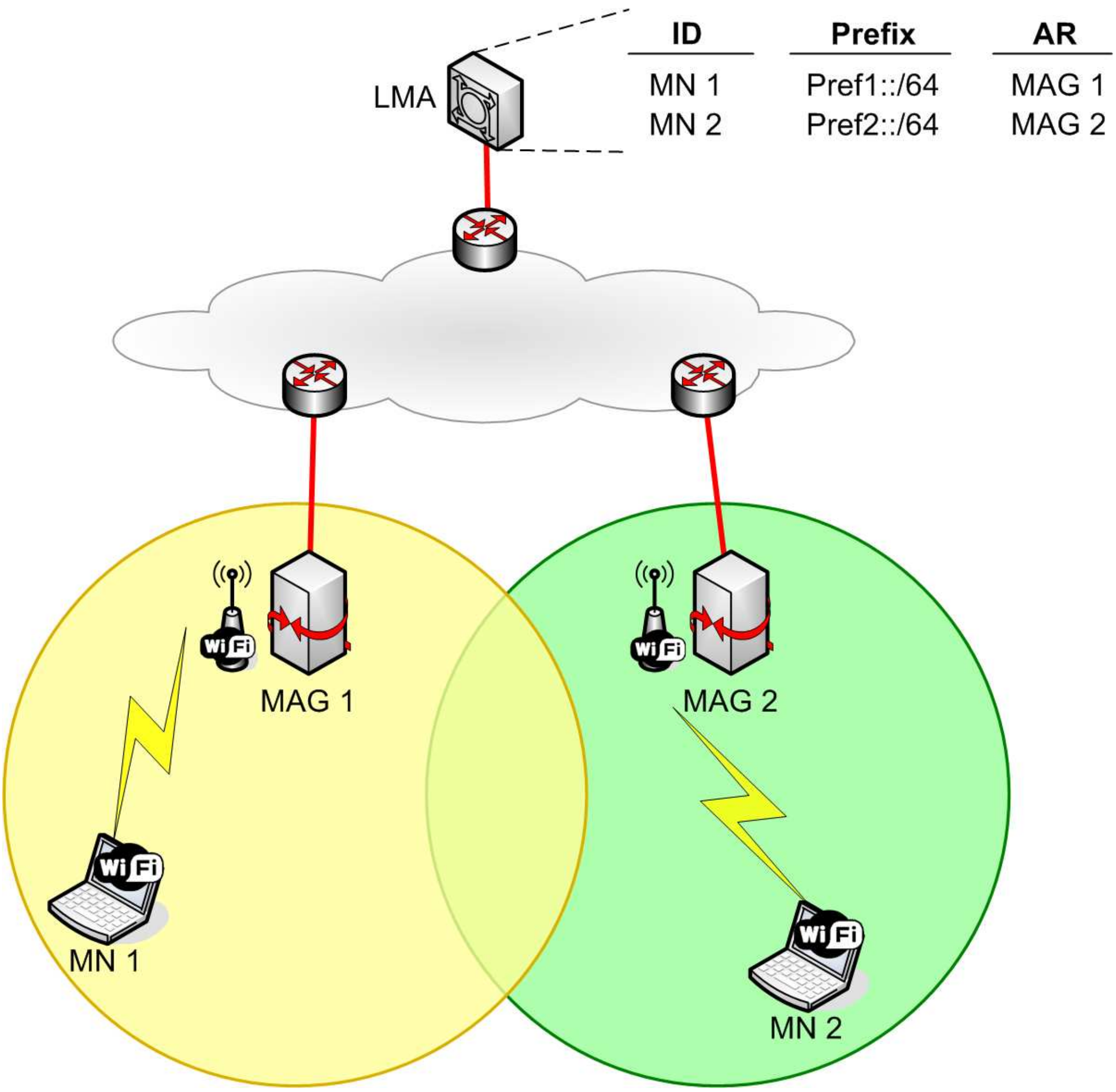


Figure 2.9. Proxy Mobile IPv6 domain

(HNP) – to the MN (i.e., it allocates a prefix for the attached interface). The LMA creates a Binding Cache Entry (BCE), shown in Figure 4.2), which main fields are the Mobile Node Identifier (MN-ID), the prefix assigned and the MAG's IP address visible from the LMA (the Proxy Care-of Address (P-CoA)). Then, the LMA establishes on its side a bi-directional tunnel to the MAG for the MN's traffic forwarding, and it replies to the MAG with a Proxy Binding Acknowledgment (PBA) message, including the prefix assigned to the MN. Once the PBU/PBA handshake is over, the MAG configures the P-CoA as the second end-point of the tunnel with the LMA, and unicasts a RA message to the MN specifying the prefix to be used

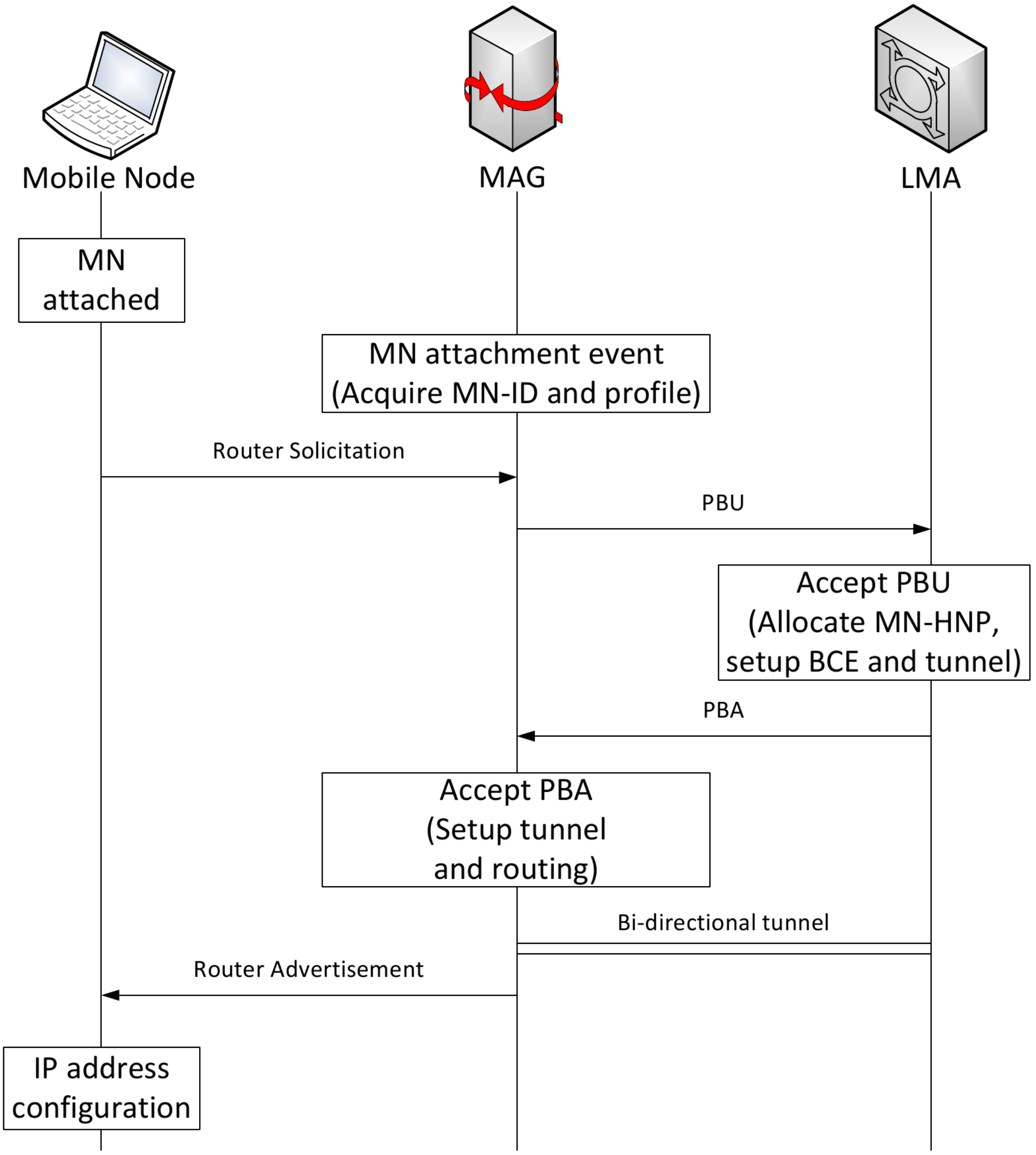

Mobile Node
MAG
LMA
MN attached
MN attachment event (Acquire MN-ID and profile)
Router Solicitation
PBU
Accept PBU (Allocate MN-HNP, setup BCE and tunnel)
PBA
Accept PBA (Setup tunnel and routing)
Bi-directional tunnel
Router Advertisement
IP address configuration

Figure 2.10. Registration to a Proxy Mobile IPv6 domain

for the IP connectivity. Now the MN is able to configure one or more addresses from the assigned prefix and the registration procedure is over (see Figure 2.10). The routing state created to forward messages to/from the MN comprises a routing entry for downlink at the LMA indicating that packets destined to the prefix(es)

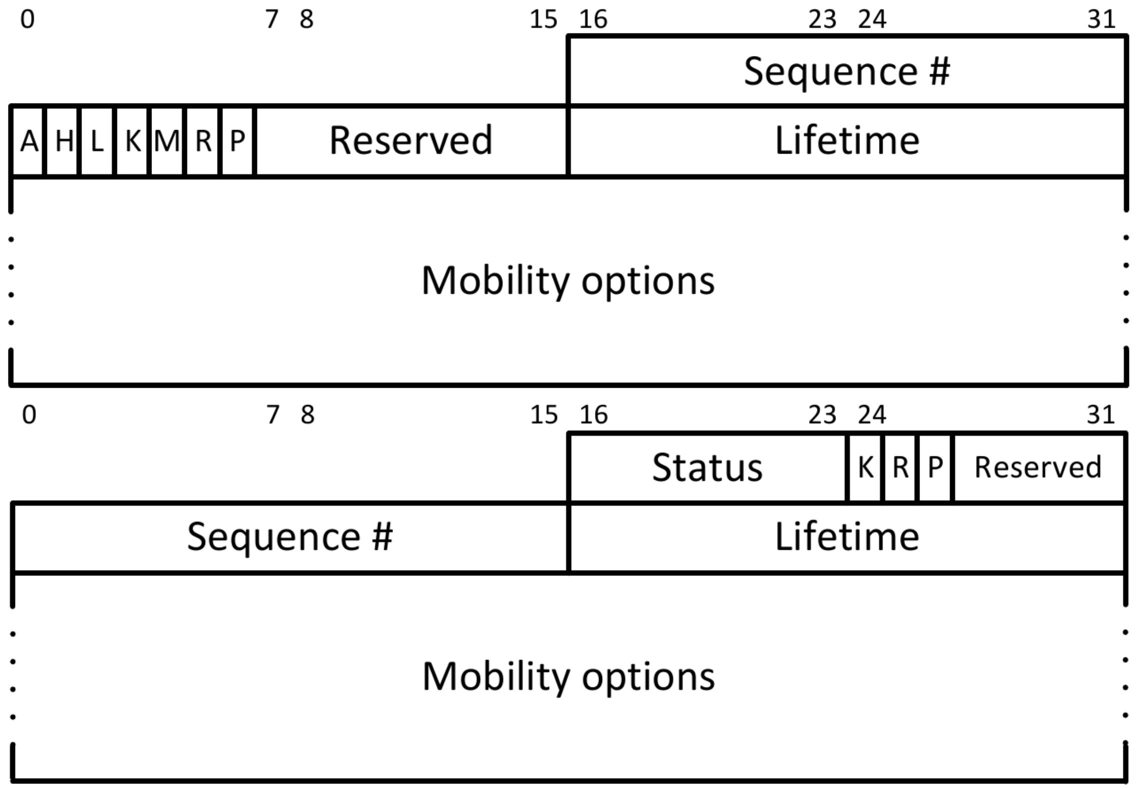


Figure 2.11. Proxy Binding Update and Acknowledgment message format.

assigned to the MN must be forwarded through the tunnel established with the serving MAG; the corresponding instruction at the MAG indicates that the MN is on the same link. For the uplink, the packets received by the MAG containing the MN's prefix(es) as source must be redirected through the tunnel towards the LMA (source-based routing). This way, the path used for the MN's traffic can be identified by the LMA-MAG tunnel set up with the MAG serving the MN.

Whenever the MN moves, the new MAG updates the MN's location in the LMA by means of a PBU/PBA handshake, and advertises through a unicast RA the same prefix to the MN. The new MAG shows the same layer-2 and layer-3 identifiers to the MN, thereby making the IP mobility transparent to the MN. Thus, the MN always keeps the address configured when it first entered the LMD, even after changing its point of attachment to the network.

In the context of Proxy Mobile IPv6 specification, the term mobility session refers to the creation or existence of state associated with the mobile node's mobility binding on the local mobility anchor and on the serving mobile access gateway. If the mobile node connects to the PMIPv6 domain through multiple interfaces, simultaneously, each of the attached interfaces will be assigned a unique set of home

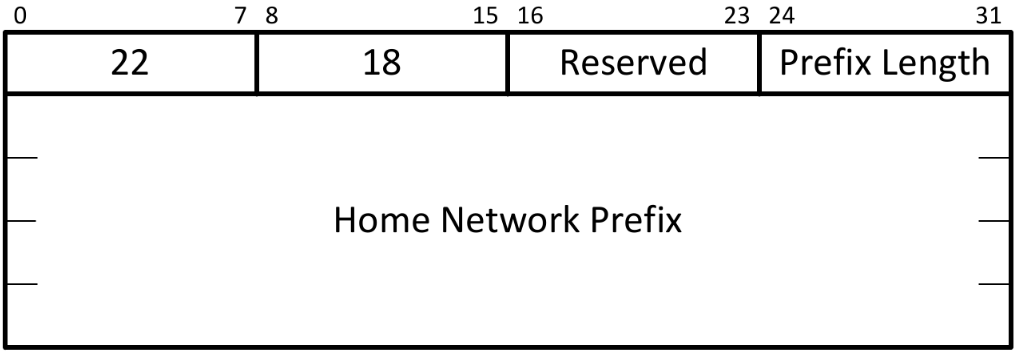


Figure 2.12. Home Network Prefix mobility option format.

network prefixes, and all the prefixes assigned to a given interface of a mobile node will be managed under a single mobility session, but separately to the session created for the other network interfaces.

The PBU and PBA messages are generated in a similar way as the BU and BA messages described in Section 2.2.2, i.e., a mobility header of the format illustrated in Figure 2.6 conveys additional fields similar to those defined by the BU and BA messages, except for a P (proxy) flag as ancillary field (see Figure 2.11). Moreover, extra information are included in the form of mobility options: Figure 2.12 depicts the format of the HNP mobility option. The advantages gained with the modularity introduced for the mobility messages is fully exploited in this thesis for the design of the flow mobility support in PMIPv6 domains, as described in Chapter 4.

2.3.3 Security

The security mechanisms in PMIPv6 are split into two levels related to *i)* the accounting of an MN and *ii)* the authentication of control messages. Indeed, whilst in MIPv6 the secure association between MN and HA converges the procedures into one, as the sender of control messages is the HA or the MN itself, in PMIPv6 the MN is first authenticated and authorized for the service by the MAG, and, next, mobility messages are authenticated between MAG and LMA.

Upon a MN's attachment to the network, either an authentication mechanism is deployed on the access link, or the MAG performs an Authentication, Authorization, and Accounting (AAA) check querying a dedicated infrastructure. RFC 5213 recommends the use of *Radius* [28] or *Diameter* [29] for this purpose. In both cases, the MN results to be authorized for the PMIPv6 service and provided with an unique identifier in the domain – the MN-ID. Moreover, the messages between MAG and LMA are secured with IPsec in a similar way as in MIPv6 (see Section 2.2.3).

However, in the development of this work, security issues have not been taken into consideration, thus next chapters do not consider this topic anymore.

Chapter 3

Overview on Multihoming and Flow Mobility

We are witnessing that the number of wireless mobile subscribers accessing data services does not stop increasing. This is motivated by a variety of different reasons: 3G access is widely available (coverage reaches almost 100% of dense populated areas in developed countries) and affordable by users (most mobile handsets are 3G capable, USB modems are quite cheap and operators offer flat rates to their customers). Besides, the number and popularity of applications designed for smart-phones that make use of Internet connectivity is getting larger every day, contributing to an increase of market penetration of such devices (e.g., iPhone, Android, Blackberry and Windows Mobile phones), which results in growing demands for 3G connectivity everywhere.

Driven by this continuous growth on the users' demand for connectivity and the high costs of 3G deployment (mainly because the radio spectrum is limited), operators are challenged to enhance their network deployments: the use of disparate heterogeneous access technologies – what is commonly referred to as 4G [30] – is considered as a mechanism to expand network capacity. This extension is not only achieved in terms of effective coverage (i.e., one particular access technology might not be offered in certain locations, while others could be deployed as an alternative way of accessing the network) but also in terms of simultaneously available bandwidth (i.e., the effective data rate that could be achieved by using two or more access technologies at the same time). User devices equipped with multiple radios (also known as *multi-mode* terminals) would be potentially capable of improving the connectivity experience they provide by using more than one single access technology at the same time. Mobile operators see today an opportunity of reducing the average cost per offered Megabyte (and therefore an increase of their revenue) by introducing an intelligent resource management mechanism that allows to offload traffic from the 3G network into other access candidate networks (mainly WLAN due to is high penetration and rate) when available. This optimizes the operator's network use, while keeping the users' QoE unaltered.

3.1 Multihoming

The scenario in which an end-host in connected through multiple interfaces to the Internet might lead to the configuration of several IP addresses (usually belonging to different subnets) for the same node. This is true unless some solutions as the logical interface are adopted [6, 7, 31].

Although the term *multihoming* is usually associated to a practice by which a stub Autonomous System is connected to different ISPs to guarantee resiliency in terms of Internet connectivity [32], here we introduce a *multihomed terminal* as an MN that configures different addresses on different interfaces. This is the default operation mode in PMIPv6 (see Chapter 4), where a distinct prefix is advertised each time the MN establishes a link with a new interface. Indeed, there are no means for the LMA to group together the MN's interfaces as a single node, thus they are seen as separate hosts. Anyway, a multihomed MN results with a set of different links to be chosen when starting a communication, with the chance of gaining performance in some Quality of Service (QoS) parameters as throughput or delay.

The SHIM6 protocol [33] and LISP Mobile Node [34] propose an architecture to allow multihoming for end-nodes, while [35] presents a comprehensive study on the benefits that can be gained by exploiting multihoming and a technique to achieve them. However, SHIM6 and LISP-MN require a great intervention on legacy terminals, whilst the solution described by Prof. Akella in [35] is suitable for powerful machines as modern routers, rather than for small devices as smartphones. However, whenever a host wants to exploit multihoming for purposes more sophisticated than a simple backup link, then an extra intelligence needs to be installed on the terminal. Generally speaking, this additional functionality can be named *Connection Manager (CM)*, as its task is to provide an interface between the network connections and the network applications. Nevertheless, one of the reasons that pushed the development of this work is how to exploit the benefits of a multihomed mobile terminal in a network controlled way, and, hence, we limit the implementation of the Connection Manager to a very simple tool that reflects the decisions taken by the network, and thus restricting the actions on the user's side. The details of such implementation are given in Chapter 5, but here we introduce in advance the two operation modes considered for a mobile node:

1. *Single interface visible from the IP stack.* Certain link-layer implementations can hide the use of multiple physical interfaces from the IP stack [7]. The *logical interface* [6, 31] at the IP layer is the most complete approach, as it allows

both sequential and simultaneous use of different physical media. In practice, applications send their data to a virtual interface which appends the IP address (if only one address is configured on the logical interface, otherwise the most appropriate one according to the desired destination) and selects the outbound interface. Therefore, the logical interface is able to send packets generated by different applications with the same source IP address but through different physical interfaces.

2. *Multiple IP interfaces or multihoming.* In this case, the IP address identifies the MN's interface, so, when a connection is set up with a given destination, an appropriate IP address is picked according to the routing entries, and the corresponding interface is used to forward those packets. The terminal may follow the *weak host* or the *strong host* model [8, 9]. The former does not limit the traffic reception at a host to only those IP packets whose destination address matches the IP address assigned to the interface receiving the packets, but allows the host to receive and process packets whose IP destination address corresponds to that of any of the local interfaces of the host. The similar concept can be applied for the sending case. Conversely, the strong model impose the host to receive or send only at the interface in which the desired IP address is configured. We have performed some tests with different operating systems, and the results show that both Linux (tested with Linux-2.6.26) and Mac OS X (tested with Leopard version) implement the weak host model for both IPv4 and IPv6 traffic. We have not performed exhaustive tests with Windows, but some results have been reported in [36]. Windows XP and Windows Server 2003 use the weak host model for all IPv4 interfaces and the strong host model for all IPv6 interfaces, not being possible to modify this behavior. The Next Generation TCP/IP stack in Windows 7 and Windows Server 2008 supports the strong host model for both IPv4 and IPv6 by default on all interfaces but in this case, the stack can be configured to use the weak host model.

Figure 3.1 depicts a multihoming scenario in a PMIPv6 domain: the MN is connected to the network through a 3G and a WiFi interface with a distinct IPv6 prefix configured per interface; each radio access is managed by a different MAG. This example is the reference scenario used throughout the thesis, further details are given in next chapters.

Next section merges the concept of multihoming with that of flow mobility, i.e., the ability of moving selected IP flows to one terminal's interface to another, with the purpose of using a different access network or path, according to the operator or ISP policies.

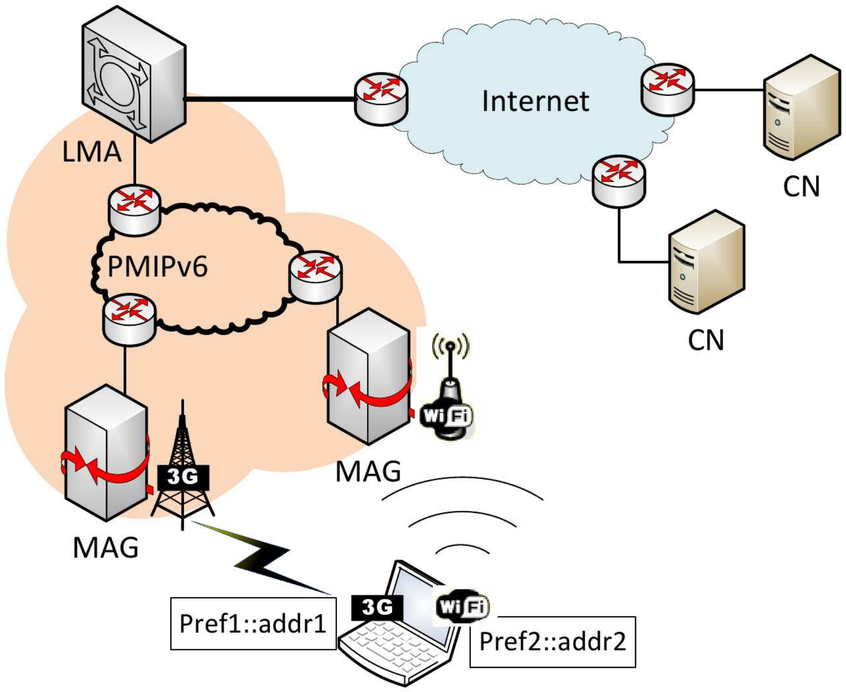


Figure 3.1. Multihoming in PMIPv6

3.2 Flow Mobility

Fully exploiting heterogeneity in the network access – e.g., enabling 3G offload – has proved to be difficult. Most of today's solutions enable the use of different technologies (e.g., 3G and WLAN) by adopting one of the following approaches (or a combination of them): *i)* manual user-based switching, or *ii)* application-based switching. In the former case, users decide to switch on a network interface based on their preferences (e.g., cost, required bandwidth for the applications being used, WLAN availability, etc.), while in the latter, applications decide to turn on and off interfaces based on predefined preferences and network availability. Both approaches, when the terminal is multihomed, involve a change on the IP address seen by the applications, and therefore rely on them surviving that change (or re-establishing the sessions). Operators are not satisfied with any of these approaches, as they leave the mobility control on the final users and/or the application developers. Additionally, the QoE obtained by users in this case may not be good enough, as it depends on the application behavior or it requires the sessions to be restarted.

The 3GPP and IETF are currently working towards the definition and specification of much richer solutions which aim at enabling true flow mobility. Flow mobility refers to the movement of selected IP flows from one access technology to another, minimizing the impact on the users' QoE. As already mentioned in the Introduction and in the previous section, flow mobility support enables the simultaneous and dynamic use of the terminal's interfaces, offering an extra ability to modern devices in addition to the *portability* and *mobility* concepts defined in Chapter 2. A simple use-case scenario is given by the possibility of using the WiFi connectivity available at home or in public hotspots (e.g., in airports, railway stations, buses) to maintain the connections started by a smartphone on the 3G link. A more sophisticated technique is described in [37], where a TCP connection is established through the WiFi in downlink and through the 3G in uplink (basically for the TCP ACKs). In this way the throughput for the TCP connection is boosted, as it is not necessary to follow the WiFi MAC procedure to acquire the channel to send the small ACK packets.

The concept of flow mobility has been extensively analyzed for client-based mobility protocols, and there already exist standardized solutions, such as the flow bindings extensions for Mobile IPv6 [38, 39]. The use of this kind of client-based solution has been proposed as a mechanism to enable mobile operators to offload data from their 3G networks [40], and there even exist approaches based on the IP Multimedia Subsystem (IMS) framework [41]. We argue that client-based solutions have several disadvantages, since they require to modify the users' devices to include an IP mobility stack, which also has to be provisioned with proper configuration and security credentials (in addition to those required to access the operator's network). This additional requirements might limit the usability of a solution due to the difficulties involved in its deployment.

As PMIPv6 is the standardized solution for network-based mobility management, the 3GPP and the IETF are currently working on the design of PMIPv6 extensions to enable flow mobility. The NETEXT Working Group[1] of the IETF has been recently re-chartered to work on extensions to enable inter-technology handovers and flow mobility. An early version of the solution described in this work has been presented in the IETF [5], being one of the first addressing the flow mobility issue that was presented and discussed there (the draft version -00 appeared before the NETEXT group was actually re-chartered to work on flow mobility). More recently, [42] refers to the latest output produced by the WG. There are other solutions which tackle the same problem, although no standard solution exists yet. We next summarize some of the most relevant existing proposals and compare them with the solution we have presented and evaluated in this work.

Koodli et al. propose in [43] new signaling between the LMA and the MAG to enable the LMA control flow mobility. Two messages are defined: the Flow Handover Request (FHRQ) – that is sent by the LMA to the MAG set up forwarding for one or more flows to an MN – and the Flow Handover Reply Flow Handover Reply

[1] http://datatracker.ietf.org/wg/netext/charter/

(FHRP)) – sent by the MAG in reply to a FHRQ message. While this signalling can be used to bind particular flows of an MN to specific MAGs, authors do not include any considerations on the mobile node behavior/support, nor provide any validation result or report on experimental tests.

Hui et al. propose a similar approach in [44] and [45], consisting on a extension of the BCE format at the LMA so the same HNP can be bound to several MAGs. The Binding Update List Entry (BULE) data structure is also modified to include the service flow information at the MAG. As opposed to [43], the handover control is on the MN and not on the LMA, and therefore it can be considered as an approach less attractive for mobile operators.

Additional material can be found in [46, 47, 48, 49, 50], although the work proposed in [42] represents the major output by the NETEXT WG on the field, merging efforts and authors of the mentioned drafts.

However, other designs are currently under investigation, sometimes endorsed by simulations as in [51, 52], but, to the author's knowledge, [10] and this thesis represent the first flow mobility support design for PMIPv6 including validation results based on real prototype experimentation.

It is worth noting how flow mobility support still constitutes a present and attractive research area, especially in those countries, as, for instance, South Korea, where different radio accesses as WiFi, WiMax and 3G are widely deployed. This trend is not likely to stop in the future, with the development of recent technologies as LTE and LTE-Advanced.

Chapter 4

Solution for PMIPv6

A first step required in order to support flow mobility is the capacity to use several physical network interfaces. Proxy Mobile IPv6 allows an MN to connect to the same PMIPv6 domain through different interfaces, though in a very limited way. There are three possible scenarios [53]:

- *Unique set of prefixes per interface.* This is the default mode of operation in PMIPv6. Each attached interface is assigned a different set of prefixes, and the LMA maintains a mobility session (i.e., a binding cache entry) per MN's interface. PMIPv6 only allows to transfer all the prefixes assigned to a given interface to another one attaching to the same PMIPv6 domain, and does not fully specify how a MAG can figure out if a new mobile node wants to get a new set of prefixes assigned (i.e., having simultaneous access via multiple interfaces) or if the mobile node is performing a handover (i.e., the MN wants to transfer the prefixes bound to a previous interface to the new one).

- *Same prefix but different global addresses per interface.* In this case the same prefix is assigned to multiple interfaces, though a different address is configured on each interface. This mode is not completely supported by PMIPv6. It either requires two different mobility sessions (as in the previous scenario) or only one but two separate host route entries. In any case, this scenario creates a multi-link subnet as the same prefix is advertised over different point-to-point links. This kind of scenario presents some issues as documented in [54].

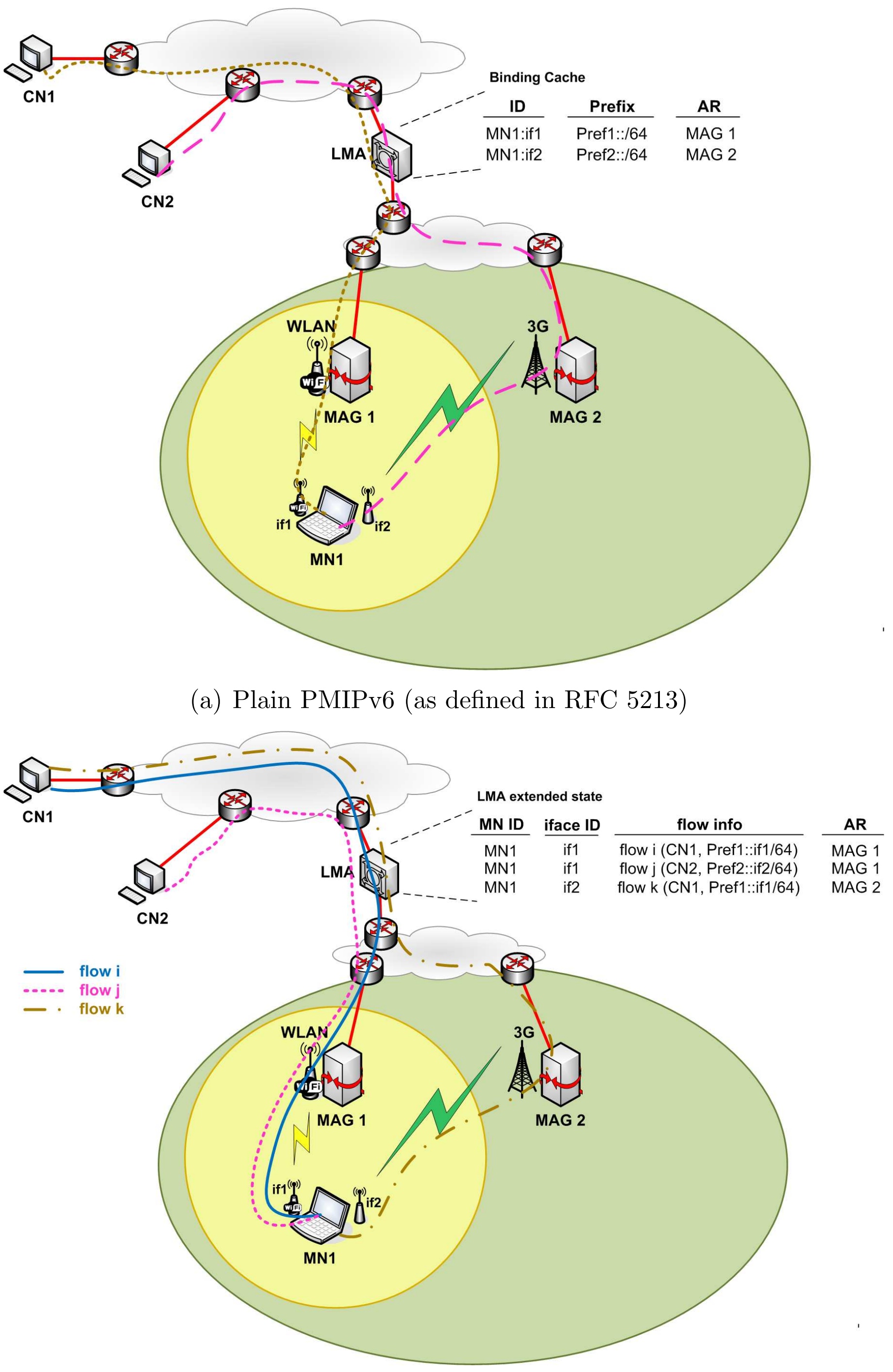
- *Shared address across multiple interfaces.* In this scenario, the MN is assigned the same IP address across multiple interfaces. This enables applications on the terminal to see and use only one address, and therefore the MN could be able to benefit from transparent mobility of flows between interfaces. This scenario is not supported by current PMIPv6, it requires one mobility session per terminal and some kind of flow filters/routes at the LMA to be able to

> forward packets via the appropriate MAG. Besides, ensuring that multiple IP interfaces of the same device configure the same IP address is not easy to achieve (e.g., IPv6 specifications assume that unique IPv6 addresses are configured per interface, as guaranteed by running Duplicate Address Detection (DAD)) nor to operate (not all Operating Systems support assigning the same IP address to multiple interfaces, and the multi-link subnet issue also appears here). One approach to mitigate this is to make use of link layer implementations that can hide the actually used physical interfaces from the IP stack [7]. For instance, the *logical interface* solution at the IP layer may enable packet transmission and reception over different physical media [6] [31].

PMIPv6 as defined in [4] cannot provide flow mobility in any of the previously described scenarios. We next identify and describe what functionality is missing from PMIPv6 to support flow mobility, by making use of an example. Fig. 4.1 shows a potential use case of interest involving a multi-mode terminal attached to a PMIPv6 domain. The MN is attached to MAG1 through its WLAN interface (`if1`), and to MAG2 through its 3G interface (`if2`). With current PMIPv6 specification (*plain* PMIPv6, see Fig. 4.1(a)), each interface is assigned a different prefix by the LMA (to allow simultaneous access), and two different mobility sessions (i.e., two separate BCEs) are maintained at the LMA. PBU/PBA signalling is used to keep alive the bindings at the LMA or to completely transfer the whole set of assigned prefixes from one interface to another. In order to support flow mobility, the state at the LMA needs to be extended (*extended* PMIPv6, see Fig. 4.1(b)), so the LMA is able to group mobility bindings referring to the same MN. Additionally, flow state should be introduced at the LMA, so it can forward packets differently (i.e., through different MAGs) on a per-flow basis. The MAG behavior needs also to be modified, since the MAG should be aware of all the MNs' IP addresses that are reachable through the point-to-point link it has set up with the MN. In order to transfer this information, the PMIPv6 signalling between the MAG and the LMA has to be extended as well.

The mobile node behavior needs also to be considered. In the plain PMIPv6 scenario, the IPv6 addresses assigned to `if1` (`addr1`) and `if2` (`addr2`) are different (`Pref1::if1/64` and `Pref2::if2/64`, respectively). Packets addressed to `addr1` will always arrive via `if1` (and the same for packets addressed to `addr2`, arriving via `if2`). In a flow mobility-enabled scenario, `addr1` and `addr2` may belong to different prefixes, belong to the same one, or even be the same IP address. Moreover, packets addressed to `addr1` may arrive at `if2` (and the other way around), and should be processed by the MN normally.

In this chapter we present the design of a solution enabling flow mobility for Proxy Mobile IPv6. An overview of the proposed mechanism (Section 4.1) is followed by the detailed description of the solution (Section 4.2).



Binding Cache

ID	Prefix	AR
MN1:if1	Pref1::/64	MAG 1
MN1:if2	Pref2::/64	MAG 2

(a) Plain PMIPv6 (as defined in RFC 5213)

LMA extended state

MN ID	iface ID	flow info	AR
MN1	if1	flow i (CN1, Pref1::if1/64)	MAG 1
MN1	if1	flow j (CN2, Pref2::if2/64)	MAG 1
MN1	if2	flow k (CN1, Pref1::if1/64)	MAG 2

(b) Extended PMIPv6 (flow mobility enabled)

Figure 4.1. Flow mobility in PMIPv6

We first define the term *flow*. A flow is intended as a stream of packets that traverses the LMA to/from the MN, regardless of which entity started the communication or which transport protocol is being used. A flow is univocally identified by 6 parameters – also referred to as flow 6-tuple:

1. Source IP address;
2. Destination IP address;
3. IPv6 flow label field;
4. IPv6 next header field (transport);
5. Source port;
6. Destination port.

Packets belonging to a bi-directional communication (e.g., a TCP session) are considered part of the same flow, as they carry the same parameters with the reversed role of the source and the destination in uplink and downlink. Nevertheless, since the uplink direction of the flow cannot be controlled directly by the network, in our implementation the LMA inspects only the packets in downlink, destined to the MNs.

4.1 Protocol overview

As outlined in previous paragraphs, a solution enabling flow mobility for Proxy Mobile IPv6 requires, on the one hand, extensions on the mobility signaling between the LMA and the MAG and, on the other hand, modifications to the behavior and data structures maintained by the LMA and the MAG. The basic idea is to create in the domain a suitable routing state for the MN that the LMA could control with the aggregated view of the multiple MN's links connected to the network.

Given its control role, the LMA is also the decision entity in the proposed approach. It performs flow routing based on operator policies – which may be dynamic to allow performing flow balancing to adapt to the network load – and/or other external triggers. The LMA enforces in this way which interface is used by the MN to receive downlink data traffic. For the uplink traffic, there are potentially several different approaches that the MN may follow. For example, the decision can be taken by the MN itself, selecting which interface to use independently of the LMA, however this could lead to asymmetric routing in the uplink-downlink paths[1]. So,

[1] The main problem here would not be the asymmetry in the paths followed by packets – IP routing does not guarantee symmetric routing – but the different access network delays imposed by different technologies, which could have an impact on the performance, e.g., of TCP flows.

we propose that the MN uses to send uplink traffic the same interface that is used to receive downlink packets belonging to the same flow. Following this approach, the MN *replicates* the decisions made by the LMA for the downlink traffic when sending uplink traffic, and consequently replicating any posterior changes that the LMA may perform during the flow lifetime.

Due to the fact that PMIPv6 does not require the MN to implement nor participate in any mobility protocol, considerations about how the terminal behaves are very relevant. As already stated in Section 3.1, in this work we consider two different kinds of IPv6 mobile nodes:

1. *Terminals with a single interface visible from the IP stack.* The *logical interface* [7] [31] at the IP layer allows both sequential and simultaneous use of different physical media, hiding the use of multiple physical interfaces from the IP and upper layers.

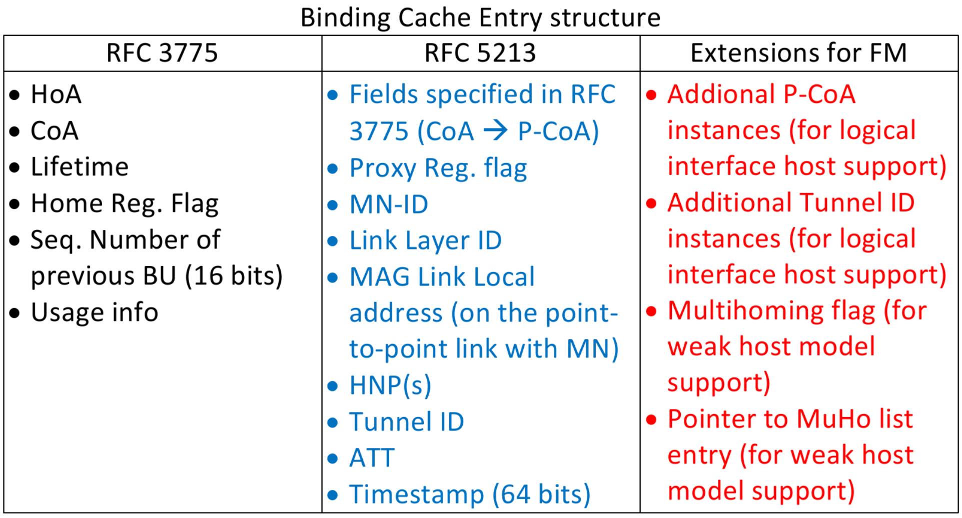
   For this kind of terminal, our preferred solution is based on the LMA delegating the same prefix (or set of prefixes) to the MN, regardless of the physical interface that is getting attached to a MAG, since there is only one interface visible from the IP layer. In fact, this basically means that from the viewpoint of the network, the MN is sharing the same IP address(es) across multiple physical interfaces, although the addresses are not really configured on the physical interfaces but on the logical one. The LMA decides – on an IP flow basis – through which MAG data traffic is forwarded to the MN, and consequently through which physical interface the MN receives traffic.

2. *Terminals with multiple IP interfaces.* In case the mobile terminal does not implement the logical interface concept (or an alternative link-layer approach that hides the use of multiple media to the IP layer), it is still possible to enable full flow mobility if the terminal follows the *weak host* model [8] [9].

   For this kind of terminal, our solution is based on the LMA delegating a unique prefix (or set of prefixes) per interface (as in plain PMIPv6). The LMA performs flow-based routing while the MN is able to process received packets at any of its interfaces, thanks to the use of the weak host model.

4.2 PMIPv6 Extensions

In the following paragraphs, we elaborate more on the specific protocol extensions that are required to enable flow mobility in a PMIPv6 domain for the two kinds of terminals supported by our solution. We observe that, compared to alternative designs, ours is intended to apply the least changes to the legacy protocol, keeping unaltered the state-machine, the order of operations and the lookup criteria defined in the specifications.



Binding Cache Entry structure

RFC 3775	RFC 5213	Extensions for FM
• HoA • CoA • Lifetime • Home Reg. Flag • Seq. Number of previous BU (16 bits) • Usage info	• Fields specified in RFC 3775 (CoA → P-CoA) • Proxy Reg. flag • MN-ID • Link Layer ID • MAG Link Local address (on the point-to-point link with MN) • HNP(s) • Tunnel ID • ATT • Timestamp (64 bits)	• Addional P-CoA instances (for logical interface host support) • Additional Tunnel ID instances (for logical interface host support) • Multihoming flag (for weak host model support) • Pointer to MuHo list entry (for weak host model support)

Figure 4.2. Binding Cache Entry format. The fields specified in RFC 5213 extend the structure defined in RFC 3775, which, with the exception of the CoA (that becomes the P-CoA) and the Lifetime fields, is not necessarily filled with `NON_ZERO` values in Proxy Mobile IPv6. For instance, the sequence number field is overridden by the timestamp field, the HoA field can be set to `ALL_ZERO` value, etc.

4.2.1 Single IP interface case: logical interface model

When an MN uses a logical interface to connect to the same LMD via multiple physical interfaces, it appears to the rest of the network as a set of different endpoints with the same Layer-2 and Layer-3 addresses. In PMIPv6, once an MN has attached one of its interfaces and has been registered in the LMA, subsequent attachments via other interfaces to new MAGs might be identified as handovers, as an update request for an already registered prefix will be coming from a different access network. Our approach:

1. extends the original PMIPv6 messages to allow the MAG specify – upon attachment of a mobile node – that the attaching physical interface belongs to a logical interface[2];

2. modifies the conceptual data structure at the LMA, so it stores information about all the MAGs that lead to the same host (that is, the Proxy CoAs and the tunnel-IDs). One extra instance of these parameters should be added for

[2] This information can be stored in the MNs' profiles database and retrieved by the MAG during the AAA process. The profile can be configured by the user only once, indicating the terminal's feature. Note that current operators have a detailed knowledge of the user's terminal (vendor, model, capabilities, etc.) and sometimes they provide (e.g., with an SMS) the parameters for the setup of a correct terminal-specific configuration for the data service access.

each physical interface (grouped under the same logical interface), so that the LMA is able to create tunnels and routes without deleting the existing one.

Figure 4.2 shows the fields required for a Binding Cache Entry according to the specifications of MIPv6 (bullets in black), PMIPv6 (bullets in black + blue) and flow mobility support (bullets in black + blue + red).

The above description (to simplify the explanation of the protocol procedures) takes into account the assignment of a single HNP per logical IP interface. In case the LMA assigns a pool of HNPs to the logical IP interface (from the LMA perspective this is a standard IP interface) all the logic still holds. The LMA will need to store all the HNPs for the specific mobility session. From a MAG point of view there may be different protocol choices:

- *Unique HNP (or set of HNPs) per physical interface.* In this case the LMA, upon attachment of each physical interface, assigns a different HNP (or set of HNPs). That is, the MAGs providing network connectivity to the MN know only the on-link prefix(es). To enable flow mobility, the LMA – during the PBU/PBA protocol exchange – should inform the MAGs about all the HNPs associated to the MN. The PBA should carry the HNPs that should be reachable via the on-link HNP. This procedure is similar to the one described in the weak host section allowing the MN to receive packets to any HNP (irrespective of the on-link configuration) as long as they are properly assigned to the logical IP interface. The PBA message contains a specific option and upon parsing, the MAG installs the required routing state.

- *Multiple shared HNPs per physical interface.* In this case the LMA behaves according to the original PMIPv6 specification [4] and assigns a pool of HNPs to the logical physical interface. The same prefixes will be assigned when the MN attaches a second physical interface.

The implementation details and the experimental results presented in next chapters describe the single HNP per logical IP interface. We argue that, from a session continuity point of view, this is the most interesting scenario: the node configures a single global, always-on reachable IP address from that HNP. Moreover, in a 3GPP context the HNP is the IP prefix assigned by the mobility anchor to the MN upon network attachment allowing seamless mobility of IP flows across heterogeneous access[3]. We observe that, in this case, the routing state created at the LMA for the logical interface consists in multiple entries for the same prefix, with the only difference of the outbound interface used (the tunnels to the MAGs). Hence the first

[3] The 3GPP SA2 working group will be standardizing for Rel-10 mechanisms for seamless WLAN offload from the LTE wireless access. Such technologies are currently based on DSMIPv6, but studies show the strong interest of mobile operators in the deployment of network-based solutions.

MN's interface attached is considered as primary, since the corresponding tunnel (included in a routing entry) appear first in the routing table.

4.2.2 Multiple IP interfaces case: weak host model

With regular PMIPv6, when an MN attaches to an LMD via more than one interface, it receives a different prefix for each one of them. Each interface is treated as if it was a completely different MN (i.e., separated mobility sessions). Our solution solves this issue by enabling the LMA to group together all the mobility state that it has referring to the same MN in a new conceptual structure called *flow-mob list*. One list's entry points to all the BCEs that are refferred to the same MN-ID, so that, by inspecting the flow-mob list, it is possible to retrieve all the prefixes assigned to the MN's interfaces. The actual implementation of the data structure and how it is related to the Binding Cache is postponed to Section 5.1.

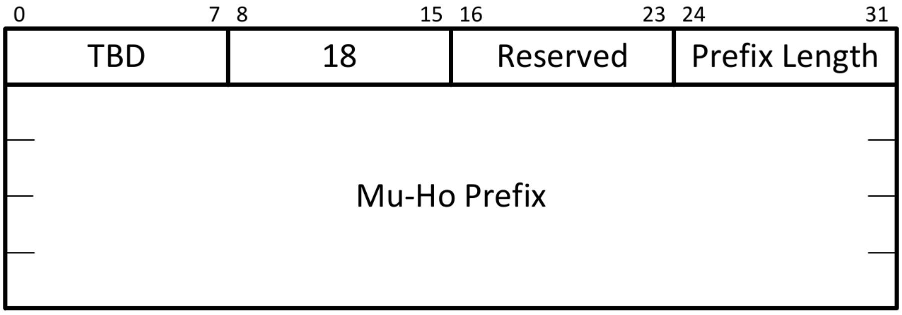


Figure 4.3. Multihoming mobility option format

The MAG, upon detecting MN attachment, checks whether the MN is authorized for PMIPv6 service. If so, the MAG prepares the PBU with the acquired MN-ID[4] in the MN-ID option and the MAC address in the Link Layer ID (LL-ID) option. When the PBU is received, the LMA registers a new BCE following the PMIPv6 standard procedure (because the HNP and the LL-ID are new), and in addition it checks whether the MN-ID is already present in the flow-mob list. If so, the LMA then builds a PBA with the prefix assigned to the new interface (standard PMIPv6 behavior), including a new extra option – which has the same format of the HNP prefix option, as can be seen by comparing Figures 4.3 and 2.12 – that carries the prefix(es) retrieved after the flow-mob list lookup, i.e., those assigned to the previously attached interface(s). This allows the MAG to install routes to all the prefixes assigned to the MN for each of its interfaces attached to the same LMD.

[4] We use the MAC address as MN-ID because this is what it is supported by our current implementation. Nevertheless, a different approach, such as the use of Network Access Identifiers (NAIs) could be followed instead, and in this case a conversion mechanism would not be necessary.

We observe that the above behavior is similar to the one described for the logical IP interface when multiple HNPs are delegated to the MN.

4.2.3 Final remarks

The advantage of the logical interface with respect to the weak host model is that, in the reference scenario where an unique single prefix is delegated to the attached interface, the same routing state is installed in the MAG, both when the MN is connected via a single link or through multiple ones. For the support of weak-host terminals, a MAG needs to know the prefixes that were formerly configured by the MN on the other IFs, so that a correct route entry for them is defined. In fact, a MAG by default installs a route entry only for the prefix configured on the "on-link" IF (i.e., directly connected to the MAG), without considering the other IFs and the prefixes assigned for them. The additional route entries are necessary to instruct the MAG to forward packets containing the other prefixes via the on-link MN's IF, as if the MN's IF were a next-hop destination.

This procedure allows the MAGs to get aware of other prefixes in an incremental way, i.e., the first MAG only knows the first prefix, whereas the latest acquires them all. However, when a BCE for an old prefix is about to expire, the refreshment mechanism permits the MAG to retrieve all the prefix currently active for the MN.

Chapter 5

Implementation of the Solution and Testbed Setup

In this chapter we provide a comprehensive description of the implementation of the different components developed to enable seamless flow mobility in PMIPv6.

In order to be able to conduct real experiments that allow us to evaluate the feasibility and performance of our proposed solution, we deployed an IPv6 network setup featuring one LMA, three MAGs, a machine acting as remote server (i.e., the CN) connected to the LMA and two mobile nodes: one implementing the weak host model (*weak host MN*) and one implementing a particular realization of the logical interface concept, based on the bonding driver[1] for Linux OS (*bonding MN*). All these nodes run an Linux distribution (Ubuntu 9.04 with kernel 2.6.31). Figure 5.1 depicts the functional boxes installed in our machines and the associated software modules, which, node by node, are the subjects of the following sections.

5.1 Local Mobility Anchor implementation

The machine acting as Local Mobility Anchor runs the PMIPv6 protocol daemon and the Flow Manager tool.

The PMIPv6 protocol is coded from the UMIP daemon[2], an open source implementation of MIPv6. Unfortunately, the version of the PMIPv6 code we started from presented some bugs to fix, due to the premature closure of the project working on it, and there is not a complete release of the daemon yet available for the community. However, thanks to the thorough work made by the UMIP developers and the corrections we brought, the customized PMIPv6 daemon results in a extremely accurate implementation of the RFC specifications. Indeed, the PMIPv6 engine

[1] http://www.linuxfoundation.org/collaborate/workgroups/networking/bonding/

[2] http://www.umip.org

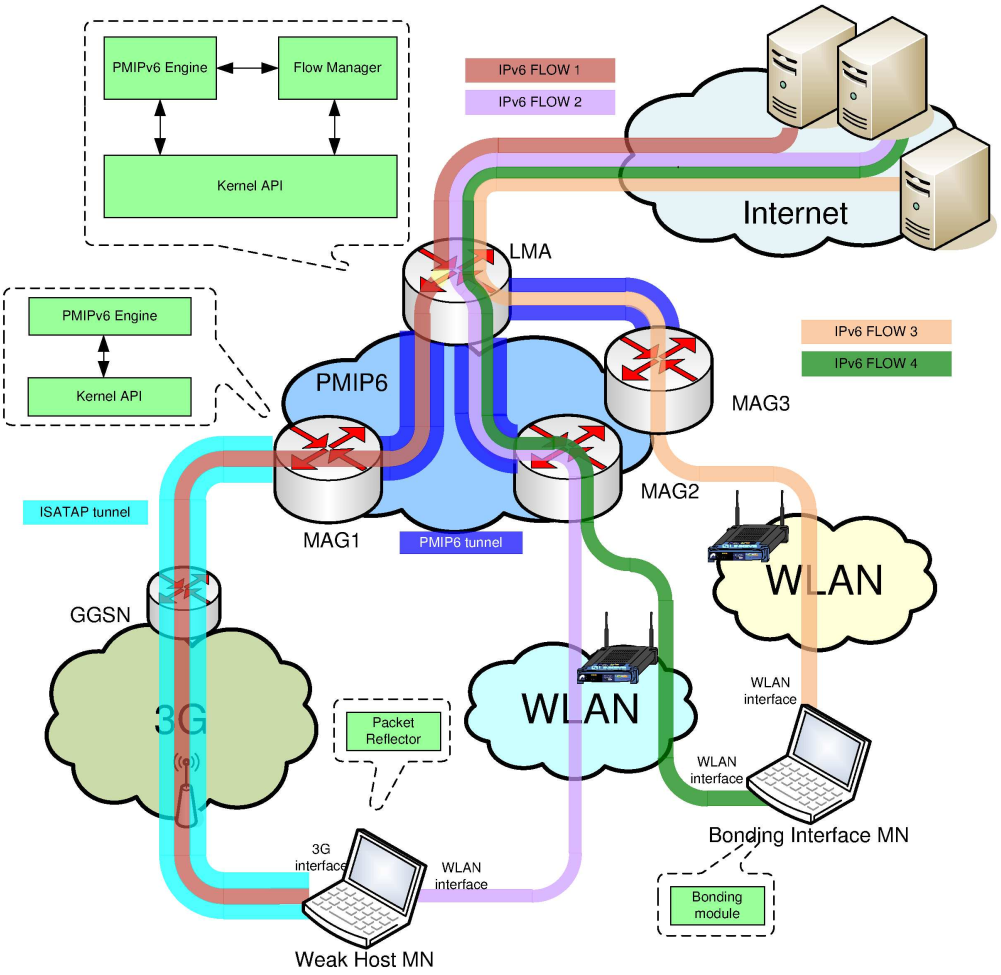


Figure 5.1. Testbed Setup

inherits from the original software the core functional blocks, and, in particular, it provides a set of functions to interact with the operative system in order to intercept the mobility messages as PBU/PBA, create the tunnels, install the routes and rules to forward packets according to the mobility protocol. Moreover, the Binding Cache data structure is maintained at the LMA, with all the necessary timers used to compare timestamps, assign lifetimes and check when a BCE is about to expire.

Our contribution to the daemon, besides the preliminary debug phase, was to further extended it to support flow mobility both for the weak-host and bonding terminals, including the enhancements described in Section 4.2. for instance, some changes are applied in the BCE structure (see Figure 4.2: a flag to indicate whether the terminal is multi-attached, pointers to the flow-mob list entries, additional instances of the fields defined for the entry. These information are provided in the

PBU by the MAG after a MN's profile retrieval, see Section 5.2, and they are crucial to propagate the correct routing instructions for the MN in the other protocol elements (nodes and data structures).

Weak-host model

This is the case in which the terminal is multihomed, i.e., the MN's multiple links to the PMIPv6 domain belong to different subnets. To support this scenario we added:

- The *flow-mob list* data structure;
- the *mu-ho* mobility option.

For the latter, it was enough to add a function to create the desired option when necessary, append it to the PBU/PBA and to properly read it during the message processing (we used the value 28 in the *type* field).

Regarding the former modification, an entry of the flow-mob list was created – called *Mu-Ho entry* – as a set of pointers to the BCEs referring to the same MN-ID, parameter used as search key in the list. In this way, a Mu-Ho entry groups together the mobility sessions related to the same terminal, while in the BCE we included an additional pointer to make accessible the Mu-Ho entry from the BCE itself.

The reverse-pointing feature is introduced to accommodate the interaction between the BC data structure and the flow-mob list, necessary for the multihoming support. Indeed, during the *i)* registration, *ii)* de-registration and *iii)* handover phases, the standard behavior is a HNP-based lookup in the BC, but, of course, this does not fulfill our purposes, as it does not perform a parallel lookup in the flow-mob list to update the multihoming MN's status.

In the following we describe how to explore consistently the two lists for the operations mentioned above. When a BCE is created with the multihoming flag set, i.e., the HNP is allocated for the first time but the MN is multihoming-enabled, the flow-mob list is checked if the same MN-ID was registered before (the flag is set if the MAG sends a void Mu-ho option in the PBU). In case of fresh registration, the flow-mob list is filled with a new Mu-Ho entry, reporting the MN-ID and pointing to the BCE. Accordingly, a pointer to the flow-mob entry is added in the BCE. Otherwise, if the MN-ID is already present in the flow-mob list, obviously it is not necessary to create a new Mu-Ho entry, but to update the pointers only, appending a new one towards the just created BCE, and, again, another in the BCE, pointing back to the list's entry. By following the pointer in the Mu-Ho entry, the LMA is able to send to the MAG the prefixes already assigned to the MN, communicated by means of the Mu-Ho mobility option, in order to install at the MAG proper routes for flow mobility support.

In case of de-registration, the LMA receives a PBU with lifetime 0 for an already registered HNP, thus the corresponding BCE is deleted, and, going after the pointer in the BCE itself, the related Mu-Ho entry can be updated.

When a handover occurs, the LMA receives a PBU from a new MAG and, the LMA might have or not obtained a previous de-registration PBU for that MN's interfaces. In the former case, the scheme for a new BCE creation is followed, as described before, while in the latter, the BCE is updated with the new P-CoA and tunnel-ID, and, following the pointer in the BCE, the Mu-Ho entry provides the prefixes acquired by the MN.

Logical interface model

Since the bonding host does not require multiple addresses configuration to support multi-attachment and flow mobility, the changes applied before are no longer necessary. However, some modifications are still needed, and they regard the BCE format (see Figure 4.2):

- Additional instances of the P-CoA field;
- Additional instances of the Tunnel-ID field.

When a host uses a bond IF to connect simultaneously to more points of the network, it will appear to the rest of the network as a bunch of different endpoints with the same L2 and L3 addresses. In the PMIPv6 scenario, after registering the first IF, all the requests coming from other IFs (that is, from other MAGs) might be seen as handover requests. In fact, the LMA receives an identical PBU for each IF, but coming from a MAG whose address differs from the Proxy CoA specified in the BCE for that prefix. If the Handoff Indicator (HI) in the PBU message is not properly set (for instance, HI value 4 stands for "unspecified"), the LMA may misunderstand the request and move the registration to the new IF, deleting the routes for the previous one. A wise use of the Access Technology Type (ATT) mobility option in conjunction with a bond value for the HI field, leads to a replication of some BCE's parameters, suitable to allow multi-interface hosts management. The LMA, indeed, needs to store in the BCE all the infos about the MAGs that lead to the same host, that is, the corresponding Proxy CoAs and the tunnel-IDs. One extra instance of the mentioned parameters should be added for each IF in bonding mode, so that the LMA is able to create tunnels and routes without deleting the existing one. It is worth noting the following about the routes setup, because the LMA sets one route towards the same prefix through each tunnel. In this case the first route set will be always chosen, i.e., it is the primary, since all the other routing parameters, like the priority, are identical. A rule mechanism might be adopted to choose which MN's interface will be the addressee of the communication, as we'll see in the Flow Management section.

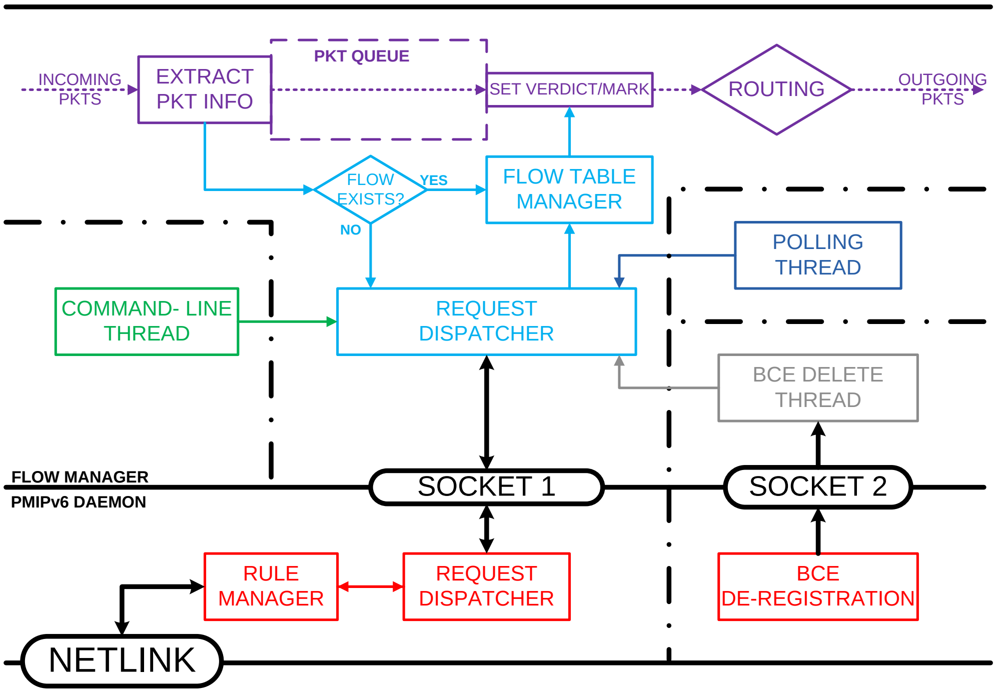


Figure 5.2. Flow Manager internal architecture

5.1.1 Flow management

Flow management is kept detached from the PMIPv6 daemon and performed by a separated process referred as *Flow Manager (FM)*. The FM is based on the `libnetfilter_queue` library, an extension provided within the `netfilter` framework for packet handling[3]. The two processes communicate through the use of UNIX sockets, as depicted in Figure 5.2, which, on the PMIPv6 side, are handled by dedicated threads, not present in the original PMIPv6 daemon, written ad-hoc for this purpose.

The basic tasks performed by the FM are:

- Packet classification into flows;
- Flow list maintenance;
- Flow routing.

. However, these tasks are executed by 5 threads, represented in different colors in the picture, which do not reflect the aforementioned logical division of jobs, but were rather implemented taking into consideration the communication between processes

[3] Available for linux platforms at http://www.netfilter.org/

and the development of the application itself. Each thread comprises one or more functional blocks, which may take part into several logical tasks. For instance, the light-blue thread is the core engine, responsible of maintaining the list of active flows. Also, in this thread, the *Request Dispatcher (RD)* module is a crucial entity interacting with all the other FM threads: it receives the action requests from them and translate the commands to the homologous block within the PMIPv6 daemon, in charge of executing the actions. The following paragraphs describe how the FM works, introducing the rest of threads and functional blocks.

The purple thread is responsible for handling flows packet by packet. Incoming packets are intercepted using the command[4]

```
# ip6tables -t mangle -A PREROUTING -j NFQUEUE --queue-num 0
```

which result is to push packets at user-space level, so that a `libnetfilter_queue`-based application listening on `queue-num 0` can manipulate them. The first purple block is thus a Deep Packet Inspection (DPI) module that extracts the 6-tuple parameters (see Chapter 4) from the intercepted packets. The 6-tuple is then passed to the core engine to check how to classify it. If the 6-tuple refers to a new flow, i.e., it is not present in the flow table, an “add flow” request is sent to the RD which translates it via socket 1 into an analogous request to the homologous module in the PMIPv6 daemon. The PMIPv6 RD checks if the destination or source address contains a prefix consistent with those stored in the Binding Cache. The reason for this check is that we want to classify only those flows that involve the MNs, and drop the initiative for the rest of the traffic. If the lookup succeeds, the RD on the PMIPv6 side replies to that on the FM side, including the flow-ID generated and the tunnel-ID used for that flow, retrieved from the BCE; otherwise, a void indication is provided, meaning that the flow cannot be processed. Upon receiving the reply, FM stores in the flow table this new stream with the related parameters, i.e. the 6-tuple, the flow ID and the tunnel used.

In the meantime, that is, while the classification stage is taking place, the packet is queued, waiting for a signal by the core engine. The signal can be a “mark verdict”, if a suitable flow-ID is provided (in this case the mark will be exactly the flow-ID), or a “void verdict” in case of empty response by the PMIPv6 daemon. The *mark* does not modify the content of the packet, it is a kernel-level label appended by `netfilter`, that can be used next for routing, as we show later, and/or for filtering by applications based on the same framework (e.g., `ip6tables`); the label is removed when the packet is forwarded to the network cards. In case the 6-tuple corresponds to an already existing flow, then the communication with the PMIPv6

[4] As observed in the paragraph before Section 4.1, we are interested in the downlink direction only, hence the command should include an additional filter for this purpose (e.g., by specifying the interface used for inbound traffic, etc.)

daemon is not necessary and the packet is marked with the related flow-ID obtained from the flow list.

Besides the “add flow” request, the RD in the FM accepts also a “move flow” command. As stated before, the flow table stores the ID of the tunnel traversed by the flow. When the flow has to be moved (two threads can issue the command and the input parameter is the flow-ID), the RD on the FM side sends a request to the PMIPv6 one, indicating the flow-ID to be moved, the prefix involved and the tunnel in use. The RD on the PMIPv6 daemon side checks the availability of tunnels for that MN by inspecting the flow-mob list, if the host is multihomed, or the BCE’s additional instances for the bonding terminal. If the additional path exists, the *rule manager* block adds a “fwmark-rule” pointing to a different routing table from the default (or *main*) one.

Routing rules management. The scope of the routing rules is to deviate from the standard routing procedure, by which the main routing table is inspected, for those packets that match a given condition (in our case we are using the mark, but, for instance, it can be the source address, for source routing). Indeed, the Linux kernel offers some options at the time of compiling to add more features to the default network stack. The IP stack for the kernel we configured comes with 256 routing tables, of which number 0 is called `local` and number 255 is `main` – the default one. In a shell console, when the command

```
# ip -6 route [add,del,show]
```

is issued, the main table is manipulated by default, but the other tables can be accessed as well, by explicitly defining the desired one (the ID can be a number, or a symbolic name):

```
# ip -6 route [add,del,show] table [table-ID]
```

The routing tables can contain opposite instructions for a desired prefix destination, allowing applications or users to select the most suitable path by inserting the aforementioned rules. The example below shows how to use three different default gateways to redirect packets through distinct paths: marked packets through wireless, and, in particular, those with mark `0x1` towards a default destination `2001::1/64`, mark `0x2` to `2002::2/64`; the unmarked packets are sent via the wired path to `2003::3/64`.

```
# ip -6 rule show
0:      from all lookup local
16383:  from all fwmark 0x1 lookup t1
16383:  from all fwmark 0x2 lookup t2
32766:  from all lookup main
```

```
# ip -6 route show table t1
default via 2001::1 dev wlan0 [...]

# ip -6 route show table t2
default via 2002::2 dev wlan1 [...]

# ip -6 route show
[...]
default via 2003::3 dev eth0 [...]
[...]
```

In our case it was only necessary to specify the tunnel interface (`ip6tnl`) through which forward the marked packets:

```
# ip -6 rule show
0:      from all lookup local
1000:  from all fwmark [FLOW-ID1] lookup tunnel1
1000:  from all fwmark [FLOW-ID2] lookup tunnel2
1000:  from all fwmark [FLOW-ID3] lookup tunnel2
1000:  from all fwmark [FLOW-ID4] lookup tunnel1
[...]
32766:  from all lookup main

# ip -6 route show table tunnel1
default dev ip6tnl1 [...]

# ip -6 route show table tunnel2
default dev ip6tnl2 [...]

# ip -6 route show
[...]
standard entries for the MNs' prefixes
Pref1:MN1::/64 dev ip6tnl1 [...]
Pref2:MN2::/64 dev ip6tnl1 [...]
Pref3:MN3::/64 dev ip6tnl2 [...]
Pref4:MN4::/64 dev ip6tnl3 [...]
[...]
```

so that, for instance, if packets with destination prefix `Pref1:MN1::/64` are marked with `FLOW-ID2`, the usual forwarding instruction (through `dev ip6tnl1`) is overridden (through `dev ip6tnl2`). In order to overcome the main table, a rule needs to be set with a priority (the number on the left) lower than 32766. Indeed the routing tables are inspected according to the matching rule with the lowest priority.

Getting back to the FM working scheme, after the fwmark-rule is set by the rule manager, the packets are forwarded through the new tunnel, bypassing the default forwarding method, based on longest prefix matching of the routing entries in the main table.

The last command accepted by both dispatchers is the “del” request to delete a flow, by which a flow is removed from the flow table, if present, and the corresponding rule is removed as well.

Besides the purple thread, three additional threads have access to the dispatcher. The command line (green) thread's main operation is to interpret manual-typed instructions (of the three types described before, plus an instruction to print the flow table) and to send the relative request to the dispatcher. This thread offers the possibility to monitor, reset and adjust the system if something went wrong with the automatic management.

The automatic management is performed by the polling (blue) thread, which monitors the flow table looking for expired flows (that is, flows no longer active after a configurable interval), and, more interesting, it determines the congestion on the tunnels, and executes a corresponding action, based on the desired policies. This behavior has been tested by setting a low bit-rate capacity over the tunnels using the `tc qdisc` utility. The FM periodically checks the tunnels' packet drop ratio, and, when the ratio crosses a given threshold, the FM moves the highest bit-rate flow on the congested tunnel to another one. Unfortunately, this is a coarse implementation, with many hard assumption for the sake of simplicity to quickly achieve some results and show the feasibility of the design. However, further implementations can refine this mechanism in order to achieve a better response; in fact, a more sophisticated tool might estimate systematically some communication parameters and move flows when necessary. Our simplistic policy is provided as starting point to show the possible usage of the FM, but, in principle, many applications can benefit of a tool that monitors and classifies the traffic traversing a node.

The last examined thread is the BCE Delete (grey) Thread, that is the only one triggered by the PMIPv6 daemon, and for which a separate communication is provided. This thread listens to BCEs' de-registration events and takes an appropriate action (deletion or moving) for those flows that carry the deleted prefix. A de-registration is triggered because the BCE lifetime is about to expire, and the host does not respond to a subsequent neighbor solicitation for that prefix. Hence, the LMA interprets it as the host's interface lost wireless connectivity. Unfortunately, the radio link might go down much before the LMA realizes it, because the BCE lifetime can be configured excessively long, and the actual detachment is not detected promptly. Nevertheless, some layer-2 mechanisms (as those described in

Section 5.2) might be applied to quickly react when an radio link is found to be active or inactive. Regarding the flow deletion phase, this is done when no other MN's IFs are connected to the network, and this operation optimizes the performance of the polling thread, because it anticipates the deletion that the polling thread would have done later on. However, if a link is lost, but still the MN is connected through another IF, it would be desirable to move all the flows to the active IF, to preserve the seamless mobility service. Unfortunately, this feature is very useful for the bonding terminal model, whilst it cannot always be guaranteed to the weak-host terminal. In fact, in the bonding model, all the host's interfaces share the same prefix, whereas, in the weak host model, if one interface loses connectivity, it may drop the address, hence the prefix (due to prefix-lifetime expiration or because the IF was turned off). In this case the weak host model does not hold anymore, since the host sees packets with a destination prefix that does not belong to any of its interfaces and thus discards the packet.

5.2 Mobile Access Gateway implementation

The PMIPv6 mobility support is enabled on the LMA and the MAGs. The wireless connectivity is offered by deploying

- WiFi (IEEE 802.11g) access, provided by two Linksys WRT54GL v1.1 Access Point (AP), running OpenWRT Kamikaze 7.09 as firmware[5], one connected to MAG_2 and the other to MAG_3 via an Ethernet cable.
- 3G (HSDPA) access, supplied by the Alcatel-Lucent in-house network by connecting via Ethernet cable the Gateway GPRS Support Node (GGSN) and MAG_1. A Point-to-Point Protocol (PPP) is used between the MN and the GGSN when the Packet Data Protocol (PDP) context is setup.

The MAGs implement the PMIPv6 engine to form PBUs, parse PBAs and install the required routing state for packet delivery. The PMIPv6 daemon does not enforce any AAA procedure; following the AAA mechanism specified by the PMIPv6 protocol is out of scope for our thesis, so we limited to develop the retrieval of the MN's profile, upon attachment to the domain, from a static configuration file. The profile contains the most important parameters for the MN, i.e., the MN-ID and the terminal's IFs characteristics, as the MAC address and the prefix(es) to be assigned to with the corresponding lifetime. Therefore, the weak-host terminal's profile contains two interfaces, while the bonding host's profile only includes one.

Furthermore, we assume that each MAG can only handle a single radio technology, that is, MAG_1 is only in charge of 3G access, while MAG_2 and MAG_3 only for

[5] https://openwrt.org/

WiFi. Since the 3G network only provides IPv4 connectivity, we setup an Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) [55] connection to convey IPv6 packets over the PPP IPv4 3G connection. That is, the in-house GGSN has been connected to MAG_1 and upon ISATAP establishment, the Router Solicitation generated by the MN is conveyed to the MAG through the ISATAP tunnel. As mentioned before, in addition to Router Solicitation messages, the WLAN MAGs are able to receive Layer-2 attachment triggers from the AP and start the PBU/PBA protocol exchange, thanks to a special communication between the APs and the PMIPv6 daemon. This is not possible to obtain for 3G as no interventions are allowed in the GGSN.

When a MN joins the network establishing a radio link with a MAG, the gateway retrieves the MN's profile and prepares a PBU with the necessary options:

- the MN-ID option (type 8);
- the MN Link Local ID option (type 25), containing the IF's MAC address;
- the HNP option (type 22), containing the prefix assigned to the MN's IF;
- the timestamp option (type 27);
- the ATT option (type 24);
- the Mu-Ho option (type 28, not IANA), if the MN's profile presents more than one IF (weak-host).

Some remarks are necessary.

1. In the first PMIPv6 daemon version we used, the the MN-ID was the IF's MAC address, and the LL-ID option was not used. This solution does not support multi-attachment, unless the MN configures the same MAC on all the interfaces, as different interfaces produce different mobility sessions at the LMA. In order to overcome this limitation, we assigned an arbitrary 48 bits MN-ID in the MN's profile, different from the IFs' MAC addresses, so that we could safely re-use the methods of the code, just changing the variables passed to the functions.

2. We suggested a possible usage of the HI option in Section 5.1 to distinguish a logical interface attachment from a handover. Unfortunately, we did not further develop this topic, but we claim that the most effective and suitable solution to fill this option in a consistent and dynamic way is to use a dedicated handover infrastructure as IEEE 802.21 [26, 56]. Hence the HI option is not included in the mobility messages.

3. Since we assume that a MAG only manages a single access technology, the ATT option is filled with a fixed value per each MAG. In particular, MAG_1 carries value 2 (PPP), MAG_2 is associated to value 4 (Wireless LAN - IEEE 802.11 a/b/g) and MAG_3 to value 1 (Logical Network Interface). In this way, when the bonding host establishes the second radio link, as the mobility options are identical for both interfaces except the ATT, the LMA is able to take the appropriate action, i.e., enabling multi-attachment support instead of handover management.

4. The Mu-Ho option is appended only when the MN's profile contains more than on interface. The option is sent in the PBU with all the fields set at 0, indicating that the MN is multihoming-capable. In the PBA, the option is omitted if no other interfaces are attached (i.e., no prefixes are registered at the LMA for that MN-ID), or filled with the corresponding prefix. In case a valid prefix is received back by the MAG, the gateway sets a route to the prefix via the MN's link-local address seen by the MAG in the RS message.

It is worth mentioning that a MAG implements the source-based routing, mentioned before in Section 5.1.1: all the packets containing as source a prefix allocated for a MN are forwarded through the tunnel to the LMA. If a MN is having a communication session with a CN connected to the same MAG, this scheme leads to sub-optimal routing, as the packets are forwarded unnecessarily to the LMA. This behavior can be avoided by setting the local routing mode in the MAG, but we are not considering this option, since we prefer to classify also this kind of IP flow at the LMA.

5.3 Mobile Node implementation

We tested our solution with the following configuration: the weak host terminal is connected to MAG_1 (3G) and MAG_2 (IEEE 802.11g), while the logical interface terminal is connected to MAG_2 and MAG_3 (both via IEEE 802.11g). Next sections give the details for each terminal configuration.

5.3.1 Weak host model

The weak host model is set by default in the IP stack in Linux-2.6 kernels both for IPv4 and IPv6. This model allows hosts to receive (send) packets from any interface as long as the packets' destination (source) is a valid address for one of the host's interfaces. The weak host MN has one WLAN interface and one 3G interface (Novatel USB dongle). As mentioned before when treating the MAG implementation, the Point-to-Point Protocol is used between the MN and the GGSN when the PDP

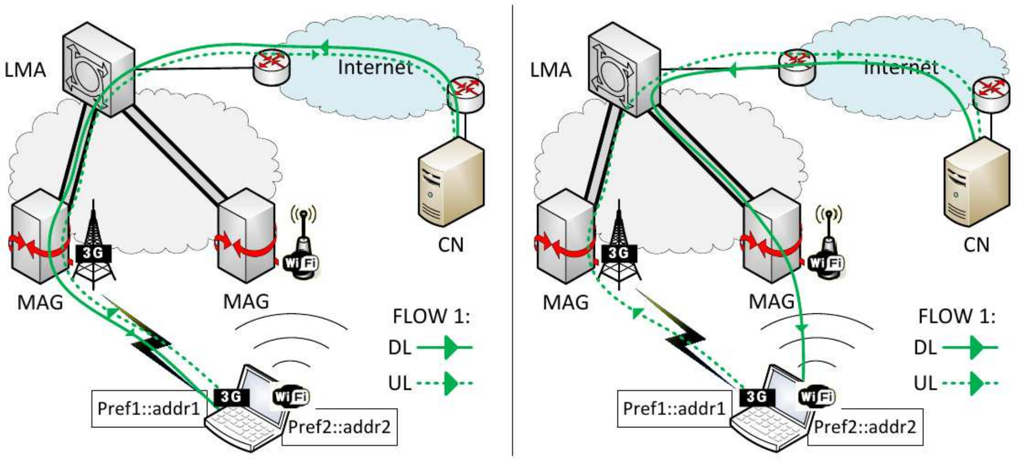


Figure 5.3. Downlink and uplink paths before (left) and after (right) flow mobility

context is setup. Thus, a PPP interface is configured on the MN (ppp0) and used as default one to reach the Internet. From a protocol behavior and flow management point of view, the use of the ISATAP tunnel over the PPP and GGSN-MAG links has no impact.

The main logic running in the MN is the *Packet Reflector*, a very simple implementation of a more general tool usually referred as Connection Manager (see Section 3.1). When the weak host MN performs network attachment, it receives two HNPs, one on each interface and the module assures that Uplink (UL) and Downlink (DL) packets belonging to a same flow are sent through the same interface. We use an example to explain its motivation and behavior. Let's assume the MN has started a communication with a CN through the 3G IF. When the MN attaches the second interface (WLAN) to another MAG, the LMA detects that the MN is multi-homing capable (see the left picture in Figure 5.3). The LMA may then decide to move the communication towards the new MAG which is now able to route for both prefixes. After moving the flow (right picture in Figure 5.3), we will observe that the DL stream is received by the second IF, while the UL stream is still sent through the first one, i.e. the IF formerly involved in the communication.

Indeed, for locally generated traffic, the applications choose the outgoing interface and the source address by inspecting the main routing table: the select route for the destination gives an indication of the interface that must be picked and its address is specified as source address in the packet header (there are a number of limitations with current source address selection left out of scope in this thesis, since we are interested in studying network-controlled flow mobility procedures and their performance). In our case, each time an IF gains IP connectivity, the MN adds a default gateway, represented by the on-link MAG through the IF itself, resulting in

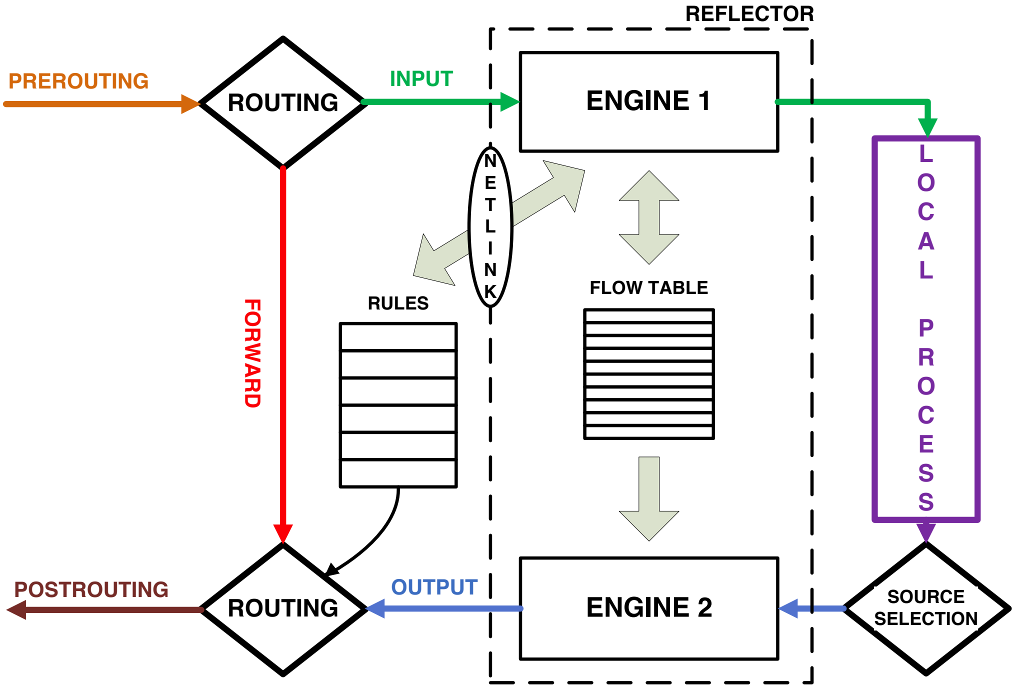


Figure 5.4. Packet Reflector

several default gateways entries in the routing table. Hence the first IF connected will be always considered the default outbound IF, as the entry appears first in the list, unless a more specific route for the destination is present.

Hence, we wrote a small module that takes care of monitoring at which interface IP packets are received, making sure that the response packets are sent by the MN through the same IF. The “Packet Reflector” module avoids the mismatch of the DL and UL paths by running two separate engines (see Figure 5.4). The first engine intercepts all the incoming packets (i.e., it is listening to every interface) and classifies them into separate flows using the flow 6-tuple matching criteria. All the collected flows are stored in a table and are associated with a receiving IF-ID field and with an unique flow-ID field. Note that, for network-controlled flow mobility, this flow-ID can be local to the MN, completely uncorrelated with that stored at the LMA for the same flow. This first engine also sets a “fwmark-type” rule indicating that the packets marked with a specific flow ID must be transmitted through the interface associated to that flow.

The second engine collects all the outgoing packets and checks whether they belong to a known flow. If the lookup succeeds, the packets will be marked with the correspondent flow ID, determined by the first engine when populated the flow table, and thus they are transmitted to the proper interface according to the rule set before.

Therefore, in the use case described above, we force the uplink and downlink streams for a given IP-flow to use the same path. If the LMA moves the flow again, the reflector detects that an already stored flow has changed incoming interface and thus upgrades the flow entry with the new IF and changes the rule for outbound sending.

Again, the packets interception is obtained with `netfilter_queue`, a tool that provides a method to pass packets from kernel-space to user-space applications. It reads packets from a particular data structure named `NFQUEUE` that is filled using `ip6tables` and makes them available to user manipulation (refer to Section 5.1.1). In the reflector we create two `NFQUEUE`s, the first one hooks in the `INPUT` chain and the second in the `OUTPUT` chain, which, thanks to the `netfilter` framework, collect packets respectively addressed to and sent by the host. We fill the queues by invoking

```
# ip6tables -t mangle -a INPUT  -j NFQUEUE --queue-num 0
# ip6tables -t mangle -a OUTPUT -j NFQUEUE --queue-num 1
```

and each engine works on its correspondent queue.

5.3.2 Bonding model

The bonding MN features the Linux bonding module, suitably modified to install our specific transmitting policies. The Linux bonding module (depicted in Figure 5.5) creates a virtual interface (`bond0`, `bond1`, ...) that groups several physical network interfaces (called "slaves") into one network device. In the standard activation mode, the virtual interface configures its MAC and link local address from the first enslaved device and these parameters will then be shared by all the other enslaved interfaces by substituting their own parameters. The bonding interface will then configure a valid IP address. This procedure creates a set of cloned interfaces, all having the same MAC and IP address without conflicts with each other.

In our scenario, the bonding terminal is created "enslaving" two wireless network interfaces, each of them connected to the WLAN AP. It should be noted that the APs run special purpose software (on top of the OpenWRT distribution) to perform network attachment/detachment detection of WLAN stations. That is, upon successful Layer-2 association, the AP sends to the MAG an *AttachmentTrigger* to bootstrap the PMIPv6 registration procedure. After the attachment of the two wireless physical interfaces, the MN has an HNP configured on the bonding device and can receive packets on any of the two physical interfaces.

From the receiving point of view, this mechanism provides different physical accesses to the host with the same IP and MAC address, while, from the sending point of view, different policies are pre-defined to choose the transmitting interface. Since

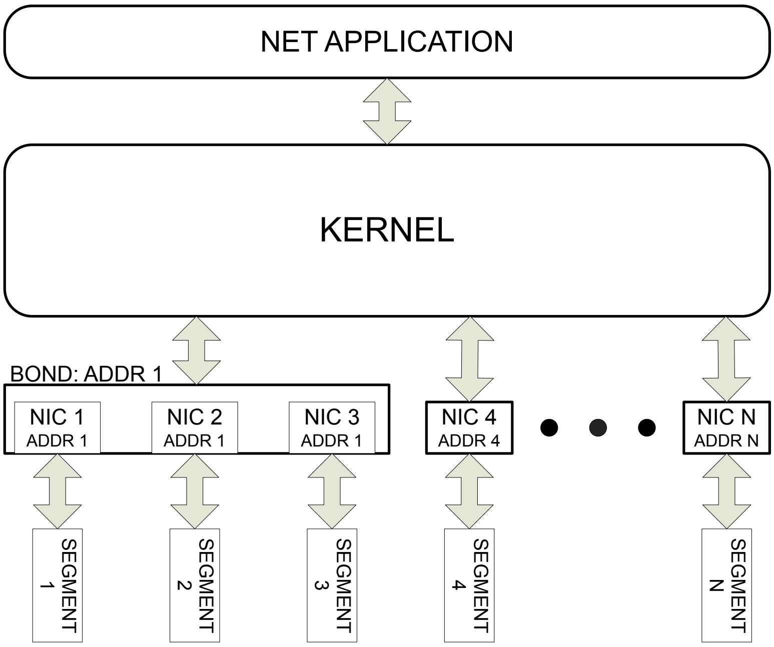


Figure 5.5. Bonding module

these policies do not meet our constraint of dynamically choose the transmitting media (i.e., to replicate the network decision), a custom extension to the module is coded, with which a transmitting slave can be chosen according to the source port number.

The extension is developed using the Packet Reflector's design, but, unlike the packet reflector's automatic response to the network decision, the IF selection is executed by an external trigger. Therefore, the module stores the incoming flows in a table, indicating the destination port number and a default slave interface to be used when transmitting with that port as source. The table is made accessible by an user-space application, allowing manual intervention to change the interface to be used, thus forcing the bonding module to transmit packets through the specified IF. Unfortunately, manual intervention produces a gap between the instant the flow is moved by the network, and the time the MN reacts to the change, but we argue that a more advanced kernel module might easily solve the issue.

Chapter 6

Evaluation Tests and Results

This chapter provides an experimental analysis of the mechanisms designed to enable flow mobility in PMIPv6 domains. Different tests were performed to validate the feasibility of the proposed approach. We consider two main situations in our experimental evaluation:

1. QoS triggered flow mobility. The movement of a flow (or set of flows) from one interface to another is triggered by QoS reasons. For example, the access network to which an interface is attached might not be able to cope with all the traffic, so the operator decides to offload a flow (or set of flows) to an interface connected to a less congested access network. This type of mobility is typically proactive.

2. Interface outage triggered flow mobility. A completely different situation appears when all the flows bound to a given interface have to be moved because the interface has just gone down. This might happen because the user has just manually switched down an interface (e.g., to save some battery life or money) or because of radio coverage. This type of mobility is typically reactive.

As explained throughout the thesis,two different types of mobile nodes are supported by our solution, following different paradigms: the logical interface and the weak host model. Although from a conceptual viewpoint our solution should behave quite similarly with both approaches, due to the particular implementations that we use for the experiments, there are some limitations that have an impact on the type and number of the tests that can be performed:

- The logical interface based MN is implemented by using the Linux bonding driver. This driver is designed for physical Ethernet interfaces only. Although other Ethernet-based technologies, such as WLAN, are also supported, it is not possible to bond (i.e., group under the same logical interface) 3G interfaces, as a logical PPP interface is brought up when 3G is enabled and the bonding module does not support non-physical interfaces.

- The weak host model does not allow the prefixes assigned to an interface to survive if the interface is shut down, as they are bound to the physical interface. Because of this limitation, we do not perform tests with the weak host MN in which an interface is completely turned down (this actually would correspond to a complete handover). Note that with some support from the terminal, this limitation might be overcome by not fully shutting down the interface, but just turning the radio off.

We want to emphasize that the main goal of this chapter is not to characterize quantitatively the behavior of the two approaches, nor to compare them, but rather to experimentally validate the design of our solution, by conducting different experiments with a real implementation.

6.1 QoS triggered flow mobility handovers

We analyze in this section the behavior of the flow mobility procedures when the Flow Manager (located at the LMA) receives QoS related triggers. We first proceed to analyze the WLAN to WLAN scenario for the bonding MN and then compare the obtained results with the WLAN to WLAN scenario for the weak host MN. The goal is to show that there is no difference from a flow management point of view. We then proceed to analyze the more compelling WLAN to 3G flow mobility scenario. It should be noted that the latter scenario is the baseline for any optimization algorithm aiming at offloading the 3G network.

6.1.1 WLAN-WLAN scenario

These experiments are performed using an MN which operates through two identical WLAN interfaces. It is worth noting, in order to understand the experiment, that the delay between the LMA and each interface of the MN is the same, without adding any artificial delay between both entities. As TCP is the predominant type of traffic in the Internet nowadays, we use TCP flows in the tests. During this experiment we simulate a degradation of the link used by the flow under inspection, triggering a handover due to an increase in the number of packet losses. In order to do so, we use the `tc` (traffic control) tool of the Linux OS for access control. By using the traffic shaping feature (`tc qdisc`, “queue discipline”) we are able to decrease the capacity of the tunnel between the LMA and the MAG, forcing a flow handover once the packet loss reaches a given threshold.

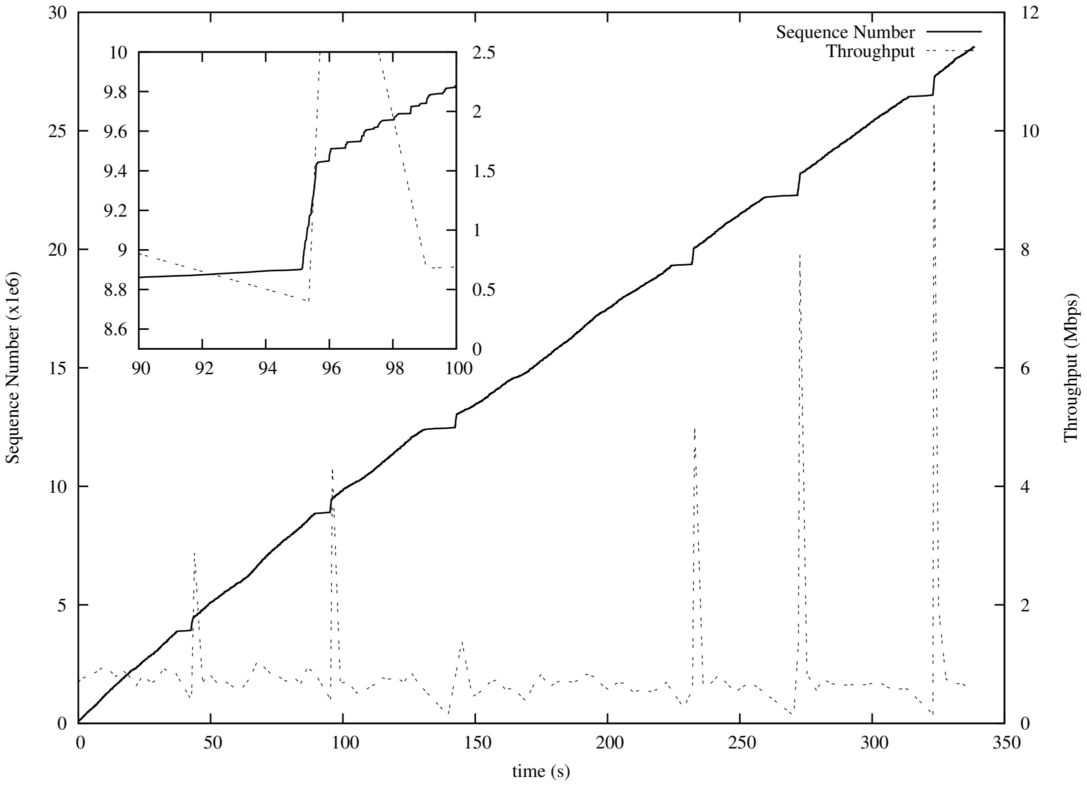


Figure 6.1. Bonding MN, QoS scenario, TCP sequence number and Average throughput vs Time

Fig. 6.1 presents the plot of TCP sequence number and throughput vs. time for the scenario explained before and using a bonding MN. It can be observed how the sequence number graph presents six step regions, starting in 37, 89, 130, 224, 257 and 314 seconds. These step regions correspond to the packets losses due to the effect of the traffic shaping. Once the flow is moved appropriately, the TCP sequence number starts increasing again since in the new path no losses occur. The same effect can also be appreciated in the throughput. At the same time intervals when the sequence number graph reduces its slope, the average throughput depicted in the figure decreases, since packets are lost at the receiver, and data segments are retransmitted. A close-up of one of the step regions is also presented in Fig. 6.1 for better understanding. It shows that the step region is not continuously flat as packets are being dropped by the traffic shaper progressively. Note that the mechanisms used to emulate congestion and to detect packet losses are not perfect. Some packets need to be lost before detecting the congestion of a particular path, and then triggering the subsequent flow mobility handover. This has an impact on the performance experienced by the user, which could be reduced by deploying more

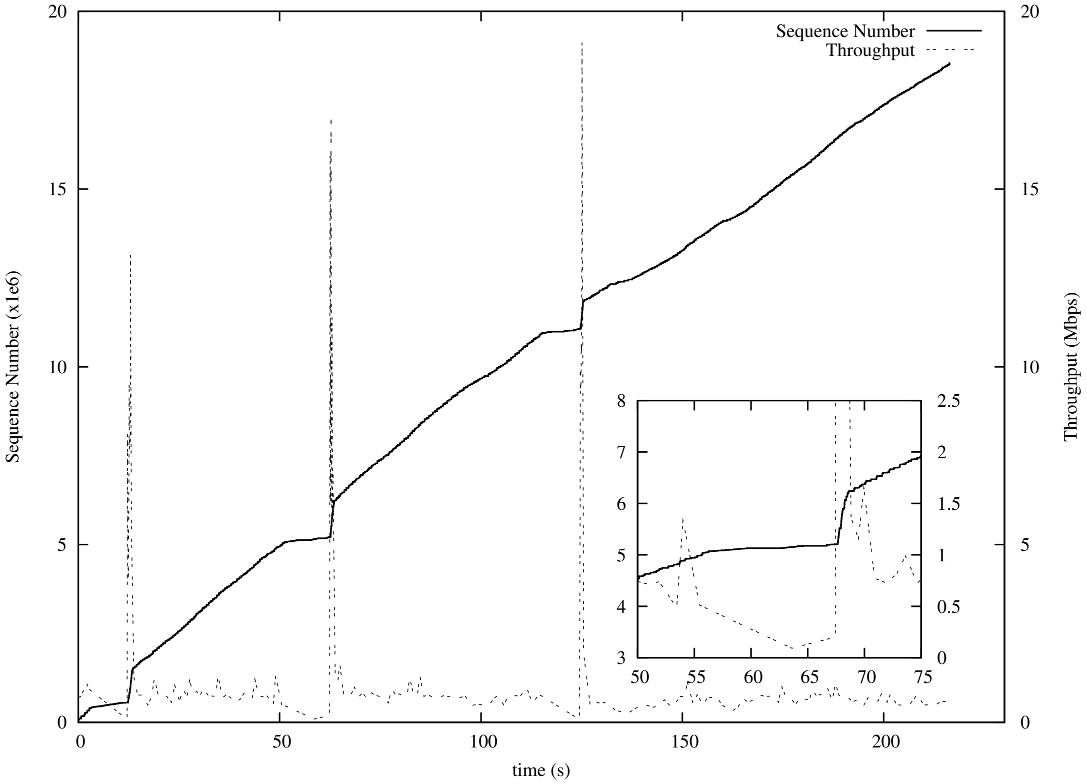


Figure 6.2. Weak Host MN, QoS scenario, TCP sequence number and Average throughput vs Time

intelligent network congestion mechanisms. In a real operator's network there are more complex tools available that could be used to help triggering flow mobility in a more effective way (i.e., shorter – close to zero – service disruption times). These experiments, however, might serve as starting point for a performance comparison for future implementations of these mechanisms.

In order to have a qualitative comparison between the weak host model and the bonding interface for flow mobility due to QoS constraints, we perform the same experiment using two WLAN cards also on the weak host MN (results are shown in Fig. 6.2). Looking at Fig. 6.1 and Fig. 6.2, it can be concluded that there are no significant differences between the observed behavior, which supports the idea that the performance of our solution is not affected by the type of MN (weak host or bonding one), but rather on the reaction time of the congestion detection mechanism. This conclusion is further motivated by the graph in Figure 6.3, which illustrates the overlapping traces obtained when running once more the test for each of two terminals. The picture shows that the qualitative behavior is identical, whereas the quantitative mismatch is due to the fact that congestions were not simulated at the

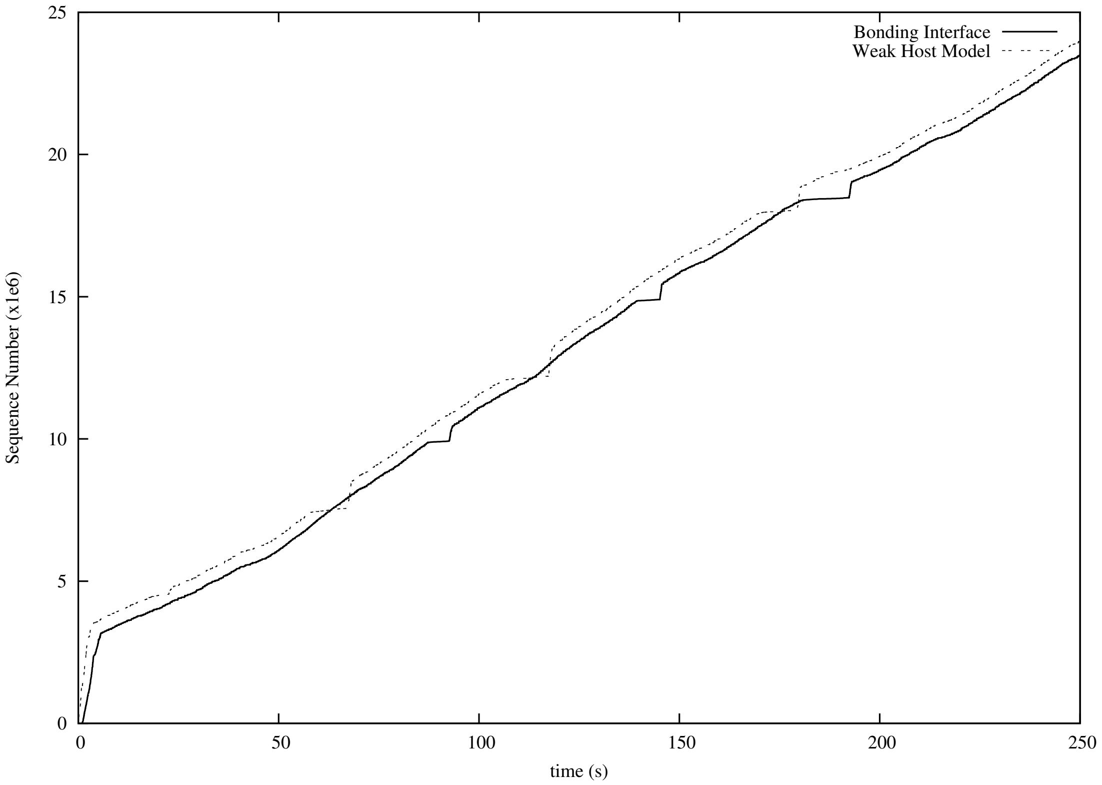


Figure 6.3. Bonding and Weak Host terminals comparison, QoS Scenario, Sequence number vs time

same times, and the reaction time after a congestion (i.e., the length of the flat trace) is responsibility of the the FM at the LMA, not due to the MNs.

6.1.2 WLAN-3G scenario

This experiment explores the inter-technology flow mobility due to QoS changes. The experiment setup is similar to the one previously depicted, but herein we focus on the relevant aspects of the handover between two different technologies. The experiment consists in the streaming – using TCP – of a video to an MN connected to two different MAGs through WLAN and 3G. As in the previous tests, the quality of the links between the LMA and MAG is affected by the use of the traffic shaping characteristics of the Linux Kernel, through the `tc qdisc` command. Fig. 6.4 presents the results obtained.

Fig. 6.4 shows the sequence of the different handovers, triggered by the packet loss ratio crossing a configured threshold. Again, we should note that in a real operator's scenario, the network would be able to predictively trigger flow mobility handovers,

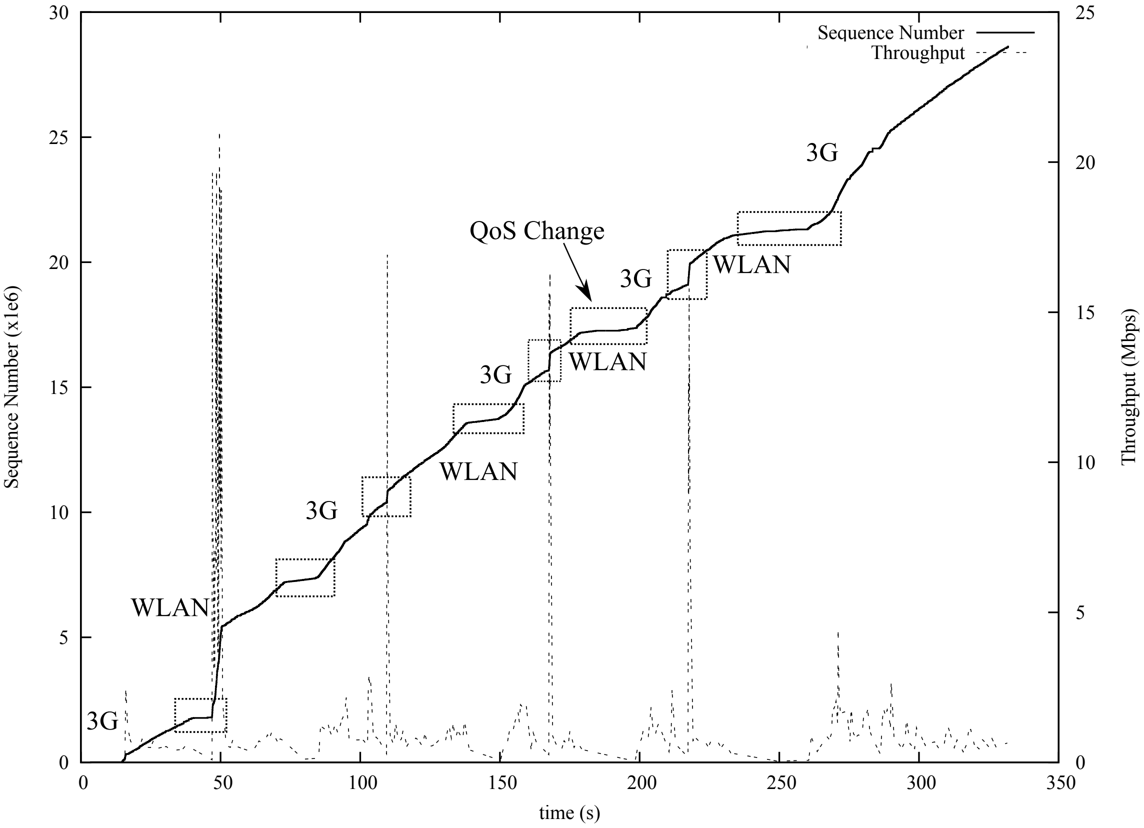


Figure 6.4. Weak host MN, WLAN-3G QoS scenario, TCP Sequence number and Average throughput vs Time

without needing to wait for a reaction upon packet losses. The experiment starts with the MN attached to the 3G network, since this is the interface defined as default. A total of eight handovers are performed in this test, each one moving the flow from the congested access network to the one without QoS constraints. As in the WLAN to WLAN experiment, the sequence number graph does not remain completely flat during the retransmissions, since the interface is affected by losses, but it never goes completely down. The instants where a flow is moved from one interface to another can be easily identified due to the fact of the average throughput decreases during the handover (this would not be the case for handovers triggered predictively by the network). Once the handover is performed, we can see an abrupt increment in the sequence number graph caused by the TCP retransmissions.

Finally, from Fig. 6.3 and Fig. 6.4, we can conclude that the designed solution is feasible and works in a real environment. Therefore, our approach could be used by network operators to provide seamless inter-technology flow mobility, fulfilling operators desires while not impacting the final user's experience. Note that while in our experimental validation handovers have been triggered upon reaction to packet losses, in a real QoS-enabled mobile operator scenario, the network would be able to predict path congestion, and therefore react accordingly to solve this by issuing flow mobility.

6.2 Interface outage triggered flow mobility

This section describes the flow mobility procedures when the LMA receives Proxy Binding Update messages with a lifetime value set to zero (in terms of protocol operations it means that an MN has disconnected from the sending MAG). Due to the limitations explained before, we first test the scenario for the WLAN to WLAN case using the bonding MN. We argue however that from a protocol operation point of view the same considerations apply to weak host terminals. We finally relate an out of coverage scenario to one in which a weak host MN performs a WLAN to 3G flow handover triggered manually. It should be noted that there is no impact on the protocol operation (only the trigger changes).

6.2.1 WLAN-WLAN scenario

As in the previous experiment, herein an MN with two identical WLAN interfaces is considered and no artificial delay is added to any of the paths between the LMA and the MN. This experiment analyzes the flow mobility when triggered by an out of coverage scenario of the interface serving the flow. When the MN's currently active interface is switched off, the flow is automatically moved to the remaining active interface (thanks to the Layer-2 attachment/detachment code, which allows the MAG quickly detect the MN's interface de-association). We then move back and forth the flow by alternating the active interface.

Fig. 6.5 presents the TCP sequence number and Average throughput vs time graphs. As in the scenario presented in the previous experiment, four step regions can be identified in the sequence number vs time graph. These step regions start at 26, 51, 76 and 100 seconds respectively. If we analyze the close-up of the figure, it can be seen how in this case the region is completely flat, in contrast with the results shown in the previous experiments (QoS triggered flow mobility handovers). There is no progressive loss of packets, since the interface is abruptly turned down. The different step regions are for all cases shorter than the ones presented in the previous experiments, because in this case the interruptions correspond to the time required by the network to detect and signal the interface disconnection, and then to re-route appropriately the affected flow.

It is worth noticing that we only perform this experiment for the bonding MN, for the reasons highlighted at the beginning of this section regarding the weak host MN.

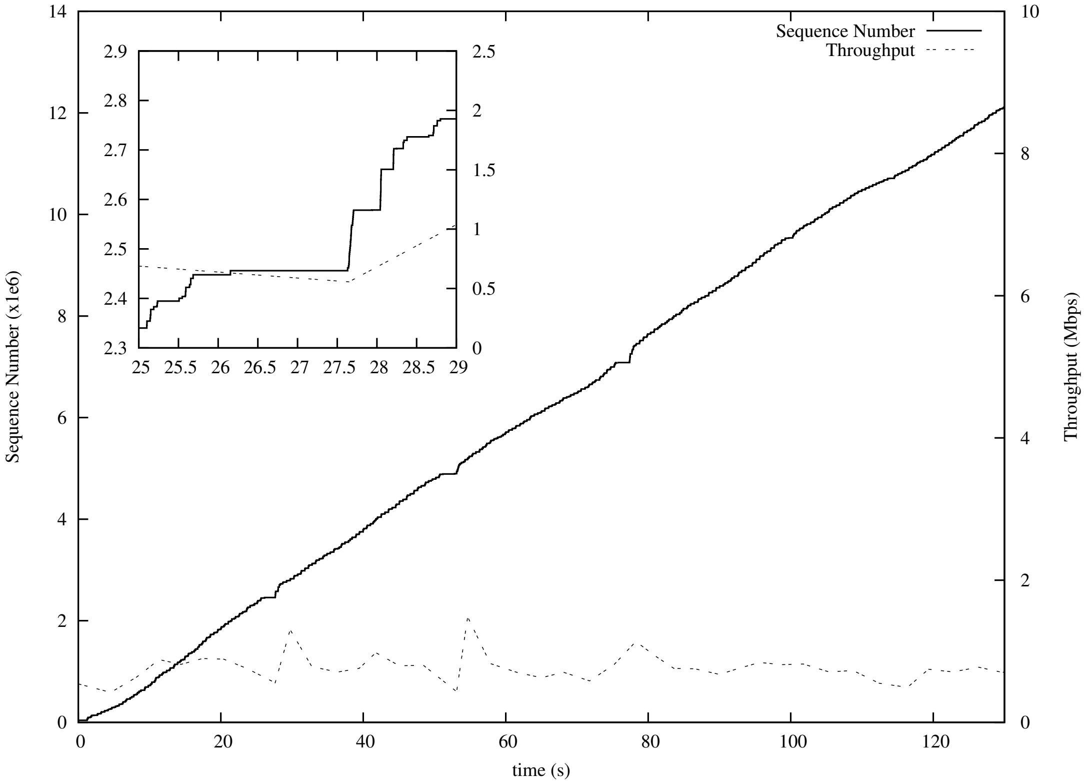


Figure 6.5. Bonding MN, Outage scenario, TCP sequence number and Average throughput vs Time

In the case of the bonding terminal, the IP prefix is delegated to the unique logical IP interface, instead to each individual IP (physical) interface as in the case of the weak host MN. This difference yields to a strange behavior of the weak host terminal when the interface is turned down, removing the IP prefix from the shutdown interface. Hence the outage experiment cannot not be conducted using the weak host model node. However, this same situation can be emulated by manually moving the flows from one interface to other, using the command line console offered by the FM. Next subsection is devoted to describe this.

6.2.2 WLAN-3G scenario

This experiment considers an MN which has an IEEE 802.11a/b/g card as one of its interfaces, while the second interface is a standard 3G modem. Herein we focus on the evaluation of a handover case emulating an out of coverage scenario. The MN starts a TCP video flow in the 3G interface and this flow is manually switched to the WLAN and 3G back and forth. Fig. 6.6 presents the results of this test. As shown in the figure, the bandwidth requirements of the video are quite low, hence the video does not suffer from congestion while being transmitted/received at any of the interfaces. We select this scenario since we want to assess the impact of changing the underlying technology to a standard traffic without QoS constraints. The observed results show that the handover between both technologies is almost transparent from the viewpoint of the flow performance. In the case of WLAN to 3G handover, we find that for each handover, some retransmissions occur, as the bandwidth of the 3G interface is lower than the WLAN one, and its delay is higher. This decrease in the performance would be hardly noticeable due to the low requirements of the traffic being used and the fact that the TCP pace is recovered quickly. For the case of the 3G to WLAN handover we find the inverse behavior, observing an increase in the speed of the sequence number growth. Observed results show that our design does not impose any penalty in the performance of the flow apart from the effect of changing the characteristics of the underlying technology, which is known to affect the TCP performance. Nevertheless, the flow handover itself is seamless and transparent for the involved communications peers.

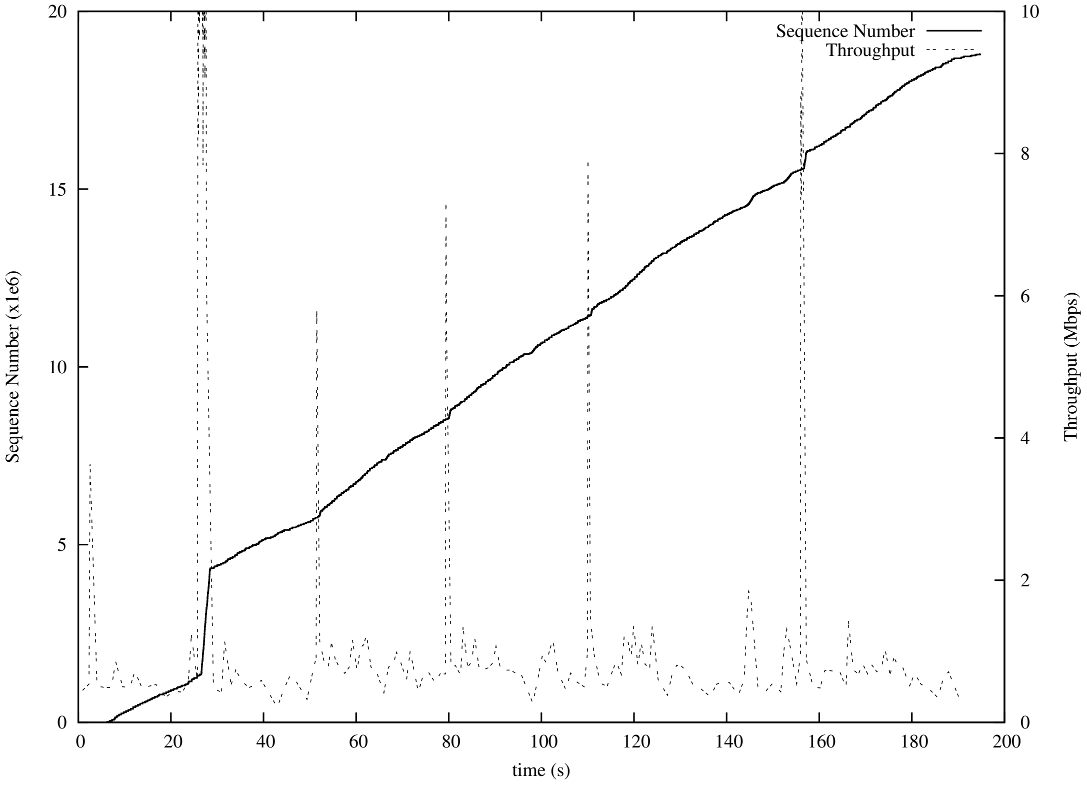


Figure 6.6. [
WLAN-3G,]Weak host MN, WLAN-3G Handover Scenario, TCP sequence number and Average Throughput vs time

Chapter 7

Conclusions

In this thesis we present an end-to-end system design featuring flow mobility extensions for the Proxy Mobile IPv6 protocol. As network-based solutions for mobility has shown to be the most interesting for deployment in the next generation all-IP mobile architectures, we started from ongoing discussions in the 3GPP and IETF standardization fora, to finally derive the required design choices covering both network components and multi-mode mobile devices. We hence adopt, as protocol extensions, what is currently the most widely accepted solution on this field, that is the draft produced by the NETEXT IETF working group [42].

The purpose of the thesis is to prove the feasibility of the proposed design by means of a real testbed setup and the related software implementation. Thus, when describing and developing the solution, we emphasize the implications of flow mobility support on hand-held devices, for which two different configurations (single logical IP interface and multiple IP interfaces) have been presented and validated. Indeed, having worked on the realization of the IETF draft's ideas gave us insights to further improve the draft itself, or some related work, as the logical interface concept.

However, while setting up the prototype for the experiments, we faced some issues due to the availability of software for our setup, so some simplifications limit the scope of our tests, as reported in the last chapters. Fortunately, the realization of the PMIPv6 protocol we started from presented a large number of functions to replicate almost all RFC 5213 specifications, and, in particular, the missing features were not necessary nor relevant for flow-mobility purposes (e.g., the use of IPsec on the LMA-MAG links). Nevertheless, we exploited the lack of code to introduce some interesting enhancements as the implementation of a customized layer-2 attachment detection and the subsequent MN's policy profile retrieval, suitable to simulate an AAA procedure with the multi-homing and multi-attachment requirements constraints.

With the PMIPv6 domain up and running, the most relevant feature of our design is the Flow Manager engine. Indeed, besides the extension applied to the mobility protocol to support the solution, the FM is the core functional block that in fact enables flow mobility. Given that PMIPv6 relies on a central entity, the LMA, cardinal node for both the control and data plane, as it manages the mobility sessions and it is traversed by users data traffic, we decided to provide the LMA with an additional functionality to filter, classify and separately route the IP flows forwarded by the node. These features are realized by the FM, which, due to implementation decisions, is presented in this thesis as a detached module from the LMA, but, still, needs to interact with it to provided mobility simultaneously on a host-basis and flow-basis.

The tests show that flow mobility in PMIPv6 based networks is feasible for TCP based data traffic. It is worth noting that the testbed setup features a real 3G in-house network compounded by WLAN coverage, and that experiments have been conducted with commercially available devices (e.g., 3G USB dongle). The implementation work is thoroughly documented in a dedicate chapter witnessing the effort in combining standard PMIPv6 routing with enhanced procedures for flow management. The reader should be comfortable in reproducing a similar setup if required.

This thesis is based on a work by the same author [10] in collaboration with Dr. Telemaco Melia, Prof. Carlos J. Bernardos, Dr. Antonio de la Oliva and Prof. Maria Calderon, where the details of the implementation were omitted, but, instead, authors investigated whether the simultaneous use of two or more wireless interfaces can be a blocking factor to the wide adoption of seamless IP flow mobility management, due to the additional battery consumption. To show its feasibility, authors have analyzed the energy consumption of a simultaneous use of multiple network interfaces, focusing on WLAN and 3G access. The tests, conducted on an experimental platform, successfully justify our choices and the proposed end-to-end design.

To the best of authors' knowledge this is one of the first and most complete studies on flow mobility support for the PMIPv6 protocol. The thesis combines an up to date review of current standardization activities with an extensive implementation effort, which also resulted in satisfactory output. Indeed, the testbed model was presented in two international events as the World Mobile Congress held in Barcelona, Spain, in February 2011, and at the Future Network and Mobile Summit in Warsaw, Poland, in June 2011.

The former event is an international exposition for vendors; the demo stand presented there included several enhancements for the FM provided by a professional team for network and software solutions accounting researchers from Alcatel-Lucent and InterDigital. The main scope was to improve the FM, enriching the features and ameliorate the congestion detection mechanisms.

The latter is an international summit for European projects where we proposed

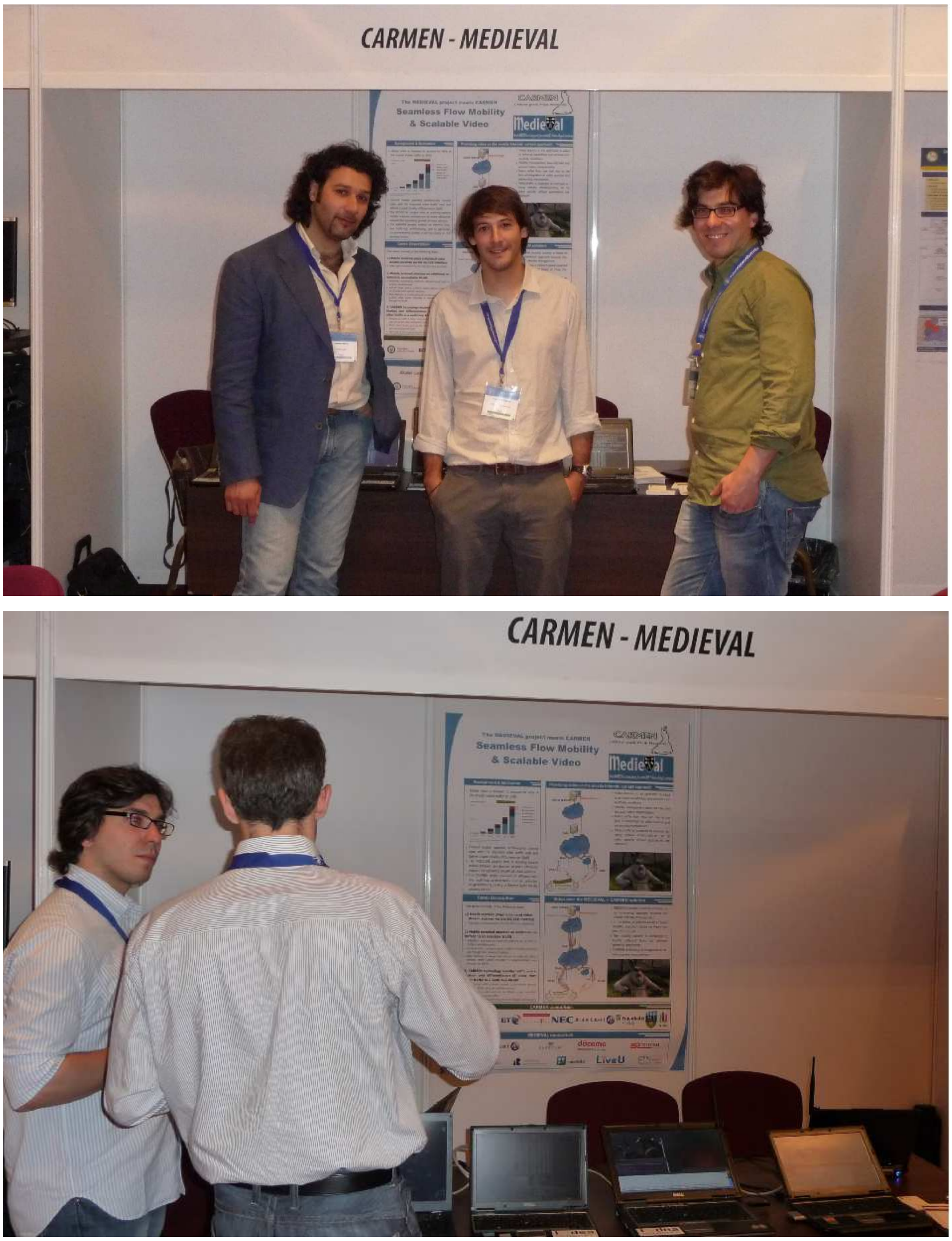


Figure 7.1. Flow Mobility demo-stand at Future Network and Mobile Summit, Warsaw, Poland, 15–17 June 2011

a flow mobility usage demonstration with the exact same prototype described in this thesis. The testbed was intended as platform to show some features envisioned by the MEDIEVAL[1] European project, therefore we implemented mechanisms for enhanced video delivery to a multihomed terminal. We used an SVC video codified into two layers, one with very low bitrate (basic layer) and the other with high bitrate (enhancement layer). Without intervening in the streaming server, the LMA was able to filter the video layers into two flows and route them separately, according to the availability of bandwidth in the access network, i.e., the enhancement video layer was dropped as long as the terminal was connected through 3G only, but, after the detection of the MN attachment via WiFi also, both layers could be delivered through the two access networks, resulting in an augmented perceived quality.

The next steps for flow mobility consist on promoting further the ideas proposed in [7] and [42] at the NETEXT IETF working group while evolving the platform as the standard itself will evolve. Also, it is interesting to explore the integration of flow mobility with dedicated handover control infrastructures as IEEE 802.21, with the purpose of achieving seamless technology and flow handovers without affecting the users' QoE, and in parallel, offering to operators working solutions for enhancing their network equipments.

[1] www.ict-medieval.eu

Figure 7.2. Flow Mobility demo-stand at Future Network and Mobile Summit, Warsaw, Poland, 15–17 June 2011

Bibliography

[1] D. Johnson, C. Perkins, and J. Arkko, “Mobility Support in IPv6,” RFC 3775 (Proposed Standard), Internet Engineering Task Force, June 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3775.txt

[2] H. Soliman, “Mobile IPv6 Support for Dual Stack Hosts and Routers,” RFC 5555 (Proposed Standard), Internet Engineering Task Force, June 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5555.txt

[3] J. Kempf, “Problem Statement for Network-Based Localized Mobility Management (NETLMM),” RFC 4830 (Informational), Internet Engineering Task Force, Apr. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4830.txt

[4] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, “Proxy Mobile IPv6,” RFC 5213 (Proposed Standard), Internet Engineering Task Force, Aug. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5213.txt

[5] CJ. Bernardos, T. Melia, P. Seite, J. Korhonen, “Multihoming extensions for Proxy Mobile IPv6,” Internet Engineering Task Force, draft-bernardos-mif-pmip-02.txt (work-in-progress), March 2010.

[6] R. Wakikawa, S. Kiriyama, and S. Gundavelli, “The applicability of virtual interface for inter-technology handoffs in Proxy Mobile IPv6,” *Wireless Communications and Mobile Computing*, 2009.

[7] S. Gundavelli and T. Melia, “Logical Interface Support for multi-mode IP Hosts,” Internet Engineering Task Force, draft-ietf-netext-logical-interface.support-02.txt (work-in-progress), March 2011.

[8] R. Braden, “Requirements for Internet Hosts - Communication Layers,” RFC 1122 (Standard), Internet Engineering Task Force, Oct. 1989, updated by RFCs 1349, 4379, 5884. [Online]. Available: http://www.ietf.org/rfc/rfc1122.txt

[9] D. Thaler, “Evolution of the IP Model,” RFC 6250 (Informational), Internet Engineering Task Force, May 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6250.txt

[10] T. Melia, C. J. Bernardos, A. de la Oliva, F. Giust, and M. Calderon, “IP Flow Mobility in PMIPv6 Based Networks: Solution Design and Experimental

Evaluation," *Wireless Personal Communication*, vol. Special issue, 2011.

[11] M. Riegel and M. Tuexen, "Mobile SCTP," Internet-Draft (work in progress), draft-riegel-tuexen-mobile-sctp-09.txt, Nov. 2007.

[12] C. Perkins, "IP Mobility Support for IPv4," RFC 3344 (Proposed Standard), Internet Engineering Task Force, Aug. 2002, obsoleted by RFC 5944, updated by RFC 4721. [Online]. Available: http://www.ietf.org/rfc/rfc3344.txt

[13] ——, "IP Encapsulation within IP," RFC 2003 (Proposed Standard), Internet Engineering Task Force, Oct. 1996, updated by RFC 3168. [Online]. Available: http://www.ietf.org/rfc/rfc2003.txt

[14] S. Deering, "ICMP Router Discovery Messages," RFC 1256 (Proposed Standard), Internet Engineering Task Force, Sept. 1991. [Online]. Available: http://www.ietf.org/rfc/rfc1256.txt

[15] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," RFC 2827 (Best Current Practice), Internet Engineering Task Force, May 2000, updated by RFC 3704. [Online]. Available: http://www.ietf.org/rfc/rfc2827.txt

[16] G. Montenegro, "Reverse Tunneling for Mobile IP, revised," RFC 3024 (Proposed Standard), Internet Engineering Task Force, Jan. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3024.txt

[17] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104 (Informational), Internet Engineering Task Force, Feb. 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2104.txt

[18] C. Perkins, D. Johnson, and J. Arkko, "Mobility Support in IPv6," RFC 6275 (Proposed Standard), Internet Engineering Task Force, July 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6275.txt

[19] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861 (Draft Standard), Internet Engineering Task Force, Sept. 2007, updated by RFC 5942. [Online]. Available: http://www.ietf.org/rfc/rfc4861.txt

[20] J. Arkko, V. Devarapalli, and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents," RFC 3776 (Proposed Standard), Internet Engineering Task Force, June 2004, updated by RFC 4877. [Online]. Available: http://www.ietf.org/rfc/rfc3776.txt

[21] V. Devarapalli and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture," RFC 4877 (Proposed Standard), Internet Engineering Task Force, Apr. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4877.txt

[22] S. Kent, "IP Encapsulating Security Payload (ESP)," RFC 4303 (Proposed Standard), Internet Engineering Task Force, Dec. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4303.txt

[23] V. Manral, "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)," RFC 4835 (Proposed Standard), Internet Engineering Task Force, Apr. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4835.txt

[24] P. Nikander, J. Arkko, T. Aura, G. Montenegro, and E. Nordmark, "Mobile IP Version 6 Route Optimization Security Design Background," RFC 4225 (Informational), Internet Engineering Task Force, Dec. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4225.txt

[25] H. Soliman, C. Castelluccia, K. E. Malki, and L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)," RFC 4140 (Experimental), Internet Engineering Task Force, Aug. 2005, obsoleted by RFC 5380. [Online]. Available: http://www.ietf.org/rfc/rfc4140.txt

[26] LAN/MAN Committee of the IEEE Computer Society, "IEEE Std 802.21-2008, Standards for Local and Metropolitan Area - Part 21: Media Independent Handover Services," 2008.

[27] I. Soto, C. J. Bernardos, M. Calderon, and T. Melia, "PMIPv6: A Network-based Localized Mobility Management solution," *The Internet Protocol Journal*, vol. 13, no. 3, Sept. 2010.

[28] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865 (Draft Standard), Internet Engineering Task Force, June 2000, updated by RFCs 2868, 3575, 5080. [Online]. Available: http://www.ietf.org/rfc/rfc2865.txt

[29] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter Base Protocol," RFC 3588 (Proposed Standard), Internet Engineering Task Force, Sept. 2003, updated by RFCs 5729, 5719. [Online]. Available: http://www.ietf.org/rfc/rfc3588.txt

[30] S. Y. Hui and K. H. Yeung, "Challenges in the Migration to 4G Mobile Systems," *IEEE Communications Magazine*, vol. 41, no. 12, pp. 54–59, December 2003.

[31] H. Yokota, S. Gundavelli, T. Trung, Y. Hong, and K. Leung, "Virtual Interface Support for IP Hosts," Internet Engineering Task Force, draft-yokota-netlmm-pmipv6-mn-itho-support-03.txt (work-in-progress), March 2010.

[32] I. van Beijnum, *BGP*. O'Reilly, 2002.

[33] E. Nordmark and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6," RFC 5533 (Proposed Standard), Internet Engineering Task Force, June 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5533.txt

[34] D. Farinacci, D. Lewis, D. Meyer, and C. White, "LISP Mobile Node," Internet-Draft (work in progress), draft-meyer-lisp-mn-05.txt, May 2011.

[35] A. Akella, B. Maggs, S. Seshan, and A. Shaikh, "On the performance benefits of multihoming route control," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 1, pp. 91 –104, Feb. 2008.

[36] M. Wasserman and P. Seite, "Current Practices for Multiple Interface Hosts," Internet Engineering Task Force, draft-ietf-mif-current-practices-12.txt (work-in-progress), July 2011.

[37] C.-L. Tsao and R. Sivakumar, "On effectively exploiting multiple wireless interfaces in mobile hosts," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, ser. CoNEXT '09, 2009, pp. 337–348.

[38] G. Tsirtsis, G. Giaretta, H. Soliman, and N. Montavont, "Traffic Selectors for Flow Bindings," RFC 6088 (Proposed Standard), Internet Engineering Task Force, Jan. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6088.txt

[39] G. Tsirtsis, H. Soliman, N. Montavont, G. Giaretta, and K. Kuladinithi, "Flow Bindings in Mobile IPv6 and Network Mobility (NEMO) Basic Support," RFC 6089 (Proposed Standard), Internet Engineering Task Force, Jan. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6089.txt

[40] Qualcomm, "3G/Wi-Fi Seamless Offload (whitepaper)," March 2010. [Online]. Available: http://www.qualcomm.com/common/documents/white_papers/3G-WiFi_Seamless_Offload.pdf

[41] N. Imai, M. Isomura, and A. Idoue, "Coordination path control method to reduce traffic load on NGN access gateway," in *13th International Conference on Intelligence in Next Generation Networks, 2009 (ICIN 2009)*, October 2009.

[42] CJ. Bernardos, Ed., "Proxy Mobile IPv6 Extensions to Support Flow Mobility," Internet Engineering Task Force, draft-ietf-netext-pmipv6-flowmob-01.txt (work-in-progress), September 2011.

[43] R. Koodli and K. Chowdury, "Flow Handover for Proxy Mobile IPv6," Internet-Draft (work in progress), draft-koodli-netext-flow-handover-01.txt, Oct. 2009.

[44] M. Hui and H. Deng, "PMIPv6 Multihoming Extension and Synchronization in LMA and MAG," Internet Engineering Task Force, draft-hui-netext-multihoming-01.txt (work-in-progress), March 2010.

[45] M. Hui, G. Chen, and H. Deng, "Service Flow Identifier in Proxy Mobile IPv6," Internet-Draft (work in progress), draft-hui-netext-service-flow-identifier-03.txt, July 2010.

[46] C. Larsson, M. Eriksson, and P. Arvidsson, "Simultaneous Multi-Access and Flow Mobility Support for PMIPv6," Internet-Draft (work in progress), draft-larsson-netext-pmipv6-sma-flow-mobility-00.txt, Mar. 2009.

[47] Y. Han, J. Lee, B. Ahn, and Y. An, "Host Initiation for Flow Mobility in PMIPv6," Internet-Draft (work in progress), draft-han-netext-host-initiation-flow-mobility-00.txt, Dec. 2010.

[48] F. Xia and B. Sarikaya, "Flow Binding in Proxy Mobile IPv6," Internet-Draft (work in progress), draft-xia-netext-flow-binding-02.txt, June 2010.

[49] B. Sarikaya and F. Xia, "PMIPv6 Multihoming Support for Flow mobility,"

Internet-Draft (work in progress), draft-sarikaya-netext-fb-support-extensions-00.txt, Feb. 2010.

[50] T. Tran, Y. Hong, and Y. Han, "Flow mobility support in pmipv6," Internet-Draft (work in progress), draft-trung-netext-flow-mobility-support-01.txt, Oct. 2010.

[51] H.-Y. Choi, S.-G. Min, and Y.-H. Han, "PMIPv6-based Flow Mobility Simulation in NS-3," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*, 30 2011-july 2 2011, pp. 475 –480.

[52] T. M. Trung, Y.-H. Han, H.-Y. Choi, and H. Y. Geun, "A design of network-based flow mobility based on proxy mobile IPv6," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, April 2011, pp. 373 –378.

[53] V. Devarapalli, N. Kant, H. Lim, and C. Vogt, "Multiple Interface Support with Proxy Mobile IPv6," Internet Engineering Task Force, draft-devarapalli-netext-multi-interface-support-00.txt (work-in-progress), March 2009.

[54] D. Thaler, "Multi-Link Subnet Issues," RFC 4903 (Informational), Internet Engineering Task Force, June 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4903.txt

[55] F. Templin, T. Gleeson, and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)," RFC 5214 (Informational), Internet Engineering Task Force, Mar. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5214.txt

[56] T. Melia, F. Giust, R. Manfrin, A. de la Oliva, C. J. Bernardos, and M. Wetterwald, "IEEE 802.21 and Proxy Mobile IPv6: A Network Controlled Mobility Solution," in *Future Network and Mobile Summit 2011 Conference Proceedings*, June 2011.

Acknowledgements

There are several people who helped me in making this thesis possible: these paragraphs are devoted to address them the most grateful thanks.

I would like to start with my supervisor at Alcatel-Lucent Bell Labs Dr. Telemaco Melia for his strong support and daily encouragements. It was a real pleasure to work side by side every day at the office desks and in the cold lab rooms (at 14^{o}C!). I was driven in this project with enthusiasm and energy, and what is left after the 6-months internship is much more than protocols' working schemes! There will not come the time I will stop thanking you! Also, thank you so much for having never talked to me in French!

I am grateful to prof. Carlos J. Bernardos and Dr. Antonio de la Oliva, not only for their helpful discussions, but also for the AP codes and the GNUplot work! They significantly contributed to the results presented in this thesis and to the publication of my first article.

Thanks to my supervisor at Università di Padova, Prof. Michele Zorzi, for having accidentally started what is now my career as researcher and for having so patiently waited for this thesis that eventually has passed through the printer!

Thanks also to all the people at Alcatel-Lucent Bell Labs in Villarceaux, France, who directly or indirectly contributed to the development of this thesis and with whom I shared the Bell Labs offices for 6 months.

Finally, I would not be here without the comprehensive support of my parents, who did not say a word when I told them I would no longer have lived in Italy. Thanks to my French uncle, aunt and cousins, who welcomed me as a son for my stay in the cold (both for the weather and the people) Paris.

A very special thank is due to Chiara, who suffered the challenge of living separated by thousands of kilometers, knowing that we could meet once a month at best. You never dropped your confidence nor the will to keep on week after week. I really love you for what you did!