UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

Distributed Mobility Management for a Flat Architecture in 5G Mobile Networks: Solutions, Analysis and Experimental Validation

Autor: Fabio Giust

Director: Prof. Dr. Carlos Jesús Bernardos Cano

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

Leganés, Madrid, Marzo de 2015

Distributed Mobility Management for a Flat Architecture in 5G Mobile Networks: Solutions, Analysis and Experimental Validation

A dissertation submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Prepared by
Fabio Giust

Under the advice of
Prof. Dr. Carlos Jesús Bernardos Cano

Date: March 2015

Departamento de Ingeniería Telemática, Universidad Carlos III de Madrid

This work has been partially supported by IMDEA Networks Institute.

TESIS DOCTORAL

Distributed Mobility Management for a Flat Architecture in 5G Mobile Networks: Solutions, Analysis and Experimental Validation

Autor: Fabio Giust

Director: Prof. Dr. Carlos Jesús Bernardos Cano

Firma del tribunal calificador:

Firma:

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de

Acknowledgements

This PhD thesis is the result of a four years work with my advisor Carlos Jesús, to whom I address my most profound gratitude for all he could transmit and the gratifications he gave me throughout this period. Then, I would like to thank his colleague Antonio De La Oliva for the extensive collaboration that the three of us had (and hopefully will have in the future) to achieve this result.

I would like also to thank the IMDEA Networks Institute and its directors during these years, Arturo Azcorra and Albert Banchs, for supporting this thesis.

Besides, a special thanks goes to the people at UC3M and at IMDEA with whom I spent most of the time and shared all the Spanish adventures: Isabel and José Pablo for the long term friendship (and co-housing), Camilo, Ignacio, Marco, Andrés, Andra, Luca, Iñaki, Pablo, Gregorio, Goyo, Alberto and all who passed through LAB 4.1C01.

Then, I would like to mention Telemaco for encouraging me to start my PhD, and the *Integration Team* guys of the MEDIEVAL project, Daniele, Gerald, Bessem, Loris, Rui, Philippe, Sérgio and Alberto.

I thank my parents, Aurelia and Renzo, who have always accepted all my decisions without posing any objection.

Un saluto agli amici della *sinistra Piave* che non ho mai smesso di sentire presenti nonostante i molti Km di lontananza, Lupin, Wally, Pichichi, Bubu, Elisa, Eva, Alberto, Della, Riki, Ale, Martyn e il Nonno.

Finally, I would like to thank you, Chiara, for having always supported me with your precious love and trust and, of course, for keeping me *rockin' in a free world!*

Resumen

El despliegue comercial de los servicios de datos en las redes móviles ha evolucionado rápidamente en los últimos años, proporcionando tecnologías de acceso radio más avanzadas y arquitecturas de red más eficientes. Los usuarios ya pueden disfrutar de los servicios de banda ancha desde sus dispositivos móviles, como smartphones y tablets, aprovechando la conectividad de las modernas redes 4G. Sin embargo, la evolución tecnológica sigue trazando su camino hasta el desarrollo de las redes de próxima generación, o 5G, en previsión del enorme aumento del tráfico de los años futuros.

Una de las innovaciones bajo estudio aborda la arquitectura de las redes móviles, con el objetivo de diseñar un sistema plano. Efectivamente, el sistema actual se basa en una estructura centralizada y jerárquica, en la cual múltiples redes de acceso se conectan al núcleo central, dónde residen funciones cruciales para el control de la red y facturación, así como la gestión de la movilidad, que es el tema central de esta tesis. En un sistema con gestión centralizada de la movilidad, se agregan los flujos de tráfico en algunos nodos claves situados en el núcleo de la red, llamados *anclas de movilidad*. De este modo, un ancla puede fácilmente redirigir los flujos al lugar donde se halla el usuario, pero *i)* supone problemas de escabilidad, *ii)* representa un punto único de fallo, y *iii)* el encaminamiento es en general sub-óptimo. Estos problemas se pueden resolver pasando a una arquitectura plana, cambiándose a un sistema de gestión distribuida de la movilidad (Distributed Mobility Management – DMM), donde no hay anclas centralizadas.

Esta tesis se desarrolla dentro el marco propuesto por DMM, presentando el diseño, el análisis, la implementación y la validación experimental de varios protocolos de movilidad distribuida. Se describen soluciones basadas en el cliente y en la red, así como una solución híbrida. El funcionamiento de las soluciones ha sido estudiado analíticamente, para evaluar los costes de señalización, el coste del transporte de los paquetes y la latencia para gestionar el traspaso de los usuarios de una red a otra. Finalmente, la validez de los protocolos ha sido demostrada con experimentos sobre un prototipo donde se implementan algunas de las soluciones utilizando el equipamiento de nuestro laboratorio.

Palabras claves. Gestión de la movilidad centralizada y distribuida, protocolos de movilidad, redes móviles y inalámbricas, redes IP, evaluación analítica y experimental.

Abstract

In the last years, the commercial deployment of data services in mobile networks has been evolving quickly, providing enhanced radio access technologies and more efficient network architectures. Nowadays, mobile users enjoy broadband and ubiquitous wireless access through their portable devices, like smartphones and tablets, exploiting the connectivity offered by the modern 4G network. Nevertheless, the technological evolution keeps moving towards the development of next generation networks, or 5G, aiming at further improving the current system in order to cope with the huge data traffic growth foreseen in the future years.

One of the possible research guidelines aims at innovating the mobile networks architecture by designing a flat system. Indeed, current systems are built upon a centralized and hierarchical structure, where multiple access networks are connected to a central core hosting crucial network functions, e.g., charging, control and maintenance, as well as mobility management, which is the main topic of this thesis. In such a central mobility management system, users' traffic is aggregated at some key nodes in the core, called mobility anchors. Thus, an anchor can easily handle user's mobility by redirecting traffic flows to his/her location, but *i)* it poses scalability issues, *ii)* it represents a single point of failure, and *iii)* the routing path is in general suboptimal. These problems can be overcome moving to a flat architecture, adopting a Distributed Mobility Management (DMM) system, where the centralized anchor is removed.

This thesis develops within the DMM framework, presenting the design, analysis, implementation and experimental validation of several DMM protocols. In this work we describe original protocols for client-based and network-based mobility management, as well as a hybrid solution. We study analytically our solutions to evaluate their signalling cost, the packet delivery cost, and the latency introduced to handle a handover event. Finally, we assess the validity of some of our protocols with experiments run over a network prototype built in our lab implementing such solutions.

Key words. Centralized and Distributed Mobility Management, mobility protocols, mobile and wireless networks, IP networks, analytic and experimental evaluation.

List of Abbreviations

3G, 4G, 5G	3rd, 4th, 5th Generation Mobile Networks
3GPP, 3GPP2	3rd Generation Partnership Project, –2
AA	Agent Advertisement
AAA	Authentication, Authorization and Accounting
AP	Access Point
ARP	Address Resolution Protocol
AS	Agent Solicitation
ATT	Access Technology Type
BA	Binding Acknowledgment
BC	Binding Cache
BCE	Binding Cache Entry
BGP	Border Gateway Protocol
BSC	Base Station Controller
BU	Binding Update
BUL	Binding Update List
BULE	Binding Update List Entry
C-DMM	Client-based DMM
CDN	Content Delivery Network
CGA	Cryptographically Generated Address
CKT	CoA Keygen Token
CM	Connection Manager
CMD	Central Mobility Database
CN	Correspondent Node
CoA	Care-of Address

CoT	Care-of Test
CoTI	Care-of Test Initialization
DAD	Duplicate Address Detection
DASH	Dynamic Adaptive Streaming over HTTP
DL	Downlink
DMM	Distributed Mobility Management
DMM-GW	DMM Gateway
DNS	Domain Name System
DPI	Deep Packet Inspection
DSMIPv6	Dual-Stack Mobile IPv6
EDGE	Enhanced Data rates for GSM Evolution
eNB	evolved NodeB
EPC	Evolved Packet Core
EPS	Evolved Packet System
eUTRAN	evolved UMTS Terrestrial Radio Access Network
FA	Foreign Agent
FHRP	Flow Handover Reply
FHRQ	Flow Handover Request
FM	Flow Manager
GERAN	GSM EDGE Radio Access Newtork
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GTP	GPRS Tunneling Protocol
HA	Home Agent
H-DMM	Hybrid DMM
HeNB	Home eNodeB
HI	Handoff Indicator
HMIPv6	Hierarchical Mobile IPv6
HNP	Home Network Prefix
HO	Handover or Handoff

HoA	Home Address
HSPA	High Speed Packet Access
HTTP	HyperText Transfer Protocol
IANA	The Internet Assigned Numbers Authority
IEEE	The Institute of Electrical and Electronics Engineers
IETF	The Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP, IPv4, IPv6	Internet Protocol, IP version 4, IP version 6
ISP	Internet Service Provider
IPsec	Internet Protocol Security
IXP	Internet eXchange Point
LAN	Local Area Network
LMA	Local Mobility Anchor
LMD	Localized Mobility Domain
LANP	Locally Anchored Network Prefix
LTE	Long Term Evolution
MAC	Medium Access Control
MAG	Mobile Access Gateway
mCDN	mobile CDN
MH	Mobility Header
MIH	Media Independent Handover
MIHS	Media Independent Handover Services
MIIS	Media Independent Information Service
MIPv4	Mobile IPv4
MIPv6	Mobile IPv6
MME	Mobility Management Entity
MN	Mobile Node
MN-ID	Mobile Node Identifier
MNO	Mobile Network Operator
NAT	Network Address Translation
ND	Neighbor Discovery

N-DMM	Network-based DMM
NetLMM	Network-based Localized Mobility Management
NFV	Network Functions Virtualization
ODMM	Open platform for DMM solutions
PBA	Proxy Binding Acknowledgment
PBU	Proxy Binding Update
P-CoA	Proxy Care-of Address
PDN	Packet Data Network
PDP	Packet Data Protocol
PGW	PDN Gateway
PHKT	Permanent HoA Keygen Token
PMIPv6	Proxy Mobile IPv6
PoA	Point of Access
PoC	Proof of Concept
PoS	Point of Service
PPP	Point-to-Point Protocol
QoE	Quality of Experience
QoS	Quality of Service
RA	Router Advertisement
RADIUS	Remote Authentication Dial In User Service
RAN	Radio Access Network
RD	Router Discovery
RFC	Request For Comments
RNC	Radio Network Controller
RO	Route Optimization
RR	Return Routability
RS	Router Solicitation
RTT	Round Trip Time
SAE	System Architecture Evolution
SDN	Software Defined Networking

SGSN	Serving GPRS Support Node
SGW	Serving Gateway
SLAAC	StateLess Address Auto-Configuration
TCP	Transport Control Protocol
UE	User Equipment
UDP	User Datagram Protocol
UL	Uplink
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
VoIP	Voice over IP
VoLTE	Voice over LTE
WCDMA	Wideband Code Division Multiple Access
WiFi	IEEE 802.11 WLAN
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless LAN

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The 21st century has brought a radical change to Internet users in the way they live their Internet experience. We are not concerned about the great source of information, entertainment, interactivity and productivity that the Internet has become in these years, but, instead, we regard at how these services originally consumed by users sitting in front of their computers are now available from portable mobile devices, granting the so-called *anywhere, anytime* connectivity.

This revolution has been brought of course by the great advances in the wireless communications, and, to a broader extent, by mobile networks in general. More than a decade has indeed passed since the commercialization of the first WiFi-branded products, leading to a massive penetration of wireless devices for residential and enterprise environments, and the leap to broadband wireless data access brought by the mobile telephony network was about to let fly. In fact, voice services dominated the Mobile Network Operators (MNOs) offer, and so the voice traffic in mobile networks, until the fixed-access usage changed to wireless access usage and, in turn, to the ultimate mobile access.

MNOs began then to pay more and more attention to the development of their mobile networks, providing their infrastructure with enhanced packet switched architectures to support the increasing demand of data services, as for instance the evolution brought by the 3rd generation of mobile networks, the Universal Mobile Telecommunications Systems (UMTS) and the high data rates achieved in the radio link with High Speed Packet Access (HSPA) technologies.

Although the rapid changes witnessed so far, nowadays worldwide consumers are getting familiar with yet another revolution: the 4G system is indeed a further enhancement to mobile networks guided by the high data rates delivered by the Long Term Evolution (LTE) in the Radio Access Network (RAN) and by the Evolved Packet Core (EPC) architecture designed for the packet core network.

The combined action of broadband wireless access, the increasing number of relatively cheap mobile devices and cloud services is speeding up the mobile traffic growth with a surprising Compound Annual Growth Rate (CAGR) of 61% estimated for the period 2013-2018. This means three times faster than the fixed IP traffic growth over the same period! The source is the latest Cisco Visual Networking Index, published in June 2014 [1]. According to the same report, video media content will be the major driver of the mobile traffic growth: with a CAGR of 70%, it will account for over 50% of total mobile data traffic by the end of 2018.

Therefore, mobile networks are expected to support a huge amount of data as never seen before, but current architectures present some drawbacks for which their are not suitable to cope with it. Indeed, a general architecture for mobile networks is structured in a hierarchical and centralized way, where multiple RANs are connected to the central core network. The core network hosts key entities, e.g., for charging, network control and maintenance, as well as the IP gateway, placed at Internet exchange points to connect the mobile network to the rest of IP-based networks. Therefore, in the data plane, the users' data packets must traverse both the RAN and the core to be routed to and from the Internet through the IP gateway. Such gateway is also the mobility anchor, that is, the node that maintains the reachability of the IP addresses allocated to the mobile nodes (MNs) after their movements. Being the anchor deployed in the core network, it has a privileged vantage point to redirect traffic to the access network where the MN currently is. In this way, the data flows destined to the MNs are always intercepted by the anchor that in turn re-routes them to the terminal's location. Nevertheless, in this approach, the anchor

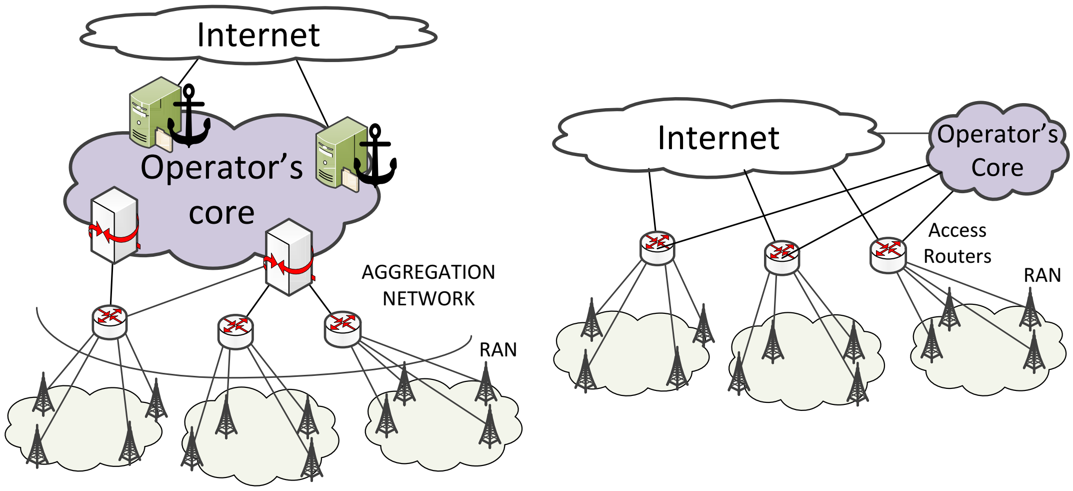
i) represents a single point of failure,

ii) poses scalability issues (i.e., it handles traffic for millions of users),

iii) leads to sub-optimal paths between the mobile nodes and their communications peers,

iv) handles mobility support with a coarse granularity (i.e., on an MN basis), even for motionless terminals or for applications that do not need mobility support.

Mobile operators are already working on solutions, both on the access side and in the core, capable of tackling the tremendous traffic growth foreseen for the future years. The enhancement of the current mobile architecture is currently gaining momentum with a plethora of solution being proposed in the research community, spanning from the coverage expansion with extremely dense and heterogeneous wireless access networks, to traffic offload techniques to closer breakout points connected to the Internet. Besides, in their future deployments, operators are not willing to exclude the flexibility and programmability features offered by Network Functions Virtualization (NFV) tools and the Software Defined Networking (SDN) paradigm. Moreover, Content Delivery Networks (CDNs) are foreseen to transport over than 50% of IP traffic by the end of 2018. For this reason, CDN-like solutions targeting specifically the mobile networks (sometimes called mobile CDN – mCDN) are expected to be key supporters tackling the avalanche of traffic.



(a) Hierarchical architecture

(b) Flat architecture

Figure 1.1: Hierarchical vs. flat architectures.

The purpose of this thesis is to introduce the concept of flat architecture, inspired by the Distributed Mobility Management (DMM) paradigm, as a valuable framework to design next generation mobile networks. The aim of DMM is to realize an architecture for mobile networks without any centralized entity in charge of routing traffic and handling the mobility of a large amount of users (10-20 millions subscribers is the typical population for the main European MNOs). Thence the flat network concept: the hierarchy inherent to the data forwarding present in current architectures is broken, letting IP flows to be routed in a more flexible way.

A typical DMM scenario envisions mobility-enabled access routers being the main actors among the data plane and control plane entities. These routers are located at the edge of the operator's network, close to the users, and they are provided with links to external IP networks (local breakouts) that do not traverse the core, as well as the functionality needed to handle users' mobility.

Fig. 1.1 illustrates a general hierarchical architecture (left picture) and a flat one (right picture). One can immediately observe the distributed nature of a flat system: instead of aggregating traffic at the core gateway(s), IP flows are split among the access routers and their links. By doing so, the potential congestion in core links and nodes is reduced, and better-suited data paths can be established for the IP flows by using standard routing algorithms.

It is worth observing that the operator's core network is not necessarily removed in a flat design: the core may still host vital control services, e.g., related to the subscribers' profiles, as well as traffic gateways for an hybrid centralized/distributed system. The key concept is that the edge network can simultaneously access both the Internet and the core without any intermediate network entities beyond plain routers and/or switches.

The contributions brought by the present thesis are multiple, including the design of original DMM protocols for a flat architecture based on IPv6, their analytic evaluation and the experimental validation of some of them. This work is organized in three parts:

Part I: *Current solutions and technologies for mobile networks.* It provides the background on the state of the art technologies for mobile networks, to understand where the DMM paradigm comes into play. Chapter 2 showcases the main efforts produced by standardization bodies to design current mobility management protocols and architectures. In particular, we focus on mobility support for IPv6 networks, as specified by the Internet Engineering Task Force (IETF) mobility protocols, namely Mobile IPv6 (MIPv6) and Proxy Mobile IPv6 (PMIPv6). Then, we explore the architectures for cellular networks defined by the 3rd Generation Partnership Project (3GPP) and the Media Independent Handover Services (MIHS) standard by the Institute of Electrical and Electronics Engineers (IEEE). At the end of the chapter we identify the major reasons that led to the introduction of the DMM paradigm. Chapter 3 includes a review of the DMM related work available in the scientific literature.

Part II: *Distributed Mobility Management solutions.* It develops the thesis core. Chapter 4 and Chapter 5 present several DMM solutions, respectively within the client-based and network-based categories, whereas Chapter 6 proposes a hybrid solution combining a client-based and a network-based DMM protocol. The client-based approach was first presented at a conference [2] and next as an Internet Draft to the IETF community [3]. Similarly, the network-based solutions first appeared in [4] and next collected in the Internet Draft [5]. The hybrid approach has been introduced in [6] and [7]. The performance of our DMM protocols have been studied analytically in Chapter 7 and experimentally in Chapter 8. Chapter 7 follows the analytical methodology that we developed in [8] and extends it to all the DMM protocols described in this thesis. Chapter 8 collects the experimental results from tests conducted over a prototype platform that we implemented to validate our DMM protocols. The platform and the results were originally published in [8]. In addition, the platform has been exhibited in public demonstrations, at an international conference, [9], and at two IETF meetings (83rd and 87th IETF). Moreover, the results obtained from the prototype have been compared to those from the experiments on two additional testbeds that we built to implement an SDN-based and a routing-based DMM solution. This outcome is published in [10], and an extension paper with more detailed results is currently under submission [11]. The last part of Chapter 8 details a real DMM use case scenario where a CDN system has been integrated within a DMM mobile network. This scenario has been used as a fundamental part of the EU project MEDIEVAL[1], validated analytically in [12] and part of a work accepted for publication [13].

Part III: *Conclusions and future work.* It is the part dedicated to conclude the work, highlighting the main outcome of the thesis and identifying the next research steps in the DMM domain.

At the end of the thesis, two appendices include additional elements related with the topic presented in the thesis body. Appendix A contains the message format for the mobility signaling defined for MIPv6, PMIPv6 and the network-based DMM solutions. Appendix B presents a design and implementation study case combining PMIPv6 with MIHS. Both chapters are useful to understand the technical solutions proposed in the thesis and to provide details for those interested in the implementation.

To the author's best knowledge, the prototypes presented in Chapter 8 represent the only DMM implementations available today. Such resources have been collected in our ODMM project, Open Platform for Distributed Mobility Management solutions, and made available as open source code in the ODMM website, `http://www.odmm.net/`.

[1] `http://www.ict-medieval.eu/`

Part I

Current solutions and technologies for mobile networks

Chapter 2

Background on mobility protocols and mobile networks

In these last years, we have witnessed the explosion of a “mobile revolution”, mainly driven by the massive market penetration of powerful mobile handsets, and the deployment of faster heterogeneous radio access technologies.

In this chapter, we provide the current state of the art mobile network architectures, leaving aside the RAN part to focus primarily on the evolution of the packet core networks. In addition, since our work is pretty much oriented to mobility management, we limit our studies to control plane and data plane solutions within this field, dedicating little or no attention to other areas that are not less important but out of the scope of this thesis, like billing, charging policies, etc.

We start describing the mobility problem in IP-based networks, reporting the IP mobility protocols standardized by the IETF. Then, we move to 3GPP cellular networks, with a brief overview on the UMTS and EPS architectures, and on mobility management within the EPS. Finally, we briefly introduce networking with the Media Independent Handover protocol suite specified in the IEEE 802.21 standard.

2.1 Mobility in IP networks

When it comes to data networks, the de-facto standard nowadays is represented by the TCP/IP stack, with the network layer being dominated almost exclusively by the **Internet Protocol (IP)**, in both its variants IPv4 and IPv6.

Tackling mobility in IP-based networks is therefore a crucial step in order to build the infrastructure that provides connectivity to mobile users. Indeed, when a terminal changes point of access, it might produce as a consequence the non-reachability of the IP address configured by the host, because it becomes topologically incorrect in the new access network. It is obvious that the packets carrying that IP address as destination 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, or if the routing infrastructure changed the rules to re-establish the data path towards the new location. Unfortunately, both conditions are hard to achieve, because the sender might not be aware of the new recipient's address, or because it is not possible and/or desirable to change the routing entries at the intermediate routers in the data path.

With this simple description we have just introduced the limitation imposed by 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 when connecting to a new network, 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 considerable inconvenience to the quality of service 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 radio access, IEEE 802.11, etc.). Moreover, from the IP layer point of view, such a solution can be applied only when the terminal is roaming within the same technology.
- **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 [14]). 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.

In the remainder of this section we describe the solutions standardized by the IETF for mobility at the IP layer, stressing the two mainstream approaches investigated so far, i.e., the client-based set of solutions (we focus on Mobile IPv6) and the network-based approach (with Proxy Mobile IPv6). These protocols represent the starting point used in this thesis to develop the extensions and enhancements proposed following the Distributed Mobility Management paradigm.

2.1.1 Mobile IPv6

In the late '90s, the IETF community developed solutions to address the mobility problem in IPv4 networks, resulting in the first IP mobility protocol standardized in 2002 with the name of "IP Mobility support for IPv4" simplified in **Mobile IPv4** or **MIPv4** (RFC 3344, [15]). This solution is rather a theoretic exercise, as it has not seen much real commercial deployment so far, but, still, it is a very important milestone, as it has been taken as starting point for the extensions and changes that came later with other protocols.

A step forward in the design of an efficient mobility protocol was achieved with the solution for IPv6 networks, after releasing RFC 3775 "Mobility Support in IPv6" [16] (or simply **Mobile IPv6 - MIPv6**), then updated by RFC 6275 [17]. Mobile IPv6 inherits most of the mechanisms introduced by its IPv4 predecessor, but, also, it brings several enhancements due to the wise exploitation of IPv6 features[1].

The basic principle of MIPv6 is that a moving terminal configures a permanent globally reachable IPv6 address when it is in its *home network*, and a temporary one when it is away, connected to a *foreign* (or *visited*) *network*. In the home network resides a special node in charge of mapping the MN permanent IPv6 address with the temporary one. The protocol's fundamental concepts are already there: next we detail the entities involved and the operations. Note that most of the terms introduced in the following paragraphs are common to other mobility protocols, and thus are used throughout the work with the same meaning, except when stated otherwise.

Here is the list of nodes comprised in the MIPv6 architecture.

MN - Mobile Node. 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[2]. The MN configures two types of addresses:

[1] MIPv4 is not covered in this thesis for the sake of consistency with the DMM solutions presented, that deal IPv6. However, the list of differences between the two mobility protocols for IPv4 and IPv6 can be found in RFC 6275.

[2] For the sake of completeness, in the documents and specifications produced by the 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 UMTS network stack, while, in the IETF, the term Mobile Node is intended to be technology agnostic, referring in particular to the IP layer.

- **HoA - Home Address**. 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.
- **CoA - Care-of Address**. It is the temporary IP address configured by the MN to maintain connectivity when in a foreign network. The MN acquires a new CoA at each visited network.

HA - Home Agent. This node is in charge of storing an association between the HoA and CoA (a *binding*) for each MN, and to reroute the packets for the MN when it is not at home. Indeed, when the MN is at its Home Network, the HA is not necessarily involved in the packets delivery to the MN. Otherwise, the MN registers a CoA at the HA when it is away, and next the HA starts intercepting the packets destined to the HoA, and it encapsulates them in a tunnel with an additional IP header, containing as destination the CoA. This procedure, necessary to properly route the packet to the intended recipient, is known also as *IP tunneling* [18].

CN - Correspondent Node. It is the other endpoint of a communication with the MN. It can be a fixed or moving node. It is involved in the mobility signaling when the MN attempts a Route Optimization (RO) procedure.

The operations related to bind the MN's HoA to the MN's CoA are grouped by a procedure called *registration* or *binding*. The registration procedure 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 close to expiration, and/or *iv)* de-register when they return home or wish to end the mobility session. A registration message is typically exchanged between the MN and the HA, but also between the MN and the CN for route optimization.

The messages defined for this procedure are the *Binding Update (BU)*, sent by the MN to the HA to register the new location at the CoA, and the *Binding Acknowledgement (BA)*, sent back by the HA to confirm or reject the registration. The BU and BA messages are forged in a modular way by adding the byte format of the BU (or of the BA) to a *mobility header*. Then, a set of *mobility options* is appended. The resulting message is then encapsulated in an IPv6 packet and sent over the appropriate link layer. This modularity makes it possible to re-use the mobility header and options also for other IPv6 mobility protocols, rather than defining new messages. The message formats are detailed in Appendix A at the end of the thesis, in Section A.1.

The registration process is depicted in Fig. 2.1, and it consists on 4 steps:

1. The MN, attaches to a foreign network, and configures a CoA valid in the new access network. This can be realized through standard Neighbor Discovery (ND) [19] operations and *Stateless Address Auto-Configuration (SLAAC)* [20].

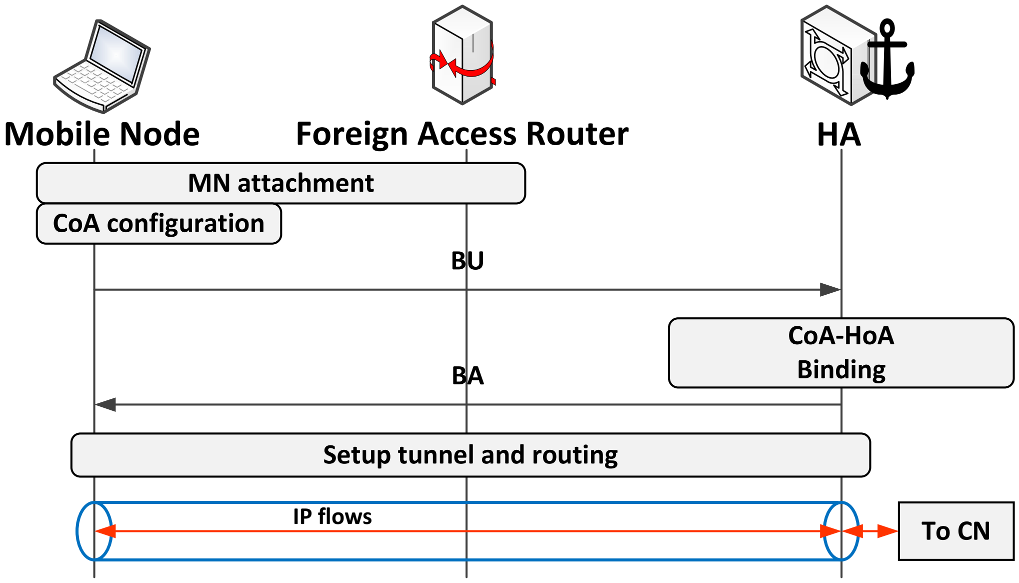


Figure 2.1: Registration in Mobile IPv6.

2. When the MN detects that the CoA differs from the HoA (or from a previous CoA), it sends a BU message to the HA to notify the location change. The message contains the HoA and the CoA, and, additionally, 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 in a data structure called *Binding Cache Entry (BCE)*.
3. The HA sends back a BA message to the MN's CoA, with the affirmative response and the lifetime. Moreover, it sets an IPv6-in-IPv6 tunnel configuring as endpoints its own address on the link towards the MN and the MN's CoA.
4. Upon receiving the BA message, the MN establishes a tunnel with the HA in the reverse direction, so that the tunnel created between them is bidirectional.

The bidirectional IPv6-in-IPv6 tunnel created between the HA and the MN's CoA establishes the data path for the MN enabling communications with any CN using the HoA as MN's IPv6 address (see the illustration in Fig. 2.2). Therefore, in downstream, a packet destined to the MN's HoA first arrives at the MN's home network, due to standard routing infrastructure, where it is intercepted by the HA (step 1). Next, this latter encapsulates the packets in a tunnel with another IPv6 header, filling the source address with its own, and the destination address with the CoA (step 2). The packet are then received by the MN which decapsulates and delivers them to the upper layers. In uplink, packets follow the reverse direction through the tunnel (step 3). 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 (RO)* is the procedure devised to register the MN's

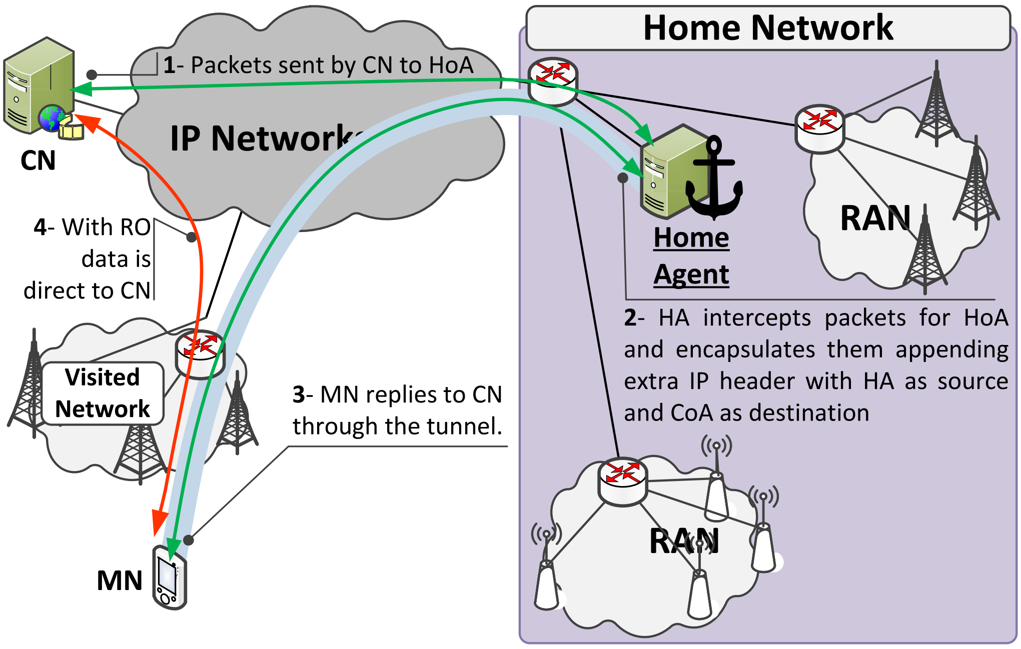

Home Network
1- Packets sent by CN to HoA
IP Network
CN
Home Agent
RAN
4- With RO data is direct to CN
2- HA intercepts packets for HoA and encapsulates them appending extra IP header with HA as source and CoA as destination
Visited Network
3- MN replies to CN through the tunnel.
RAN
MN

Figure 2.2: Data forwarding in MIPv6.

CoA at the CN as well, so that the communication can be migrated (i.e., packets are forwarded) through a shorter path to destination (see step 4 of Fig. 2.2). 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 2.1.2).

Mobile IPv6 enables global reachability and session continuity, as the mobility tunnel can be established from any MN location, as long as there is a routing infrastructure able to connect the MN's CoA to the HA's global IPv6 address. *Global mobility* is thus an opposite concept from the *localized mobility* that is the scope of the solution discussed next, Proxy Mobile IPv6.

However, before moving to the next protocol, it is worth spending few words on security. Indeed, security mechanisms are crucial in a mobility protocol design. 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 *redirection attack*. A bogus node registers its 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 attracting their packets to its terminal. The opposite scenario is when the misbehaving node registers its HoA associated to a third party's CoA and starts several IP sessions with the purpose of flooding the victim's links and/or draining its resources.

Therefore, the security mechanisms adopted in MIPv6 leverage the IPsec suite [21, 22] in Encrypted Security Payload mode [23, 24] to secure the MIPv6 signaling. In this way, the MN and HA authenticate each other and the BU and BA messages are protected against external manipulation attempts.

However, an additional security mechanism is necessary for the route optimization procedure. Indeed, RO is vulnerable to redirection attacks and IP spoofing, because, with this procedure, a CN basically stores a binding between the MN's HoA and CoA as an HA does, but while the HA–MN interface is secured by IPsec, the CN–MN connection may be 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, most of the details are skipped in order to highlight the elegance of the basic idea (a comprehensive study is carried out in [17, 25]). The MN starts the procedure sending a request through both paths to the CN, i.e., through the direct one and 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 (the 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.

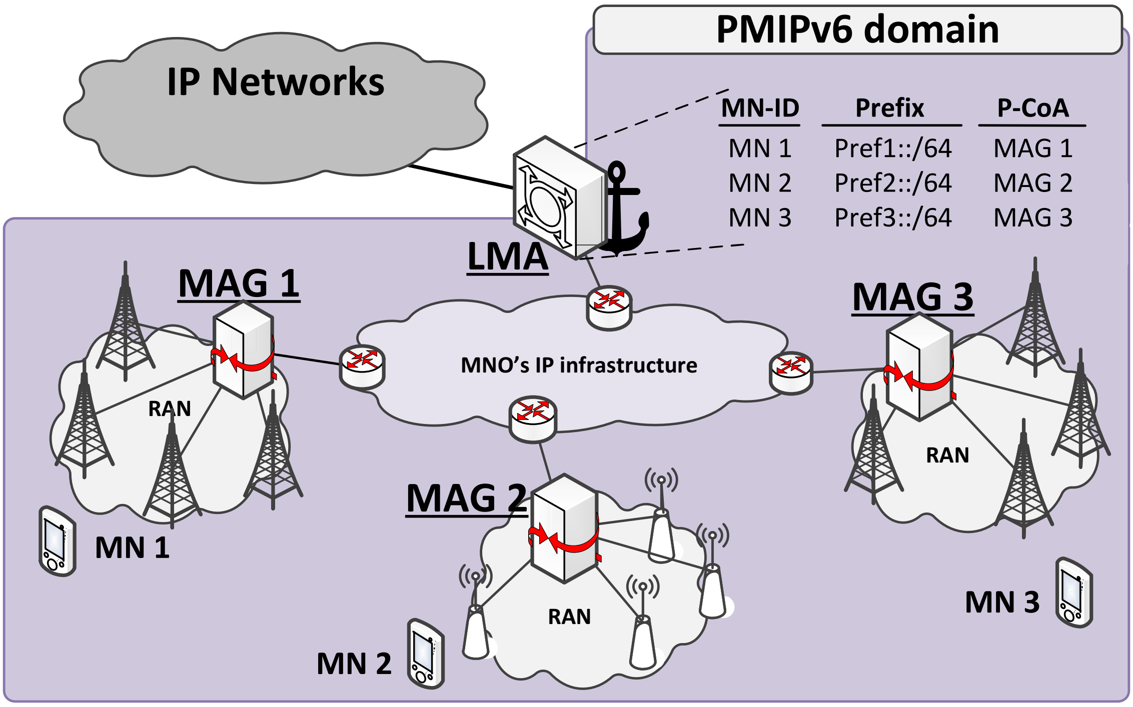
2.1.2 Proxy Mobile IPv6

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** [26].

The motivation behind this new research area comes with bringing the mobility management closer to the MN, in order to reduce the latency associated to the binding signaling and the drawbacks of the triangukar routing. To achieve so, the scope of the mobility support is reduced as well, to a limited portion of the network called *Localized Mobility Domain (LMD).* Thence, in this case, the MN benefits from mobility support only when it moves within the LMD, whereas MIPv6 is intended for global mobility.

The improvement achieved with PMIPv6 is provisioning the mobility service within such an LMD without involving the MNs. Hence the terminology *network-based* and *localized* mobility management. MIPv6 requires the MN to implement the mobility module as an IPv6 stack add-on, while PMIPv6 does not; moreover, it is clear that relieving the terminal from mobility operations represents saving the over-the-air signaling, and, therefore, battery energy.

The network based scheme is achieved by relocating relevant mobility management



MN-ID	Prefix	P-CoA
MN 1	Pref1::/64	MAG 1
MN 2	Pref2::/64	MAG 2
MN 3	Pref3::/64	MAG 3

Figure 2.3: Proxy Mobile IPv6 domain

functions from the MN to the network. In a network-based LMD, the network learns through standard terminal operation, such as ND [19] or by means of link layer support [27], 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 [28]. 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.

The core functional entities in the PMIPv6 infrastructure are (see Fig. 2.3):

MN - **Mobile Node**. It is the moving host. Differently from MIPv6, it is a legacy IPv6 stack host.

MAG - **Mobile Access Gateway**. 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 movements on the access network. There are multiple MAGs in an LMD.

LMA - **Local Mobility Anchor**. 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). A route points to the MAG managing the link where the MN is currently located. Data packets are routed to and from the MN through a tunnel between the LMA and the corresponding MAG. The LMA is also responsible for assigning IPv6 prefixes to the 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.

The sequence of operations in PMIPv6 is quite similar to that drawn in MIPv6, except for the relocation of the mobility client from the MN to 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 Router Solicitation (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)* message, to associate its own address with the MN's identity (e.g., an ID related with the MN's authentication in the network). Upon receiving this request, the LMA assigns a prefix to the MN – called MN Home Network Prefix (MN-HNP). The LMA creates a Binding Cache Entry (BCE), which main fields are the MN-ID, the MN-HNP and the MAG's IP address visible from the LMA (the Proxy-CoA – 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 Acknowledgement (PBA)* message, including the MN-HNP. 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 Router Advertisement (RA) message to the MN specifying the prefix to be used for the IP connectivity. The MN is now able to configure one or more addresses from the assigned MN-HNP and the registration procedure is over (see Fig. 2.4). The routing state created to forward messages to/from the MN comprises a set of routing entries at the LMA and MAG: in downlink, one entry at the LMA indicates that packets destined to the MN-HNP must be forwarded through the tunnel established with the serving MAG; a corresponding route at the MAG indicates that the MN-HNP is on one of its access link. In uplink, a route at the MAG forces packets containing the MN-HNP as source to 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 location in the LMA by means of a PBU/PBA handshake, and advertises through a unicast RA the same MN-HNP 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.

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

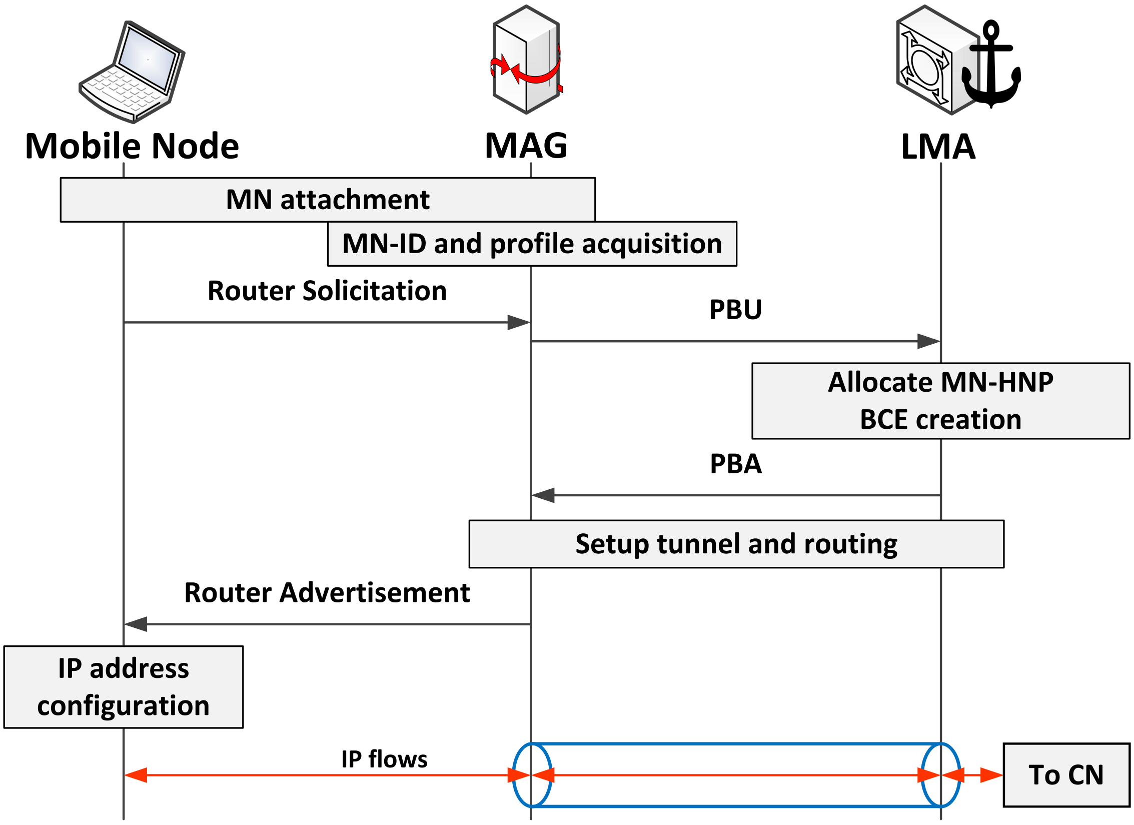


Figure 2.4: Registration to a Proxy Mobile IPv6 domain

two procedures converge, 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 an MN's attachment to the network, either an authentication mechanism is deployed on the access link, or the MAG performs an AAA check querying a dedicated infrastructure. RFC 5213 recommends the use of *RADIUS* [29] or *Diameter* [30] 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 the last paragraph of Section 2.1.1).

2.2 3GPP architectures for mobile networks

Mobile networks for telephony started being deployed and available as a commercial service much before the mobility problem was laid down in IP data networks. During the '90s, the GSM cellular networks spread all around the world producing a profound change in the society. The next step was bringing data services along with voice, and the GSM architecture evolved to accommodate the Global Packet Radio Service (GPRS).

The Information Technology era was about to make a big step forward, but still there were big news coming.

Afterwards, when the 3$^{\text{rd}}$ generation cellular network specifications were defined, both data and voice services were encompassed at the same time by the designers, leading to the UMTS architecture. Both the radio and the network part were improved, granting higher rates to data services and an overall enhanced quality perceived by the user. Nevertheless, although packet switching was a native technology in the new architecture (rather than a patch, as for the addition of GPRS to the GSM system), in UMTS there are two separated domains for voice and data service, namely the *circuit switched core* and the *packet switched core*. This separation, originally motivated by the need to ensure backward compatibility with previous generation devices and network entities, soon became inefficient for MNOs, for its expensive deployment and maintenance costs. Further improvements and enhancements were added with the introduction of the IP Multimedia Subsystem (IMS), with the objective of creating a common platform for multimedia services available to both mobile and fixed access. IMS enables the UMTS packet switched domain to transport voice calls, enabling Voice over IP in mobile networks. In this architecture, the UMTS packet domain is used for the data plane (i.e., forwarding the "voice" packets), and the IMS is used for the control plane (i.e., establishing and releasing the voice call with the correspondent node). That is, IMS does not replace the packet domain, but it is rather an add-on for multimedia services.

Fig. 2.5 depicts the GSM and UMTS simplified architectures: the orange lines represent the connections the circuit switched domain (i.e., for voice services), whereas the blues ones stand for the packet-switched domain; the IMS is not included to keep the diagram simple, but it would be connected to the packet switched domain. Each segment bears the communication interface name. In the top part of the diagram, we have the Base Station Controller (BSC) in the GSM radio access network (GERAN) connected to the Mobile Switching Centre (MSC) for voice services and to the Serving GPRS support node (SGSN) for data services. Similarly, the Radio Network Controller (RNC) in the UMTS terrestrial radio access network (UTRAN) is connected to the same nodes, but using different interfaces. The SGSN and the Gateway GPRS Support Node (GGSN) were originally designed as part of the GPRS architecture. From the picture, it is clear how the UMTS architecture re-uses the previous network entities, rather than introducing new ones.

The overview given above is in no way to be considered exhaustive of the whole architecture, thence many details are skipped on purpose. However, if we examine a bit the GPRS architecture, we first observe that the cellular mobile network architecture is centralized and hierarchical. Indeed, the packet switched infrastructure forwards packets through a pair of gateways before being routed to destination, and these entities are generally connected in a pyramidal manner, with the GGSN on top. In particular, the

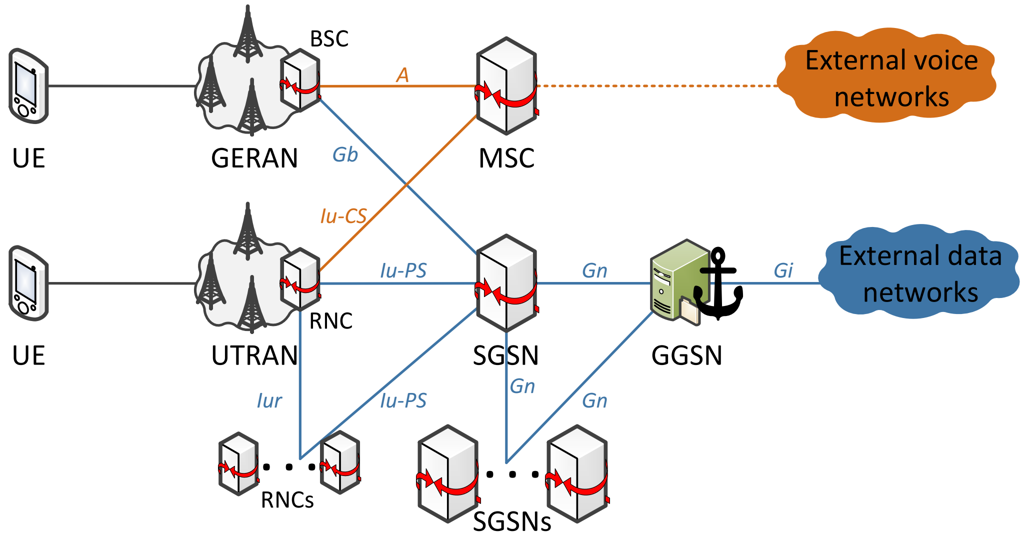


Figure 2.5: GPRS simplified architecture for UMTS and GSM.

GGSN acts as connection point between the MNO's network and the rest of external data networks, while the SGSN is a sort of "door" from several RAN entities to the core.

There is thus a division in the network data path into two transport segments: from the RNC/BSC to the SGSN and from the SGSN to the GGSN. This latter segment, specified by the *Gn* interface, is realized through the GPRS Tunneling Protocol (GTP) [31]. GTP creates transport tunnels for user traffic on top of an IP infrastructure. More in general, GTP is used in the MNO's backbone between any pair of GSNs, and the *Gn* interface is common to both the GSM and UMTS architectures. On the contrary, the SGSN connects to the BSC and to the RNC via two different interfaces, respectively the *Gb* interface, and the *Iu-PS* interface. The former relies on the creation of permanent virtual circuits between the SGSN and the RNCs. The latter is based on GTP as well, since the UMTS specifications devise the use of IP also in the backhaul infrastructure between the RAN and the core. In any case, we can generalize these concepts stating that there are two transport segments clearly defined and separated in the data path. This separation allows (among other additional features) to handle mobility by simply switching the tunnels or virtual circuits used to forward user traffic. For instance, in the UMTS case, when a terminal moves under another RNC's coverage area, then the SGSN-RNC GTP tunnel is switched accordingly, and when the UE connects in an area controlled by another SGSN, besides the SGSN-RNC tunnel, also the GGSN-SGSN tunnel is moved[3]. The GTP is split into a control plane protocol, GTP-C, that is in charge to signal how tunnels are switched, and a data plane protocol GTP-U, that defines how to encapsulate the user

[3] Mobility in cellular networks is a wide topic, and reporting the techniques for all the mobile architectures is outside the scope of this work. Thus, we limit our focus only to mobility management in the Evolved Packet System, see next Section 2.2.1. However, we remark that the 3GPP systems share the general principles.

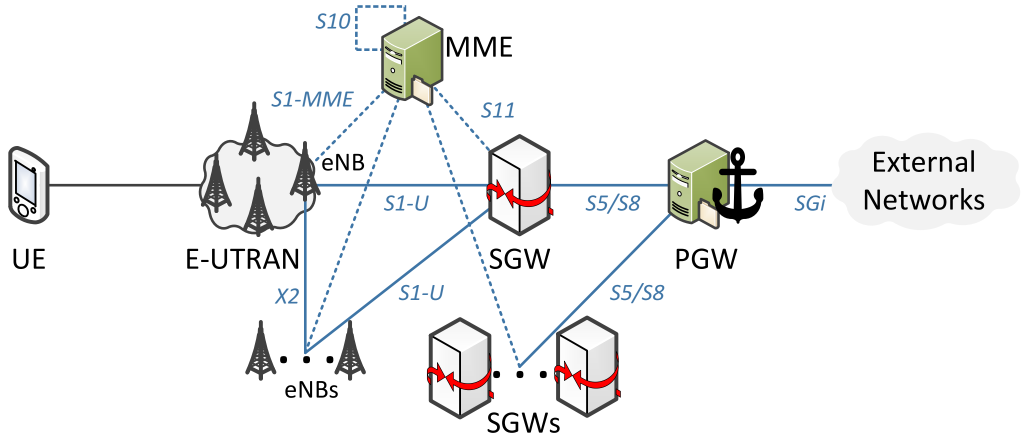


Figure 2.6: EPS simplified architecture.

packets. The RNC, SGSN and GGSN are directly involved in both the control (GTP-C) and user (GTP-U) planes. Specifically, the SGSN is responsible for paging a terminal and handling the handoff procedures when the mobile equipment disconnects from one Base Station and attaches to another. With the High Speed Packet Access (HSPA) technology, the two tunnels can be optionally merged into one, from the GGSN to RNC, in the so-called “direct tunnel” mode. In this scenario, the SGSN is by-passed by the data plane, but still is involved in the signalling with GTP-C.

2.2.1 The Evolved Packet System

The full transition to a packet switched domain only for both voice and data is achieved with the current evolution of the mobile architecture, the Evolved Packet System (EPS). The EPS, defined in the 3GPP's Release 8, devises the Long Term Evolution (LTE) radio access interface and an all-IP core network, called Evolved Packet Core (EPC). EPC is a fundamental cornerstone of the mobile broadband revolution; with it, the cellular mobile architecture definitely migrated to a full packet switched infrastructure based on IP. The EPS and its improvements, specified in 3GPP's Release 8, Release 9, Release 10 and Release 11, represent the current state of art technology for mobile network architectures. The simplified EPS architecture for LTE only access is shown in Fig. 2.6, with the most relevant elements of the EPC[4]. The EPC is the first effort carried out to provide a common packet core architecture for a plethora of access technologies. Apart from the obvious support for 3GPP radio access, like WCDMA (used in UMTS) and LTE technologies, it supports non-3GPP radio access too, like, among others, the 3GPP2 CDMA, the IEEE 802.16 (WiMAX) and IEEE 802.11 (WiFi).

In the EPS architecture (we consider the simplified subsystem as in Fig. 2.6), the

[4] In the EPS, voice services are provided through the use of IMS – not shown in Fig. 2.6 – and they are usually referred to as the Voice over LTE (VoLTE) system. Nevertheless, the EPS specifies a fallback mechanism to the old circuit switched domain for voice, to help MNOs in the transition to a full EPS.

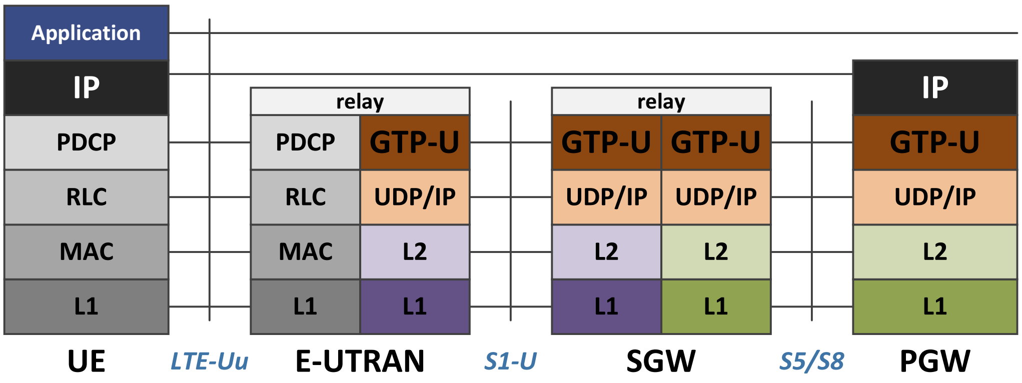


Figure 2.7: EPS stack model, GTP option.

Packet Data Network Gateway (PGW) acts as gateway between the operator's network and external IP networks, generally referred as Packet Data Networks (PDN) in the 3GPP terminology for the EPS. The PGW also aggregates the traffic from several edge networks, conveyed by intermediate nodes called Serving Gateways (SGW). We observe that this is not different from what the GGSN/SGSN pair does in UMTS.

Also, by taking a look at the EPS communication stack model in Fig. 2.7, we notice that the user traffic is conveyed by means of tunneling from an EUTRAN element (i.e., an eNodeB – eNB) to the SGW, and also in the SGW–PGW segment. GTP is still the encapsulation method designed to transmit the IP packets within the EPC[5]. This chain of tunnels creates the transport infrastructure for the so-called *PDN connection*.

A PDN connection is an IP point-to-point link between the User Equipment (UE) and a PGW, that enables the UE to access the target PDN through the PGW. Therefore, a separate PDN connection is set up for each PDN that the UE can access, and each PDN connection is bound to a specific PGW according to the target PDN[6]. The UE is allocated one and only one IPv4 (or IPv6) prefix per PDN connection by the PGW (e.g., dual stack hosts require separate PDN connections for IPv4 and IPv6). By assigning a whole network prefix to the UE, it is ensured the point-to-point nature of the IP link between the UE and the PGW, as no one else can be on the same subnet and therefore it is not necessary to perform the mapping between Layer-2 and Layer-3 identifiers[7].

All the interfaces' specifications changed from UMTS to EPS in order to take into account the all-IP nature of the transport infrastructure. The RNC entity is incorporated in the Base Transceiver Station forming the eNB, and the SGSN is split into a control plane entity, the Mobility Management Entity (MME), and a data plane entity, the SGW. The

[5] Proxy Mobile IPv6 [26] can be used as alternative to GTP for the interface between the PGW and the SGW.

[6] For instance, two typical PDNs deployed by an MNO are for "Internet" and "Multimedia Messaging Service (MME)".

[7] For instance, mobile terminals are typically assigned a /30 IPv4 prefix, thus leaving room only for two addresses (one for the terminal itself and the other for the PGW).

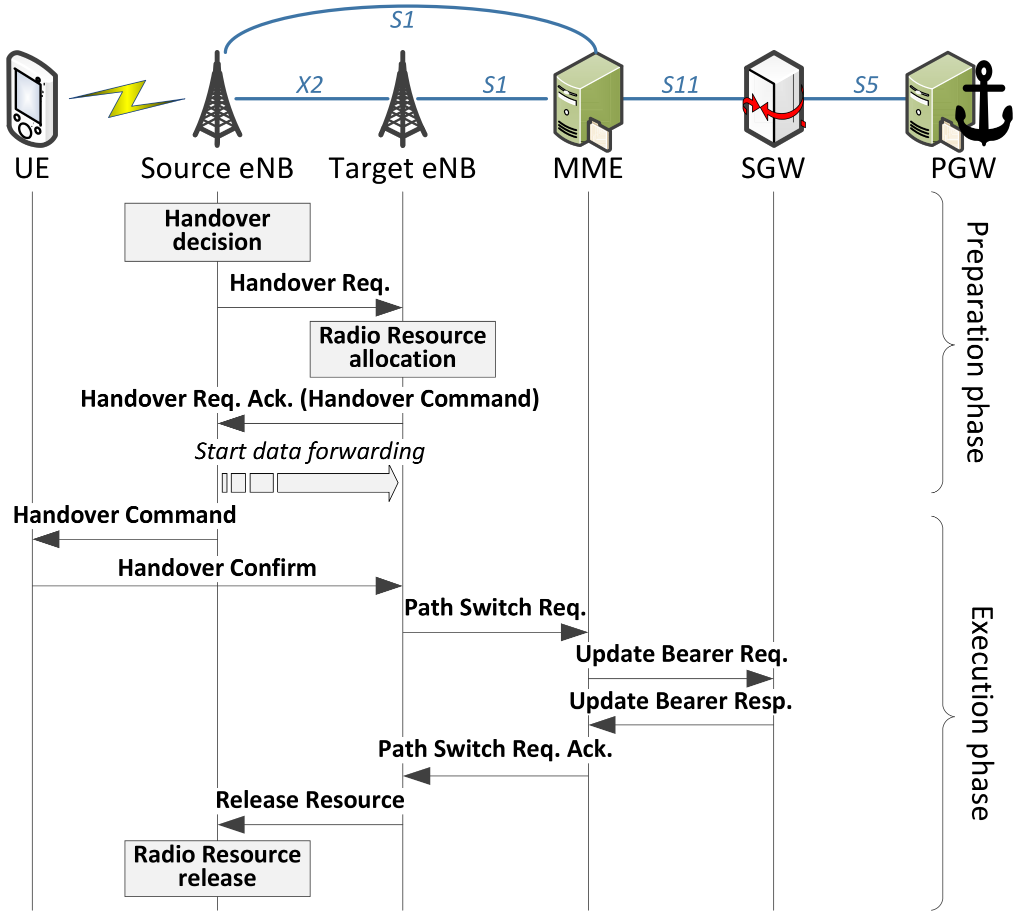


Figure 2.8: Handover in *active* mode, with *X2* support and no EPC relocation.

MME has been introduced as the main actor in the GTP-C's mobility procedures when a terminal moves from one eNB to another, therefore taking over this functionality from the SGSN. The MME thus coordinates the SGW–eNB tunnel switch after a handover.

Mobility principles in the EPS. As mentioned earlier, the EPC attempts to provide a common packet core to a variety of heterogeneous access technologies, therefore the handover specifications cover a wide set of operations to support different mobility mechanisms and mobility *between* different technologies. It is worth noting that the EPS is the first realization of multi-access convergence, providing full mobility support, access network discovery and selection for any type of access network. Nevertheless, in this section we reduce the scope of our analysis to the intra-technology (LTE) mobility case, focusing on the mobility management for the terminal only when in *active* state.

Conversely to the *idle* state location update, when in *active* state the terminal is engaged in a communication, thus a cell transition must be notified timely to the network to guarantee service continuity[8].

[8] When in active state, the network knows exactly the location of the UE, i.e., the eNB where the UE

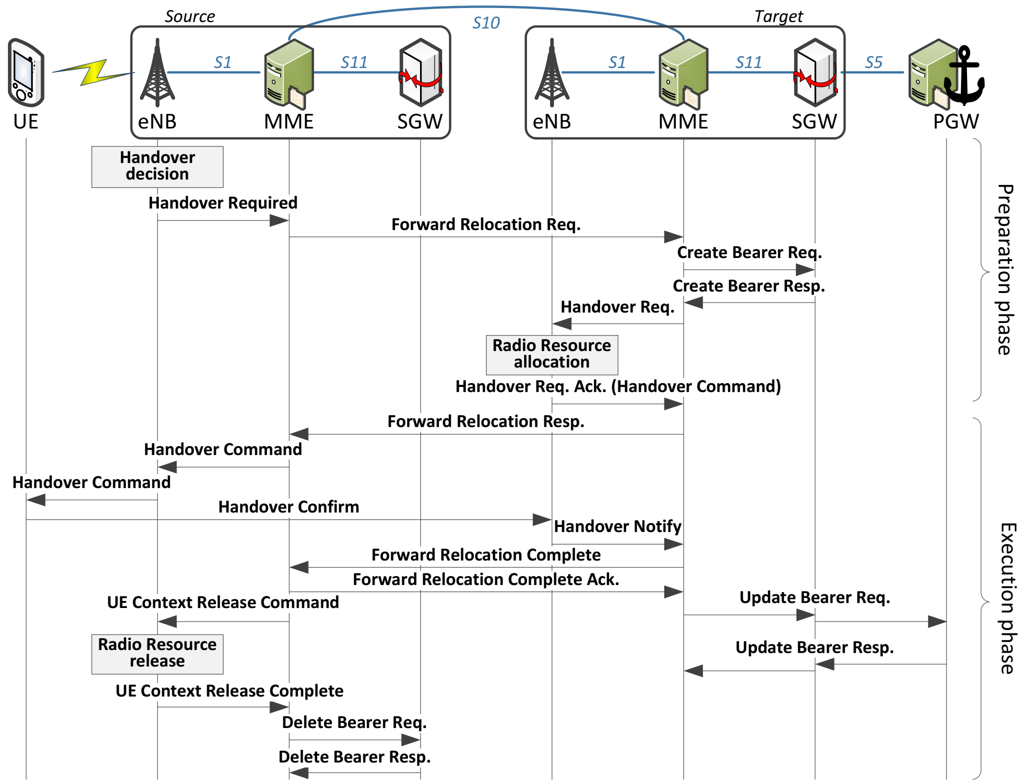


Figure 2.9: Handover in *active* mode, with EPC relocation.

Depending to the entity of the cell transition, the target eNB might be still connected to the same MME and SGW as the source eNB (handover with no EPC relocation) or to a new pair (handover with EPC relocation). Besides, the handover without EPC relocation might be supported or not by the *X2* interface between the source and target eNB. Fig. 2.8 depicts the procedure for the handover with no EPC relocation and with *X2* support. In this scenario, the source and target eNB can directly exchange handover related information to prepare the radio resources and to forward packets to the target eNB to minimize packet loss. Then the handover is executed by sending the command to the UE and by updating the data path from the SGW to the target eNB. The handover is concluded when the radio resources at the old eNB are released. It is worth noting the role of the MME in the procedure: it receives the notification from the target eNB and next it commands the SGW to switch the tunnel (called *bearer*) to point to the target eNB. Simply put, a bearer maps a PDN connection for an UE to a transport tunnel. Without *X2* support, the procedure is similar, but the signaling between the eNBs must pass through the MME using dedicated messages.

In case the handover implies a relocation of the MME, or the SGW, or both, then the

is attached. On the contrary, in idle state, the network knows the UE location in a broader area defined as a set of eNBs. A comprehensive description of the *paging* and handover procedures when the terminal is in idle state and for inter-technology handoff can be found in [32] and in [33].

signaling load is larger. We detail for instance the case of the MME/SGW pair relocation, being it the most general case, but the other scenario share the same principles[9]. The procedure is illustrated in Fig. 2.9. During the preparation phase, the target MME coordinates the resources preparation in the target eNB and also the bearer creation in the SGW. When the target entities are ready the handover is executed. After establishing the new link, the new MME instructs the old one to release the radio resources and bearers in the origin entities, and it triggers a bearer update to the SGW, which in turn updates the information at the PGW. When this procedure is over, both the EPC tunnel (from the PGW to the SGW) and the access tunnel (from the SGW to the eNB) are aligned to the new path for the UE.

User traffic anchoring. From a mobility perspective, the user's data traffic is anchored at the SGW as long as the terminal is moving within the SGW's management area, but, to a wider extent, the user's flows are in any case always anchored at the PGW. This is due to the point to point nature of a PDN connection: the UE "sees" always the same PGW as the next hop towards a given PDN. With this solution the network load lays completely upon the PGW: although this is a convenient solution to easily handle mobility, it brings several limitations, as we describe in more detail at the end of this chapter.

2.3 IEEE 802.21 Media Independent Handover Services

Modern mobile handsets, laptops and similar devices are equipped with different wireless interfaces, usually 3G/4G, WiFi and Bluetooth, and it is likely to have a larger coexistence of heterogeneous access technology in the future, mainly from the IEEE 802, 3GPP and 3GPP2 families. In this context, the Media Independent Handover Services (MIHS), defined in the IEEE 802.21 standard [27, 34], enable the optimization of handover between heterogeneous IEEE 802 networks and facilitate handover between IEEE 802 networks and cellular networks.

IEEE 802.21 is a framework that provides service continuity while an MN switches between heterogeneous link layer technologies by defining a set of operations to prepare, execute and conclude the handover in an assisted and controlled manner. Nevertheless, MIHS is not involved in any Layer-3 or higher mobility management procedure, for which it relies on the presence of a dedicated mobility management protocol within the network elements' stack. The MIHS framework consists of the following elements:

- A set of handover-enabling functions within the protocol stacks of the network elements and a new entity created therein called the MIH Function (MIHF).

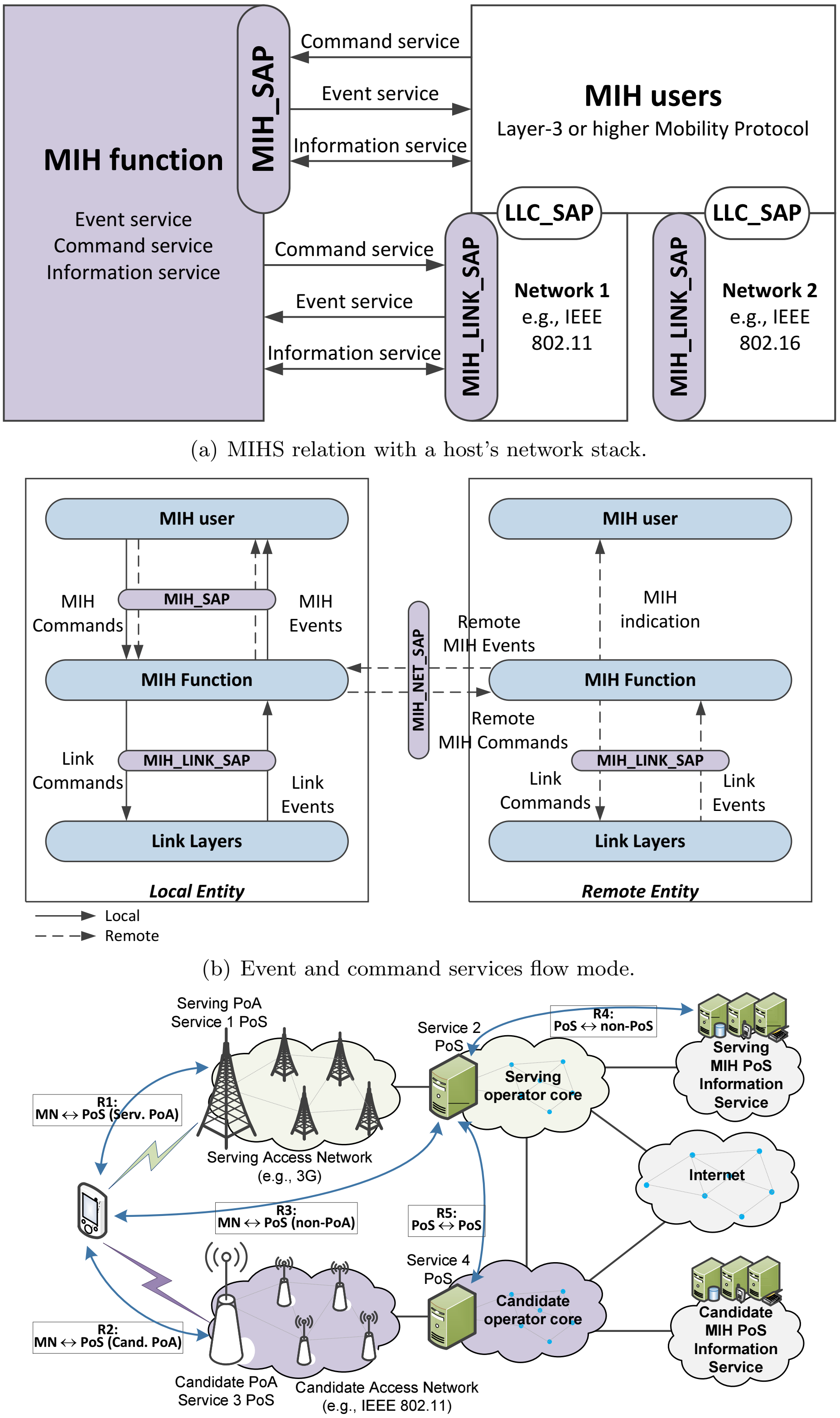
[9] Depending on the operator deployment choice, a UE might hand off to a eNB sharing the same MME as the previous but not the SGW, or the same SGW but not MME, or none of them.

- A media independent handover service access point (called the MIH_SAP) and associated primitives to provide MIH users with access to the services of the MIHF. The MIHF provides the following services:
  - The *media independent event* service (MIES) that detects changes in link layer properties and initiates appropriate events (triggers) from both local and remote interfaces.
  - The *media independent command* service (MICS) provides a set of commands for the MIH users to control link properties that are relevant to handover and switch between links if required.
  - The *media independent information* service (MIIS) provides the information about different networks and their services thus enabling more effective handover decision to be made across heterogeneous networks.
- The definition of new link layer service access points (MIH_LINK_SAPs) and associated primitives for each link layer technology. The new primitives help the MIHF collect link information and control link behavior during handovers.

Fig. 2.10(a) shows the MIHS relation with a multiple interfaced host's network stack. MIHS provide link layer hooks (MIH_LINK_SAPs) through which a MIHF communicates with the network interfaces. In addition, MIH_SAPs permit to a MIHF to interact with Layer-3 or higher mobility protocols. In addition to the elements represented in the stack model of Fig. 2.10(a), MIHS define another interface, the MIH_NET_SAP, used between MIHFs residing in different network hosts. Such interface enables remote communication between peer MIH users, or between a MIH user and a link layer, in different network elements. Fig. 2.10(b) illustrates the communication flows for the event and command services, both locally and remotely.

Now that we have introduced the MIHS communication and service flows, let's examine how network entities interact following the MIH procedures. The scenario of Fig. 2.10(c) illustrates the MIHS network and communications reference model, that includes the following entities:

- MIHF on the MN.
- MIH Point of Attachment (MIH PoA or PoA). It is the endpoint of an L2 link that includes the MN as the other endpoint.
- MIH Point of Service (MIH-PoS or PoS). It is the network side MIHF that has MIH communication with the MIHF on the MN. Note that an MN can exchange MIH messages with multiple network entities:
  - PoS on the network entity that includes the serving PoA of the MN;
  - PoS on the network entity that includes a candidate PoA for the MN;
  - PoS on a network entity that does not include a PoA for the MN (PoS non-PoA).



(a) MIHS relation with a host's network stack.

(b) Event and command services flow mode.

(c) MIHS communication and network reference model.

Figure 2.10: MIHS: stack model, services and reference model.

- MIH non-PoS. It is a network entity that has no MIH communications with the MN. Note that a given node might be a PoS for an MN and a non-PoS for another.

The network entities in the reference model interact through the following communication *reference points*:

- Reference point R1 – MN ↔ PoS (Serving PoA). It groups the communications between the MIHF on the MN and the PoS on the network entity that includes the serving PoA.
- Reference point R2 – MN ↔ PoS (Candidate PoA). It refers to the procedures between the MIHF on the MN and the PoS on the network entity that includes a candidate PoA.
- Reference point R3 – MN ↔ PoS non-PoA. It is used for the communications between an MN and a PoS on a non-PoA network node.
- Reference point R4 – PoS ↔ non-PoS. It is related to interactions between a PoS and a non-PoS.
- Reference point R5 – PoS ↔ PoS. It refers to the procedures between two different PoSs located ad different network entities.

In Appendix B we describe how MIH services behave when applied to a real use case scenario. We develop, as an example, the combination of IEEE 802.21 in a PMIPv6-operated network, presenting the design and implementation as reported in [35].

2.4 Final remarks. The need of Distributed Mobility Management

The recent advances brought by the Evolved Packet System let foresee a trend to design flatter systems for mobile architectures. For instance, the eNodeB, beyond being the Base Transceiver Station, implements the functionalities that were formerly executed by the Base Station Controller (GERAN) or the Radio Network Controller (UTRAN). Fig. 2.5 and Fig. 2.6 clearly show the the transition from the old base station to a smarter entity. In addition, eNBs are now interconnected through a dedicated interface (*X2*, see Section 2.2.1) used to forward traffic betweeen them during a handover. Nevertheless, such flat system is still limited in the EPS by the centralized traffic anchoring, due to the tunnels chain that carries data traffic within the mobile infrastructure.

Indeed, Centralized Mobility Management (CMM) is conceptually simple to design, because it relies on a mobility-enabled IP gateway, that is always traversed by users' data flows as long as the terminals roam within a large part of the MNO's network (if not the whole). Then tunnel-based packet forwarding enables easy traffic redirection by switching the tunnel used to serve the MN, and the operation requires little signaling. Also, it meets a legacy design principle in 3GPP networks that envisions the connection-oriented communication between the terminal and the network. This fact is reflected in the EPS by the PDN connection permanently established between the UE and the PGW: by using tunneling, the PDN connection is decoupled from the transport infrastructure. Finally, at deployment and maintenance level, CMM benefits from a long experience since it is a mature technology, widely used in mobile networks since their birth.

Nevertheless, in CMM, packet tunneling/encapsulation and mobility contexts are set up even for static nodes and for short-lived communications (like HTTP or SMTP), resulting in unnecessary processing overhead upon the network entities to grant mobility support in a scenario where it is not required. A more efficient mobility management technique should dynamically provide mobility support when needed, and handle with plain routing the rest of traffic demands. Besides, aggregating traffic from a huge number of users at the anchor and its transport links poses scalability and bottleneck issues that can be addressed only with expensive dimensioning and redundancy engineering. Also, this scenario clearly suffers from the single point of failure problem.

Another important problem inherent to CMM is due to sub-optimal routing. Data packets are conveyed to the network core, even when the user is closer to the correspondent node than to the anchor. This is true for instance in the voice calls case, which majority is originated and terminated at local level [36], but it can be applied also to data services, given the increasing spread of Content Delivery Networks, and other IP-based services available at the user location. Reducing the data path length potentially yields to lower communication latency and higher throughput.

These concerns have been originally raised by Bertin, Bonjour and Bonnin in [37], after the publication, in 2008, of a couple of proposals for dynamic and distributed anchoring, [38] and [39]. Next, the IETF community brought this discussion within the MEXT and NETEXT working groups (those related to extensions to mobility protocols, respectively for the client-based and network-based families), and some members published a sort of DMM *manifesto* in the article [40] that served next as basis for the DMM working group chartering, in March 2012[10]. At the time of writing, the WG has published the DMM problem statement and requirements document in RFC 7333 [41], together with RFC 7429 [42], that identifies the current practices for DMM and the gaps with the DMM requirements. No standard solutions are available yet.

With the above paragraphs, we have claimed that Distributed Mobility Management plays a key role in the design of a flat system for mobile architectures, which motivation is supported by the CMM weak points summarized in the following list:

i) sub-optimal routing,

ii) scalability,

iii) single point of failure,

iv) lack of finer granularity for the mobility support.

[10] `http://datatracker.ietf.org/wg/dmm/documents/`

In the next chapter, we provide an overview of existing DMM proposals presently available in the scientific literature, whereas Part II is devoted to the presentation and analysis of our DMM solutions.

Chapter 3

Related work on Distributed Mobility Management

This chapter surveys Distributed Mobility Management proposals available in the scientific literature. We collected IETF Drafts, 3GPP documents and scientific articles that either explicitly address the DMM problem, or can be related indirectly to the topic. Therefore, since the solutions span in a wide plethora of possibilities, we grouped the proposals based on their shared features, and we identified four different categories, extending the classification proposed in [40]:

1. *architecture-dependent approaches*, such as the different efforts initiated in the 3GPP to offload and/or anchor some traffic flows closer to the user [43];
2. *extension approaches*, proposing extensions and/or modifications to existing IETF protocols;
3. *peer-to-peer (P2P) approaches*, distributing the mobility management functionality across a P2P network;
4. *clean-slate approaches*, proposing novel network architectures tackling the root of the problem, as opposed to the evolutionary one.

The categories are ordered so that each increments the level of modifications that must be applied in order to deploy the solution. We start with solutions that can be applied to the legacy Evolved Packet System and those that require the deployment of modified versions of IETF protocols, then we move to the P2P distribution of the mobility management functions to finally arrive at proposals with brand new network systems in mind.

A key distinguishing feature of a mobility protocol is the main entity in charge of performing the operations on the mobile side. This is the basis for the classical division in *client-* and *network-based* solutions, which also holds for DMM. In addition, DMM network-based solutions can be further classified according to the level of distribution of the control plane [41, 42]:

- *Partially distributed solutions*, which are characterized by distributing the data path among several anchors deployed closer to the end user, but still keeping the control plane centralized.
- *Fully distributed solutions*, which completely distribute both the data and control planes (there is no centralized control entity).

At the end of the survey, Table 3.1 lists the proposals described in this chapter, highlighting their main characteristics and classifying them following the proposed taxonomy.

3.1 Architecture dependent solutions

Regarding the *architecture dependent* category, the most relevant work is being performed within the 3GPP. The solutions currently under study are specifically tailored for the EPS, aiming at reducing the volume of user data traffic traversing operators' core networks by providing enhanced mechanisms for local breakouts and offloading. Rather than being actual DMM proposals, these solutions offer complementary mechanisms to avoid binding the IP connections to the PGW, and they are already available as technical reports in the latest releases of the 3GPP system architecture specifications. Such mechanisms are the Local IP Access (LIPA) [44], the Selected IP Traffic Offload (SIPTO) [45], the LIPA Mobility and SIPTO at the Local Network (LIMONET) [46] and the co-ordinated PGW Change for SIPTO (CSIPTO) [47].

LIPA enables direct access to IP resources co-located with a residential/enterprise access (e.g., femtocells). In this way, the flows generated for local services (e.g., home media servers, intranet resources, etc.) from the residential/enterprise access points (called Home eNB - HeNB) are not routed to the EPC, but, instead to a Local Gateway (LGW), either co-located with HeNB itself or placed close by.

SIPTO enables the dynamic deviation of selected traffic to an alternative data path that does not traverse the EPC. The deviation point can be placed at the RAN level or above, and in all cases it consists in relocating the original SGW/PGW pair to a gateway entity that better fits for offload purposes. SIPTO is available also from residential/enterprise access, taking as deviation point the LGW.

LIMONET provides basic mobility support for LIPA and SIPTO when moving among femtocells connected to the same LGW.

In the LIPA/SIPTO mechanism, either a NAT box performs the address translation that is required to switch the data paths, or the MN is assigned an IP address bound to another PDN connection that refers to the LGW or another gateway entity. The simultaneous use of several IP addresses is a key concept common to many DMM proposals, and we take it into consideration in more detail in next chapters when discussing our DMM proposals.

CSIPTO enables the simultaneous use of two PDN connections to the same PDN through different PGWs (e.g., one PGW at the local network and the other in the core). With CSIPTO a terminal can bind IP flows from applications that can survive an IP address change to a local PGW (if available) that ensures a more efficient data path but cannot provide IP address preservation when the terminals move outside the gateway scope. The IP flows from applications that require IP address continuity are bound to the PGW in the the core.

The deployment of the above mechanisms and the design of new ones are fostered by the increasing employment of tools for Network Functions Virtualization (NFV), that allow to create virtual instances of individual network entities, or group of elements that concur to a specific function, or even to virtualize the whole EPC (virtual EPC - vEPC) [48]. Indeed, by employing NFV, operators may find cheap solutions to deploy multiple SGW/PGW pairs closer to the edge of the network, so as to exploit the benefits from LIPA, SIPTO, LIMONET and CSIPTO.

Apart from the outcome within the 3GPP, Hahn *et al.* propose in [49] and [50] two complementary solutions for PGW relocation within the 3GPP Release 10 specification. The main idea proposed by both works is the definition of new mechanisms for application aware non-optimal path detection, so that the UE's PGW can be dynamically changed.

The work in [51] explores the deployment of client and network based DMM solutions in the EPS architecture, providing a detailed description of the required operations and the re-use of the architectural elements and interfaces.

A similar concept is developed in [52]. The basic idea is to deploy distributed PGWs located at the edge of the network and to leverage the MME entity to run the key control functions. This approach builds a flat network as the traffic is deviated to the edge PGWs that act also as mobility anchors. The paper discusses also the necessary interfaces that need to be added to complement the legacy EPS with the proposed architecture.

3.2 Extension to existing protocols

There are several benefits inherent to extending already established protocols to support DMM, such as an easier backwards compatibility and simpler design. Since the two mainstream IETF protocols for mobility support are MIPv6 and PMIPv6, most of the following propose upgrades to these.

DMM solutions focusing on Mobile IPv6 try to reduce the impact of the triangular routing on the overall performance. In [53], the ADA (Asymmetric Double Agents) extension to Mobile IP is presented to optimize handover latency and communication delays. These improvements come at the cost of introducing two new entities in the network, the local mobile proxy (LMP), that takes care of the functionality of the home agent, but is located closer to the mobile node; and the correspondent mobile proxy (CMP), which is located near the correspondent node to provide an optimized route towards the LMP.

A different approach for reducing the HA–MN delay is taken in [54]. This work proposes a solution that enables the use of multiple home agents distributed through the Internet, interconnected by high speed links and reached through anycast routing. Hence these nodes can be placed near the mobile node, in order to reduce the limitations of centralized deployments.

The design in [55] proposes to deploy a home agent in each access router of the network, and to dynamically switch home agent as the MN moves from one access network to another. In this way, old IP flows are anchored by the visited home agents, while the new flows are routed by the current home agent without encapsulation. The signaling to maintain the mobility sessions is derived from MIPv6. This design reflects the principles of our work previously published in [2], but the two differ for minor aspects related to security.

Many works on DMM network-based mobility support come from Proxy Mobile IPv6. In [56], Chan proposes to deploy several PMIPv6-like domains forming a large mobility super-domain, and to enable inter-domain mobility by re-routing packet from one mobility anchor to another. The first domain where the MN attaches is the home network, whereas next become the visited networks. When the MN is at home, typical PMIPv6 data and control planes are used. When the MN moves, the home mobility anchor intercepts the packets destined to the MN and tunnels them to the mobility anchor where the MN currently is. In parallel the home mobility anchor instructs the ingress node where packet for the MN are coming from to forwards such packets with a tunnel to the current visited mobility anchor. The proposed solution still maintains hierarchy levels of traffic gateways, hence a real flat architecture is not achieved.

The previous design is enriched with signaling details and a performance analysis based on simulations in [57]. Nevertheless, the signaling mechanism is based uniquely on the case in which the CN is connected to another mobility anchor of the same super-domain. More results from the same design are available in the article [58], written by the same authors. The same design is exploited to provide network mobility (NEMO) support in [59].

In [60], PMIPv6 route optimization is proposed. In this solution, the MAGs serving the MN and CN leverage on the information stored at the LMA to establish a direct tunnel between them, so a better path can be used for the communication. This mechanism still makes use of a tunnel for the whole duration of the data session. The solution is only applicable to the case in which the CN is attached to the same PMIPv6 domain as the mobile node.

Another route optimization technique for PMIPv6 is discussed in [61]. The proposed protocol either needs the CN to be connected to the PMIPv6 domain or to be able to interpret some modified PMIPv6 signaling messages.

The proposal described in [62] suggests to split the functionality of the localized mobility anchor (LMA) of PMIPv6 into two distinct nodes: a control plane LMA (CLMA) and a data plane LMA (DLMA). The former maintains the mobility sessions for the MNs, whereas the second is the anchor for the MNs' traffic. The CLMA also assigns the most suitable DLMA to the MNs. This proposal relieves the LMA's burden, but, in general, does not fit for flat architectures, as the DLMA/MAG hierarchy is preserved, along with the tunnels, which are established for the whole duration of a data session. Besides, it is envisioned an operating mode by which, if the MN and CN are under the same CLMA's administration, route optimization can be set up between the corresponding MAGs.

Three mobility schemes are proposed in [63]: signal-driven PMIPv6, data-driven distributed PMIPv6 and signal-driven distributed PMIPv6 which explore respectively one partially and two fully distributed solutions. The three mechanisms rely on control/data planes split for the partially distributed solution, multicast and peer-to-peer communication for the remaining fully distributed ones.

In the article [64], the authors present an extension for Proxy Mobile IPv6 that enables the local mobility anchor to select an intermediate anchor (IA) to handle a mobile node's IP flows. The anchoring function is in charge of establishing tunnels with the old and the new MAGs, as the MN moves from one to another.

The Dynamic Mobility Anchoring (DMA) in [38] is a generic solution in which mobility management is offered on a per IP flow basis. The design encompasses two roles for an access node: it is a visited access node (VAN) when the MN is connected to it, and it is an anchor access node (AAN) when it is in charge of anchoring MN's IP flows after the MN has moved to a different VAN. The anchoring is based on redirecting the MN's IP flows to the MN's VAN by means of an IP tunnel. The paper does not discuss a mobility signaling among the access nodes. On the contrary, a VAN learns the corresponding AAN through packet inspection of uplink traffic: It the packet belongs to an old IP flow then it carries the IP address assigned by the AAN, and thus is tunneled to AAN. Packets of new IP flows carry the IP address assigned by the VAN and they are forwarded using standard routing. In parallel, an AAN learns the current VAN when receiving encapsulated traffic. In case of no uplink traffic, the mobile node is required to send void packets to timely recover connectivity with an AAN. The side effect of this approach is the introduction of unnecessarily latencies at handover execution. This proposal is evaluated in [65] through simulations, and in [66] through analytical models but no validation via implementation is documented.

The same design is extended in [67] to support a prefix relocation mechanism, capable of relocating the old prefixes used by the mobile node to prefixes allocated by the serving access router. This requires mobile node modifications to indicate to the network the best moment to perform the relocation.

An evolution of DMA comes by the same authors in [68], intending to embrace PMIPv6 as the basis to develop a mobility signaling for DMA. The modified DMA solution relies on mobility capable access routers (called MARs) that exchange PBU and PBA messages to update the MN location and IP addresses. Before the signaling, MARs interact with a central database to retrieve the mobility sessions and coordinate the routing state for the MNs. This approach reflects with slight modifications the partially distributed design that we proposed in [5]: the main difference is on the active role than we envision for the central database (somehow similar to the coordination tasks of the MME in the EPS), whereas in [68] the database is a passive store for the mobility contexts. An analytic evaluation of such protocol is provided in [69, 70]. Besides, the same authors published in [71] a joint analysis of DMA with their client-based DMM solution [55].

The authors of [62] analyze in [72] the performance of a general DMM approach where the data plane is the same as DMA. The most interesting part of the analysis is the estimation on how many IP flows need tunnel encapsulation out of the total. The traffic model is based on flow durations measured during 24 hours at a campus. Results tell that for a moving user in 500 meters radius cells at 5.6 Km/h, less than 15% of flows are tunneled. When moving at 10 Km/h, then the ratio rises up to 25% of flows, and at 56 Km/h up to 65% of flows requires tunneling.

Last, the following two articles envision a DMM-like scheme for the network mobility support. In [73], many distributed home agents (DHA) are deployed to facilitate a mobile router to establish an optimized path with the correspondent node. The coordination of the home agents is achieved through the home agents location registration agents (HALRA), which are responsible also for assigning an HA to the mobile router.

Conversely, the authors of [74] propose to use a PMIPv6-based DMM solution to provide mobility support to a moving network. This solution inherits many features from [68].

3.3 Peer-to-peer solutions

One of the key aspects of the DMM concept is the distribution of the mobility management functionality across multiple entities. *Peer-to-Peer* (P2P) paradigms naturally apply to this kind of distributed interaction.

In [75], the authors present m-Chord, a protocol used to distribute the home agent and foreign agent functionalities of Mobile IPv4 using the Chord P2P technology [76]. Their performance analysis concludes that only in some cases their solution performs faster than standard Mobile IP, although in general there is a performance drawback from the use of the P2P technology.

Similar to the previous work, also [77] presents a solution based on Chord. Here, MNs and CNs are enabled with a MIPv6 module and interact through the P2P overlay. During handover, the MN sends a BU to inform all the CNs about the new mobility parameters. The authors argue that one of the main drawbacks of using P2P overlays for mobility management is the lack of coherence between the overlay and the actual physical topology of the nodes. Hence they propose to extend the P2P protocol to consider physical information in order to optimize the update and query performance.

[39] is another mobility management protocol based on distributed hash tables (DHT), called Distributed IP Mobility Approach (DIMA). Again, the home agent functionality is split and spread across different nodes that share a common binding distributed database. The data traffic towards the mobile node is intercepted by one of these nodes, which anchors the mobile node's home address. Differently from MIPv6, the MN does not take active part in handling location updates, as the set of home agents are in charge of transmitting the Binding Update and Acknowledgement signaling.

Finally, the work [78] also describes a DMM solution that leverages a DHT storing the MNs' ID/location pairs. Nevertheless this can be accounted as a client-based solution, because the entities located at the edge of the network are responsible to handle the MN's mobility context coordinated by the messages exchanged with the MN itself, which employs a dedicated hand-off module. The session continuity during handover is granted by the bi-casting mechanism.

3.4 Clean slate proposals

In this section we present those proposals that break with the traditional approach to mobility management. A quite representative *clean slate* DMM solution can be found in [79, 80], where the authors propose the use of routing updates between routers to manage the mobility of the nodes within the domain. It relies on Domain Name System (DNS) updates and lookups to detect the prefix assigned to the node and Border Gateway Protocol (BGP) signaling to update the internal routing within the domain. Global roaming is also supported by issuing BGP route updates between several Autonomous Systems (ASs). Although the proposal has been discussed within the IETF, a deep performance analysis is still missing. Regarding the client- network-based classification criterion, this solution can be considered as fully distributed network-based, as the routing is updated based on BGP signaling exchanged between the routers. Note, however, that the mobile node has the responsibility to update the IP address in the DNS record if it happens that the MN changes the IP address after a handover.

In [81], authors propose an architecture called Access Independent Mobile Service (AIMS) to improve scalability of the network management service. The proposal is a network-based mobility management protocol where the date plane and control plane are decoupled. Data plane nodes run mobility control functions that are in charge to establish the data paths to transport users' traffic, for instance creating IP in IP tunnels in case a handover takes place. This work can be counted in the partially distributed category, because the control plane functions rely on the Mobility Information Control Server (MICS), that acts as a central controller. Also, this is a clean slate approach because authors do not re-use any existing protocol to define their network entities nor to handle the signaling.

Other existing clean slate approaches leverage the concept of identifier/locator split to provide flatter architectures. In [82], the authors present a novel approach called HIMALIS (Heterogeneity Inclusion and Mobility Adaptation through Locator ID Separation) that advocates for mobility management built on top of the concept of a locator address and an identifier address. End host traffic is routed through the optimal direct path by the swapping of the locators used in the communication, while the connection is not closed as the identifiers are kept constant. The functionality provided by HIMALIS resembles existing approaches such as the Host Identity Protocol (HIP) [83] or SHIM6 [84], which are not just a proposal to enable mobility management, but rather a new network architecture. In the same way as the HIP/SHIM6 approaches, the main drawback of HIMALIS is on the difficulties to deploy it, given that the hosts' IP stack is considerably changed. A similar approach is followed in [85], where the locator/identifier split is obtained through the use of the Locator Identifier Separation Protocol (LISP) [86].

Last, it is worth mentioning the current trends envisioned by Software Defined Networking (SDN) in the DMM context. SDN enables centralized programmability of network entities, e.g., to build the routing infrastructure, without manual intervention on the nodes, or without running distributed routing algorithms. Even if SDN was primarily thought for fixed networks, recently it has emerged as a valuable resource also for wireless and mobile networks [87]. Indeed, SDN brings the advantages of fast and flexible data plane configuration, thence it suits very well to design distributed mobility management solutions. An example appears in [88], where SDN is applied to the Evolved Packet System. In this scenario, the mobile network comprises several PGWs, each of them responsible for their access network. An SDN controller maps the PGW where the MN is connected, so that the MN traffic flowing into the network from an ingress point is redirected to the corresponding PGW, seen as an egress point. Traffic redirection is performed through Network Address Translation between the ingress and egress nodes. This protocol is a partially distributed solution for the presence of the SDN controller, responsible of determining the data path for the user traffic.

Table 3.1: Summary chart of main DMM proposed solutions

Solution	Client-based	Network-based	
		Partially	Fully
Architecture dependent solutions			
Hahn [49, 50]			PGW relocation in 3GPP EPC
Bernardos [51]	DSMIPv6-based for 3GPP EPS		GTP/PMIPv6-based for 3GPP EPS
Kim [52]		MME-based edge PGWs	
Extension of existing protocols			
Liu [53]	MIPv6 based		
Wakikawa [54]	MIPv6 based		
Ali-Ahmad [55]	MIPv6 based		
Chan/Ernest [56–58]			PMIPv6 based
Chan/Ernest [59]			PMIPv6-based for NEMO
Chan/Ernest [60]		RO for PMIPv6	
Xue [61]		RO for PMIPv6	
D-PMIPv6 [62]		LMA split into CLMA and DLMA	
Jung [63]		PMIPv6 based	PMIPv6 based
Anchor PMIPv6 [64]		LMA for control plane IAs for data plane	
DMA-Seite/Bertin [38, 68]		PMIPv6 based	
Li [73]	Distributed HAs for NEMO		
Do [74]		PMIPv6 based for NEMO	

Continues on next page

Table 3.1: *Continues from previous page*

Solution	Client-based	Network-based	
		Partially	Fully
P2P appraches			
m-Chord [75]			multiple HAs and FAs interact through Chord
Zhai [77]	MIPv6 and Chord based		
DIMA [39]			DHT updated with BU and BA messages
Yu [78]	Loc./ID pairs stored in DHT		
Clean slate approaches			
McCann [79, 80]			BGP/DNS based
AIMS [81]		Data and Control plane separation	
HIMALIS [82]	Loc./ID split		
Zhang [85]		LISP-based	
Valtulina [88]		SDN-based in EPS	

3.5 Final remarks

We have explored in the previous sections several ideas to realize a DMM protocol, spanning from little changes to existing mobility support mechanisms, to the deployment of a completely novel network architecture.

The solutions that we propose in the next chapters belong to the *Extensions to existing protocols* category, as they are designed based on MIPv6 and PMIPv6. Therefore, our solutions share some features with those within the same category, but still, there are several elements that make our contributions original.

The main difference concerns the feasibility of such proposals. Many cited works are conceptual designs that require the specification of additional signaling, non-standard mechanisms and other custom operations. Indeed, the performance evaluation reported for these works is either based on mathematical analysis or simulations. The purpose of such methodology is rather starting a stub model to be optimized with future work than proposing a functional protocol.

On the contrary, our solutions are inspired by the wise adaptation of MIPv6 and PMIPv6 aimed at not introducing any non-standard signaling nor network entities beyond adding flags or options in modular data structures and message formats. Sometimes we incorporate or leverage protocols from other RFCs, but without touching the specifications therein defined. We argue that this approach produces the smallest impact on legacy equipment and specifications, thus simplifying a possible adoption as standard.

Moreover, our protocols come with the goal to provide all the necessary functional details that permit an implementation by taking existing MIPv6/PMIPv6 open source code and extending it following the new specifications. The already mentioned ODMM project offers indeed the platform where we made available our DMM implementations based on the solutions described in this thesis.

Part II

Distributed Mobility Management solutions

Chapter 4

Client-based DMM solution

The Distributed Mobility Management approach tries to overcome the limitations of the traditional centralized mobility management by bringing the mobility anchor closer to the MN.

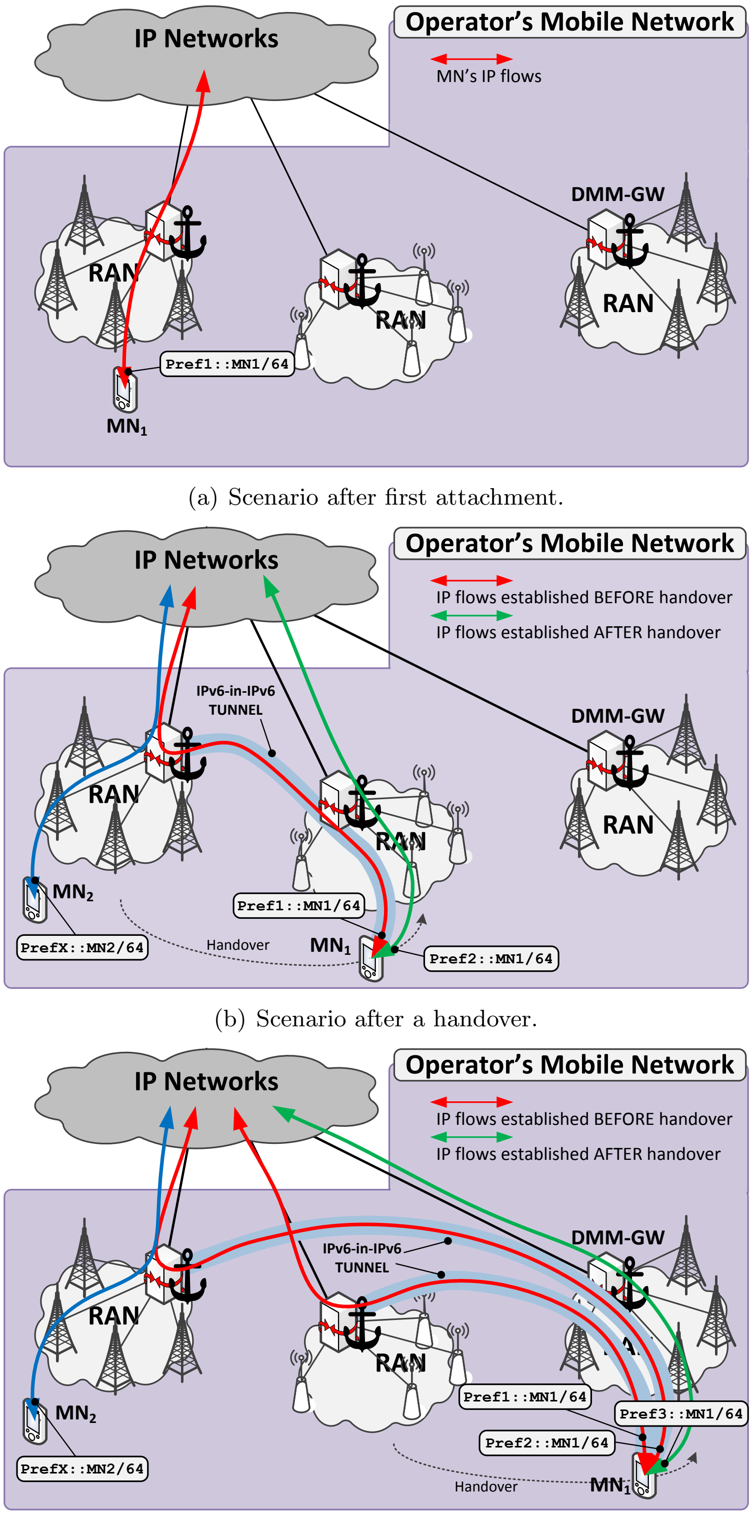
Following this idea, in this chapter we present a client-based mobility protocol derived from Mobile IPv6 in which the home agent is moved from the core to the edge of network, being deployed in the access router and default gateway of the mobile node.

This solution has been first published in [2] under the name of Flat Access and Mobility Architecture (FAMA), and its evolution is currently being maintained as an individual IETF Draft within the DMM Working Group [3]. In the following we refer to it as the client-based DMM solution, or simply **C-DMM**.

4.1 Solution Overview

In C-DMM, the MN coordinates its own mobility with the access routers that it visits while roaming among different access networks. Therefore access routers are provided with the mobility functions needed to interpret the mobility signaling with the MN, and to execute the required actions. In this context, and throughout the rest of the book unless stated differently, a mobility enabled access router is called **DMM Gateway (DMM-GW)**.

A DMM-GW possesses a private pool of IPv6 prefixes (i.e., not shared with any other DMM-GW in the domain), out of which it assigns one to an MN that attaches to any of its access links. The DMM-GW is the first IP hop seen by the MN, and its default gateway. Moreover, the DMM-GW ensures the reachability for the prefix it delegates regardless the fact that the MN with such prefix is still directly connected to a local access link, or moved to another DMM-GW's access network. Thence, according to the mobility terminology, the DMM-GW is a “mobility anchor”, in the sense that it always attracts the packets containing the anchored prefix, despite the current MN location. Due to the



(a) Scenario after first attachment.

(b) Scenario after a handover.

(c) Scenario after a second handover.

Figure 4.1: C-DMM architecture and example scenarios.

locally-anchored nature of the prefix delegated to an MN from the DMM-GW's pool, we refer to such as a **Locally Anchored Network Prefix (LANP)**, and we remark that, despite its name, the LANP is used to configure a *global* IPv6 address.

As a consequence, when the MN is connected to a DMM-GW's access link, packets containing the LANP are forwarded by the DMM-GW as a plain access router would do: in downlink, the packets are delivered to the MN through the direct link, whereas in uplink they are forwarded according to the shortest path computed by the domain's routing protocol (see the picture in Fig. 4.1(a)). Therefore the routing path is optimal for the MN's IP flows, and no extra processing due to packet encapsulation is spent. When the MN hands off to another access network, it connects to another DMM-GW, thus the LANP configured previously is no longer topologically correct. Therefore, an IPv6-in-IPv6 bi-directional tunnel is established between the MN and the previous DMM-GW to grant the LANP reachability in the new location. Thus, a DMM-GW routes a LANP in a standard way if the MN associated to the LANP is on a DMM-GW's link, whereas it performs encapsulation and decapsulation if the MN is connected to another DMM-GW. Besides, this double role can be played simultaneously on a LANP-basis, depending to the MN location (see the how flows belonging to MN_1 and MN_2 are handled by the leftmost DMM-GW in Fig. 4.1(b)).

In parallel, the new DMM-GW assigns another IPv6 prefix to the MN from its own LANPs pool. By doing so, the IP flows established before the handover are re-routed through the tunnel, whereas new flows can be started using the just configured address, ensuring an optimal path for these latter (see Fig. 4.1(b)). Thus, the MN eventually has multiple anchors, one for each visited DMM-GW as in the scenario in Fig. 4.1(c).

As we already mentioned, the MN is responsible for its own mobility management. Indeed, the tunnel is established, and therefore the prefix reachability is maintained upon handover, after a pair of mobility control messages exchanged between the MN itself and the old DMM-GW. In fact, after the handover, the MN configures a topologically correct (i.e., usable and reachable in the new access network) IPv6 address from the LANP advertised by the new DMM-GW. Such new address is used as source IPv6 address to pack a *Binding Update (BU)* message and send it over to the old DMM-GW, to report the MN current location. According to the MIPv6 terminology, after the handover, the previous DMM-GW regards its LANP as that used to form the Home Address (HoA), while the new prefix is part of the Care-of Address (CoA). The DMM-GW then stores the HoA–CoA pair for the MN and replies back with a *Binding Acknowledgement (BA)* message.

Note that it is not mandatory for the MN to keep an old LANP reachable, as this decision is dynamic and for example can be done on an application basis. In any case, this architecture basically enables a mobile node to simultaneously handle several IPv6 addresses – each of them anchored at a different DMM-GW – ensuring their continuous reachability by using Mobile IPv6 in a distributed fashion (i.e., each access router is a potential home agent for the LANP it delegates, if required). This distributed address anchoring is enabled on demand and on a per-address granularity, which means that depending on the user needs, it might be the case that all, some or none of the IPv6 addresses that an MN acquires while moving within a C-DMM domain are kept reachable and used by the mobility client.

4.2 Protocol Description

In traditional Mobile IPv6, the signaling between the MN and the HA is secured through IPsec [22], as mentioned in Section 2.1.1. Following a similar approach in C-DMM is difficult due to the large number of security associations that would be required, since any DMM-GW can play the role of a home agent for any mobile node within the domain. In order to overcome this problem and provide authentication between the DMM-GW and the MNs, we propose the use of Cryptographically Generated Addresses [89] (CGAs), as introduced in [90].

CGAs are basically IPv6 addresses for which the interface identifier part is generated by computing a cryptographic one-way hash function from a public key and the IPv6 prefix[1]. The binding between the public key and the address can be verified by re-computing the hash function and comparing the result with the interface identifier. To authenticate a message, the packet is signed with the corresponding private key, hence the receiver is able to authenticate the message with the knowledge of the address and the public key. CGAs are a powerful mechanism allowing packet authentication without requiring any public-key infrastructure, and hence it is well-suited for this application.

Following the basic idea presented above, next we describe how C-DMM works along with the CGA security system. Fig. 4.2 illustrates the message sequence chart for the mobility operations. When an MN attaches to a DMM-GW, let it be DMM-GW$_1$, it configures a CGA from the LANP anchored at that DMM-GW (for instance `Pref1::/64`), obtained after a Router Solicitation (RS) and Router Advertisement (RA) messages exchange. RS and RA messages are part of the usual IPv6 Router Discovery procedure (RD) [19], with the RA extensions introduced by MIPv6 that enable a router to advertise its global IPv6 address. The address acquired by the MN, indicated as `Pref1::CGA1/64`, can then be used to establish a communication with a remote Correspondent Node (CN) while attached to that particular DMM-GW. Let's recall that such IP flow is handled without encapsulation.

If the mobile then moves to DMM-GW$_2$, it obtains a second LANP after the RD process with the new DMM-GW, and it configures a second CGA address from it, that

[1] There are additional parameters that are also used to build a CGA, in order to enhance privacy, recover from address collision and make brute-force attacks unfeasible.

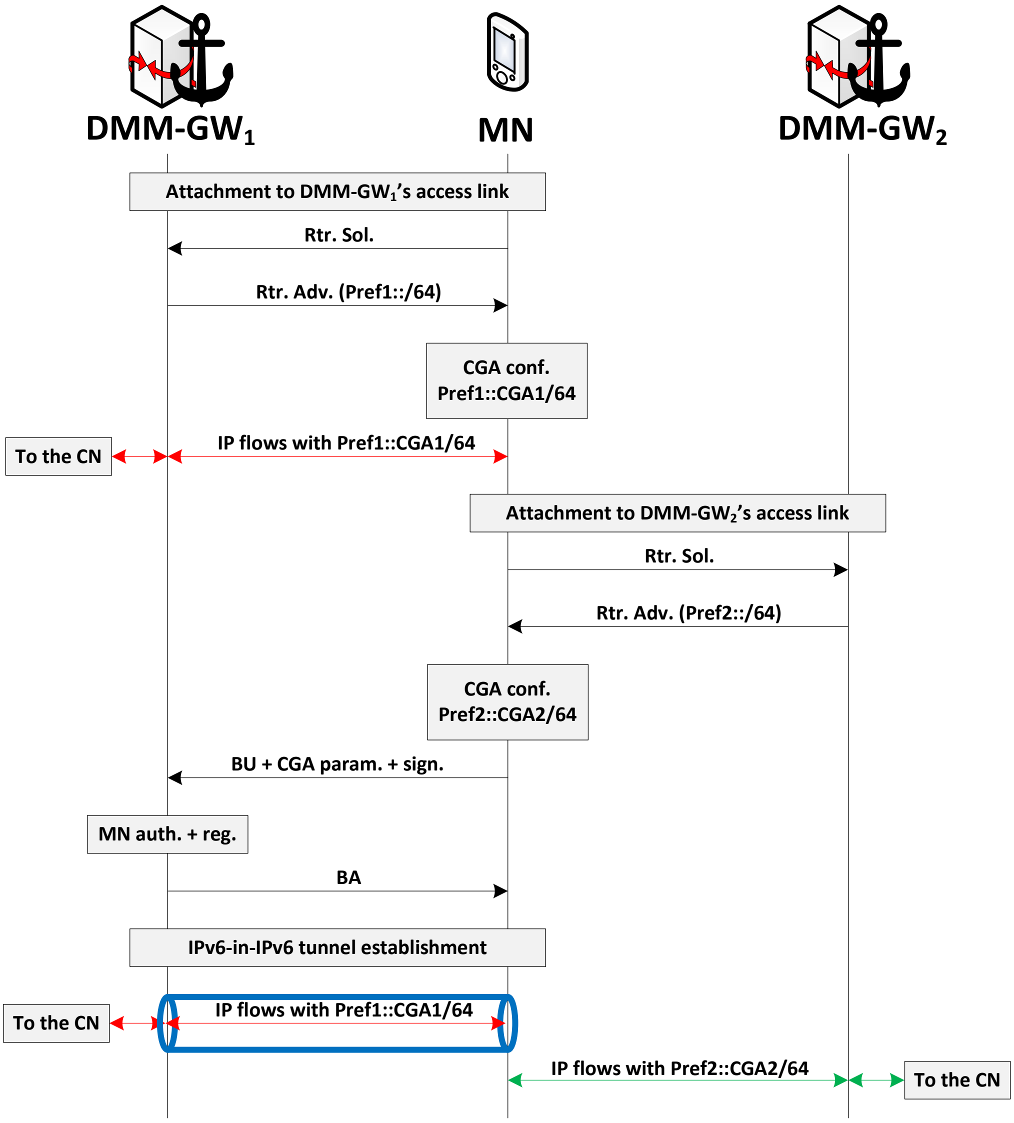


Figure 4.2: Signalling between the MN and the DMM-GWs.

is `Pref2::CGA2/64`[2]. After the handover, the following two cases are possible: *i)* there is no need for the address `Pref1::CGA1/64` to survive the movement: in this case no further action is required; *ii)* the mobile node wants to keep the reachability of the address `Pref1::CGA1/64`. In this latter case, the MN sends a Binding Update message to DMM-GW$_1$, using the address `Pref1::CGA1/64` as HoA and the address configured at the new DMM-GW, `Pref2::CGA2/64`, as CoA. This BU includes the CGA parameters and signature for `Pref1::CGA1/64`, which are used by DMM-GW$_1$ to identify the MN as the legitimate owner of the address. The BU might optionally include the CGA parameters

[2] Note that, even if the MN uses a single public key used to generate all the CGAs, and in fact we can safely assume so for simplicity, the IPv6 address interface identifier (i.e., the last 64 bits) results different. Indeed the CGA generation takes as input also the address prefix, i.e., the LANP, that is different in every access network.

and signature also for `Pref2::CGA2/64`, but it is not strictly necessary, so we omit this option. The handover operations conclude after the BA transmission from DMM-GW$_1$ to the MN, so that the bi-directional IPv6-in-IPv6 tunnel can be established. After the mobility operations, the IP flows started with `Pref1::CGA1/64` can be resumed redirecting the packets through the tunnel, whereas the new flows, started with `Pref2::CGA2/64`, transit through the optimal route via DMM-GW$_1$.

We remark that an MN can keep multiple IPv6 addresses active and reachable at a given time, requiring – every time the MN moves – a BU message to all the previous DMM-GWs that are anchoring the IP flows that the MN wishes to maintain. For instance, extending the example depicted in Fig. 4.2, if the MN moves to a third DMM-GW, it would send a BU to DMM-GW$_1$ containing `Pref1::CGA1/64` as HoA, and a BU to DMM-GW$_2$, with HoA `Pref2::CGA2/64`.

We conclude this subsection with a last consideration on the signaling, that results useful for the analysis conducted in Chapter 7. LANPs are assigned with an associated timer, that is the prefix's usability lifetime. If the LANP is associated to an on-link MN, then the renewal comes for free after the periodic IPv6 Neighbor Discovery process: if the MN replies, then the route for the LANP is maintained, otherwise it is deleted. If the MN is not on-link, then it is the MN itself responsible to maintain active the LANP by sending periodic BUs with a non-zero lifetime. If such BU does not arrive to the DMM-GW within the LANP's expiration time, then the DMM-GW assumes that the MN is no longer active with respect to its LANP and deletes the binding. Additionally, the MN can terminate the mobility session with a DMM-GW by sending a zero-lifetime BU for the corresponding LANP. In this way, the MN and network's resources are optimized to accommodate only the mobility sessions that are necessary.

4.3 Security considerations

Although the mechanism described above reduces the burden of the IPsec signalling overhead, a malicious node can still exploit it to export an incorrect CoA in the BU message. We mentioned that a BU might convey the CGA parameters and signature for the CoA as well, but still this does not bring any proof that the MN is really connected at the visited network through the CoA announced. Indeed, the CGA approach assures that the BU message has been sent by the legitimate HoA's (and optionally CoA's) owner but it does not guarantee that the same MN is *reachable* at the provided CoA. For instance a malicious node could injected a BU packet carrying the CoA as source address from an incorrect location, thence vanishing the purpose of the mobility tunnel that would be created, wasting resources at the DMM-GW. In order to provide a more robust solution, we propose a *Return Routability (RR)* procedure similar to the one defined for the MIPv6 *Route Optimization (RO)*.

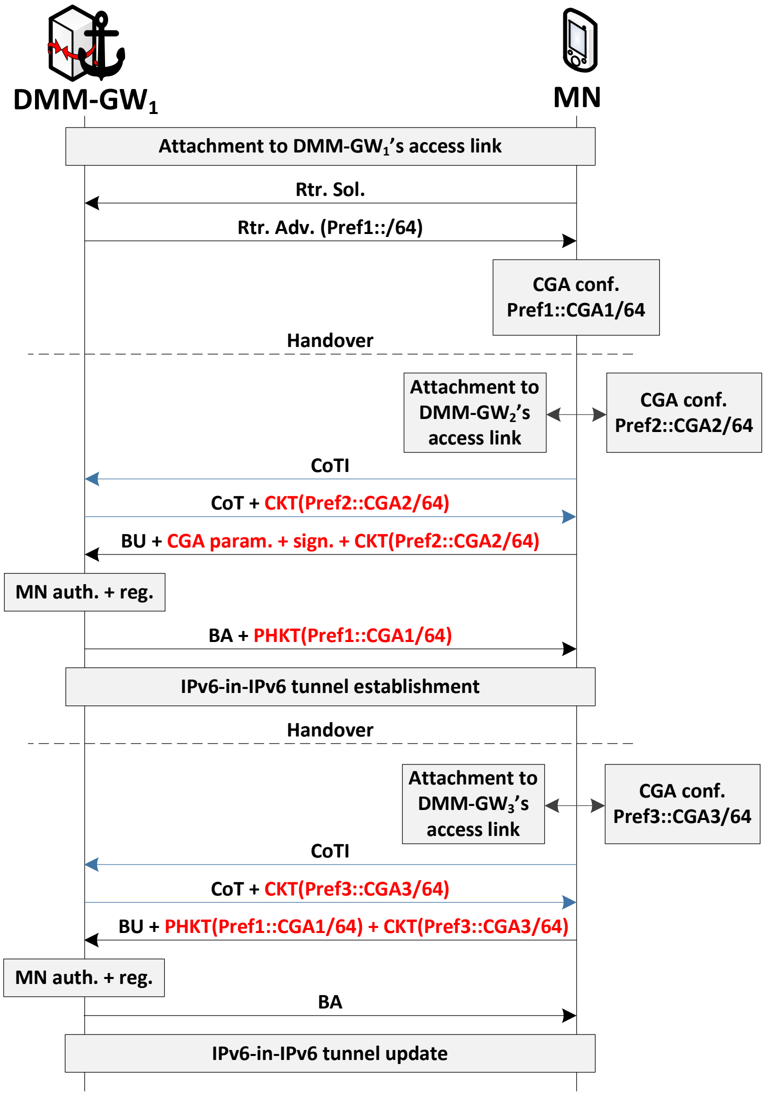

DMM-GW$_1$
MN
Attachment to DMM-GW$_1$'s access link
Rtr. Sol.
Rtr. Adv. (Pref1::/64)
CGA conf. Pref1::CGA1/64
Handover
Attachment to DMM-GW$_2$'s access link
CGA conf. Pref2::CGA2/64
CoTI
CoT + CKT(Pref2::CGA2/64)
BU + CGA param. + sign. + CKT(Pref2::CGA2/64)
MN auth. + reg.
BA + PHKT(Pref1::CGA1/64)
IPv6-in-IPv6 tunnel establishment
Handover
Attachment to DMM-GW$_3$'s access link
CGA conf. Pref3::CGA3/64
CoTI
CoT + CKT(Pref3::CGA3/64)
BU + PHKT(Pref1::CGA1/64) + CKT(Pref3::CGA3/64)
MN auth. + reg.
BA
IPv6-in-IPv6 tunnel update

Figure 4.3: Detailed signalling with the enhanced security mechanisms.

The signaling for the enhanced security mechanism is illustrated in Fig. 4.3. The RR procedure starts once the MN configures the CoA after the handoff. Before sending the BU message, the MN sends a Care-of Test Initialization message (CoTI) to the old DMM-GW. This message is replied by the DMM-GW with a Care-of Test message containing a CoA Keygen Token (CKT) associated to such CoA (`Pref2::CGA2/64` in our example). Upon receiving the CKT, the MN sends a BU signed with the CGA method described above and includes a secret generated with the CKT. The authenticated BU message and the knowledge of the secret CKT is a proof for the DMM-GW that the MN is the legitimate node who has sent the BU and it is also reachable at the CoA indicated.

As most of the security improvements, also the one proposed incurs in a performance penalty, in this case an increased handover delay. Specifically, this enhanced security approach requires two more control messages, the CoTI and CoT, to be exchanged between the MN and the DMM-GW, resulting in one additional Round Trip Time (RTT) in terms of handover latency. Moreover, even though the use of CGAs does not impose a heavy burden in terms of performance, depending on the number of MNs handled by the DMM-GW, the processing of the CGAs can be problematic. To reduce the overall computational complexity, we suggest an alternative mechanism to authenticate any subsequent signaling packets exchanged between the MN and the DMM-GW (in case the mobile performs a new attachment to a different DMM-GW). This alternative method relies on the use of a Permanent HoA Keygen Token (PHKT) associated to the HoA delegated by the DMM-GW. The PHKT is forwarded to the MN in the Binding Acknowledgment message, sent in reply to the first received BU. The MN will use it to generate an Authorization option that is included in all the next Binding Update messages.

Therefore, when the MN performs any subsequent movement, the BU messages are secured using the secrets computed using the PHKT and the last CKT received, reducing the computational load at the receiving DMM-GW. Note that each DMM-GW visited by the MN delivers a different PHKT to the MN, so that each mobility session maintained by the MN is associated to a specific (DMM-GW, HoA, PHKT) triple.

4.4 Final Remarks

In this chapter we have described C-DMM, a distributed mobility management solution based on Mobile IPv6.

Compared with its centralized mobility counterpart solution, C-DMM brings the advantages of a distributed solution. Indeed, C-DMM offers:

- *shorter data paths*, because the anchor is located closer to the MN, hence the data traffic is not forced to traverse the operator's core network where an HA usually resides;
- *better scalability*, as a DMM-GW should handle traffic from a reduced subset of MNs, compared to what an HA does; therefore, the computational and bandwidth provisioning for a DMM-GW is reduced as well;
- *reliability*, since the C-DMM architecture does not suffer the single point of failure problem;
- *shorter handover latency*, because the entities involved in the handover signaling are close to each other;
- *better control on the mobility granularity*, as a result of the anchoring split based on the prefixes allocated to the MN.

The solution is compliant with the security mechanism envisioned by the original MIPv6, but, in addition, it proposes an enhanced security mechanism based on Cryptographically Generated Addresses to reduce the computational load at the DMM-GW that does not require the use of IPsec.

The C-DMM protocol studied in this chapter intrinsically lays down the concept of Localized Mobility Domain (LMD), which typically belongs to network-based mobility protocols like PMIPv6. Indeed, the set of DMM-GWs deployed by an MNO form the domain where the MN can move while benefiting from optimal C-DMM mobility support. Nonetheless, the PMIPv6 domain and the C-DMM are quite different in the sense that if the MN happens to move to an access network that is not part of the PMIPv6 domain, then the MN ceases completely to receive the PMIPv6 mobility support. On the contrary, with the C-DMM protocol, the MN can still establish mobility sessions with the old DMM-GWs, even if it attaches to an access network where there are no DMM-GWs deployed. However, the MN cannot enjoy a full service, because the new IP flows started in the foreign access network are not supported by the C-DMM protocol, unless tunneled to the old DMM-GWs. We explore this scenario in more details in Chapter 6, where we suggest a DMM solution for inter-domain mobility, sometimes known as the *roaming* case.

As a final word, it is worth mentioning that a first C-DMM implementation is available from the ODMM project, Open Platform for Distributed Mobility Management Solutions, at `http://www.odmm.net/c-dmm/`.

Chapter 5

Network-based DMM solutions

Network-based mobility management is a mobility support method that does not require the MN to maintain any mobility session with the anchor nor to explicitly notify its location changes. Indeed, the network nodes are responsible to track the MN movements and to re-configure the routing map accordingly. With respect to the IP mobility solutions space, we have already mentioned in Section 2.1 that, with network-based IP mobility, the MN is transparent to the mobility management, being the terminal unaware of the changes at the IP layer that a handover may produce. In addition, a client mobility solution modifies the host's IP stack, whereas a network-based IP mobility solution supports legacy IP stacks.

In this chapter we explore several solutions for network-based distributed mobility management, and we refer to them as the **N-DMM** solutions. As MIPv6 was the reference centralized solution for C-DMM, for the N-DMM designs we consider as starting point Proxy Mobile IPv6, the network based protocol standardized by the IETF that we described in Section 2.1.2. The N-DMM solutions that we propose next are modifications to PMIPv6 to operate in a distributed manner, and they basically follow two different approaches:

- the *Partially distributed* solutions;
- the *Fully distributed* solutions.

In both categories, the basic principle is to remove the PMIPv6's LMA from the data plane, whereas the control plane is designed to include or not a centralized role. As the name suggests, in the partially distributed approach the data plane is distributed but the control plane is kept centralized, being it bound to a controller similar to the PMIPv6's LMA. In the fully distributed scheme, both the data and control planes are distributed among the protocol entities without a centralized coordination.

Since all the solutions envisage the same data plane scheme, we start from its description and then move to the control plane management for each solution. All the network-based solutions that follow, partially and fully distributed, share the concept of

DMM Gateway (DMM-GW) introduced in Chapter 4, that is, a mobility-enabled access router that possesses an IPv6 prefixes pool for the assignment to the connected MNs.

The protocols described in this chapter are currently maintained within the IETF DMM working group as an individual draft submission [5].

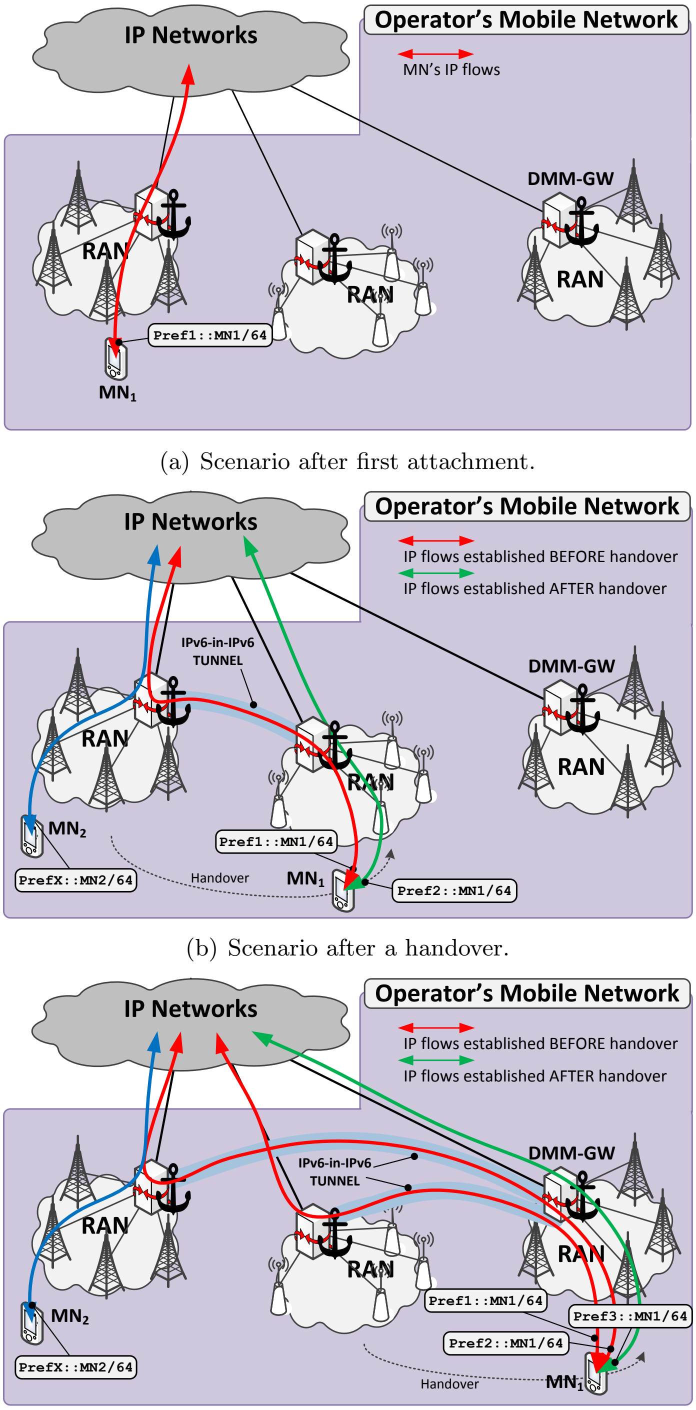
5.1 Data plane management

A serving DMM-GW provides IPv6 connectivity to the mobile node connected to one of its access links through the **Locally Anchored Network Prefix (LANP)** delegated to the MN. The LANP is an IPv6 prefix belonging to the DMM-GW's prefixes pool, and it is assigned to the MN on a per MN basis. The MN then configures an IPv6 address from the LANP using Stateless Address Auto configuration (SLAAC) [19]. The DMM-GW forwards packets carrying the LANP without encapsulation, as a plain IPv6 access router, when the MN is connected to any DMM-GW's access link. This holds both in downstream and upstream directions, as illustrated in Fig. 5.1(a).

If the mobile node moves to another DMM-GW's access network, a new LANP is obtained from that DMM-GW, and another IPv6 address is configured. This latter address is preferred by the MN to start new IP flows, so packets benefit from optimal routing. Ongoing data sessions are recovered by re-routing them through an IPv6-in-IPv6 bi-directional tunnel created between the new DMM-GW and the previous one. In this way the reachability of the old IPv6 address is preserved and ongoing communications are not disrupted (see Fig. 5.1(b)). Borrowing the PMIPv6's terminology, the DMM-GW where the MN is currently connected behaves as a MAG for the old flows, and the previous DMM-GW as an LMA. On the contrary, for the fresh IP flows, the serving DMM-GW is a plain IPv6 router.

The MN may have hence a number of flows directly routed by the new DMM-GW to and from the global Internet without encapsulation, and another set of streams anchored at the previous DMM-GW. Depending on the MN's movement within the domain and the active IP sessions, this situation might be replicated for multiple DMM-GWs, like in the scenario of Fig. 5.1(c).

The data plane described here is thus pretty similar to what was examined for C-DMM in Section 4.1, with the difference that, in C-DMM, the tunnel endpoints terminate at a DMM-GW and at the MN, whereas in N-DMM the tunnels are established between DMM-GWs only. Nevertheless, at the control plane level, there is no longer an intelligence in the MN able to notify old DMM-GWs about the location change, hence the key aspect in N-DMM is how the DMM-GWs coordinate among each other to track the MN location. This task belongs to the control plane, and in the following, we develop this for the partially distributed scheme first.



(a) Scenario after first attachment.

(b) Scenario after a handover.

(c) Scenario after a second handover.

Figure 5.1: N-DMM architecture and example scenarios.

5.2 Control plane management: the partially distributed approach

In the partially distributed scheme, the data and control planes are split to operate separately. In this approach, whilst the data plane is distributed, the control plane is kept centralized and it leverages the PMIPv6 control plane to store the mobility sessions for all the MNs at a central entity. We call this entity the *Central Mobility Database (CMD)*, to distinguish it from the legacy LMA, because, even if they share many functionalities, they differ for some crucial aspects. Indeed, similar to an LMA, the CMD stores the mobility sessions of the MNs in a data structure called *Binding Cache Entry (BCE)*, one for each MN. However, the CMD extends the legacy BCE structure by applying the following changes (see Section A.3 for the format details):

- The PMIPv6 Mobile Node Home Network Prefix (MN-HNP) is no longer a valid concept in N-DMM, as each DMM-GW assigns a different LANP from their prefixes pool. Therefore the legacy BCE's MN-HNP field is renamed into the **LANP** field, and it contains the prefix assigned to the MN in the current access network;
- the legacy Proxy CoA (P-CoA) field is renamed **Serving DMM-GW** field, and it holds the IPv6 address of the DMM-GW where the MN is currently attached;
- a new item is added to the BCE structure, called the **Previous DMM-GWs List**. Each element of such list is composed by a DMM-GW/LANP pair, as it holds the IPv6 address of a previously visited DMM-GW that is still anchoring MN's IP flows along with the LANP delegated by such DMM-GW.

With these changes the new BCE of an MN maintains the notion about which DMM-GW is currently serving the MN, and which are anchoring (if any) the MN's old flows. Also, the BCE associates each stored DMM-GW to the LANP it assigned, hence keeping track of the most valuable information of the MN mobility context.

The BCEs at the CMD are created, updated and deleted upon the mobility events that take place in the access networks. The DMM-GWs are the entities in charge to notify the mobility events to the CMD, by means of *Proxy Binding Update (PBU)* and *Proxy Binding Acknowledgement (PBA)* messages. Such signaling is imported from the PMIPv6 specifications, but a CMD has the intelligence to process and forge both PBU and PBA messages: this is another difference with respect to the LMA, which cannot send PBU messages nor receive PBA messages. Also, two new mobility options are added within the scope of N-DMM, and a legacy one is simply renamed:

- the *Serving DMM-GW mobility option* contains the IPv6 address of the DMM-GW where the MN is currently attached. This option is used to notify an old DMM-GW about the new MN location.

- the *Previous DMM-GW mobility option* contains the IPv6 address of a DMM-GW previously visited by the MN that is still anchoring MN's IP flows and the LANP it delegated to the MN for those IP flows.
- the PMIPv6 legacy Mobile Node Home Network Prefix mobility option is renamed into *LANP mobility option.*

More details and the mobility option message format can be found at the end of the thesis in Section A.3.

Next sections are devoted to describe the protocol operations in more details, but first we want to remark that solutions from a PMIPv6-based partially distributed approach have been presented also by other authors, and their works are quite different from ours. For instance, [68] and [74] propose a similar partially distributed approach, where a server acting as a mobility sessions store is queried by the new serving anchor to retrieve the old MN locations. With such information at disposal, the serving anchor then sends to the previous anchor(s) a control message and the proper routing configuration can be set for the MN. [68] focuses on host mobility, while [74] applies to a NEMO scenario, but both share the same principles. On the contrary, our CMD is not simply a passive session store, for it takes an active key role in the signaling. Indeed, being the CMD in possession of the whole picture associated to the MN mobility through the extended BCE, it can exploit its vantage point to command the DMM-GWs to react accordingly to the location change[1] (e.g., to create the appropriate data paths for the MN's prefixes). We explore in the following three partially distributed solutions, each attributing a different role to the CMD:

1. the CMD behaves as a PBU/PBA relay, or the **"relay"** solution,
2. the CMD behaves as a DMM-GW locator, or the **"locator"** solution,
3. the CMD behaves as a PBU/PBA proxy, or the **"proxy"** solution.

For each solution we focus on the three main phases in which a mobility management protocol is typically split: *i)* the **initial registration**, *ii)* the **handover management** and *iii)* the **de-registration**.

5.2.1 The "relay" solution

In this solution the CMD receives a notification from the new serving DMM-GW through a PBU message and then forwards the PBU message with a slight adjustment to the old DMM-GWs. These DMM-GWs in turn reply to the CMD with a PBA message that eventually delivers the last acknowledgement back to the DMM-GW that originated the notification. Since the CMD stands in the middle of a DMM-GW to DMM-GW communication, it acts as a "relay", therefore the solution name.

[1] The MME in the EPS has a similar role for it instructs the SGW to build the tunnel with the new eNB after a handover, see Section 2.2.1.

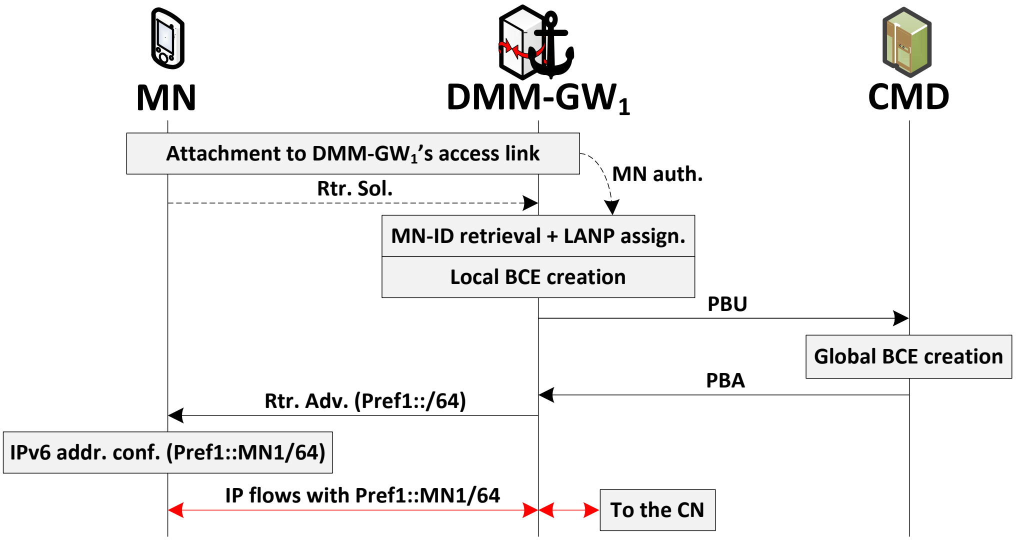

MN
DMM-GW$_1$
CMD
Attachment to DMM-GW$_1$'s access link
MN auth.
Rtr. Sol.
MN-ID retrieval + LANP assign.
Local BCE creation
PBU
Global BCE creation
PBA
Rtr. Adv. (Pref1::/64)
IPv6 addr. conf. (Pref1::MN1/64)
IP flows with Pref1::MN1/64
To the CN

Figure 5.2: Partially distributed N-DMM: initial registration.

Initial registration. The message sequence for the initial registration phase is described in the following and depicted in Fig. 5.2. In fact, this stage is the same for all the three flavours proposed as partially distributed solutions, thence it is presented only here for brevity.

Upon the mobile node attachment to a DMM-GW, say DMM-GW$_1$, the DMM-GW obtains the MN unique identifier in the domain (MN-ID) and a LANP from the DMM-GW's prefix pool is reserved for the MN (let it be `Pref1::/64`). A DMM-GW has two different options to detect the MN attachment:

i) based on link layer triggers;

ii) based on the IPv6 router discovery procedure initiated by the MN with a *Router Solicitation (RS)* message.

The former depends on the link layer technology, whilst the latter is technology-agnostic. Note that, even if method *i)* is used, the MN may still send the RS message[2].

Also, there are alternative solutions to obtain the MN-ID. Indeed, the MN-ID is a domain-wide identifier useful only within the scope of the N-DMM protocol, therefore, a network administrator can choose to bind the MN-ID retrieval to the authentication mechanism deployed in the radio access link, or to decouple the two processes.

In the former case, it is convenient to combine the MN-ID retrieval to the attachment detection, so that the wireless access point (e.g., an eNB or a WiFi AP) communicates the MN-ID to the DMM-GW after the link layer technology-specific authentication process. Such communication would serve as well as an attachment notification for the DMM-GW. For instance, for an LTE link layer, the MN-ID can be associated to the subscriber's

[2] Although the support for the RS message is mandatory in an IPv6 stack, the sending policy depends on the host's IPv6 stack implementation.

identity and stored in the terminal's Universal Subscriber Identity Module (USIM). The eNB obtains the subscriber's credentials after the LTE random access procedure initiated by the terminal to register to the network, and then the eNB can translate this piece of information into the MN-ID sent to the DMM-GW. Similarly, a WPA/WPA2-secured IEEE 802.11 AP may first authorize the radio link establishment to the MN, and next notify it to the DMM-GW, sending for instance the MN's MAC address as the MN-ID.

Alternatively, the MN-ID retrieval might be decoupled from the access security mechanism, which can be even missing, and performed *after* the attachment detection. Such technique is the only possible when the attachment detection method *ii)* is used, and it can be realized deploying an Authentication, Authorization and Accounting (AAA) infrastructure directly accessed by the DMM-GW. For instance, the AAA infrastructure can be a Diameter [30] or RADIUS [29] server queried by the DMM-GW against the MN's link layer address, or the MN's IPv6 link local address, a parameter conveyed within the RS message. Note that the AAA infrastructure can be also adopted when using the attachment detection *i)*. In order to keep the picture of Fig. 5.2 as simple as possible, we have included both attachment detection methods, but we have omitted the MN-ID retrieval.

After attachment phase, the DMM-GW includes the MN-ID/LANP pair as part of a BCE locally stored. Then, it packs the MN-ID and LANP parameters into the corresponding mobility options and sends them to the CMD within a PBU message[3]. Since the MN is attaching to the domain for the first time, the CMD has no previous BCE for it. Hence a fresh BCE is created as well at the CMD, containing as main fields the MN-ID, the LANP and the Serving DMM-GW, this latter holding the DMM-GW$_1$'s IPv6 global address. The CMD then replies to DMM-GW$_1$ with a PBA message, which is a mere copy of the options included in the PBU message received before, meaning that the mobile node's registration is fresh and no additional information was previously available at the CMD. DMM-GW$_1$ finalizes the registration of the local BCE previously created and unicasts a *Router Advertisement (RA)* message to the mobile node. The RA message includes the reserved LANP, which can be used by the MN to configure an IPv6 address with stateless address auto-configuration (e.g., `Pref1::MN1/64`). Since this address is locally anchored at the serving DMM-GW, no encapsulation nor special handling is required to route packets of IP flows started there, as illustrated in Fig. 5.1(a).

Handover management. When the MN moves from its current serving DMM-GW, i.e., DMM-GW$_1$, and attaches to another DMM-GW, say DMM-GW$_2$, the handover is handled in 5 steps (see Fig. 5.3):

1. As for the initial registration phase, DMM-GW$_2$ fetches the MN-ID (the same obtained before) and reserves a LANP for it (say `Pref2::/64`) from its local pool. Then it

[3] The PBU contains a batch of other mandatory options which description is not relevant here.

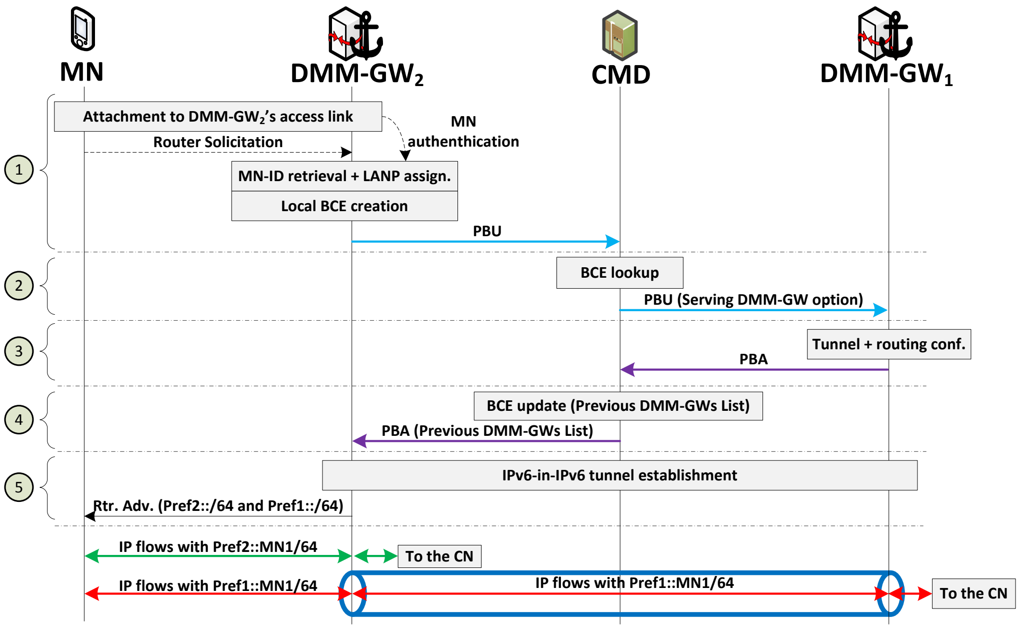

MN
DMM-GW$_2$
CMD
DMM-GW$_1$
Attachment to DMM-GW$_2$'s access link
MN authenthication
Router Solicitation
1
MN-ID retrieval + LANP assign.
Local BCE creation
PBU
2
BCE lookup
PBU (Serving DMM-GW option)
3
Tunnel + routing conf.
PBA
4
BCE update (Previous DMM-GWs List)
PBA (Previous DMM-GWs List)
5
IPv6-in-IPv6 tunnel establishment
Rtr. Adv. (Pref2::/64 and Pref1::/64)
IP flows with Pref2::MN1/64
To the CN
IP flows with Pref1::MN1/64
IP flows with Pref1::MN1/64
To the CN

Figure 5.3: Handover management in the “relay” solution.

creates a local BCE, and it sends a plain PBU to the CMD for registration.

2. Upon PBU reception, the CMD looks up the MN-ID in its binding cache, retrieving the BCE created before for the MN. Since the BCE indicates the DMM-GW$_1$'s address in the Serving DMM-GW field, the CMD forwards the received PBU message to DMM-GW$_1$, appending to the message a Serving DMM-GW option that holds DMM-GW$_2$'s IPv6 address. Also, the BCE's Serving DMM-GW field is updated pointing now to DMM-GW$_2$'s IPv6 address.
3. After the PBU reception from the CMD, DMM-GW$_1$ realizes by inspecting the Serving DMM-GW option that the MN is no longer attached to a direct link, but, instead, to DMM-GW$_2$. Hence it updates its BCE accordingly. Then, DMM-GW$_1$ sets up an end-point for the bi-directional tunnel towards DMM-GW$_2$ and adds the required rule to route `Pref1::/64` through the tunnel. DMM-GW$_1$ informs the CMD that these steps have been accomplished by sending a PBA message.
4. The CMD, after receiving the PBA, proceeds to populate the BCE's Previous DMM-GWs List. As we already mentioned, an element of the Previous DMM-GWs List is composed by a DMM-GW/LANP pair, so, in our example, the Previous DMM-GWs List holds as first element the DMM-GW$_1$'s address and `Pref1::/64`. Note that such information was already available to the CMD in the BCE after the MN initial registration, so, with this operation, the parameters are simply moved from one BCE's field to another. Finally, the CMD sends a PBA to the current serving DMM-GW, including an instance of the Previous DMM-GW option for each item of the Previous DMM-GWs List (in this case only one).

5. The PBA message enables the serving DMM-GW to finally establish the correct routing state, i.e., the bi-directional tunnel with DMM-GW$_1$ and the routing entries for `Pref1::/64`, both in uplink (through the tunnel) and downlink (through the access link). The Serving DMM-GW concludes the operations by advertising to the MN the new LANP, locally anchored at that same DMM-GW, and also the old LANP. This latter is advertised specifying, as ancillary prefix attributes, a non-zero valid lifetime and a zero preferred lifetime. In this way, the old address can be correctly used to terminate old data sessions, but it is deprecated for using in new ones, forcing the MN to pick the address advertised by the current DMM-GW. Note that the RA message is sent with the same link-local IPv6 address used by the previous DMM-GW. In this way, the MN does not detect any change in the default router and next hop's identifiers. This practice is inherited from the recommendations specified for a MAG in PMIPv6.

Fig. 5.1(b) illustrates how old and new IP flows are routed in the domain. Any subsequent mobile node's handover follows the same procedure, involving all the previous DMM-GWs that are anchoring active flows incrementally. Indeed, when the CMD receives the first PBU message from another serving DMM-GW, be it DMM-GW$_3$, the CMD forwards a PBU to the DMM-GW pointed by the current value of the Serving DMM-GW field, (i.e., DMM-GW$_2$), and to all the DMM-GWs contained in the Previous DMM-GW list (in this scenario is populated with DMM-GW$_1$ only). All the DMM-GWs that receive a PBU update their BCE of the MN and the corresponding routing entries. They eventually reply back with a PBA message to the CMD and become part of the updated Previous DMM-GWs List. Such list is aggregated into a single PBA and sent over to the new serving DMM-GW. In this way the routing state can be re-configured in the whole domain for all the anchored flows (see Fig. 5.1(b)).

De-registration. An MN mobility session is associated to a leasing time stored in the BCE's Lifetime field, indicating how long the MN is granted mobility support within the domain. When the BCE Lifetime is about to expire, the serving DMM-GW attempts with a Neighbor Discovery (ND) process to assess whether the terminal is still attached to the access link or not. In case the MN is still connected, the leasing is refreshed for another BCE Lifetime, and the refreshment is notified to the CMD with a PBU/PBA exchange that does not involve the previous DMM-GWs. Otherwise, the timer is let expire and the MN is eventually de-registered from the domain. The de-registration is performed by sending a PBU with a zero value lifetime to the CMD, that then next propagates the information to the nodes in the Previous DMM-GWs list. The old DMM-GWs delete their BCE of the MN and the corresponding routing rule and tunnel, and next acknowledge the CMD with a PBA, which is finally delivered to the serving DMM-GW that originated the de-registration. After this stage the MN mobility session is deleted

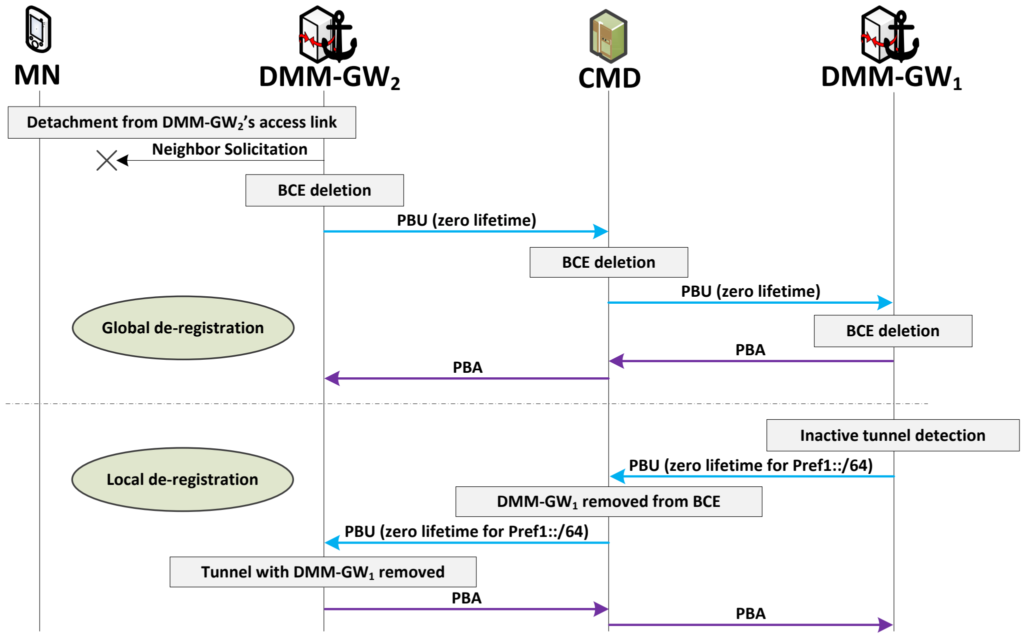

MN
DMM-GW$_2$
CMD
DMM-GW$_1$
Detachment from DMM-GW$_2$'s access link
Neighbor Solicitation
BCE deletion
PBU (zero lifetime)
BCE deletion
PBU (zero lifetime)
Global de-registration
BCE deletion
PBA
PBA
Inactive tunnel detection
PBU (zero lifetime for Pref1::/64)
Local de-registration
DMM-GW$_1$ removed from BCE
PBU (zero lifetime for Pref1::/64)
Tunnel with DMM-GW$_1$ removed
PBA
PBA

Figure 5.4: De-registration in the "relay" solution.

from all the DMM-GWs visited by the MN.

It is worth noting that only the DMM-GW currently serving the MN can state if the terminal is still attached using the ND procedure, but still, also old DMM-GWs have a leasing time associated to the prefix they allocated, but they cannot assess the MN presence in the domain. As a consequence, if an old DMM-GW deletes a local BCE of the MN, and consequently the rouging rules, ongoing flows carrying the LANP assigned by that DMM-GW cannot be delivered any longer. On the contrary, if an old DMM-GW is prevented to perform a local de-registration, then the BCE and the routing rules are maintained even if there are no flows requiring it, wasting resources at the DMM-GW. A possible solution could be the following: an old DMM-GW freezes its local leasing timer upon receiving the PBU from the CMD during the handover phase. Then it starts monitoring the activity of the MN's flows through the tunnel[4]. When the tunnel is inactive for a certain interval, then the old DMM-GW can proceed to perform the local de-registration, by deleting the local BCE and sending a zero-lifetime PBU to the CMD. The CMD removes that DMM-GW from the Previous DMM-GW list and informs also the current serving DMM-GW, that can delete the tunnel and routing rules for the prefix anchored by that DMM-GW. The procedure is over when the serving DMM-GW acknowledges the operation with a PBA, sent over to the old DMM-GW through the CMD.

[4] Flow monitoring can be performed with well known Deep Packet Inspection (DPI) strategies, for instance based on the 5-tuple formed by the IPv6 header's source address, destination address and next header fields, and the transport header's source port and destination port fields.

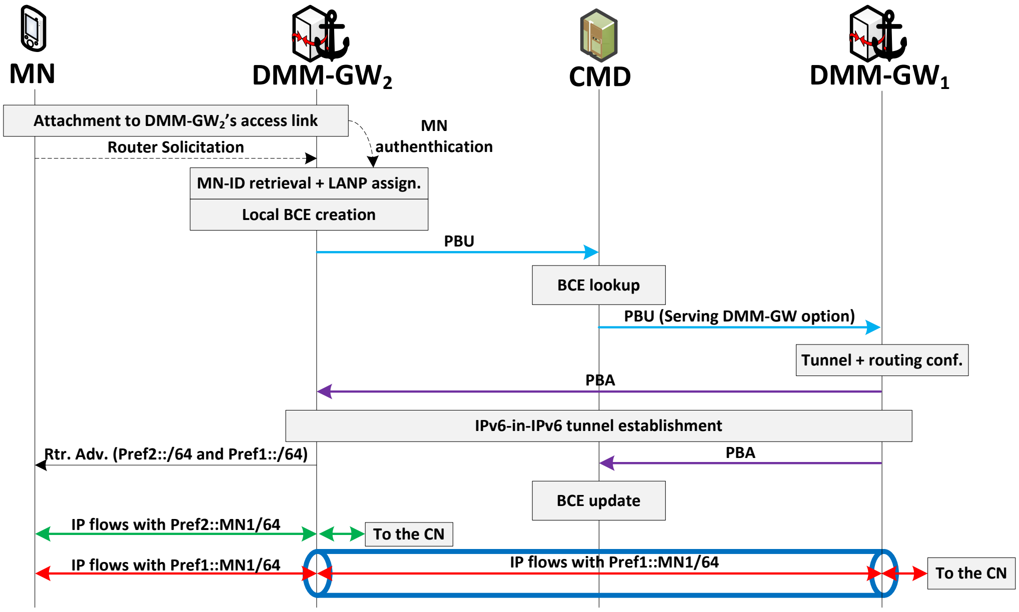

MN
DMM-GW$_2$
CMD
DMM-GW$_1$
Attachment to DMM-GW$_2$'s access link
MN authenthication
Router Solicitation
MN-ID retrieval + LANP assign.
Local BCE creation
PBU
BCE lookup
PBU (Serving DMM-GW option)
Tunnel + routing conf.
PBA
IPv6-in-IPv6 tunnel establishment
Rtr. Adv. (Pref2::/64 and Pref1::/64)
PBA
BCE update
IP flows with Pref2::MN1/64
To the CN
IP flows with Pref1::MN1/64
IP flows with Pref1::MN1/64
To the CN

Figure 5.5: Handover management in the "locator" solution.

We observe at last that, when the de-registration comes from the serving DMM-GW, the MN mobility session is deleted throughout the whole domain, hence we can call it a *global* de-registration, whereas when it comes from an old DMM-GW, it accounts only for the LANP allocated by that DMM-GW, hence it is a *local* de-registration.

5.2.2 The "locator" solution

The initial registration phase is identical to that described for the "relay" solution, thus we move to the handover management operations.

Handover management. The mobility update procedure follows the same steps defined before for the "relay" solution up to step 2, the moment when an old DMM-GW receives the PBU message from the CMD carrying the Serving DMM-GW mobility option. At this point, the old DMM-GW is aware of the new mobile node's location (because of the new DMM-GW's address pointed by the mobility option). Therefore, the previous DMM-GW signals with a PBA message directly to the serving DMM-GW the LANP it is anchoring for that MN (included in the LANP mobility option, so no extra options are necessary). A similar message is sent to the CMD too, to maintain the consistency in the global database. The routing state can be recovered and the procedure is expected to terminate quicker than the previous scheme. Fig. 5.5 illustrates the new signaling sequence.

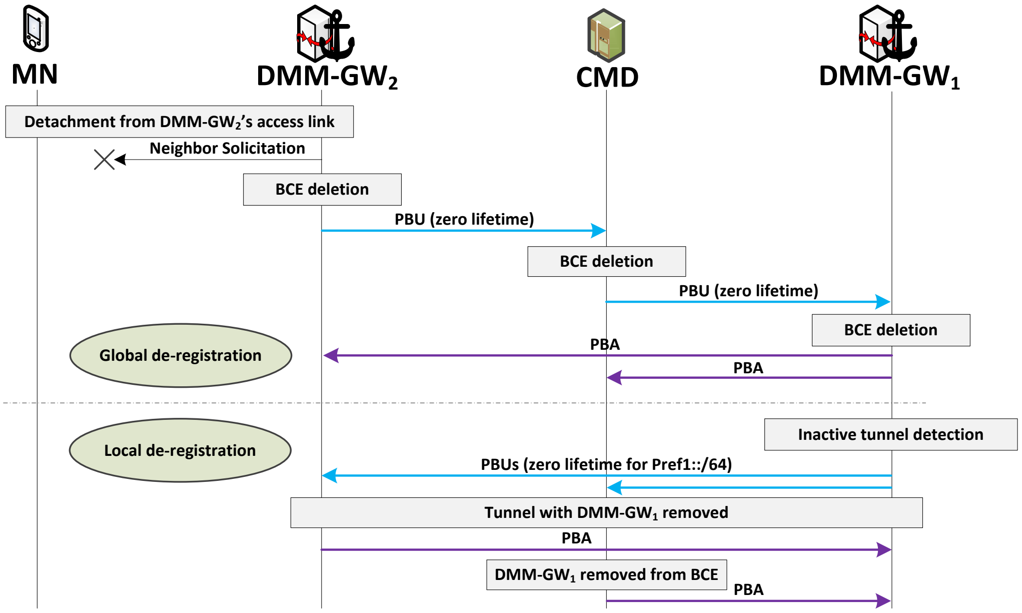


Figure 5.6: De-registration in the "locator" solution.

De-registration. The de-registration technique is conceptually similar to the one in the "relay" solution, but it reflects the message sequence envisioned for the handover management. When the serving DMM-GW proceeds to a global de-registration, it sends a zero lifetime PBU to the CMD that in turn forwards the command to the DMM-GWs in the Previous DMM-GWs List. At this point the old DMM-GWs delete their entry for the MN and the corresponding routing configuration, and eventually conclude the procedure by sending a PBA to the CMD and one to the serving DMM-GW.

In case of local de-registration, the DMM-GW anchoring an inactive LANP transmits a zero lifetime PBU to the CMD and another to the serving DMM-GW, and both recipients, after taking the corresponding actions, reply back to the old DMM-GW.

5.2.3 The "proxy" solution

The previous mechanisms can be further sped up by exploiting the privileged vantage point of the CMD upon the global mobility session produced by the MN. Again, we skip the initial registration, being it the same for all the three partially distributed flavors.

Handover Management. When the CMD receives the PBU message from the new serving DMM-GW, it already possesses the whole mobility picture of the MN, being the latest DMM-GW prior to handover stored by the current value of the BCE's Serving DMM-GW field, and the anchoring DMM-GWs pointed by the Previous DMM-GWs List. Therefore, the new serving DMM-GW is notified immediately with a PBA message, including the Previous DMM-GW List packed in the appropriate option. At the same

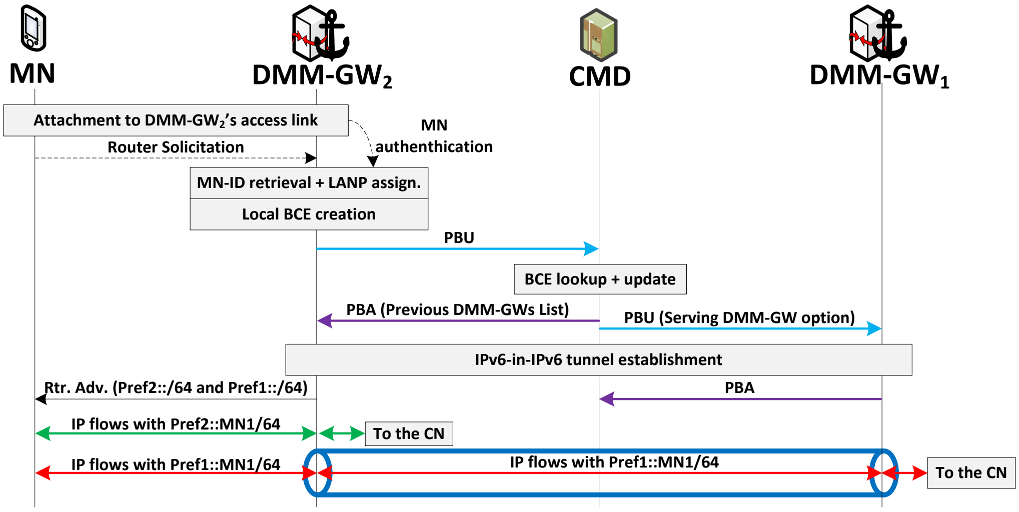

MN
DMM-GW$_2$
CMD
DMM-GW$_1$
Attachment to DMM-GW$_2$'s access link
Router Solicitation
MN authenthication
MN-ID retrieval + LANP assign.
Local BCE creation
PBU
BCE lookup + update
PBA (Previous DMM-GWs List)
PBU (Serving DMM-GW option)
IPv6-in-IPv6 tunnel establishment
Rtr. Adv. (Pref2::/64 and Pref1::/64)
PBA
IP flows with Pref2::MN1/64
To the CN
IP flows with Pref1::MN1/64
IP flows with Pref1::MN1/64
To the CN

Figure 5.7: Handover management in the "proxy" solution.

time, the CMD sends a PBU with the Serving DMM-GW option to all the previous DMM-GWs in parallel, notifying them about the new MN location. In this way, all the involved DMM-GWs can almost simultaneously establish the required tunnels and routing entries on their side. Each previous DMM-GW, after completing the update, sends a PBA message to the CMD to indicate that the operation is concluded and the state has been updated. This scheme is depicted in Fig. 5.7.

De-registration As for the previous schema, the de-registration mechanism follows the message sequence designed for the handover management. The global de-registration is realized through a zero lifetime PBU/PBA handshake between the serving DMM-GW and the CMD, and this latter in turns notifies the rest of active DMM-GWs with another PBU/PBA exchange.

Similarly, when a local de-registration is triggered, the DMM-GW originating the request performs a PBU/PBA signaling with the CMD that eventually informs the serving DMM-GW as well with another PBU/PBA session.

5.2.4 Comparison of the three partially distributed solutions

The three partially distributed solutions described before are identical to each other for the initial registration phase, that accounts for a single signaling session between the serving DMM-GW and the CMD, hence consisting in two control messages.

As part of the common set of features among the three solution, there is also the Serving DMM-GW mobility option. This option is sent by the CMD to all the DMM-GWs actively involved with MN's IP flows after a new DMM-GW happens to detect the MN attachment, and it is therefore used to announce a new serving DMM-GW.

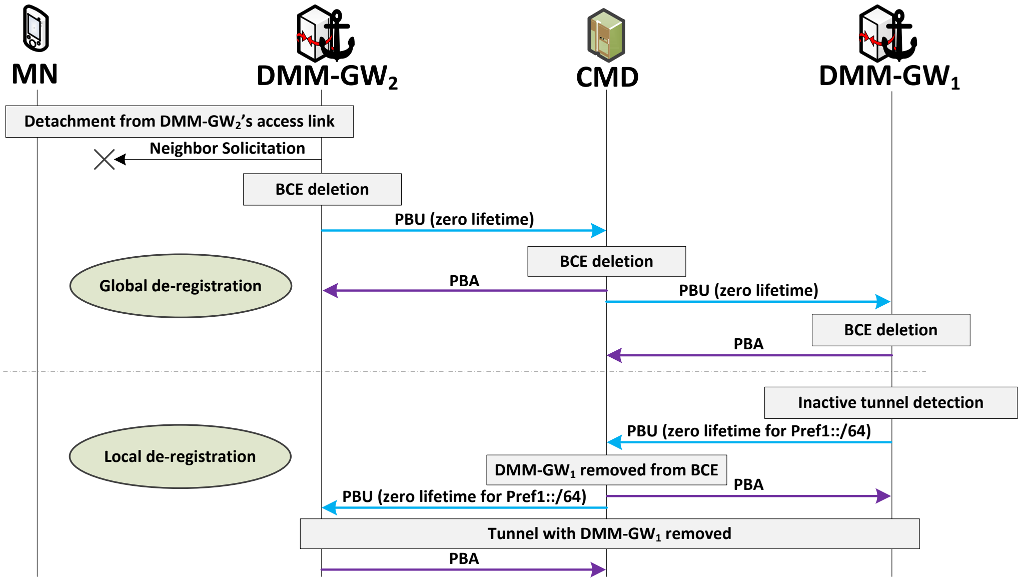

MN
DMM-GW$_2$
CMD
DMM-GW$_1$
Detachment from DMM-GW$_2$'s access link
Neighbor Solicitation
BCE deletion
PBU (zero lifetime)
BCE deletion
Global de-registration
PBA
PBU (zero lifetime)
BCE deletion
PBA
Inactive tunnel detection
PBU (zero lifetime for Pref1::/64)
Local de-registration
DMM-GW$_1$ removed from BCE
PBA
PBU (zero lifetime for Pref1::/64)
Tunnel with DMM-GW$_1$ removed
PBA

Figure 5.8: De-registration in the “proxy” solution.

The other mobility option introduced by the partially distributed approach is the Previous DMM-GW option, that bundles an item from the Previous DMM-GWs List contained in the CMD’s BCE. This option is appended to the PBA sent by the CMD to the new serving DMM-GW and there might be multiple instances. The “locator” solution does not make use of such option, since the CMD does not deliver any PBA message to the serving DMM-GW. On the contrary, the “locator” solution introduces a control plane link between DMM-GWs to transfer such PBA, that is otherwise not necessary in the other two schema.

After the signaling sequence for each design, the number of control packets transmitted for the handover management is $2+2n$ for the “relay” and “proxy” solutions, being n the number of DMM-GWs that are anchoring MN’s IP flows at the handover time, that is, the number of MN’s active LANPs prior to handover. Similarly, in the “locator” solution the number of messages scales as $1+3n$. Regarding the de-registration signaling cost, being n the number of MN’s active LANPs, the number of messages exchanged for the global procedure is $2n$ for the “relay” and “proxy” solutions, while are necessary $1+3(n-1)$ messages for the “locator” scheme. In the local de-registration procedure there are 4 messages for every solution. This study is exhaustively covered in Section 7.2.

Focusing on the handover management, which can be regarded as the most critical from the user’s perspective, the “relay” and “proxy” solutions share the same signalling path and number of control messages with the sole difference that message order is permuted so that the “proxy” speeds up the operations. Indeed, in the “proxy” case, all the DMM-GWs, including the new one that is detecting the MN attachment, are expected to be updated almost simultaneously. The expected converge time for the “locator” solution stands in between the other two solutions. More on this analysis is described in Section 7.4.

Although faster, the drawback of the “proxy” approach is that it is not efficiently protected against a signaling failure. Let’s take as an example the case in which the CMD receives a PBU from a new serving DMM-GW but, for any reason, it fails forwarding it to an old DMM-GW, after a number of attempts specified for fault protection. In the “relay” scenario, the CMD would then exclude that DMM-GW from the Previous DMM-GWs list sent over to the new serving DMM-GW: the IP flows anchored at that old DMM-GW would be disrupted but no resources in terms of processing and routing/tunnel setup would be required at the serving DMM-GW. On the contrary, in the “proxy” case, the unreachable old DMM-GW would be included in the list passed to the new serving DMM-GW, so that the upstream tunnel is created, but still the IP flows could not be recovered, resulting in a waste of resources and in an inconsistent configuration. In this latter case, the CMD, after receiving no responses from the unreachable old DMM-GW within a guard interval, should issue a message to the new serving DMM-GW to withdraw the information announced in the previous PBA. A similar inefficient handling occurs in the DMM-GW to DMM-GW control interface of the “locator” solution, as there is not a direct update/acknowledgement mechanism.

5.3 Control plane management: the fully distributed approach

In the fully distributed approach, the data plane follows the description provided in Section 5.1 and illustrated in Fig. 5.1, hence the prefix assignment and routing configuration concepts still hold. Nevertheless, the control plane no longer relies on a centralized entity that maintains a global view of the MN mobility sessions, but, instead, the control interactions take place among the DMM-GWs only. Therefore, the key point is how the new serving DMM-GW finds out if the attached mobile node has any DMM-GW anchoring active flows, and, if so, which IPv6 prefixes they assigned. Once these notions are provided, the new serving DMM-GW may establish the new routing paths with the other DMM-GWs without the involvement of other entities.

For instance, in Section 3.3 we already introduced some peer-to-peer (P2P) DMM solutions. By employing a distributed hash table (DHT) to implement the binding cache, such paradigm perfectly reflects the fully distributed approach. The operations, signaling and performance of such a solution is bound to the particular P2P protocol adopted. Given the amount of works devoted to this topic, we do not dig further. Apart from the P2P paradigm, here are some alternative mechanisms:

- Multicasting a PBU message from the new DMM-GW to the group formed by all the other DMM-GWs of the domain (or a subset of them based on the operator's policies). In case no answer (i.e., a PBA message) is received within a timeout interval, the DMM-GW may assume this is the first time the MN joins the network. Otherwise, those DMM-GWs that reply set a tunnel up with the DMM-GW that originated the notification. Unfortunately this approach is not regarded as efficient due do the flooding of signaling messages. It also might not provide a good performance in terms of handover delay as the originating DMM-GW does not know a priori how many DMM-GWs are anchoring MN's IP flows.

- Retrieving the necessary information directly from the MN, through extensions of the Neighbor Discovery protocol, or through dedicated signaling like the Node Information Queries protocol [91] or a custom one. This is an efficient mechanism to speed up the handover operations, and also to provide mobility only to those prefixes that the MN exports in the signaling. Nevertheless, it impacts the terminal supported actions, which does not grant backward compatibility or affects the host's IP stack.
- Employing Layer-2 handover support through Media Independent Handover Services specification (IEEE 802.21) [27]. The latest revisions of IEEE 802.11 and IEEE 802.16 wireless technologies already provide support to the so-called *link layer events*. Through these mechanisms, a network interface is able to indicate changes in e.g., point of attachment or re-connection. Therefore, a handover is handled by a dedicated control plane infrastructure by which the movement is prepared, executed and completed in a controlled and assisted way, according to the *make-before-break* philosophy. Additionally, the IEEE 802.21 suite is intended to allow inter-technology handovers, providing support to mobile nodes roaming within a heterogeneous environment (see Section 2.3).

According to the considerations reported in the list above, and to the originality of the solutions, we have developed a fully functional solution only for the last category, employing the IEEE 802.21 protocol suite.

5.3.1 Distributed Mobility Management with MIHS

In the Media Independent Handover Services architecture, mobility management might be performed as the superposition of MIHS and an IP mobility protocol, such as MIPv6 or PMIPv6. For instance, the MIHS specifications document [27] provides a fully functional use case scenario where a mobile terminal switches between WiFi and WiMAX access technologies and the IP mobility support is given by PMIPv6. We also describe an MIHS+PMIPv6 implementation study case in Section B. Taking such use case scenario as reference, we have extended it in order to operate with an N-DMM protocol, yielding to the solution presented in the following.

The MIHS suite provides powerful mechanisms to a user-space application to manage the system's network interfaces. In the MIHS terminology, such application is called MIH-

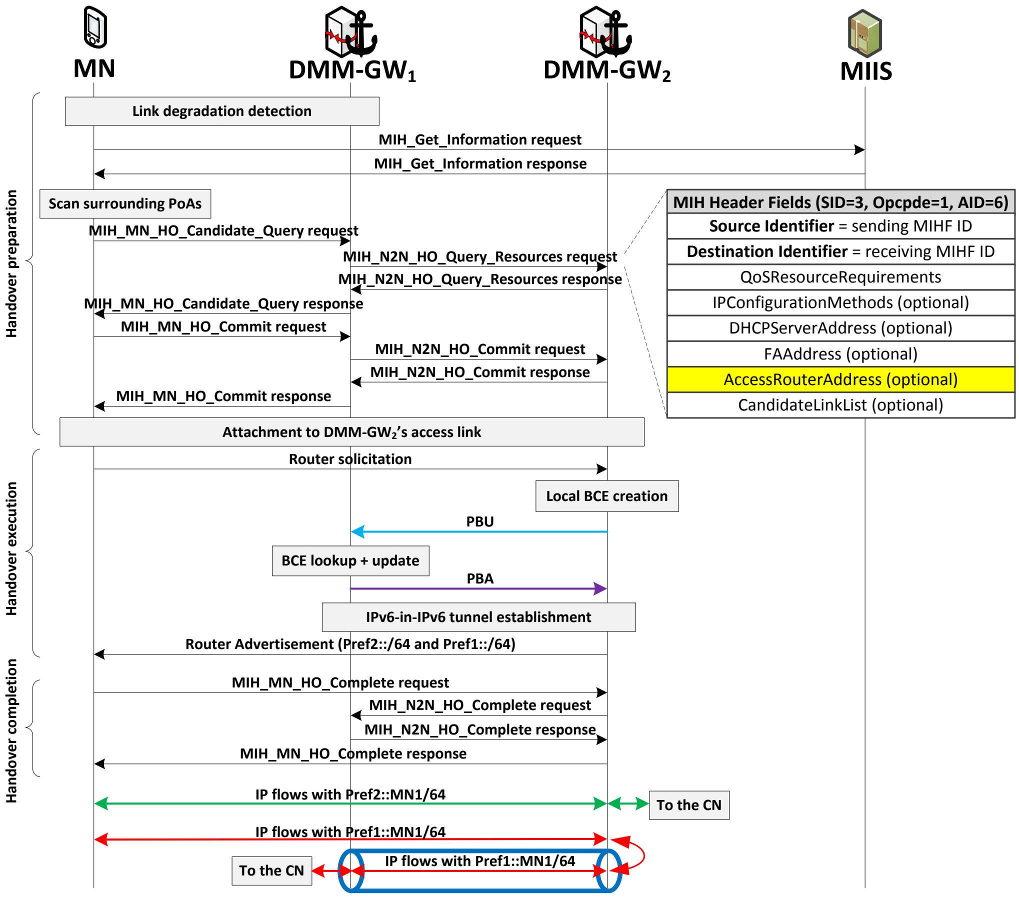

MN
DMM-GW₁
DMM-GW₂
MIIS
Handover preparation
Link degradation detection
MIH_Get_Information request
MIH_Get_Information response
Scan surrounding PoAs
MIH_MN_HO_Candidate_Query request
MIH_N2N_HO_Query_Resources request
MIH_N2N_HO_Query_Resources response
MIH_MN_HO_Candidate_Query response
MIH_MN_HO_Commit request
MIH_N2N_HO_Commit request
MIH_N2N_HO_Commit response
MIH_MN_HO_Commit response
MIH Header Fields (SID=3, Opcpde=1, AID=6)
Source Identifier = sending MIHF ID
Destination Identifier = receiving MIHF ID
QoSResourceRequirements
IPConfigurationMethods (optional)
DHCPServerAddress (optional)
FAAddress (optional)
AccessRouterAddress (optional)
CandidateLinkList (optional)
Attachment to DMM-GW₂'s access link
Handover execution
Router solicitation
Local BCE creation
PBU
BCE lookup + update
PBA
IPv6-in-IPv6 tunnel establishment
Router Advertisement (Pref2::/64 and Pref1::/64)
Handover completion
MIH_MN_HO_Complete request
MIH_N2N_HO_Complete request
MIH_N2N_HO_Complete response
MIH_MN_HO_Complete response
IP flows with Pref2::MN1/64
To the CN
IP flows with Pref1::MN1/64
To the CN
IP flows with Pref1::MN1/64

Figure 5.9: Fully distributed N-DMM: IEEE 802.21-aided message exchange sequence during handover.

user, and we designed two separate MIH-users to run in the MN and in the DMM-GW, called respectively Connection Manager (CM) and Flow Manager (FM).

The CM can be seen as an MIH-enabled extension of the connection manager that is currently available in most of the operating systems of modern hand-held mobile devices, like Android, iOS or Windows Mobile, and it does not necessarily impact the IPv6 stack implementation.

The FM operates as the network side termination for the MIH signaling with the MN, being thus the MN's Point of Service (PoS), and it is also the trigger for the DMM actions performed by the N-DMM protocol. We proceed next to the details for *i)* the initial registration, *ii)* the handover management and *iii)* the de-registration procedures.

Initial registration. At terminal bootstrap, the CM picks the default access technology to establish a radio link with the corresponding DMM-GW, and then it obtains IP connectivity with standard SLAAC. To do so, the DMM-GW assigns a LANP from its prefixes pool to the MN, and stores the MN-ID/LANP pair in a BCE stored locally. The attachment detection and the MN-ID retrieval can be performed using one of the methods proposed at the beginning of Section 5.2. Once the IP connectivity is set up, the CM performs the required MIH subscriptions at the DMM-GW's PoS, i.e., the FM. The MN is finally registered at the domain.

Moreover, the DMM-GW refreshes periodically the registration issuing ND messages to the MN, and, on top of that, the FM monitors the MN activity based on the IP flows traversing the node.

Handover management. The mobility procedures for handover handling require a deep involvement of the MIH signaling in order to prepare, execute and complete the handoff. Fig. 5.9 presents the detailed procedure, including the IEEE 802.21 signaling required to perform a fully distributed network-based handover. In the figure, only the DMM-GWs are shown, whilst the Points of Attachment (PoA) are omitted to keep the chart simple. We consider for simplicity only the mobile-initiated handover, being the network-initiated handover similar.

The handover procedure starts when the MN detects that the link quality is going down. The MN queries the Media Independent Information Service (MIIS) server for a list of possible candidate DMM-GWs and next scans the radio environment looking for them. The scan results are sent to the serving PoS residing in the current DMM-GW (`MIH_MN_HO_Candidate_Query request`) that in turns negotiates the resources with all the DMM-GW indicated in the results (`MIH_N2N_HO_Query_Resources request`). In Fig. 5.9 we have drawn for simplicity only one DMM-GW, but there might be more, yielding to multiple parallel message exchanges. The `MIH_N2N_HO_Query_Resources request` message is important for the DMM operations because it carries the IPv6 address of the querying DMM-GW (i.e., the DMM-GW where the the MN is currently connected to), in the `AccessRouterAddress` option (marked in yellow in Fig. 5.9). The MN is next reported which DMM-GW can accommodate the new radio link, and the MN commits to attach to that target (`MIH_MN_HO_Commit request`). When the response is received, the MN establishes the new radio link with the target DMM-GW. The stage up to this moment is called *handover preparation.* Once the link is up, the standard SLAAC is procedure is started, so that the DMM-GW can assign a new LANP to the MN and register it. In addition, the FM triggers the PBU/PBA message exchange with the DMM-GW where the MN was previously connected, so that the IP tunnel is created for the traffic redirection of old flows. This stage, known as *handover execution*, is concluded when the MN receives a router advertisement for the old prefix, granting the usability of it also in the new access network. The concluding stage is called *handover completion*, and it consists in releasing the resources at the previous DMM-GW (see the group of `MIH_MN_HO_Complete` and `MIH_N2N_HO_Complete` messages at the end of the chart).

At this stage, the new FM is observing two groups of IP flows attributed to the MN:

one group with the prefix announced by the current DMM-GW, and one with the prefix announced by the old DMM-GW. If both prefixes are "active" at the time a subsequent handover occurs, the future target DMM-GW must establish a PBU/PBA handshake, and consequently a tunnel, with both the DMM-GWs anchoring such prefixes. For this purpose, the current DMM-GW needs to communicate to the future target DMM-GW the old DMM-GW's address beyond its own, and this task might be accomplished including another instance of the `AccessRouterAddress` option in the `MIH_N2N_HO_Query_Resources request` message introduced before.

De-registration. As already described for the partially distributed DMM solutions, the DMM-GW that is currently serving the MN periodically issues ND messages to assess the MN presence at the access link. If the ND attempt fails, then the DMM-GW deletes the local MN registration and propagates the de-registration to the rest of DMM-GWs anchoring the MN's IP flows.

However, in the MIH-DMM fully distributed solution, the flow manager constantly monitors the MN's prefixes activity, so when the serving DMM-GW detects that a prefix is no longer active, it might directly trigger the de-registration with the corresponding anchor DMM-GW.

5.4 Security considerations

Secure mobility signaling is mandatory for all mobile networks, in order to guarantee that the information exchanged are authentic and no malicious nodes are trying to exploit breaches to attack the network entities or the rest of MNs. To this extent, in the description devoted to C-DMM, we have examined a security system for the mobility signalling between the MN and the DMM-GWs in order to enhance the performance of C-DMM with respect to a straightforward application of the legacy Mobile IPv6 security mechanism.

In this chapter we have not posed this problem, as there is no need to improve the system envisioned by PMIPv6. In PMIPv6, the LMA/MAG signaling is protected using IPsec, and so can be done in all the N-DMM solutions, applying IPsec to the DMM-GW–to–DMM-GW and DMM-GW–to–CMD interfaces. Indeed, the security associations among the entities can be statically deployed by the MNO when building the infrastructure, so it is not necessary to devise any security enhancement.

Nevertheless, in Section 5.2.1 we have suggested how to perform user authentication and authorization, along with the MN-ID fetching, and those mechanisms hold for any N-DMM protocol. We stress that from a performance perspective, it is more convenient to combine the attachment detection with the security system deployed on the access link, so that the DMM-GW obtains the MN-ID already contained in the attachment notification.

Nevertheless, for a more general and link layer independent solution, the two processes should be decoupled and performed separately.

5.5 Final remarks

In this chapter we have explored two approaches to deploy a network-based DMM architecture, namely the *partially distributed* approach and the *fully distributed* approach. Both of them bring the advantages described for the C-DMM solution in Section 4.4, and, in addition, no action is supposed to be taken by the MN, including the fact that no tunnel overhead is consumed in the over-the-air link.

A fully distributed approach, although perfectly feasible, requires a more complicated intervention as compared to the partially distributed one. Indeed, it might require support from the mobile nodes, or the deployment of a whole control infrastructure (as in the case of IEEE 802.21). This might be not desirable, but the deployment of such an architecture would yield to a more scalable and bottleneck-free operator infrastructure, where no single point of failure could bring the network down. We analyze the scalability of the N-DMM solutions in Chapter 7.

A mature N-DMM implementation is available at the ODMM project, under the name of Mobility Anchors Distribution for PMIPv6 (MAD-PMIPv6). MAD-PMIPv6 implements the partially distributed solution in the "CMD as proxy" flavor. Similarly, the fully distributed DMM solution with the MIHS suite has been implemented for the MEDIEVAL project. The description of the above experimental platforms is the object of Chapter 8, whereas the next proposes a hybrid design formed by a combination of a C-DMM and an N-DMM protocol to handle inter-domain mobility.

Chapter 6

Hybrid DMM solution

In the previous chapters we have examined how to use a client-based or a network-based DMM solution as a standalone strategy to provide mobility support. However, we argue that future mobile network operators can benefit from a framework allowing a seamless integration of solutions from both categories [92].

Actually, a dual mobility support is envisioned also by current mobile network architectures. Indeed, they typically support both network and client-based *centralized* mobility solutions, offering in this way more flexibility to the operator according to the mobility scenario. For instance, according to the 3GPP's EPS, an MNO might employ GTP for mobility in the home network with 3GPP access, PMIPv6 when the user is connected to a *trusted non-3GPP access* network and Dual Stack MIPv6 (DSMIPv6)[1] [93] for an MN from a *non-trusted non-3GPP access*.

Typically, network-based mobility protocols are used in the user's home network, that is, when moving within the network infrastructure provided by the MNO the user has subscribed to. From the technical point of view, all the mobility signaling must be secured, and an MNO can apply such security system only to the network entities that form its infrastructure. Also, this is justified by the global control that an MNO has on the users' subscriptions database, necessary to authenticate the MNs and to authorize them to connect to the domain. On the contrary, a client-based solution is adopted when the user connects to a visited network, the so-called *roaming* scenario, as the MNO cannot control the behavior of the visited network entities. This is for instance the case in which an MN attaches to the cellular network from another operator, in the same country or abroad, or attempts to connect to the EPS through the mentioned non-trusted non 3GPP access.

In all cases, the actual mobility support deployment heavily depends on the policies and commercial agreements stipulated by the MNOs of the visited networks with the home MNO. In fact, operators usually do not openly share with other MNOs their

[1] Simply put, DSMIPv6 collects the specifications to ensure IPv4 compatibility to MIPv6.

users' subscription database, which is unfortunately mandatory to enable fully transparent network-based mobility between home and visited networks. In parallel, visited MNOs do not always let user's traffic to traverse their core infrastructure, therefore roaming users' traffic is routed to the home gateways.

Therefore, a Hybrid DMM (**H-DMM**) architecture basically takes these concepts one step further, by proposing a combined usage of network and client-based mobility protocols from the Distributed Mobility Management framework.

We consider inter-domain mobility as the representative scenario of combined H-DMM deployment, and use it in this chapter to explain how our proposed hybrid scheme works.

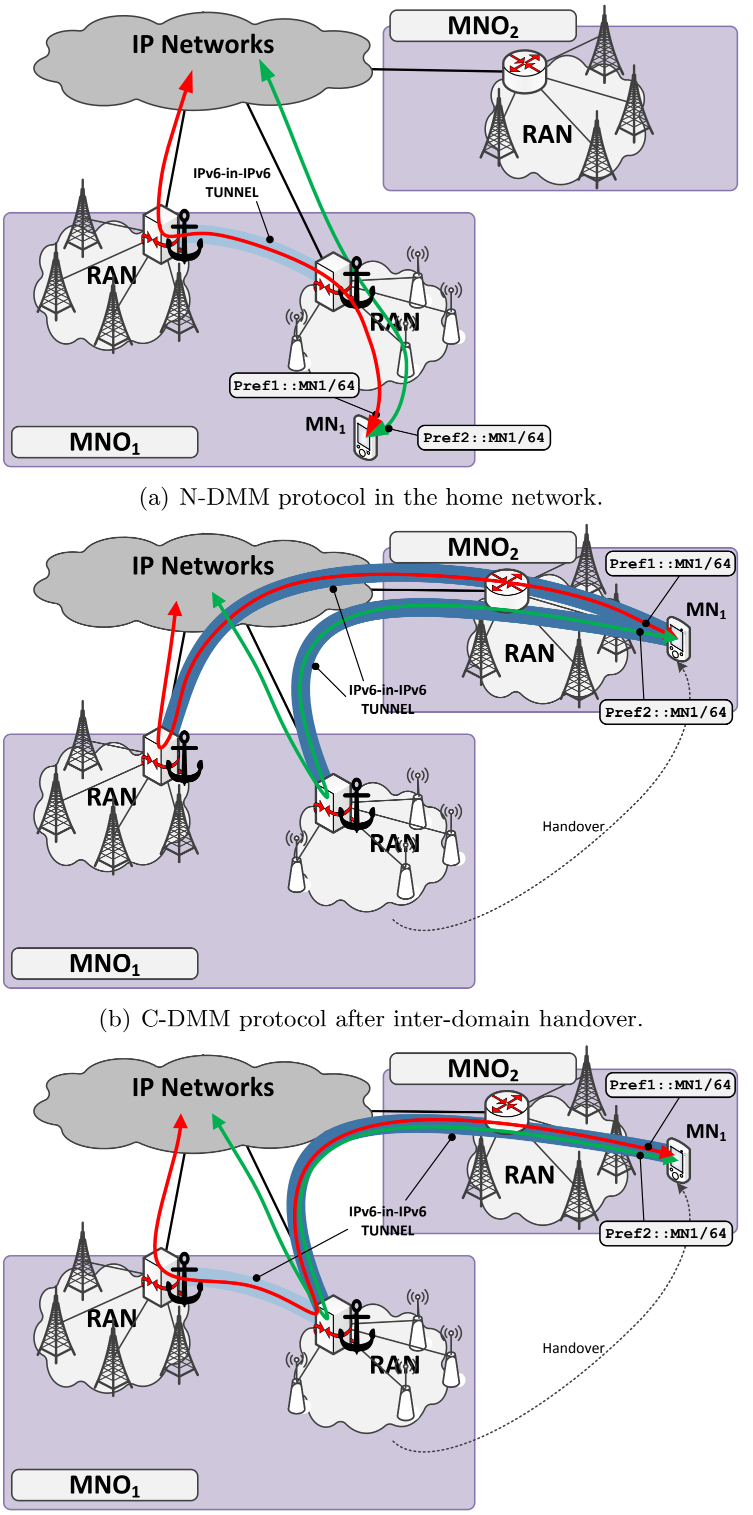
6.1 HDMM: overview

We assume that as long as a mobile node is attached to its home network, it benefits from network-based distributed mobility management (by using either the partially or the fully distributed variant described in Chapter 5).

While roaming within the same operator network, the security associations required among the involved distributed anchor routers can be easily set up (on demand or can be already pre-configured). If the mobile node moves to an access network managed by a different operator, the new operator might not even support a DMM-like mobility solution (i.e., there are no DMM-GWs deployed) or, if a DMM protocol is supported, setting up security associations that cross operator boundaries might not be possible. In both cases, using a client-based DMM approach appears as the best possible solution to provide those sessions anchored at the previous domain with session continuity. H-DMM supports this by activating the client-based component of the solution, and using the IP address configured on the new domain as care-of address where active sessions anchored elsewhere can be redirected. This operation can be executed exploiting enhanced features of the terminal's connection manager.

The connection manager is a software construct widely available in most of today's portable devices. In the last years, and due to the availability of different networks where the mobile nodes can connect to, this piece of software has gained quite a lot of relevance. The connection manager, upon detection that the target point of attachment does not belong to the home operator or it does not support any DMM flavor, can activate the mobility client at the mobile node.

Let's consider the examples shown in Fig. 6.1. A mobile node has been moving within its DMM-enabled home network operator's domain (MNO_1). This domain employs a network-based DMM protocol. Fig. 6.1(a) illustrates an initial scenario where we depicted two MN's IP flows: one uses the LANP assigned by the DMM-GW where the MN is currently connected, and the other is an old IP flow handed over by an old DMM-GW to the current DMM-GW by means of the N-DMM protocol.



(a) N-DMM protocol in the home network.

(b) C-DMM protocol after inter-domain handover.

(c) C-DMM protocol after inter-domain handover with tunneling cascade.

Figure 6.1: H-DMM architecture and example scenarios.

When the mobile node performs an inter-operator hand-off, there are three possible solutions:

i) the connection manager of the mobile node has tracked all the active DMM-GWs (i.e., all the DMM-GWs visited by the MN in the home dome domain anchoring a LANP used by an active session),

ii) the connection manager is only aware of the last DMM-GW it was attached to before roaming to a new domain,

iii) the connection manager knows none of them.

Cases *i)* and *ii)* require the DMM-GWs to include their global IPv6 address along with the Router Advertisement message sent to the MN upon attachment. In case *i)*, the mobile node can just follow a plain client-based DMM approach and update each of the DMM-GWs with its current location by sending a BU message to them. Note that this results in creating dedicated tunnels for data traffic between each of the DMM-GWs and the current location of the mobile node[2]. This approach is shown in Fig. 6.1(b) and the corresponding signaling sequence in the upper part of Fig. 6.2.

In case *ii)*, the mobile node only updates its location on the last visited DMM-GW, and sets up a single tunnel only with it. This DMM-GW is already receiving the redirected traffic from the rest of active DMM-GWs through a tunnel established with them, and builds another tunnel with the MN to deliver the packets to destination (see Fig. 6.1(c) for the network scenario and the upper part of Fig. 6.2 for the signaling sequence). This approach requires less complexity on the connection manager, but does not fully optimize the data path between the active DMM-GWs and the current location of the mobile node (i.e., packets have to traverse a chain of two tunnels in cascade: one from the MN to the last visited DMM-GW, and one from this DMM-GW to the one associated to the LANP in use).

The case *iii)* is the most complex, because the MN needs to discover a mobility support entity in the home network, e.g., a CMD or a home agent, using DNS or by looking at a preconfigured entry locally stored. The signaling sequence for this case is depicted in Fig. 6.3. There can be then different possible scenarios according to which network-based protocol is used in the home network. For instance, with a partially distributed solution, when the MN moves to a foreign access network it should be forced to initiate a CMD discovery process. Then it activates mobility support with a trigger sent to the CMD. The CMD reacts almost with the same operations, as it communicates to the MN the previous DMM-GWs and the LANPs that were active prior to handover, and, in parallel, it commands the previous DMM-GWs to redirect the IP flows to the MN's CoA through a tunnel. Next handovers may follow this same procedure or a plain C-DMM application.

[2] This does not optimize the overall route between the mobile node and the peers it is communicating with. This fully optimized route can be achieved if the communication peer supports the correspondent node Mobile IPv6 route optimization (RO) functionality.

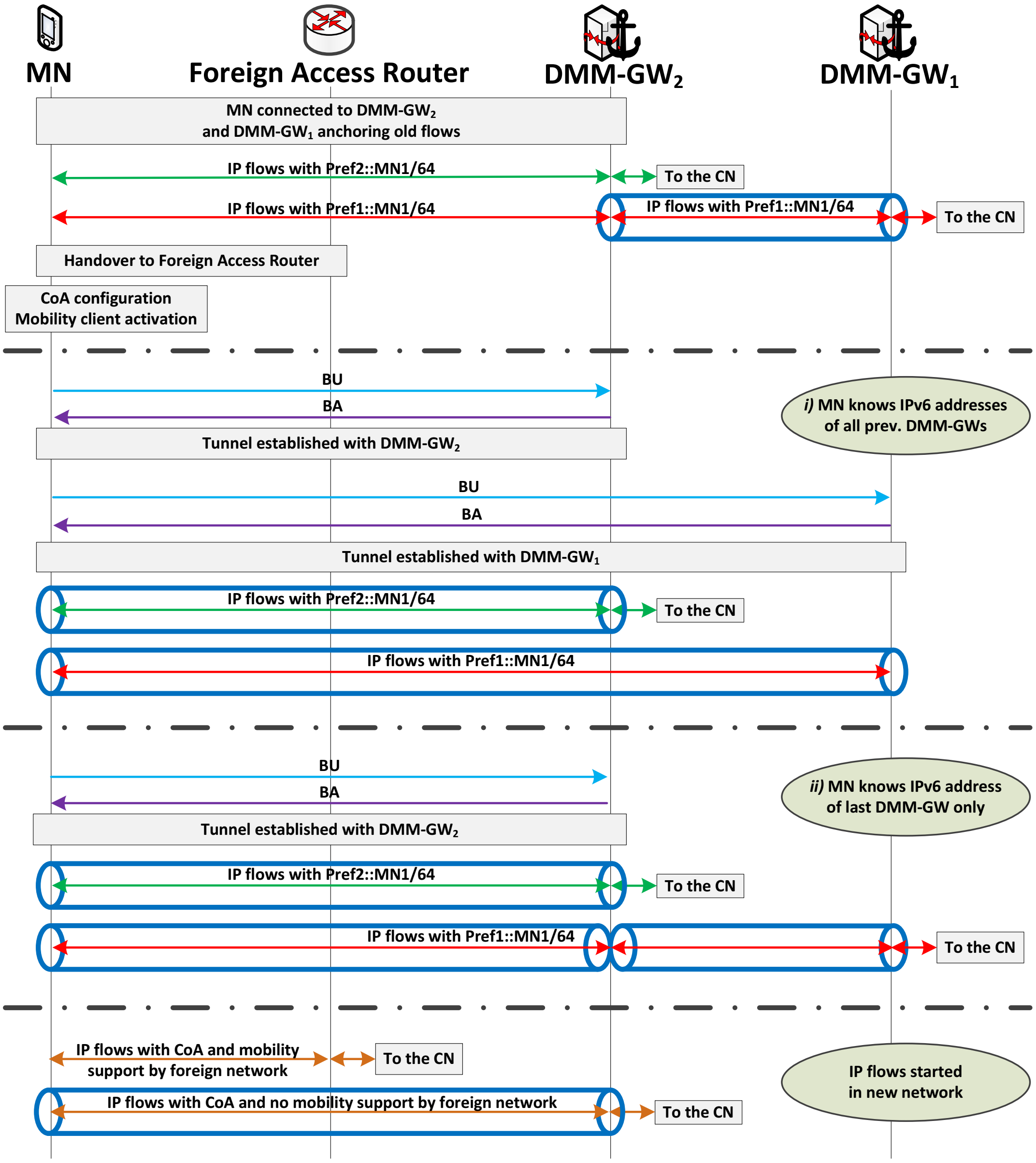

MN
Foreign Access Router
DMM-GW$_2$
DMM-GW$_1$
MN connected to DMM-GW$_2$ and DMM-GW$_1$ anchoring old flows
IP flows with Pref2::MN1/64
To the CN
IP flows with Pref1::MN1/64
IP flows with Pref1::MN1/64
To the CN
Handover to Foreign Access Router
CoA configuration
Mobility client activation
BU
BA
i) MN knows IPv6 addresses of all prev. DMM-GWs
Tunnel established with DMM-GW$_2$
BU
BA
Tunnel established with DMM-GW$_1$
IP flows with Pref2::MN1/64
To the CN
IP flows with Pref1::MN1/64
BU
BA
ii) MN knows IPv6 address of last DMM-GW only
Tunnel established with DMM-GW$_2$
IP flows with Pref2::MN1/64
To the CN
IP flows with Pref1::MN1/64
To the CN
IP flows with CoA and mobility support by foreign network
To the CN
IP flows with CoA and no mobility support by foreign network
To the CN
IP flows started in new network

Figure 6.2: Signaling sequence for handoff to foreign network in cases *i)* and *ii)*.

If the mobility support in the home network follows the fully distributed approach, then there are no central databases that the MN can directly contact to recover the mobility support. Nevertheless, the MN may still discover a mobility anchor in the home network, and establish a mobility session with it. In this event, the assigned mobility anchor would operate as a MIPv6 home agent, and a mechanism must be deployed to enable the MN mobility context acquisition from the active DMM-GWs. However, the integration of a CMM and a DMM protocol is not covered in the thesis, which is indeed a possible research area for future work (see Chapter 10).

Note that, in all cases, in order to have mobility support for the IP flows started in the foreign network, either the new domain provides it to the user, or the MN keeps using the address(es) from the LANP(s) configured while at home. For instance, if the new domain also supports H-DMM, and it is granted access to the home users authorization database, then subsequent handovers within that domain could be transparently managed by the network-based mobility solution in place, without requiring any action on the client-mobility stack running on the mobile node[3].

[3] In this case, the mobile node could actually decide if it prefers to update the care-of address used in the bi-directional tunnels established with DMM-GWs located at the other domain, or just let the network-based distributed mobility support deployed in the new domain provide address continuity to the care-of address used to set-up the tunnels.

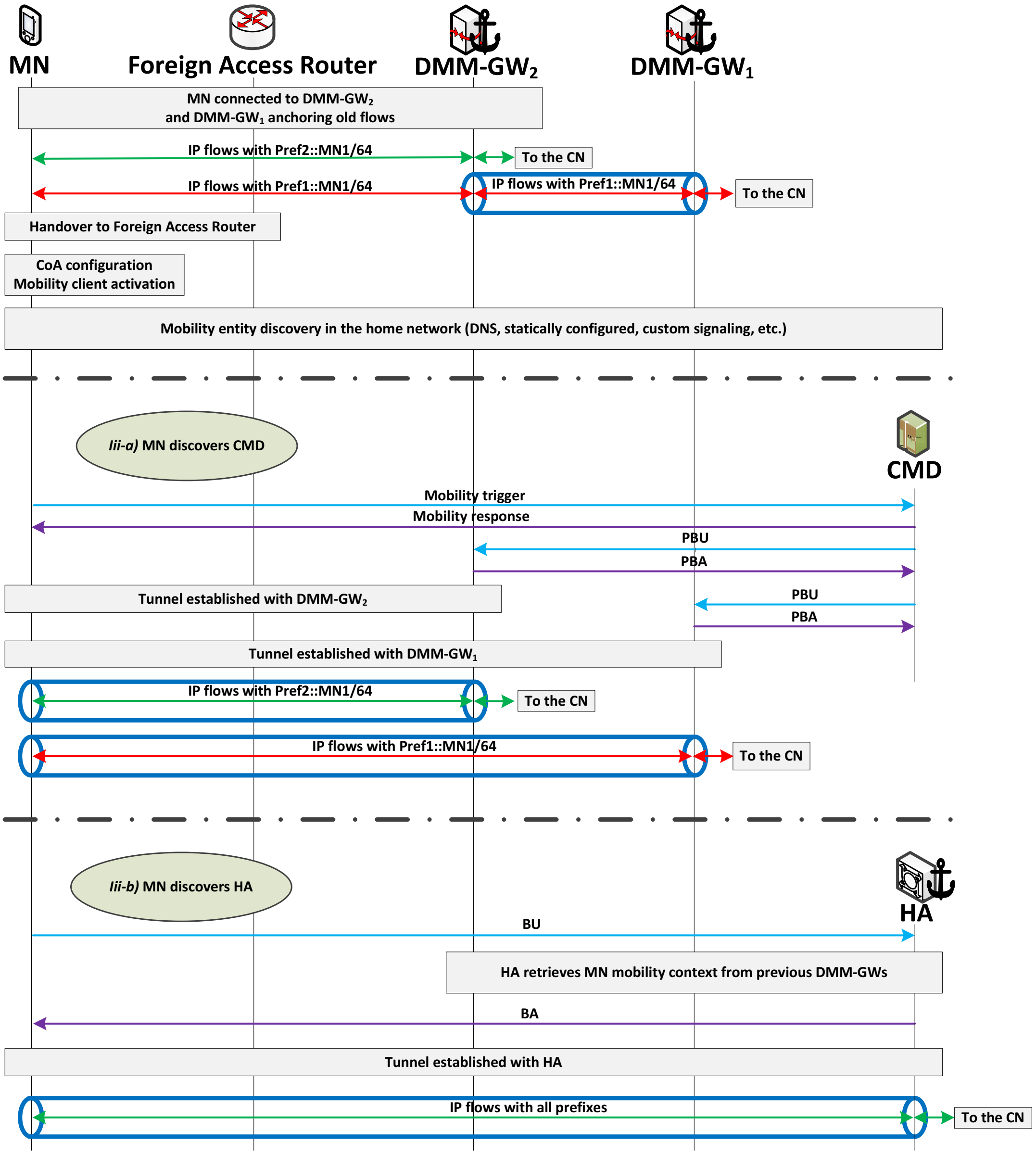

MN
Foreign Access Router
DMM-GW$_2$
DMM-GW$_1$
MN connected to DMM-GW$_2$ and DMM-GW$_1$ anchoring old flows
IP flows with Pref2::MN1/64
To the CN
IP flows with Pref1::MN1/64
IP flows with Pref1::MN1/64
To the CN
Handover to Foreign Access Router
CoA configuration
Mobility client activation
Mobility entity discovery in the home network (DNS, statically configured, custom signaling, etc.)
Iii-a) MN discovers CMD
CMD
Mobility trigger
Mobility response
PBU
PBA
Tunnel established with DMM-GW$_2$
PBU
PBA
Tunnel established with DMM-GW$_1$
IP flows with Pref2::MN1/64
To the CN
IP flows with Pref1::MN1/64
To the CN
Iii-b) MN discovers HA
HA
BU
HA retrieves MN mobility context from previous DMM-GWs
BA
Tunnel established with HA
IP flows with all prefixes
To the CN

Figure 6.3: Signaling sequence for handoff to foreign network in case *iii)*.

6.2 Final remarks

The previous use case clearly shows how each DMM-GW can be simultaneously playing – on a prefix basis – the roles of plain IPv6 access router (for prefixes locally anchored used by attached mobile nodes), as local mobility anchor (for prefixes locally anchored that are in use by mobile nodes which are no longer directly attached), as mobile access gateway (to enable address continuity for prefixes anchored at a different DMM-GW) and as home agent (for locally anchored prefixes used by mobile nodes which are no longer directly attached and that are using the C-DMM component).

Next chapter covers the scalability analysis of all the DMM protocols presented so far.

Chapter 7

Analytical evaluation

In this chapter we analyze the costs in terms of signaling overhead, packet delivery and handover latency of our proposed solutions, and we compare them with those of MIPv6 and Proxy Mobile IPv6. This type of cost analysis has received a lot of attention in the recent past, starting during the development of cellular networks [94–97], and then moving to the handover and mobility management in IP-based networks [98, 99].

In the following, we introduce the mobility and traffic model used in our analysis, to then derive the signaling, packet delivery and handover latency cost functions for both CMM and our DMM solution. The notation used throughout this section is summarized in Table 7.1.

7.1 User Mobility and Traffic Models

In an IP mobility protocol, a location update takes place whenever the MN passes from one subnet to another, that is, between two networks handled by different access routers, or, in the DMM case, by two DMM-GWs. This phenomenon is shared by all the IP mobility protocols, hence the first step is to characterize users' mobility as a common ground to develop an analytical study for each protocol.

In general, a node mobility is modelled as the time T_i spent in a cell i before moving to the next one (this is also referred to as *cell residence time*), and its related probability density function (PDF) f_{T_i}. In the following, we consider the worst case in which each cell is a different IP-subnet (SN). We assume that the wireless network is composed of statistically identical cells, so that T_i can be expressed by T_{SN} for all the cells, and with PDF:

$$f_{T_{SN}} = f_{T_i}\,,\ \forall i\,. \tag{7.1}$$

A simple method that has been extensively used in the past to characterize the statistical average residence time is known as *Fluid Flow model* [94, 100]. With such model[1],

[1] Note that the analytic evaluation performed in this section is not limited to using the Fluid Flow

Table 7.1: Notation.

$f_X(t)$, $f^*_X(s)$	prob. density function of X, its Laplace transform
T_{SN}	subnet residence time
μ_{SN}	subnet crossing rate
T_{LANP}	active prefix lifetime
$\overline{T}_{LANP}$	mean value of T_{LANP}
T_H	active prefix lifetime while the MN is attached to the prefix home network (prefix used as LANP)
T_F	active prefix lifetime while the MN is visiting a foreign network (the prefix is used as pLANP)
$1/\lambda_F$	mean value of T_F
$\overline{N}_{LANP}$	average number of used active prefixes
$\overline{N}_{pLANP}$	average number of active anchored prefixes
$\alpha(K)$	probability that K handovers occur during the interval T_F
τ	cost per packet for tunnel transmission
ω	cost per packet for wireless transmission
λ_p	packet transmission rate per active prefix
A_{SN}, L_{SN}	area and perimeter of a subnet
R_{SN}	radius of a circular subnet
$\overline{v}$	average speed of mobile node
T_{BCE}	BCE lifetime
R_{BCE}	rate of BCE refresh operations

the MN travels within a subnet with a direction uniformly distributed in $[0, 2\pi)$ and average speed $\overline{v}$. If the subnet has an area A_{SN} and a perimeter L_{SN}, then the subnet border crossing rate μ_{SN} is given by:

$$\mu_{SN} = \overline{v}\frac{A_{SN}}{\pi L_{SN}} = \frac{2\,\overline{v}}{\pi R_{SN}}\,, \tag{7.2}$$

where the second equation is obtained assuming circular subnets with radius R_{SN}. Consequently, the mean subnet residence time is given by the inverse ratio of the border crossing rate:

$$\mathrm{E}[T_{SN}] = 1/\mu_{SN}\,. \tag{7.3}$$

Model, but it is valid for any mobility model with statistically identical cells.

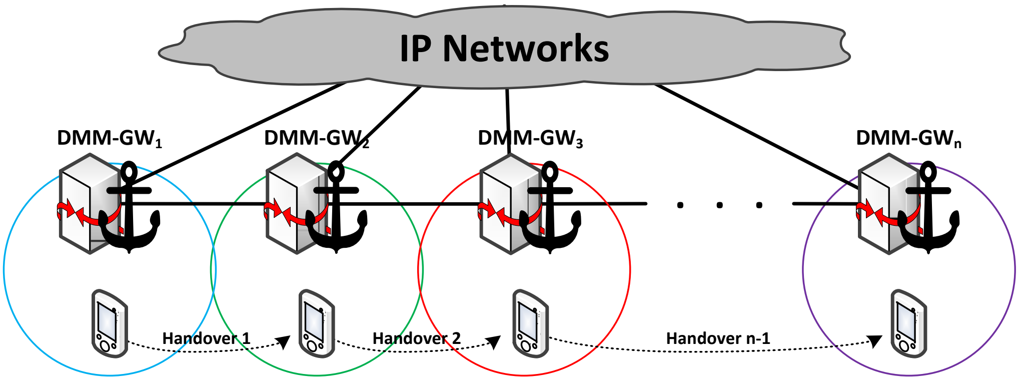


Figure 7.1: Mobility scenario for an MN.

The first order statistical description provided by Eq. (7.3) is sufficient to describe the average signalling cost in a centralized mobility protocol like MIPv6 and PMIPv6. Indeed, in such protocols, every location update produces a fixed cost given by the constant number of control messages exchanged by the mobility entities. For instance, in MIPv6, a location update consists of a BU and a BA message[2] between the MN and the HA, whereas, in PMIPv6, there is zero lifetime PBU/PBA exchange between the old MAG and the LMA, and a non-zero lifetime PBU/PBA handshake between the new MAG and the LMA.

Nevertheless, the signaling in a DMM protocol, regardless it is a client or network based one, scales also with the number of LANPs that are required to be kept reachable after the handoff. Therefore, we have to include in our model the statistical "lifetime" of a LANP, in order to derive the mean number of DMM-GWs involved in the signaling.

In the following, a LANP is considered "active" if the address derived from it is being used by at least one IP flow. Among the active LANPs, one was allocated by the DMM-GW currently serving the MN, and we refer to it as the current LANP (cLANP). The other LANPs were advertised by previously visited DMM-GWs, thus we call them previous LANPs (pLANPs). The LANPs not involved in active IP flows are eventually de-registered and no longer influence the mobility updates. Therefore, a key aspect of our cost analysis is the lifetime of a LANP, that is, how long it takes to be de-registered after the first advertisement to the MN. Indeed, the number of active LANPs determines the number of involved DMM-GWs, hence impacting on the signaling overhead, as well as on the traffic tunneled between the DMM-GWs.

In the following we evaluate the average number of active LANPs at a handover event. We consider a scenario where an MN moves as in Fig. 7.1, that is, the MN visits always a new DMM-GW, adjacent to the previous one. We shall see next that it represents the worst case scenario for the scope of our analysis. Moreover, we assume that a cLANP

[2] We consider here only the MN–to–HA interface, omitting for simplicity the route optimization procedure with the CNs.

is always maintained at least for the first handover. This assumption is in accordance with the N-DMM protocols operations, whereas it is not a mandatory procedure in the C-DMM protocol.

Hence, a LANP's lifetime consists of a whole subnet residence time, because it is used as cLANP, plus a trailing interval, in which it is used as pLANP. We refer to the former interval as the LANP *home time* (T_H), and to the latter as the LANP *foreign time* (T_F). If T_{LANP} denotes the time while a LANP is active (we name it the LANP *active time*), then we obtain:

$$T_{LANP} = T_H + T_F \, , \tag{7.4}$$

where $T_H = T_{SN}$, and T_F is the random decay interval since the MN leaves the "home" DMM-GW until the LANP expires. By denoting with $1/\lambda_F \triangleq \mathrm{E}[T_F]$ the mean value of the LANP foreign time, we can write the LANP mean lifetime as:

$$\overline{T}_{LANP} = \mathrm{E}[T_{LANP}] = \frac{1}{\mu_{SN}} + \frac{1}{\lambda_F} \, . \tag{7.5}$$

We now compute the average number of active prefixes for an MN, $\overline{N}_{LANP}$. This is is given by one (the cLANP freshly acquired) plus the average number of active pLANPs, $\overline{N}_{pLANP}$:

$$\overline{N}_{LANP} = 1 + \overline{N}_{pLANP}. \tag{7.6}$$

$\overline{N}_{pLANP}$ is the product of the average number of handovers that occur during the foreign prefix lifetime, times the prefix generation rate r, which in our case is simply $r = 1$ prefix per handover. We refer to $\alpha(K)$ as the probability that K handovers occur during the interval T_F. Therefore, a pLANP remains active on average for E[α(K)] subnet residence intervals and from the above considerations, we obtain:

$$\overline{N}_{pLANP} = r \; \mathrm{E}[\alpha(\mathrm{K})] = \mathrm{E}[\alpha(\mathrm{K})] \; . \tag{7.7}$$

We next follow the methodology devised in [97] to compute $\alpha(K)$ and E[α(K)], for any distribution of the subnet residence time and the LANP's foreign time.

Let $f^*_{T_{SN}}(s)$ and $f^*_{T_F}(s)$ be the Laplace transforms of $f_{T_{SN}}(t)$ and $f_{T_F}(t)$ respectively. By applying Theorem 1 of [97], we obtain:

$$\alpha(K) = \begin{cases} -\sum\limits_{p \in \sigma_{T_F}} \operatorname*{Res}\limits_{s=p} \dfrac{1 - f^*_{T_{SN}}(s)}{s} f^*_{T_F}(-s) & \text{for } K = 0 \\ -\sum\limits_{p \in \sigma_{T_F}} \operatorname*{Res}\limits_{s=p} \dfrac{[1 - f^*_{T_{SN}}(s)][f^*_{T_{SN}}(s)]^K}{s} f^*_{T_F}(-s), & \text{for } K > 0 \end{cases} . \tag{7.8}$$

where σ_{T_F} is the set of poles of $f^*_{T_F}(-s)$ in the right half complex plane, and $\mathrm{Res}_{s=p}$ is the residue at pole $s = p$. In a similar way, from Theorem 2 of [97], we derive the expected value E[α(K)]:

$$\mathrm{E}[\alpha(\mathrm{K})] = -\sum_{p\in\sigma_{T_F}} \operatorname*{Res}_{s=p} \frac{f^*_{T_{SN}}(s)}{s[1-f^*_{T_{SN}}(s)]} f^*_{T_F}(-s)\,, \tag{7.9}$$

with the same meaning of σ_{T_F} and $\mathrm{Res}_{s=p}$.

Eqs. (7.8) and (7.9) hold for any generic distribution $f_{T_{SN}}(t)$ and $f_{T_F}(t)$, but they may yield to complicated results if the corresponding Laplace transforms are not easy to manipulate, and numeric methods or simulations may be necessary. For this reason, approximations are commonly introduced in order to carry out closed and tractable expressions. In this sense, the generalized Gamma distribution is often used because it has a simple Laplace transform, and, more important, it does not have a specific shape, so it can fit an arbitrary distribution by choosing appropriate parameters [96].

We assume then that $f_{T_{SN}}(t)$ follows a Gamma distribution with mean $1/\mu_{SN}$ and variance $1/(\gamma\mu^2_{SN})$, and let $f_{T_F}(t)$ follow an Exponential distribution with mean $1/\lambda_F$ (an Exponential is a special case of Gamma). We obtain:

$$f_{T_{SN}}(t) = \frac{\gamma\mu_{SN}{}^{\gamma}t^{\gamma-1}}{\Gamma(\gamma)}e^{-\gamma\mu_{SN}t} \quad\Leftrightarrow\quad f^*_{T_{SN}}(s) = \left(\frac{\gamma\mu_{SN}}{s+\gamma\mu_{SN}}\right)^{\gamma}, \tag{7.10a}$$

$$f_{T_F}(t) = \lambda_F e^{-\lambda_F t} \quad\Leftrightarrow\quad f^*_{T_F}(s) = \frac{\lambda_F}{s+\lambda_F}\,, \tag{7.10b}$$

and therefore, substituting these expressions into (7.6) and (7.7), we get:

$$\begin{aligned}\overline{N}_{pLANP} = \mathrm{E}[\alpha(\mathrm{K})] &= \frac{f^*_{T_{SN}}(\lambda_F)}{1-f^*_{T_{SN}}(\lambda_F)}\\ &= \frac{(\gamma\mu_{SN})^{\gamma}}{(\lambda_F+\gamma\mu_{SN})^{\gamma}-(\gamma\mu_{SN})^{\gamma}}\,; \end{aligned}\tag{7.11a}$$

$$\overline{N}_{LANP} = \frac{(\lambda_F+\gamma\mu_{SN})^{\gamma}}{(\lambda_F+\gamma\mu_{SN})^{\gamma}-(\gamma\mu_{SN})^{\gamma}}\,. \tag{7.11b}$$

In order to further simplify the problem formulation, let's assume that also $f_{T_{SN}}(t)$ follows an exponential distribution with parameter μ_{SN}. In this scenario, the computation of $\overline{N}_{LANP}$ does not require to use the Laplace transform. In fact, the assignment of a new LANP is a Poisson process with parameter μ_{SN}. Let $N_{LANP}(t)$ denote the number of active prefixes at time t, it then follows a Poisson distribution with mean:

$$\overline{N}_{LANP}(t) = \mu_{SN}\int_0^t [1-F_{T_{LANP}}(u)]\,du\,, \tag{7.12}$$

where $F_{T_{LANP}}(u)$ is the cumulative distribution function (CDF) of T_{LANP}. By letting $t\to\infty$ in Eq. (7.12), we obtain the long run behavior of $\overline{N}_{LANP}(t)$, which represents the average number of active prefixes:

$$\begin{aligned}\overline{N}_{LANP} &= \mu_{SN}\int_0^{\infty}[1-F_{T_{LANP}}(u)]\,du \\ &= \mu_{SN}\overline{T}_{LANP} = 1+\frac{\mu_{SN}}{\lambda_F}\,. \end{aligned} \tag{7.13}$$

Similarly, the evaluation of $\alpha(K)$ is straightforward:

$$\begin{aligned}\alpha(0) &= \Pr[T_F \leq T_{SN}] \\ &= \int_0^{\infty}\Pr[T_F \leq u] f_{T_{SN}}(u)\,du \\ &= \frac{\lambda_F}{\lambda_F+\mu_{SN}}\,; \end{aligned} \tag{7.14a}$$

$$\alpha(K>0) = 1-\alpha(0) = \frac{\mu_{SN}}{\lambda_F+\mu_{SN}} \triangleq P_{LANP}\,. \tag{7.14b}$$

Iterating this computation and after simplifying the expression, we obtain:

$$\alpha(K \geq k)] = P_{LANP}^{k}\,,\; k \geq 1\,, \tag{7.15}$$

which finally yields to the probability $\alpha(K)$ and its expected value:

$$\alpha(K) = P_{LANP}^{K} - P_{LANP}^{K+1} = P_{LANP}^{K}(1-P_{LANP})\,, \tag{7.16a}$$

$$\mathrm{E}[\alpha(\mathrm{K})] = \frac{P_{LANP}}{1-P_{LANP}} = \frac{\mu_{SN}}{\lambda_F}. \tag{7.16b}$$

By merging (7.6), (7.7) and (7.16b), we finally have:

$$\overline{N}_{LANP} = 1+\frac{\mu_{SN}}{\lambda_F}. \tag{7.17}$$

We now anticipate a definition that it is used later in the chapter. Taking into account the scenario of Fig. 7.1, let $D_i(K)$ denote the number of hops between the DMM-GW current serving the MN, and the DMM-GW anchoring prefix i, after that K handovers took place during LANP i foreign time ($T_F^{(i)}$). Then we have:

$$D_i(K) = K+1. \tag{7.18}$$

Intuitively, this is explained by the fact that each prefix is maintained active at least for the time that the MN stays in the subnet.

7.2 Total signaling cost

A key operation for an IP mobility protocol is maintaining the MN's mobility session up to date. As we have described in previous chapters, such operation requires dedicated signaling, thus an important performance metric is the cost associated to it.

In the following, we refer to the total signaling cost, C_{sig}, as the sum of three main components [98, 99]:

1. the cost for the binding update after a handover (C_{update}),
2. the cost for terminating a prefix that is no longer active ($C_{de\text{-}reg}$),
3. the cost required to periodically refresh the bindings ($C_{refresh}$).

With this approach, we consider the steady-state or long run behavior, that is, we omit the initial registration phase and the de-registration when the MN definitely leaves the domain.

These operations are performed for each new visited access network. Since it takes place at a rate μ_{SN}, we have:

$$C_{sig} = \mu_{SN}(C_{update} + C_{refresh} + C_{de\text{-}reg})\,. \tag{7.19}$$

Next subsections are devoted to infer the cost components expressions for MIPv6, PMIPv6 and all our DMM solutions. The expressions are also collected in Table 7.2. The notation used is as follows: $C_{\text{X–Y}}$ represents the symmetric cost of the signalling interface between network nodes X and Y[3]; PC_{X} is the cost of processing a received packet by node X, and $\overline{N}_{LANP}$ is defined in Eq. (7.6). The term R_{BCE} accounts for the number of mobility session refreshments that are performed on average during the sojourn within the same subnet. Since the refreshment takes place every time the binding cache entry lifetime T_{BCE} is about to expire, we have

$$R_{BCE} = \left\lfloor \frac{1}{\mu_{SN} T_{BCE}} \right\rfloor\,. \tag{7.20}$$

7.2.1 MIPv6

In MIPv6, a location update consists in the BU/BA sequence, therefore C_{update} is twice the cost associated to the MN–HA interface plus the processing of such messages respectively at the HA and at the MN.

The same procedure takes place when the binding lifetime is about to expire, thus $C_{refresh}$ has the same expression of C_{update} times the number of refreshments during a residence time.

In MIPv6, the de-registration phase occurs only when the terminal ends the mobility service, so we omit the term $C_{de\text{-}reg}$.

[3] The cost metric attributed to the parameter $C_{\text{X-Y}}$ is arbitrary. For instance, it may account for the latency, the bandwidth consumed, the hop count, etc.

Table 7.2: Binding signaling costs.

	MIPv6
C_{update}	$= 2\,C_{\text{HA–MN}} + PC_{\text{HA}} + PC_{\text{MN}}$
C_{refresh}	$= R_{BCE}\,(2\,C_{\text{HA–MN}} + PC_{\text{HA}} + PC_{\text{MN}})$
$C_{\mathit{de\text{-}reg}}$	(Not necessary)
	PMIPv6
C_{update}	$= 2\,C_{\text{LMA–MAG}} + PC_{\text{LMA}} + PC_{\text{MAG}}$
C_{refresh}	$= R_{BCE}\,(2\,C_{\text{LMA–MAG}} + PC_{\text{LMA}} + PC_{\text{MAG}})$
$C_{\mathit{de\text{-}reg}}$	$= 2\,C_{\text{LMA–MAG}} + PC_{\text{LMA}} + PC_{\text{MAG}}$
	C-DMM
C_{update}	$= \overline{N}_{LANP}\,(4\,C_{\text{DMM-GW–MN}} + 2\,PC_{\text{DMM-GW}} + 2\,PC_{\text{MN}})$
C_{refresh}	$= R_{BCE}\,\left(\overline{N}_{LANP} - 1\right)(2\,C_{\text{DMM-GW–MN}} + PC_{\text{DMM-GW}} + PC_{\text{MN}})$
$C_{\mathit{de\text{-}reg}}$	$= 2\,C_{\text{DMM-GW–MN}} + PC_{\text{DMM-GW}} + PC_{\text{MN}}$
	Partially Distributed N-DMM, “relay” solution
C_{update}	$= (\overline{N}_{LANP} + 1)(2\,C_{\text{CMD–DMM-GW}} + PC_{\text{CMD}} + PC_{\text{DMM-GW}})$
C_{refresh}	$= R_{BCE}\,(2\,C_{\text{CMD–DMM-GW}} + PC_{\text{CMD}} + PC_{\text{DMM-GW}})$
$C_{\mathit{de\text{-}reg}}$	$= 4C_{\text{CMD–DMM-GW}} + 2PC_{\text{CMD}} + 2PC_{\text{DMM-GW}}$
	Partially Distributed N-DMM, “locator” solution
C_{update}	$= C_{\text{CMD–DMM-GW}} + PC_{\text{CMD}} +$ $\overline{N}_{LANP}(2\,C_{\text{CMD–DMM-GW}} + C_{\text{DMM-GW–DMM-GW}} + PC_{\text{CMD}} + 2PC_{\text{DMM-GW}})$
C_{refresh}	$= R_{BCE}\,(2\,C_{\text{CMD–DMM-GW}} + PC_{\text{CMD}} + PC_{\text{DMM-GW}})$
$C_{\mathit{de\text{-}reg}}$	$= 2\,C_{\text{CMD–DMM-GW}} + 2\,C_{\text{DMM-GW–DMM-GW}} + PC_{\text{CMD}} + 3PC_{\text{DMM-GW}}$
	Partially Distributed N-DMM, “proxy” solution
C_{update}	$= (\overline{N}_{LANP} + 1)(2\,C_{\text{CMD–DMM-GW}} + PC_{\text{CMD}} + PC_{\text{DMM-GW}})$
C_{refresh}	$= R_{BCE}\,(2\,C_{\text{CMD–DMM-GW}} + PC_{\text{CMD}} + PC_{\text{DMM-GW}})$
$C_{\mathit{de\text{-}reg}}$	$= 4C_{\text{CMD–DMM-GW}} + 2PC_{\text{CMD}} + 2PC_{\text{DMM-GW}}$
	Fully Distributed N-DMM
C_{update}	$= \overline{N}_{LANP}(2\,C_{\text{DMM-GW–DMM-GW}} + PC_{\text{CMD}} + PC_{\text{DMM-GW}})$
C_{refresh}	(Not necessary)
$C_{\mathit{de\text{-}reg}}$	$= 2\,C_{\text{DMM-GW–DMM-GW}} + 2\,PC_{\text{DMM-GW}}$

7.2.2 PMIPv6

In PMIPv6, the location update incurs in a cost C_{update}, given by two utilizations of the LMA–MAG interface due to the PBU and PBA messages and the associated processing by the LMA and MAG.

As for the MIPv6 case, $C_{refresh}$ has the same expression of C_{update} times the number of refreshments during a residence time.

However, differently from MIPv6, in PMIPv6, an old MAG previously visited by the MN eventually triggers the de-registration process for the MN. Such trigger may come during the handover phase if the MAGs employs a dedicated mechanism to detect a link layer event about the terminal detachment. Should this be the case, the de-registration and location update procedures both take place during handover. Otherwise, the de-registration is performed when the mobility session lifetime expires at the old MAG, that is, well after the handover concluded, and the notification is ignored by the LMA. In any case, given the PBU/PBA nature of the de-registration process, the cost $C_{de\text{-}reg}$ is the same as a location update cost.

7.2.3 C-DMM

As described in Section 4.3, when the MN signals a location update to a DMM-GW, there is twice a two-way handshake between the MN and the DMM-GW. One for the CoTI/CoT signaling required by the enhanced security mechanism, and another for the binding itself. Thus, C_{update} accounts for four signaling events in the DMM-GW–MN interface, plus the due processing, for each MN's active LANP. The average number of active LANPs has been derived in Section 7.1.

The $C_{refresh}$ component does not require the CoTI/CoT signaling, thence the number of control messages is reduced to two with respect to C_{update}, but, still, each LANP needs to be refreshed upon lifetime expiration, except the one assigned by the current DMM-GW.

A LANP acquired by the MN is eventually de-registered when the MN decides so, and this incurs in a cost $C_{de\text{-}reg}$ obtained from the zero lifetime BU/BA exchange with the associated DMM-GW.

7.2.4 Partially Distributed N-DMM

"Relay" solution. In this DMM solution, C_{update} consists of two utilizations of the CMD–DMM-GW interface for each DMM-GW active before the handover and two more for the new serving DMM-GW. The average number of DMM-GWs before handoff is equal to the mean number of active LANPs.

The BCE refreshment involves only the CMD and the current serving DMM-GW PBU/PBA. Hence, $C_{refresh}$ does not take the parameter $\overline{N}_{LANP}$.

An old DMM-GW eventually de-registers its pLANP upon detection that the prefix is no longer in use, incurring in a cost $C_{de\text{-}reg}$, which is twice a two-way handshake between the CMD and the DMM-GW since there are two DMM-GWs involved.

"Locator" solution. The "locator" solution introduces the DMM-GW–DMM-GW control interface with the associated cost $C_{\text{DMM-GW–DMM-GW}}$. Therefore, for every LANP, C_{update} accounts for a two utilizations of the CMD–DMM-DW interface and one for the DMM-GW–DMM-GW interface, with the necessary processing costs. In addition, C_{update} takes the first notification sent from the new serving DMM-GW to the CMD.

The refreshment procedure is performed by the serving DMM-GW with the CMD, and it is indeed the same for all the partially distributed solutions.

The cost associated to de-register a pLANP is given by two messages delivered through the CMD–DMM-GW interface and two through the DMM-GW–DMM-GW interface plus the costs to process the received packets.

"Proxy" solution. The "proxy" solution devises the same operations from the "relay" solution, except for a permutation of the message sequence. Therefore, the signaling cost is identical to the "relay" solution.

7.2.5 Fully Distributed N-DMM

Arguing that the MIHS protocol is not an exclusive feature of the fully distributed DMM solution, since it can be used along with any IP mobility protocol, the fully distributed solution is analyzed only for the contribution of the IP mobility signaling, leaving aside the MIHS part.

As the partially distributed "locator" solution, this protocol employs the DMM-GW–DMM-GW control interface with the associated cost $C_{\text{DMM-GW–DMM-GW}}$. Therefore, the location update cost C_{update} is given by the cost associated to transfer a PBU and a PBA from the new DMM-GW to each old DMM-GW anchoring an active LANP through such interface.

The refreshment phase is not necessary because the DMM-GW's FM constantly monitors the LANP active on its link to the MN (either a tunnel or a direct link).

When a LANP is no longer necessary, there is a zero lifetime PBU/PBA message exchange between the serving DMM-GW and the one anchoring the LANP that became inactive.

7.2.6 Total signaling cost comparison

This study builds on top of the results presented above and aims at providing insights about the advantages and disadvantages of deploying a DMM solution, either client or network-based, compared to its centralized counter part, i.e., MIPv6 or PMIPv6.

We thereby study the signaling operations as a result of applying Eq. (7.19) with the MIPv6/PMIPv6 and DMM cost expressions presented in Table 7.2. The objective is to compute the signaling cost ratio between any DMM solution and its centralized version. For a fair comparison, we assume that the cost components are identical for any signaling interface, and also the processing costs at any node. That is:

$$C_{\text{X–Y}} = a\,, \quad \forall\, \text{X,Y} \qquad \text{and} \qquad PC_{\text{X}} = b\,, \quad \forall\, \text{X} \tag{7.21}$$

Thus we obtain the following expressions:

$$\frac{C_{sig}^{\text{C-DMM}}}{C_{sig}^{\text{MIPv6}}} = \overline{N}_{PR} + \frac{\overline{N}_{PR} - R_{BCE} + 1}{R_{BCE} + 1}\,, \tag{7.22a}$$

$$\frac{C_{sig}^{\text{relay/proxy}}}{C_{sig}^{\text{PMIPv6}}} = 1 + \frac{\overline{N}_{PR} + 1}{R_{BCE} + 2}\,, \tag{7.22b}$$

$$\frac{C_{sig}^{\text{locator}}}{C_{sig}^{\text{PMIPv6}}} = 1 + \frac{\frac{3}{2}\overline{N}_{PR}}{R_{BCE} + 2}\,, \tag{7.22c}$$

$$\frac{C_{sig}^{\text{Full N-DMM}}}{C_{sig}^{\text{PMIPv6}}} = \frac{\overline{N}_{PR} + 1}{R_{BCE} + 2}\,. \tag{7.22d}$$

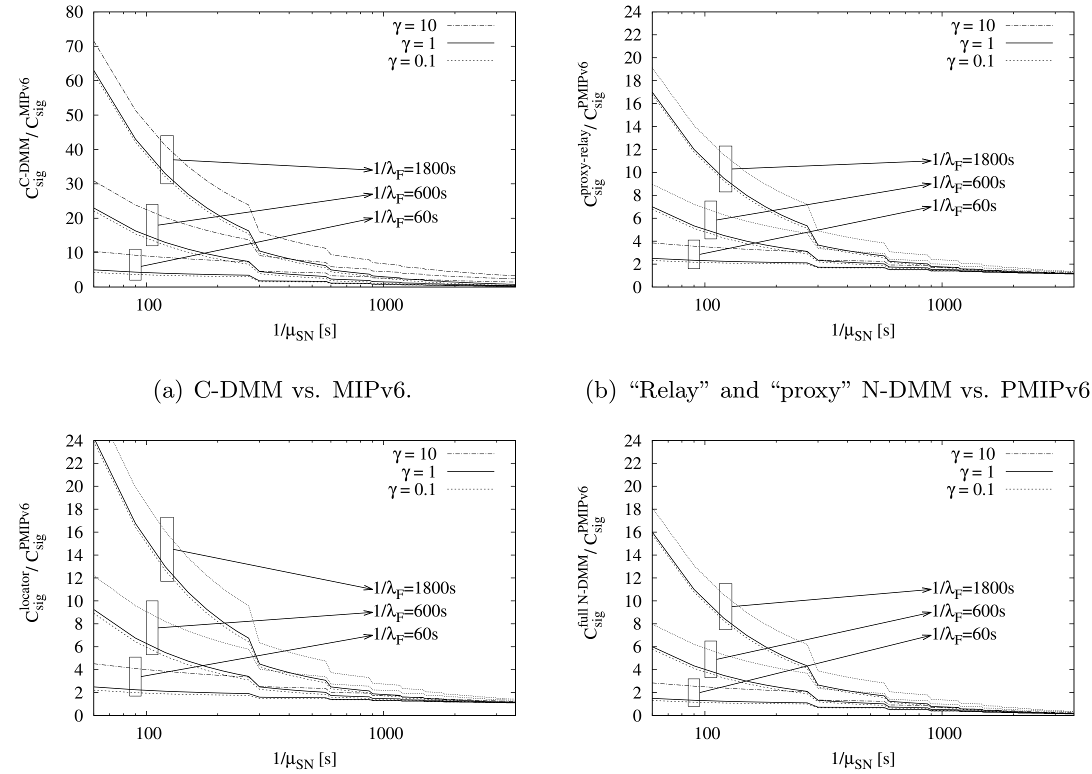
In Fig. 7.2 we depicted the plot for each of the above equations, for $T_{BCE} = 300\text{s}$[4], and $\overline{N}_{LANP}$ computed from Eq. (7.12), that is, assuming a Gamma distribution for the subnet residence time.

The graphs of Fig. 7.2 illustrate the effects of the subnet residence time, $1/\mu_{SN}$, and the prefix lifetime, $1/\lambda_F$, on the signaling cost for various values of the parameter γ (note that for $\gamma = 1$, the subnet residence time becomes an exponential distribution with parameter μ_{SN}).

The most remarkable observation is that the signaling overhead of any DMM solution is higher than a centralized approach (since several DMM-GWs must be updated), especially when the LANP lifetime is long compared to the residence time. This behavior is exacerbated by the C-DMM solution, as all the active LANPs are refreshed with a double signaling session with each anchor. Among the N-DMM protocols, the "locator" solution is clearly the one with higher costs (we already anticipated this result in Section 5.2.4), while the rest performs quite the same, even if the fully N-DMM solution is slightly better.

However, the DMM costs converge to the those of a centralized protocol for the scenarios with a longer residence time, as it is more likely that IP sessions are consumed within fewer subnets, thus producing a lower number of active LANPs. It is important to note that we are considering hypothetical mobility traces where users visit a new subnet

[4] PMIPv6 Configuration Guide, Cisco IOS XE Release 3S, `http://www.cisco.com/en/US/docs/ios-xml/ios/mob_pmipv6/configuration/xe-3s/imo-pmipv6-mag-support-xe.html/`



(a) C-DMM vs. MIPv6.

(b) "Relay" and "proxy" N-DMM vs. PMIPv6.

(c) "Locator" N-DMM vs. PMIPv6.

(d) Fully distributed N-DMM vs. PMIPv6.

Figure 7.2: Signaling cost comparison for DMM and CMM solutions.

at each handover. We remark that this is a worst case scenario, because typically users tend to move within a limited area covered by few access routers.

Finally, the signaling costs are moderately low in N-DMM when the LANP lifetime is lower than 60 seconds. We argue this is the most common scenario as the majority of popular mobile applications (e.g., web browsing, social networks, instant messaging, email[5]) typically generate many short lived sessions and they can survive an IP address change with little or no impact to the user.

7.3 Packet Delivery Cost

One of the key drivers of the development of flatter network architectures exhibiting distributed mobility management is the reduction of the costs caused by all the traffic traversing a centralized anchor. The use of a DMM solution allows reducing the communication delay between endpoints bypassing the core network, and reducing the costs

[5] Source: Wikipedia, "List of most downloaded Android applications", `http://en.wikipedia.org/wiki/List_of_most_downloaded_Android_applications/`, accessed January 2015. Wikipedia, "List of Most downloaded iOS applications", `http://en.wikipedia.org/wiki/List_of_Most_downloaded_iOS_applications/`, accessed January 2015

of using very powerful network nodes and large links that need to be dimensioned to transport all mobile nodes' data to the core. For instance, in PMIPv6, user data always traverses the centralized local mobility anchor and its surrounding links, unless the two communication endpoints reside in the same domain and the local routing feature [101] is enabled.

A DMM mechanism should mitigate the problems of mobile operators when coping with the foreseen increase in user traffic. Together with this traffic demand increase, operators also expect that most of the user data sessions will be terminated in the same region where the originating peer is located [36]. This is generally the case with voice calls, which are usually established between geographically close users, with instant messaging services and with the growing penetration of content distribution networks (CDNs).

In the following, we develop a framework to evaluate to which extent a flat architecture brings advantages over a centralized one in terms of packet delivery cost, and under which mobility scenarios. In our analysis, we consider that the transfer of a packet over link without encapsulation has a unitary cost, whilst the delivery through a tunnel incurs in a penalty $\tau > 1$; similarly, a wireless link has an extra cost $\omega > 1$. We consider λ_p as the packet rate per active LANP, and any prefix generates the same packet rate, thus packets are transferred to and from the MN at rate $\lambda = \lambda_p \cdot \overline{N}_{LANP}$ packets/second. Note that for the MIPv6 and PMIPv6 cases, only one prefix is active, hence only the aggregate packet rate λ is meaningful.

In a mobile network, the path followed by user data traffic between a correspondent node (CN) and a mobile node can be divided in three major segments:

1. S_1, from the CN to the mobility anchor, that is, a HA, an LMA or a DMM-GW, respectively for MIPv6, PMIPv6 and a DMM solution. This segment is usually external to the MNO's infrastructure, hence we omit it in the next cost analysis.
2. S_2, from the mobility anchor to the access router currently serving the MN (in MIPv6), or the MAG (in PMIPv6), or the serving DMM-GW (in any DMM protocol). This segment is usually a tunnel link.
3. S_3, from the access router to the MN, which we assume is the final wireless link. This assumption is almost never implemented in practical deployments, as radio access points and base stations are usually provided with high speed backhaul links to the router, and not co-located with it. However we disregard this observation in order to keep a simple study scenario. We also note that in MIPv6 and C-DMM, this segment is part of a tunnel.

The packet delivery cost C_{PD} is then given by the sum of the costs for each segment times the packet rate:

$$C_{PD} = \lambda \left[C\left(S_1\right) + C\left(S_2\right) + C\left(S_3\right) \right] \tag{7.23}$$

In the following we carry out the expression of Eq. (7.23) for the centralized and distributed mobility protocols. As for Section 7.2, the notation $C_{\mathrm{X-Y}}$ indicates the cost to transmit a packet from node X to Y with any arbitrary metric.

7.3.1 MIPv6

In MIPv6, without considering route optimization, data packets travel from the CN to the HA, then they are encapsulated from the HA to the MN passing through the access router where the MN is currently connected. Thus we have:

$$C\,(S_1)= C_{\mathrm{CN-HA}}\,, \tag{7.24a}$$

$$C\,(S_2)= \tau\, C_{\mathrm{HA-AR}}\,, \tag{7.24b}$$

$$C\,(S_3)= \omega\,\tau\, C_{\mathrm{AR-MN}}\,, \tag{7.24c}$$

and, finally:

$$C_{PD}^{\mathrm{MIPv6}} = \lambda\left(C_{\mathrm{CN-HA}} + \tau\, C_{\mathrm{HA-AR}} + \omega\,\tau\, C_{\mathrm{AR-MN}}\right)\,. \tag{7.25}$$

7.3.2 PMIPv6

In PMIPv6, the data path consists in the segment from the CN to the LMA, then the tunnel from the LMA to the MAG, and finally the wireless link from the MAG to the MN. Therefore, the PMIPv6 packet delivery cost is given by:

$$C_{PD}^{\mathrm{PMIPv6}} = \lambda(C_{\mathrm{CN-LMA}} + \tau\, C_{\mathrm{LMA-MAG}} + \omega\, C_{\mathrm{MAG-MN}})\,. \tag{7.26}$$

7.3.3 C-DMM

In the client-based DMM solution, the path component S_1 is given by the cost for data packets to reach the DMM-GW that anchors the LANP carried by the packet from the CN, so, assuming the same cost for any destination DMM-GW, we have:

$$C\,(S_1) = C_{\mathrm{CN-DMMGW}}\,. \tag{7.27}$$

The segment S_2 is present only for those IP flows using a LANP anchored at a different DMM-GW where the MN is, therefore for $\left(\overline{N}_{LANP} - 1\right)$ LANPs on average. Moreover, each of this $\left(\overline{N}_{LANP} - 1\right)$ LANPs is routed through a different path, depending to how far the anchoring DMM-GW is from the serving DMM-GW. Thence, we need to make some assumptions on the mobile network topology and estimate the mean value of $C\,(S_2)$.

By taking as reference the scenario of Fig. 7.1, the S_2 segment is formed by summing the links between adjacent DMM-GWs, from the serving one to that anchoring the LANP. If each link between adjacent DMM-GWs has a cost $C_{\mathrm{DMM\text{-}GW-DMM\text{-}GW}}$, then $C\,(S_2)$ for LANP i is:

$$C_{LANP_i}\,(S_2) = D_i(K)\,\tau C_{\mathrm{DMM\text{-}GW-DMM\text{-}GW}} \tag{7.28}$$

being $D_i(K)$ introduced in Eq. (7.18). $D_i(K)$ is a statistical parameter obtained by the probability that K handovers occur during the LANP i foreign time, and its value is $D_i(K) = K + 1$. Thus, in general, the value for $C_{LANP_i}(S_2)$ is given by the sum $\forall\, D_i(K) > 0$ of the costs $(D_i(K)\,\tau\, C_{\text{DMM-GW–DMM-GW}})$, weighted by the probability of the prefix to experience K handovers during a LANP foreign time. Using Eqs. (7.6), (7.7) and (7.18), we have:

$$\begin{aligned} C_{LANP_i}(S_2) &= \sum_{K=0}^{\infty} D_i(K)\,\alpha(K)\,\tau C_{\text{DMM-GW–DMM-GW}} \\ &= \sum_{K=0}^{\infty} (1+K)\,\alpha(K)\,\tau C_{\text{DMM-GW–DMM-GW}} \\ &= (1 + \mathrm{E}[\alpha(\mathrm{K})])\,\tau C_{\text{DMM-GW–DMM-GW}} \\ &= \overline{N}_{LANP}\,\tau C_{\text{DMM-GW–DMM-GW}}. \end{aligned} \tag{7.29}$$

Summing up for the $\left(\overline{N}_{LANP} - 1\right)$ LANPs we finally get:

$$C(S_2) = \frac{\lambda_p}{\lambda}\left(\overline{N}_{LANP} - 1\right)\overline{N}_{LANP}\,\tau C_{\text{DMM-GW–DMM-GW}}\,. \tag{7.30}$$

The last segment S_3 aggregates those packets that carry the LANP announced by that DMM-GW, which are not encapsulated, and those carrying the other $\left(\overline{N}_{LANP} - 1\right)$ LANPs, that are encapsulated. Thus we can write:

$$C(S_3) = \omega\, C_{\text{DMM-GW–MN}}\left[\frac{\lambda_p}{\lambda}\left(\overline{N}_{LANP} - 1\right)\,\tau + 1\right]. \tag{7.31}$$

By applying the compound cost formula from Eq. (7.23), we can finally derive the packet delivery cost for C-DMM, $C_{PD}^{\text{C-DMM}}$:

$$\begin{aligned} C_{PD}^{\text{C-DMM}} =& \lambda\, C_{\text{CN–DMM-GW}} + \lambda_p\,\left(\overline{N}_{LANP} - 1\right)\overline{N}_{LANP}\,\tau C_{\text{DMM-GW–DMM-GW}} + \\ &+ \omega\, C_{\text{DMM-GW–MN}}\left[\lambda_p\left(\overline{N}_{LANP} - 1\right)\,\tau + \lambda\right]. \end{aligned} \tag{7.32}$$

7.3.4 N-DMM

With DMM, the path followed by user data traffic is the same as in C-DMM, with the sole difference that packets are never encapsulated in the last segment S_3. This yields to the following costs:

$$C(S_1) = C_{\text{CN–DMM-GW}}\,, \tag{7.33a}$$

$$C(S_2) = \frac{\lambda_p}{\lambda}\left(\overline{N}_{LANP} - 1\right)\overline{N}_{LANP}\,\tau C_{\text{DMM-GW–DMM-GW}}\,, \tag{7.33b}$$

$$C(S_3) = \omega\, C_{\text{DMM-GW–MN}}\,, \tag{7.33c}$$

and to the aggregate cost $C_{PD}^{\text{N-DMM}}$:

$$
\begin{aligned}
C_{PD}^{\text{N-DMM}} =& \lambda\, C_{\text{CN–DMM-GW}} + \lambda_p \left(\overline{N}_{LANP} - 1\right) \overline{N}_{LANP}\, \tau C_{\text{DMM-GW–DMM-GW}} + \\
&+ \lambda\, \omega\, C_{\text{DMM-GW–MN}}\,. \qquad (7.34)
\end{aligned}
$$

7.3.5 Packet delivery cost comparison

The evaluation of the packet delivery cost for the mobility protocols studied in the previous sections drives the comparison analysis carried out in the following.

Nevertheless, taking into account the full expressions of Eqs. 7.25, 7.26, 7.32 and 7.34 would yield to a too high number of variables to derive a general result. Therefore we limit our comparison focusing only to the value of $C\left(S_2\right)$ for MIPv6, PMIPv6, C-DMM and N-DMM, which is the most relevant from an MNO's perspective, as it represents the cost of packets flowing in the network's core infrastructure. Also, it is the portion of the network where the DMM principle applies.

As it was done in Section 7.2.6, we examine the cost ratio between a DMM solution and its centralized version. Yet, for a fair comparison, we assume that the cost associated to the HA–AR interface is the same to the LMA–MAG and equal to a constant C_{core}:

$$
C_{\text{HA–AR}} = C_{\text{LMA–MAG}} = C_{\text{core}}\,, \qquad (7.35)
$$

so that the value of $C\left(S_2\right)$ is the same for both MIPv6 and PMIPv6.

Noting that $C\left(S_2\right)$ has the same expression also for C-DMM and N-DMM solution, the packet delivery cost comparison can be easily evaluated as the ratio between the cost in a DMM solution, C_{PD}^{DMM}, against the cost in a CMM solution, C_{PD}^{CMM}. We then have:

$$
\frac{C_{PD}^{\text{DMM}}}{C_{PD}^{\text{CMM}}} = \frac{C^{\text{DMM}}\left(S_2\right)}{C^{\text{DMM}}\left(S_2\right)} = \left(\overline{N}_{LANP} - 1\right) \frac{C_{\text{DMM-GW–DMM-GW}}}{C_{\text{core}}}\,. \qquad (7.36)
$$

Fig. 7.3 illustrates the plot of Eq. (7.36) for different values of the $\frac{C_{\text{DMM-GW–DMM-GW}}}{C_{\text{core}}}$ ratio. This graph provides a first snapshot to evaluate the benefits that a DMM approach offers, given the network characteristics. Indeed, the larger is C_{core} with respect to $C_{\text{DMM-GW–DMM-GW}}$, then the higher is the number of active prefixes which packets can be transported without incurring in extra costs than those due to a CMM protocol. For instance, if the network is dimensioned so that $C_{\text{DMM-GW–DMM-GW}}$ is at least five times smaller than C_{core}, then DMM performs better than CMM up to 5-6 active LANPs on average. If we recall the simplified expression for $\overline{N}_{LANP}$ from Eq. (7.17), we obtain such a value of mean active LANPs when the prefix lifetime after the first handover is on average 4-5 times larger than residence time in a subnet. We argue this is a reasonable upper bound for most of the IP sessions generated by the most popular applications. Nonetheless, from the signaling cost evaluation given in the previous section, such a sce-

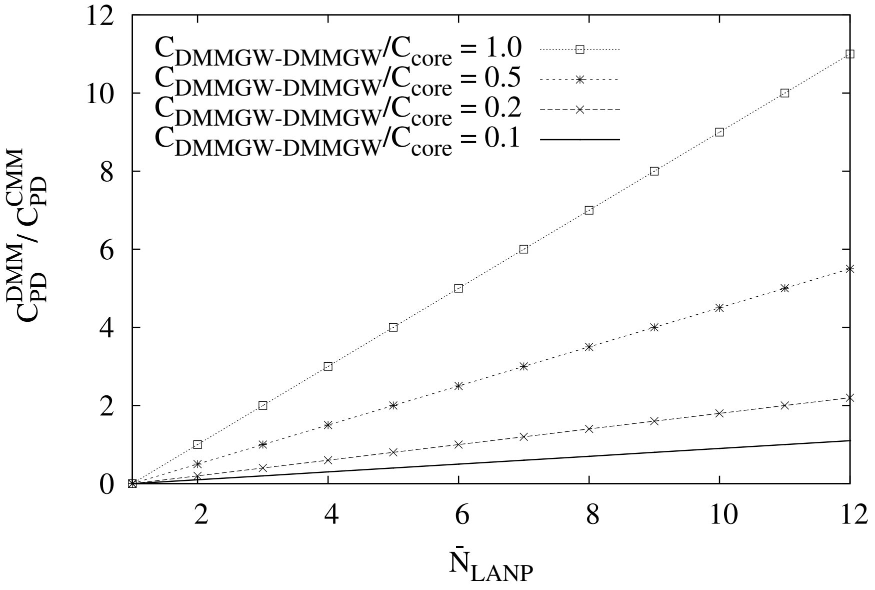


Figure 7.3: DMM vs. CMM: Packet delivery cost.

nario still has negative effects on the signaling load. It is therefore crucial for a network administrator to evaluate the trade-off between a CMM and a DMM protocol, provided the network characteristic and the traffic patterns observed.

7.4 Handoff latency and Packet loss

In this section we analyze the protocols with respect to the handover latency, that is, how long it takes for the protocol to execute the location update procedures. This metric is an important parameter to guarantee service continuity to the users. In fact, during a handover, the user terminal is disconnected from the network, therefore it cannot send nor receive data packets. For this reason, from the user's perspective, packet loss is the main issue associated to a handover. For instance, in a UDP flow, all the packets destined to the MN during the handover cannot be received, resulting in an interruption of the communication at the application level. In a TCP flow, the disconnection interval produces retransmissions and connection timeouts, therefore impacting on the overall transmission delay.

Nevertheless, handover events are a crucial part for the mobile network life and operations, that the network administrators have to cope with in the most effective manner. In the following we explore how the centralized and distributed IP mobility protocols behave, and which solution fits the best according to the possible scenarios.

For the sake of simplicity, we consider an IP flow only in the downstream direction, thus we refer to the *IP flow recovery time*, $T_{flow\text{-}rec}$, as the interval from the last data packet of a given session received by the terminal before the handover to the first packet received after the handover. Therefore, and since packet buffering is not considered, the packet loss during the MN's movement, PL, is proportional to the length of such interval and to the compound packet transmission rate, λ, of the IP flows:

$$PL = \lambda \, T_{\textit{flow-rec}} \, . \tag{7.37}$$

We characterize next the expression of $T_{\textit{flow-rec}}$ for each IP mobility protocol.

It is worth to anticipate here two sub-problems that usually come together with the handover latency study:

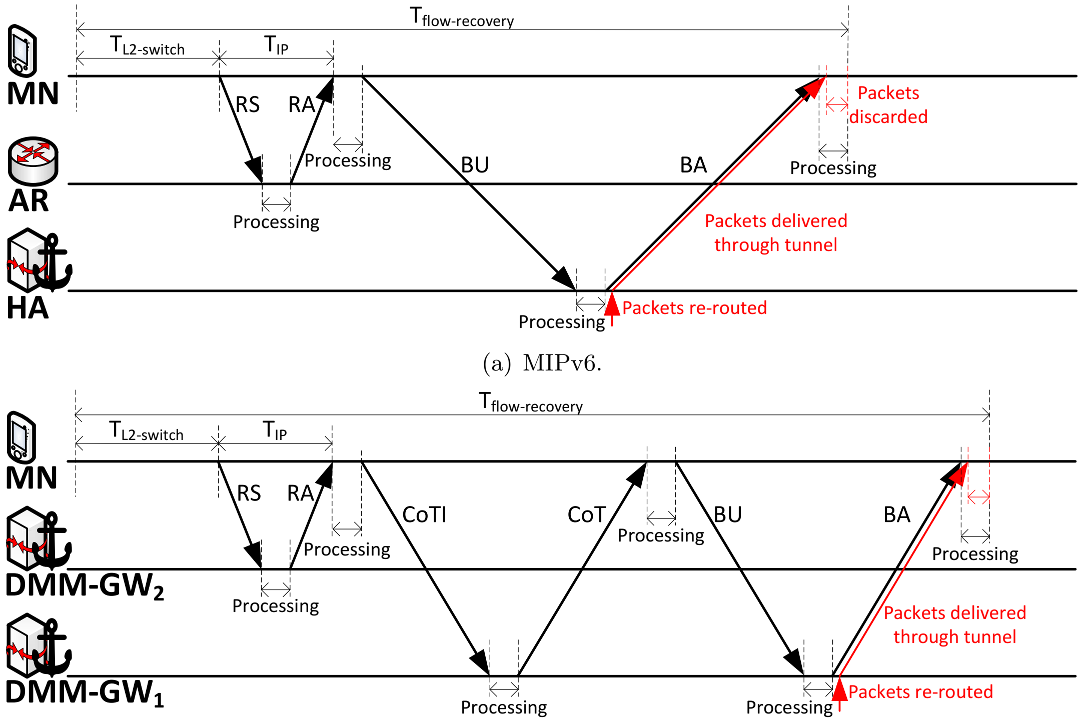
1. the *Layer-2 handover time*, T_{L2}, is the time elapsed since the old radio link is torn down until the new one is established. This term comprehends all the time required to de-activate a link and set up a new one, prior to gain connectivity at the IP layer. It depends on the specific radio technology used in the access network, but, since it does not depend on the mobility protocol at the IP level, we assume it to be a common and equal term for all the IP mobility solutions.
2. the *Layer-3 configuration time*, T_{IP}, is the time required by the MN to obtain network layer connectivity. This latter is achieved when the MN configures a global IPv6 address and is able to use it to communicate with its access router. This procedure does not necessarily implies global IPv6 connectivity, even if it is so in most cases. An additional delay contribution should be accounted for the Duplicate Address Detection (DAD) operation, performed by the MN after configuring an IPv6 address from the prefix conveyed in the RA message [19]. This step ensures that the IPv6 address configured by the MN is unique in the local link. However, we disregard this contribution as the MN's prefix is assigned uniquely in all the mobility protocols except that in MIPv6, where we might assume that the IPv6 address uniqueness is optimistically achieved (Optimistic DAD [102]). Therefore, we assume that this step is accomplished upon receiving the RA message from the sending entity (AR/MAG/DMM-GW).

The Layer-2 handover time and Layer-3 configuration time are disjoint intervals, and it is always true the following relation:

$$T_{\textit{flow-rec}} \geq T_{L2} + T_{IP} \, , \tag{7.38}$$

as the IP flow recovery time includes these two phases plus the remaining actions performed within the network to ensure IP session continuity.

In the study that follows, we assume that the Round Trip Time between nodes X and Y, denoted as $RTT_{\text{X–Y}}$, approximates the time spent to transfer a location update message and the corresponding acknowledgement between entities X and Y. Also, T^{P}_{X} indicates the processing time at node X. The set of pictures in Fig. 7.4, Fig. 7.5 and



(a) MIPv6.

(b) C-DMM.

Figure 7.4: Handover latency for client based CMM and DMM.

Fig. 7.6 displays the timeline of the operations for all the protocols, with the following assumptions:

$$RTT_{\text{MN–AR}} = RTT_{\text{MN–MAG}} = RTT_{\text{MN–DMM-GW}} = RTT_{\text{access}}\,, \tag{7.39a}$$

$$RTT_{\text{AR–HA}} = RTT_{\text{MAG–LMA}} = RTT_{\text{DMM-GW–CMD}} = RTT_{\text{core}}\,, \tag{7.39b}$$

$$RTT_{\text{access}} = \frac{RTT_{\text{DMM-GW–DMM-GW}}}{2}\,, \quad RTT_{\text{DMM-GW–DMM-GW}} = \frac{RTT_{\text{core}}}{2}\,, \tag{7.39c}$$

$$T^P_{\text{X}} = T^P, \quad \forall\ \text{X}\,. \tag{7.39d}$$

7.4.1 MIPv6

With the help of Fig. 7.4(a), we can observe that the IP flow recovery time is the interval since the MN starts the Layer-2 handover, until the MN configures its tunnel endpoint, which occurs after processing the BA message from the HA. Therefore we derive the following expression:

$$T_{\mathit{flow\text{-}rec}} = T_{L2} + RTT_{\text{MN-AR}} + T^P_{\text{AR}} + 2\,T^P_{\text{MN}} + RTT_{\text{MN-HA}} + T^P_{\text{HA}}\,. \tag{7.40}$$

Note that Eq. (7.40) includes twice the term T^P_{MN} because of *i)* the *movement detection*, that is, the operation required by the MN's mobility client to detect that the MN has configured a new IPv6 CoA, and thence to prepare the BU message to the DMM-GW;

ii) the processing of the BA message from the HA and tunnel setup. With the notation of Eq. (7.39), we can simplify Eq. (7.40):

$$T_{\textit{flow-rec}} = T_{L2} + RTT_{\text{access}} + RTT_{\text{core}} + 4\,T^{P}\,. \tag{7.41}$$

Also, the MN gains IP connectivity right after the router discovery procedure with the access router. After receiving the RA message, the MN is not only able to communicate with the AR using the CoA, but has also global connectivity. Indeed, the BU message sent next is an IP flow which source address is the CoA, destined to a global IPv6 address. However, the IP flows started with the CoA do not have mobility support in MIPv6 for next handovers. In all cases we have:

$$T_{IP} = RTT_{\text{access}} + T^{P}\,. \tag{7.42}$$

We can highlight the contribution of T_{L2} and T_{IP} in Eq. (7.41), to show that Eq. (7.38) holds:

$$T_{\textit{flow-rec}} = T_{L2} + T_{IP} + RTT_{\text{core}} + 3\,T^{P}\,. \tag{7.43}$$

7.4.2 C-DMM

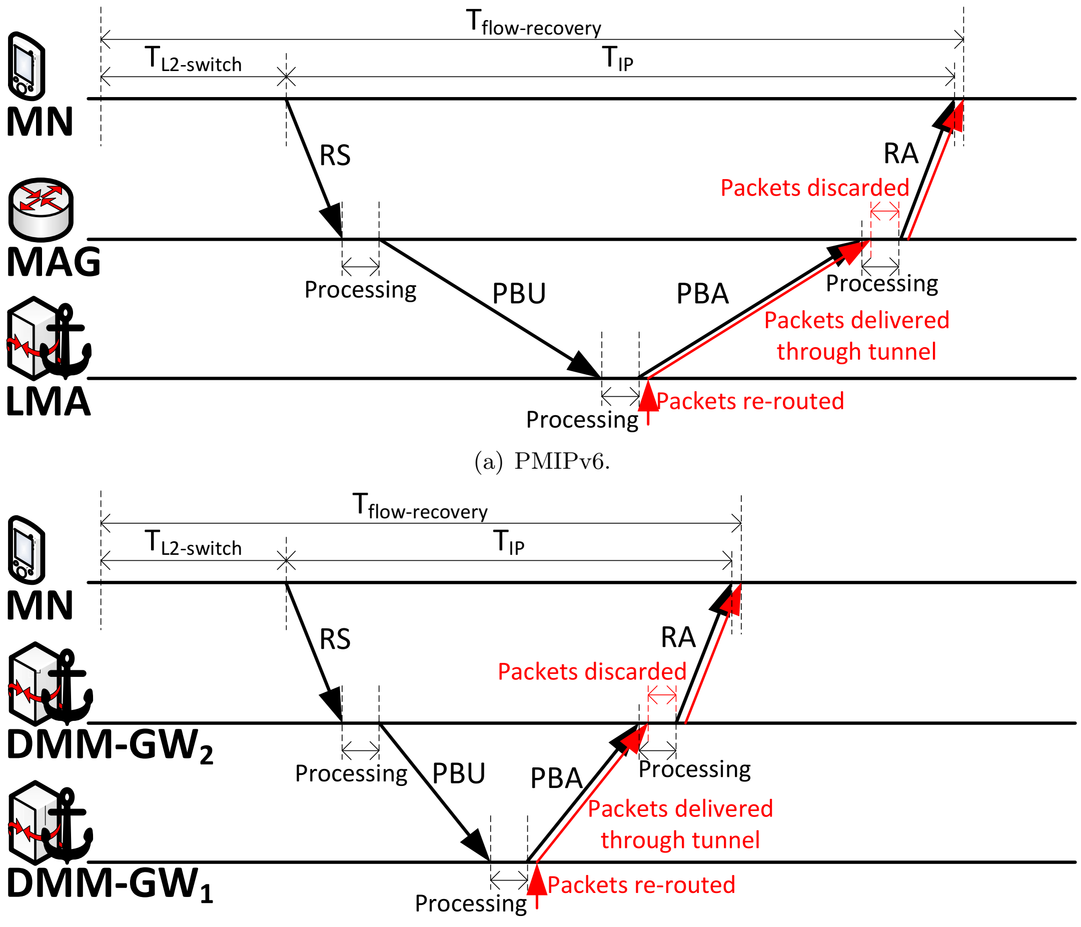
The analysis of $T_{\textit{flow-rec}}$ for the C-DMM solution can be carried out with reference to Fig. 7.4(b), where we notice twice the interaction between the MN and an old DMM-GW due to the Care-of Address Test security mechanism. Note that in the picture of Fig. 7.4(b), we have drawn for simplicity two adjacent DMM-GWs, that is, one DMM-GW–DMM-GW hop far from each other, according to the notation used throughout this chapter. However, as we did in previous sections, we consider the scenario of Fig. 7.1, where the distance from the current DMM-GW to the one anchoring the IP flow increases linearly after each handover. Thence the average distance between the two involved DMM-GWs can be obtained from the procedure followed for Eq. (7.29). Also, we remark that the packets of old IP flows cannot be delivered to the MN until the MN has correctly set up the tunnel. Thus we obtain the following:

$$\begin{aligned} T_{\textit{flow-rec}} =& T_{L2} + RTT_{\text{MN-AR}} + T^{P}_{\text{AR}} + 3\,T^{P}_{\text{MN}} + \\ & + 2\,\overline{N}_{LANP}\,RTT_{\text{DMM-GW–DMM-GW}} + 2\,T^{P}_{\text{DMM-GW}}\,, \end{aligned} \tag{7.44}$$

which can be simplified with the assumptions in Eq. (7.39), leading to:

$$T_{\textit{flow-rec}} = T_{L2} + RTT_{\text{access}} + 2\,\overline{N}_{LANP}\,RTT_{\text{DMMGW-DMMGW}} + 6\,T^{P}\,. \tag{7.45}$$

As for the MIPv6 case, when the MN receives the RA message from the new DMM-GW, it obtains local and global IPv6 connectivity through the CoA, enabling the MN to



(a) PMIPv6.

(b) Full N-DMM.

Figure 7.5: Handover latency for PMIPv6 and the full N-DMM.

start new IP sessions. Once again, the mobility control messages sent by the MN to an old DMM-GW are possible because of such global IPv6 connectivity through the CoA. Differently from the MIPv6 case, the IP flows started with the CoA do enjoy mobility support from the current DMM-GW if required by the MN. After these considerations we can write:

$$T_{IP} = RTT_{\text{access}} + T^P \,, \tag{7.46}$$

and, therefore, also

$$T_{\textit{flow-rec}} = T_{L2} + T_{IP} + 2\,\overline{N}_{LANP}\, RTT_{\text{DMMGW-DMMGW}} + 5\,T^P \,. \tag{7.47}$$

7.4.3 PMIPv6

Fig. 7.5(a) shows the timeline of a location update according to the PMIPv6 protocol. The user's data packets re-routed by the LMA to the new MAG, are discarded by this latter until the tunnel with LMA is correctly created. After this step, they are delivered to the MN and the IP flow is finally recovered. $T_{\textit{flow-rec}}$ can be expressed as:

$$T_{\textit{flow-rec}} = T_{L2} + RTT_{\text{MN–MAG}} + 2\,T^P_{\text{MAG}} + RTT_{\text{MAG–LMA}} + T^P_{\text{LMA}}\,, \tag{7.48}$$

and then simplified with:

$$T_{\textit{flow-rec}} = T_{L2} + RTT_{\text{access}} + RTT_{\text{core}} + 3\,T^P\,. \tag{7.49}$$

The MN obtains IPv6 connectivity with the MAG when it receives the RA message from this latter, and so follows the equation:

$$T_{IP} = RTT_{\text{access}} + RTT_{\text{core}} + 3\,T^P\,, \tag{7.50}$$

and also:

$$T_{\textit{flow-rec}} = T_{L2} + T_{IP}\,. \tag{7.51}$$

7.4.4 Full N-DMM

The sequence of operations of the fully distributed N-DMM solution is very similar to that of PMIPv6, except for the fact that the PBU and PBA messages are exchanged by DMM-GWs. The location update timeline is depicted in Fig. 7.5(b). As for the drawing of the C-DMM timeline, Fig. 7.5(b) reports two adjacent DMM-GWs. Still, the average distance between the current DMM-GW and the DMM-GW anchoring the IP flow to be recovered is given by $\overline{N}_{LANP}$ hops of adjacent DMM-GWs, leading to the expression below:

$$T_{\textit{flow-rec}} = T_{L2} + RTT_{\text{access}} + \overline{N}_{LANP}\,RTT_{\text{DMM-GW–DMM-GW}} + 3\,T^P\,, \tag{7.52}$$

and its simplified version:

$$T_{\textit{flow-rec}} = T_{L2} + RTT_{\text{access}} + \overline{N}_{LANP}\,RTT_{\text{DMM-GW–DMM-GW}} + 3\,T^P\,. \tag{7.53}$$

After a handover, the MN establishes IPv6 connectivity with the current DMM-GW after the usual router discovery procedure, hence after the reception the LANP contained in the RA sent by the DMM-GW. So we obtain:

$$T_{IP} = RTT_{\text{access}} + \overline{N}_{LANP}\,RTT_{\text{DMM-GW–DMM-GW}} + 3\,T^P\,, \tag{7.54}$$

and

$$T_{\textit{flow-rec}} = T_{L2} + T_{IP}\,, \tag{7.55}$$

as for the PMIPv6 protocol.

7.4.5 Partial N-DMM

The partially distributed solutions are somehow more articulated in the handover procedure, which is depicted in Fig. 7.6 for all the three variants. We next take the study of each one by one.

The "relay" solution. With the help of Fig. 7.6(a) we observe that the two DMM-GWs involved in the IP flow recovery, perform the tunnel setup task at very different times. This leads to packet dropping for the whole time it takes to the mobility signalling to travel from the old DMM-GW to the current one through the CMD. Indeed, in this interval, the tunnel is set up only at the old DMM-GW, thence, the encapsulated packets are discarded at the current DMM-GW until a tunnel interface is up and able to process them. From such considerations we get:

$$T^{\text{relay}}_{\textit{flow-rec}} = T_{L2} + RTT_{\text{MN–DMM-GW}} + 3\,T^{P}_{\text{DMM-GW}} + 2\,RTT_{\text{DMM-GW–CMD}} + 2\,T^{P}_{\text{CMD}}\,, \quad (7.56)$$

and with the simplifications of Eq. (7.39):

$$T^{\text{relay}}_{\textit{flow-rec}} = T_{L2} + RTT_{\text{access}} + 2\,RTT_{\text{core}} + 5\,T^{P}\,. \quad (7.57)$$

As for the full N-DMM protocol, the MN receives the parameters to establish IPv6 connectivity with the RA message sent by the DMM-GW, so T_{IP} for the "relay" solution is:

$$T^{\text{relay}}_{IP} = RTT_{\text{access}} + 2\,RTT_{\text{core}} + 5\,T^{P}\,, \quad (7.58)$$

and the final expression for $T_{\textit{flow-rec}}$ can be written as:

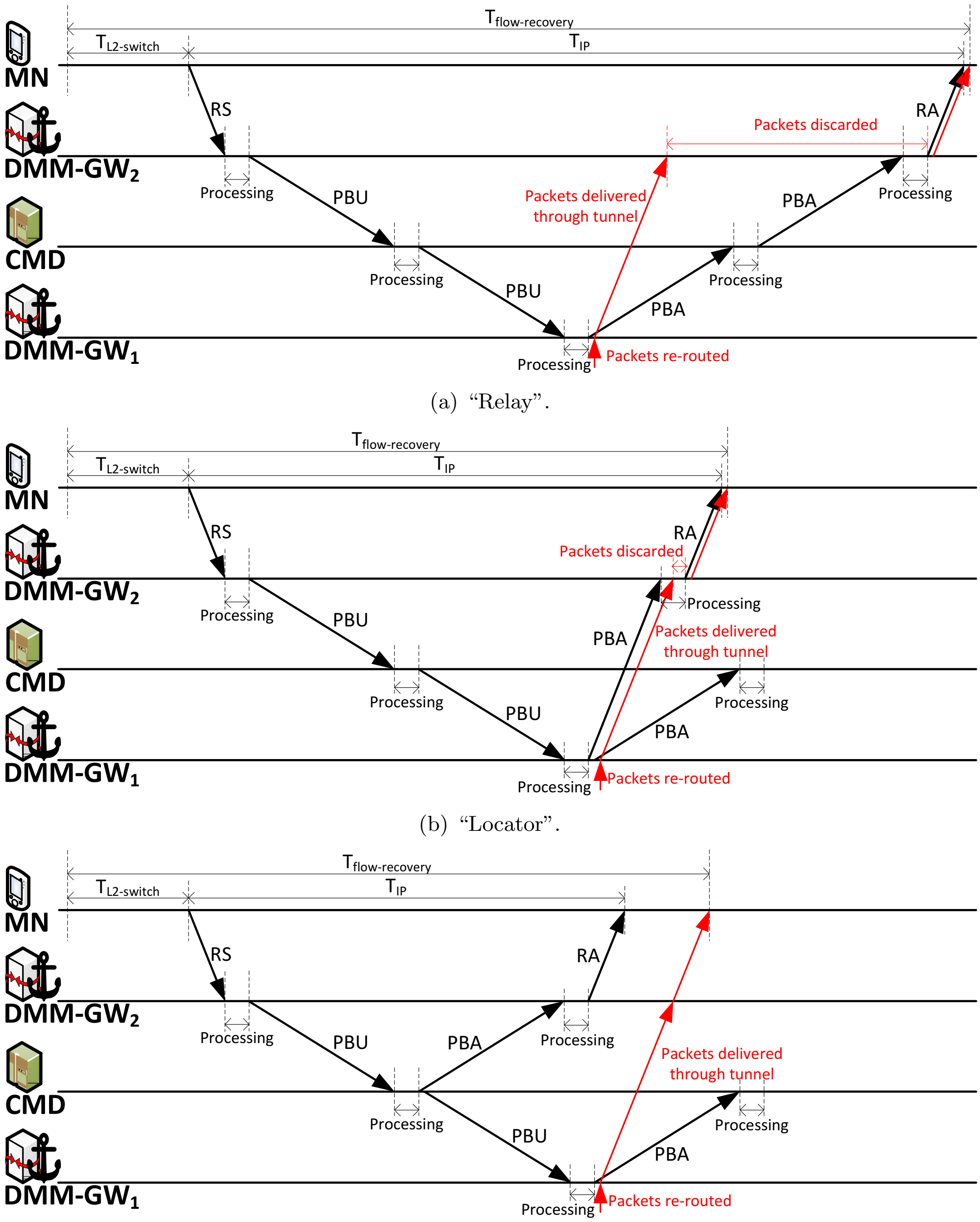
$$T^{\text{relay}}_{\textit{flow-rec}} = T_{L2} + T^{\text{relay}}_{IP}\,. \quad (7.59)$$

The "locator" solution. The "locator" solution overcomes the packets dropping problem seen in the previous paragraph for the "relay" solution as the old DMM-GW signals directly to the new one the parameters to set up the tunnel endpoint for IP flows recovery. This feature is evident from Fig. 7.6(b), upon which we can derive the following:

$$\begin{aligned} T^{\text{locator}}_{\textit{flow-rec}} =& T_{L2} + RTT_{\text{MN–DMM-GW}} + RTT_{\text{DMM-GW–CMD}} + T^{P}_{\text{CMD}} + \\ &+ \frac{\overline{N}_{LANP}}{2} RTT_{\text{DMM-GW–DMM-GW}} + 3\,T^{P}_{\text{DMM-GW}} \end{aligned} \quad (7.60)$$

and

$$T^{\text{locator}}_{\textit{flow-rec}} = T_{L2} + RTT_{\text{access}} + RTT_{\text{core}} + \frac{\overline{N}_{LANP}}{2} RTT_{\text{DMM-GW–DMM-GW}} + 4\,T^{P}\,. \quad (7.61)$$



(a) “Relay”.

(b) “Locator”.

(c) “Proxy”.

Figure 7.6: Handover latency for partially N-DMM.

For the term T_{IP} we observe that the MN must receive the LANP in the RA message to configure IP connectivity with the DMM-GW, so that

$$T_{IP}^{\text{locator}} = RTT_{\text{access}} + RTT_{\text{core}} + \frac{\overline{N}_{LANP}}{2} RTT_{\text{DMM-GW–DMM-GW}} + 4\,T^{P}\,, \tag{7.62}$$

and

$$T_{\textit{flow-rec}}^{\text{locator}} = T_{L2} + T_{IP}^{\text{locator}}\,. \tag{7.63}$$

The "proxy" solution. In the "proxy" partial N-DMM solution, the old and new DMM-GWs are instructed in parallel by the CMD, so that the tunnel endpoints are created almost simultaneously. By inspecting the timeline deployed in Fig. 7.6(c), we get

$$\begin{aligned} T_{\textit{flow-rec}}^{\text{proxy}} =& T_{L2} + RTT_{\text{MN–DMM-GW}} + RTT_{\text{DMM-GW–CMD}} + T_{\text{CMD}}^{P} + \\ &+ \frac{\overline{N}_{LANP}}{2} RTT_{\text{DMM-GW–DMM-GW}} + 2\,T_{\text{DMM-GW}}^{P}\,, \end{aligned} \tag{7.64}$$

that can be simplified into

$$T_{\textit{flow-rec}}^{\text{proxy}} = T_{L2} + RTT_{\text{access}} + RTT_{\text{core}} + \frac{\overline{N}_{LANP}}{2} RTT_{\text{DMM-GW–DMM-GW}} + 3\,T^{P}\,. \tag{7.65}$$

The "proxy" solution enables the MN to gain global IPv6 connectivity with the new DMM-GW before the "relay" and "locator" solutions, since the new serving DMM-GW receives the acknowledgement quicker from the CMD, and thus it can dispatch the RA message in advance. T_{IP} is computed as follows:

$$T_{IP}^{\text{proxy}} = RTT_{\text{access}} + RTT_{\text{core}} + 3\,T^{P}\,, \tag{7.66}$$

leading to an alternative expression for $T_{\textit{flow-rec}}$ as:

$$T_{\textit{flow-rec}}^{\text{proxy}} = T_{L2} + T_{IP}^{\text{proxy}} + \frac{\overline{N}_{LANP}}{2} RTT_{\text{DMM-GW–DMM-GW}}\,. \tag{7.67}$$

7.4.6 Handover latency and packet loss comparison

We have observed that the crucial aspect of a handover is the interval in which an MN is not able to send or receive packets belonging to ongoing IP flows, incurring in a potential packet loss if such packets are not buffered somewhere in the network (see Eq.(7.37)). In fact, our designs do not envision any mechanism to prevent packet loss to occur, so here we finally draw a comparison between the packet loss introduced by our DMM solutions and their centralized counterpart. We take into consideration first the client based mobility protocols and next the network based ones.

Client based mobility protocols: C-DMM vs. MIPv6. The potential packet loss depends on the time required by the protocol to recover ongoing IP flows during a handover. For MIPv6 and C-DMM this time is expressed by the formulae in Eqs. (7.41) and (7.45) respectively. From these equations we obtain the ratio:

$$\frac{PL^{\text{C-DMM}}}{PL^{\text{MIPv6}}} = \frac{T^{\text{C-DMM}}_{\textit{flow-rec}}}{T^{\text{MIPv6}}_{\textit{flow-rec}}} = \frac{T_{L2} + RTT_{\text{access}} + 2\,\overline{N}_{LANP}\,RTT_{\text{DMM-GW–DMM-GW}} + 6\,T^P}{T_{L2} + RTT_{\text{access}} + RTT_{\text{core}} + 4\,T^P} \tag{7.68}$$

Eq. (7.68) presents a number of variables that make it difficult to treat, so we introduce some synthetic parameters to evaluate it. In particular, the terms $T_{L2} + RTT_{\text{access}}$ are the same in both C-DMM and MIPv6 and the processing time T^P is negligible compared to the other terms. For these reasons, the following form factors may be used:

$$\rho = \frac{RTT_{\text{DMM-GW–DMM-GW}}}{T_{L2} + RTT_{\text{access}}} \tag{7.69a}$$

$$\sigma = \frac{RTT_{\text{core}}}{RTT_{\text{DMM-GW–DMM-GW}}} \tag{7.69b}$$

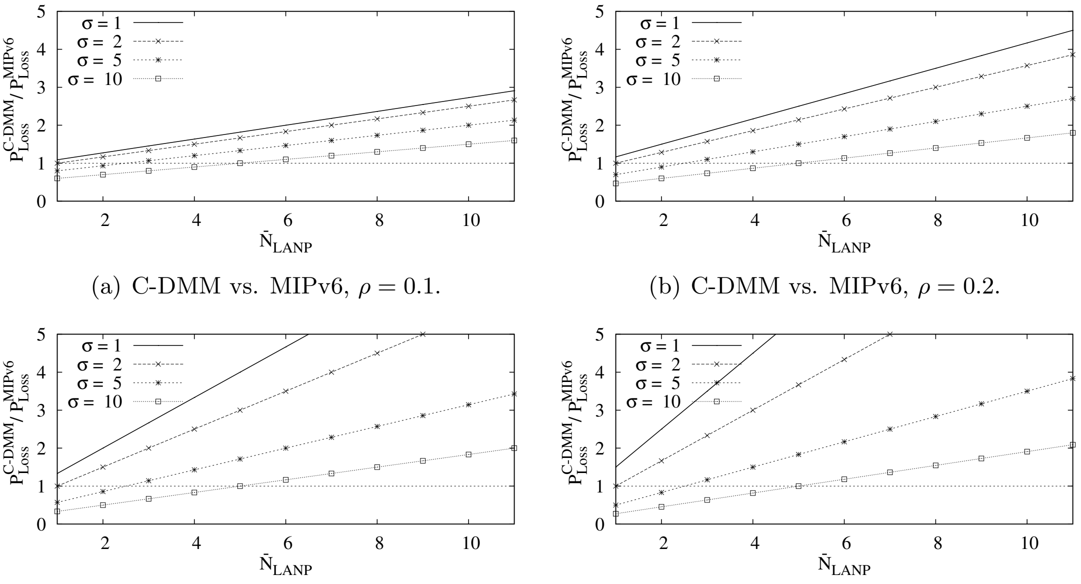
$$\rho\,\sigma = \frac{RTT_{\text{core}}}{T_{L2} + RTT_{\text{access}}} \tag{7.69c}$$

$$T^P \sim 0 \tag{7.69d}$$

so that Eq. (7.68) can be simplified into:

$$\frac{PL^{\text{C-DMM}}}{PL^{\text{MIPv6}}} = \frac{1 + 2\,\overline{N}_{LANP}\,\rho}{1 + \rho\,\sigma}\,. \tag{7.70}$$

The set of diagrams of Fig. 7.7 plots the ratio in Eq. (7.70) as a function of $\overline{N}_{LANP}$ for various values of ρ and σ. The form factor ρ reflects how well connected are adjacent DMM-GWs ($RTT_{\text{DMM-GW–DMM-GW}}$) with respect to the time necessary to perform a Layer-2 handover plus the latency on the access link ($T_{L2} + RTT_{\text{access}}$). This latter component is determined by the access technology, so that the network administrator can plan the DMM-GWs deployment based on the specific radio technology used. The form factor σ captures the size of the operator's network, as it compares the average distance between adjacent DMM-GWs and the average distance between a DMM-GW and the core entities. By observing the graphs of Fig. 7.7, it is clear that C-DMM outperforms MIPv6 in very large networks, where $\sigma \geq 5$, and in scenarios where IP flows are not much longer than the cell residence time, i.e., for $\overline{N}_{LANP} \leq 5$. We recall that our study reflects the worst case scenario where the MN keeps moving from one DMM-GW to the next in a linear way, so that the distance between the serving DMM-GW and the anchor DMM-GW keeps increasing of one DMM-GW–DMM-GW hop at each handoff (see Fig. 7.1).



(a) C-DMM vs. MIPv6, $\rho = 0.1$.

(b) C-DMM vs. MIPv6, $\rho = 0.2$.

(c) C-DMM vs. MIPv6, $\rho = 0.5$.

(d) C-DMM vs. MIPv6, $\rho = 1$.

Figure 7.7: Packet loss comparison C-DMM vs. MIPv6.

Network based mobility protocols: N-DMM vs. PMIPv6. Similarly to what was presented in the previous paragraph, the potential packet loss for the network based mobility protocol can be expressed by the Eqs. (7.49), (7.53), (7.57), (7.61) and (7.65), respectively for PMIPv6, the full DMM, the "relay", the "locator" and the "proxy" solutions.

By applying the same methodology of the previous paragraph and the parametrization of Eqs. (7.69), we obtain the following expressions for the packet loss ratio of the the N-DMM protocols with respect to PMIPv6:

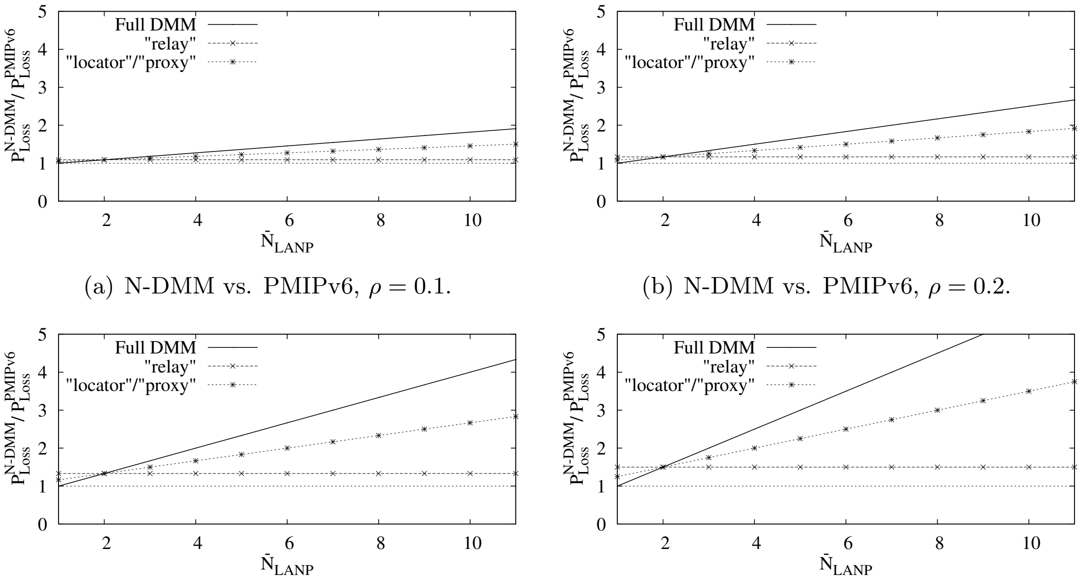
$$\frac{PL^{\text{Full}}}{PL^{\text{PMIPv6}}} = \frac{1 + \overline{N}_{LANP}\,\rho}{1 + \rho\,\sigma} \tag{7.71a}$$

$$\frac{PL^{\text{relay}}}{PL^{\text{PMIPv6}}} = \frac{1 + 2\,\rho\,\sigma}{1 + \rho\,\sigma} \tag{7.71b}$$

$$\frac{PL^{\text{locator}}}{PL^{\text{PMIPv6}}} = \frac{1 + \rho\,\sigma + \frac{\overline{N}_{LANP}}{2}\,\rho}{1 + \rho\,\sigma} \tag{7.71c}$$

$$\frac{PL^{\text{proxy}}}{PL^{\text{PMIPv6}}} = \frac{1 + \rho\,\sigma + \frac{\overline{N}_{LANP}}{2}\,\rho}{1 + \rho\,\sigma} \tag{7.71d}$$

The diagrams in Figures 7.8, 7.9, 7.10 and 7.11 depict Eqs. (7.71) for different values of the form factors ρ and σ.



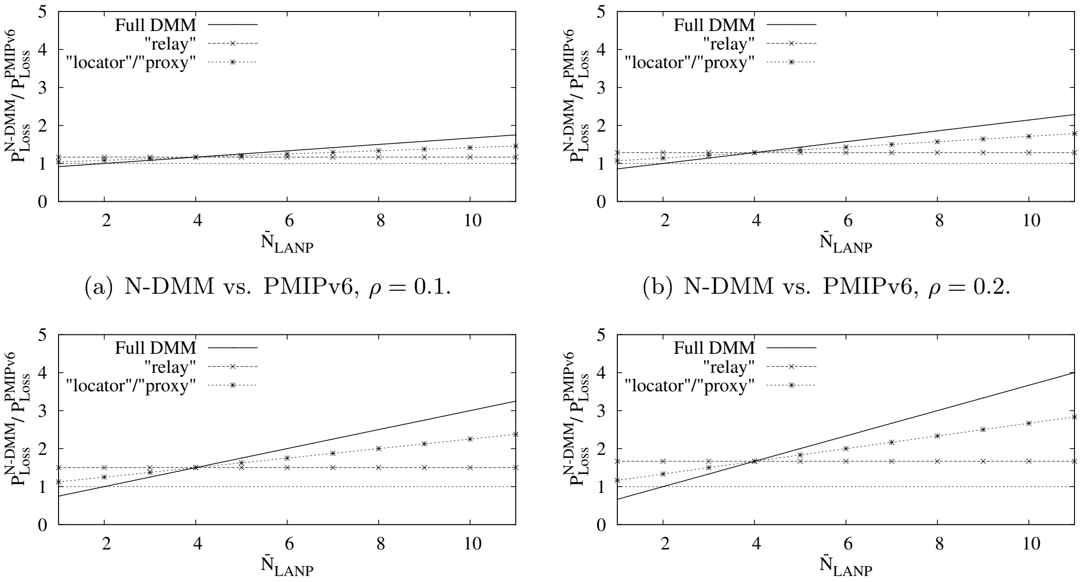
(a) N-DMM vs. PMIPv6, $\rho = 0.1$.

(b) N-DMM vs. PMIPv6, $\rho = 0.2$.

(c) N-DMM vs. PMIPv6, $\rho = 0.5$.

(d) N-DMM vs. PMIPv6, $\rho = 1$.

Figure 7.8: Packet loss comparison N-DMM vs. PMIPv6: $\sigma = 1$



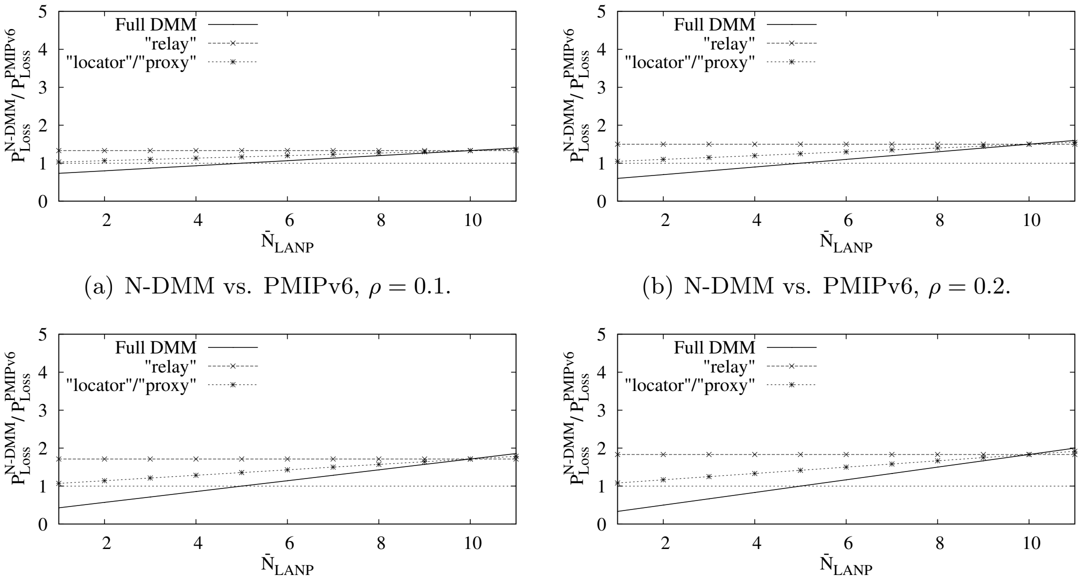
(a) N-DMM vs. PMIPv6, $\rho = 0.1$.

(b) N-DMM vs. PMIPv6, $\rho = 0.2$.

(c) N-DMM vs. PMIPv6, $\rho = 0.5$.

(d) N-DMM vs. PMIPv6, $\rho = 1$.

Figure 7.9: Packet loss comparison N-DMM vs. PMIPv6: $\sigma = 2$



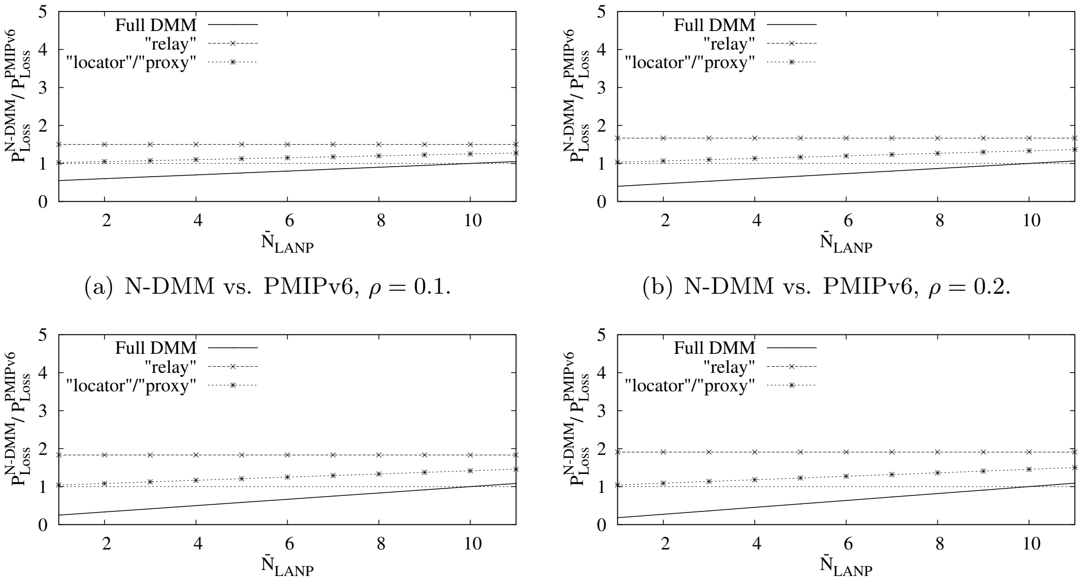
(a) N-DMM vs. PMIPv6, $\rho = 0.1$.

(b) N-DMM vs. PMIPv6, $\rho = 0.2$.

(c) N-DMM vs. PMIPv6, $\rho = 0.5$.

(d) N-DMM vs. PMIPv6, $\rho = 1$.

Figure 7.10: Packet loss comparison N-DMM vs. PMIPv6: $\sigma = 5$



(a) N-DMM vs. PMIPv6, $\rho = 0.1$.

(b) N-DMM vs. PMIPv6, $\rho = 0.2$.

(c) N-DMM vs. PMIPv6, $\rho = 0.5$.

(d) N-DMM vs. PMIPv6, $\rho = 1$.

Figure 7.11: Packet loss comparison N-DMM vs. PMIPv6: $\sigma = 10$

From the pictures we observe that, among the N-DMM protocols, only the full DMM solution can perform better than PMIPv6, but only in those networks with a large edge–core distance (see Figures 7.10 and 7.11). In particular, the larger the mobility domain, the higher the value of $\overline{N}_{LANP}$ that can be reached while still having better performance than PMIPv6. Nevertheless, the flow recovery latency in the full DMM solution increases quicker than in the rest of solutions, deteriorating the performance up to be worse than the partially DMM protocols.

All the partial DMM solutions perform worse than PMIPv6 because we assumed that the CMD resides where the LMA is. However, since the CMD does not aggregate traffic, it does not have constraints for its deployment, giving the network administrator the flexibility to place it closer to the edge, reducing the size of the DMM domain. Indeed, a future research item within the partially distributed DMM approach is how to deploy multiple CMDs, so to reduce the CMD's scope and thence be placed very close to the edge part of the network (see Chapter 10).

7.5 Final remarks

From the analyses presented in this chapter, a plethora of scenarios emerged where either DMM or CMM performs better, but there is not a clear general case where one outperforms the other. For instance, DMM fits best in large networks, where the communication latency between the edge and the core is substantially higher than the latency between adjacent access networks. Nevertheless, if an access network coverage is too small, then the DMM-GW–DMM-GW delay can be reduced, but it is more likely to have a high number of handovers, with the consequent signaling overhead to maintain many active anchors.

Nevertheless, DMM performs better in what appears to be the most common scenario for the majority of users. This scenario comprises *i)* low mobility, as users tend to transfer following defined patterns within a few Km^2 area; *ii)* short lived IP flows, generated by applications that in many cases do not need address continuity (non-real time applications); *iii)* closeness to the correspondent node, due to the nature of human interactions, but also to the increasing penetration of Content Delivery Networks.

Therefore, in order to evaluate the trade off between the costs and benefits of CMM and DMM, a network administrator has to take in consideration the network size and the typical traffic and mobility patterns observed in its infrastructure. We can argue that a Mobile Network Operator in a medium sized European country might deploy few tens of DMM-GWs, so that the probability for a user to change DMM-GW is very low, but, still, enabling the MNO to provide local breakouts to the data services without the costs to carry the traffic to the core.

Chapter 8

Experimental validation

In this chapter we describe the experimental evaluation conducted over a prototype platform realized to host some implementations inspired by the DMM paradigm.

One of the initial objectives of the DMM implementation process was to produce a completely functional prototype of the fully distributed solution in the IEEE 802.21 flavour (see Section 5.3.1), to be used in the MEDIEVAL project.

Nevertheless, the implementation process implied some intermediate stages that still gained importance *per se*, like the prototype for the partially distributed DMM solution in the "proxy" variant (Section 5.2.3), and described next in Section 8.1. This prototype served as demonstrator too, at international conference and other events.

Moreover, this chapter also collects the implementation efforts produced to compare the "proxy" solution with other designs from two different DMM approaches, namely the SDN-based approach and the routing-based approach. This study is presented in Section 8.1.3.

8.1 The "proxy" N-DMM implementation

The testbed deployed with the implementation of the "proxy" N-DMM solution comprises a set of DMM-GWs (up to 5 nodes), one CMD and one CN. These entities are interconnected as illustrated in the schematic representation of Fig. 8.1 by a transport network made up of IPv6 routers.

The DMM-GWs and CMD's features are realized by a software daemon called *Mobility Anchors Distribution for Proxy Mobile IPv6 (MAD-PMIPv6)*. MAD-PMIPv6 is developed in C from an existing MIPv6/PMIPv6 implementation from the UMIP project[1]. MAD-PMIPv6 is open source and available for download along with the documentation at `http://www.odmm.net/mad-pmipv6/`.

[1] `http://www.umip.org/`

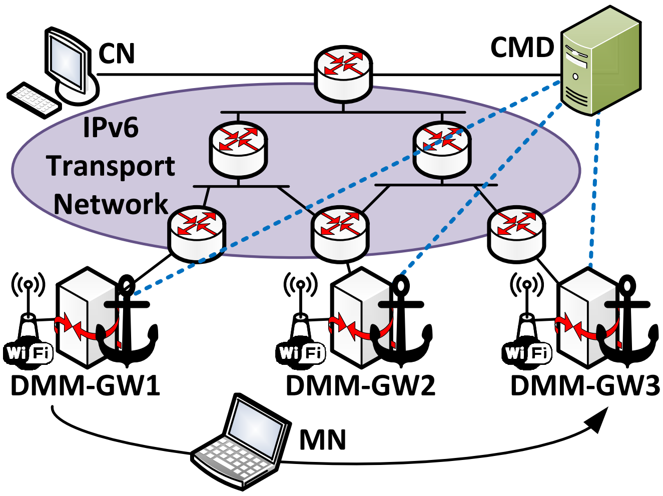


Figure 8.1: Testbed deployed for the “proxy” partially distributed solution.

Each DMM-GW is the access router for a WLAN network realized by an IEEE 802.11b/g wireless card plugged to the DMM-GW machine.

The hardware and software requirements for the MN are loose, being simply a Linux machine equipped with an IEEE 802.11b/g wireless card and a standard IPv6 stack implementation.This is due to the network-based nature of the DMM mobility solution.

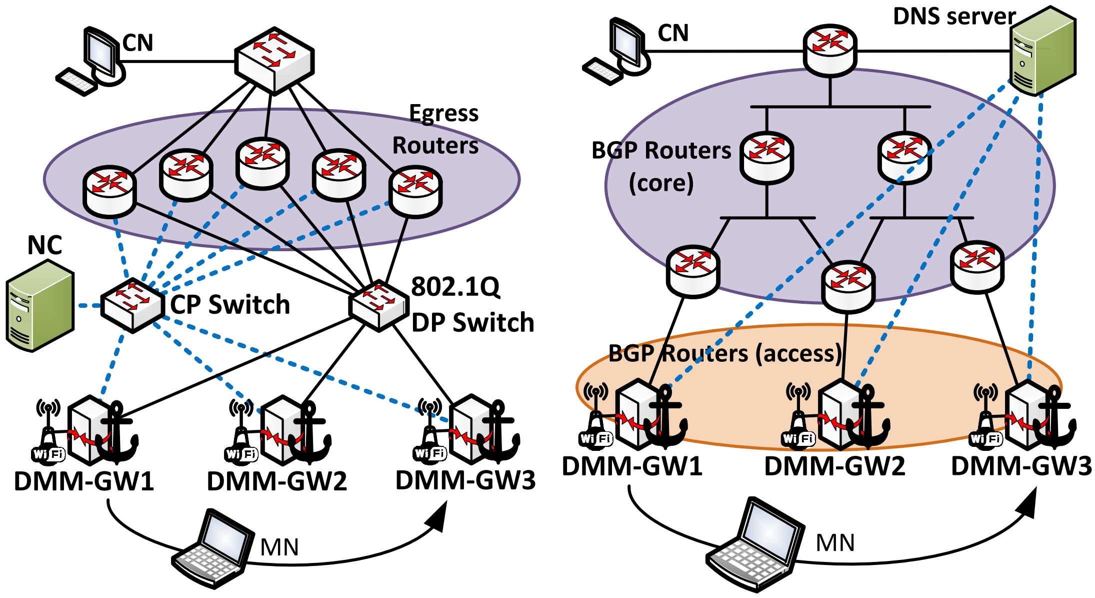
The prototype allows the MN to connect to any DMM-GW and establish IP sessions with the correspondent node.The MN is enabled to roam within the domain, i.e., the three WLANs provided by the DMM-GWs, while enjoying the IP services provided by the CN without experiencing any service interruption.

The objective of the experiments is to observe how an MN reacts when a handover occurs, that is, what happens to the data traffic when the MN moves from one DMM-GW to the other. This study has been conducted as a comparison with the “proxy” DMM prototype and two more implementations from other DMM approaches. One approach adopts the Software Defined Networking (SDN) paradigm, and thus we have designed and implemented an SDN-based DMM solution. The other approach envisions the use of IP routing protocols for mobility support, and so we have implemented the BGP-based DMM solution described in [79, 80]. We next briefly describe these two additional prototypes.

8.1.1 SDN-based solution and implementation

The SDN paradigm decouples the system making decisions about routing (i.e., the control plane) from the underlying system that forwards traffic to the selected destination (i.e., the data plane). This approach, among other advantages, allows a quicker provisioning and configuration of network connections, highly improving the flexibility of the network. Therefore, SDN is a suitable tool for mobility support.

In our SDN-based DMM solution, a central entity called Network Controller (NC) im-



(a) Testbed for the SDN-based solution.

(b) Testbed for the BGP-based solution.

Figure 8.2: Testbeds for DMM scenarios.

plements the control logic in charge of configuring the forwarding rules of all the switching nodes and routers in the network. This operation is realized using a common Application Programming Interface (API) supported by all the involved network entities. The *de facto* standard for such API is OpenFlow[2].In our design, part of the network nodes consists of the access routers providing the wireless access. We call for convenience these nodes DMM-GWs, but they should not be confused with those in the "proxy" solution, because the functions they provide are different. Then, there is a set of Egress Routers (ERs), being the routers at the edge of the SDN domain, connected to the rest of IP-based networks (e.g., the Internet). Packet forwarding within the SDN domain is based on VLANs and statically configured. Thus, pre-configured VLAN paths connect the DMM-GWs and the ERs so that each DMM-GW has a link to each ER.

In this way, forwarding the packets destined or originated from an MN is achieved by installing OpenFlow rules at all the Egress Routers, instructing these network entities to route packets using the appropriate VLAN tag. For instance, upstream packets from the MN are tagged with the VLAN that brings to the ER that should be used to get to destination; in downstream, the ER to which the packets arrive uses the VLAN tag that leads to the DMM-GW where the MN is connected. In case of handover, the NC simply needs to rewrite this tagging rule at all the ERs, selecting the correct VLAN that connects the ERs to the new DMM-GW. Note that this solution does not involve the use of any IP tunnels and it scales as the number of deployed ERs.

The testbed for the SDN-based DMM solution is depicted in Fig. 8.2(a). We deployed

[2] https://www.opennetworking.org/

3 DMM-GWs and a set of ERs accounting for up to 5 elements, connected each other through two separate networks, one for the control plane (drawn with blue lines) and one for the data plane (the black segments). In the control plane network, a simple switch realizes the interconnection among all the nodes and also with the NC (the CP Switch node in Fig. 8.2(a)). For the data plane, we deployed and configured an 802.1Q-capable switch, for the VLAN-based forwarding, that interconnects all the DMM-GWs and ERs. Moreover, the ERs have a third link used to connect the test-bed to the CN.

All the DMM-GWs and ERs run the Linux kernel version 3.10, that includes Open vSwitch[3] for the OpenFlow 1.3 interface. The NC runs Ryu[4] as OpenFlow controller. The SDN-based solution is therefore implemented as Ryu application (i.e. based on the API provided by the NC). The connection between Open vSwitch and Ryu is performed out-of-band involving TCP for the OpenFlow messages delivery.

8.1.2 BGP-based solution and implementation

The basic concept of this approach is to remove any anchor from the architecture, letting the standard routing mechanisms in the network to re-establish a new routing map when terminals move. Our solution is based on what is proposed in [79, 80], which builds on top of the Border Gateway Protocol (BGP) [103] and the Domain Names System (DNS). This solution considers a flatter network than current deployments by enabling BGP on the access routers (the DMM-GWs), so that they propagate upwards to the their BGP peers the changes in the access links.

Upon an MN attachment to any DMM-GW's access link, the access router learns the MN's DNS name through the authentication process. Next, through a DNS lookup in the `inaddr.arpa` or `ip6.arpa` space, the DMM-GW retrieves the IP address (and consequently the IPv6 prefix) associated to the MN DNS record and triggers a routing update in the rest of the network, announcing itself as next-hop to reach the MN prefix. This is done by originating a BGP UPDATE message and sending it to its parent routers in the aggregation layer. In case there are no route reflectors, the update must be sent to all BGP peers in the domain. Otherwise, the message will be reflected down to all BGP routers in the local domain, saving the number of messages sent by the DMM-GW.

Fig. 8.2(b) depicts the BGP-based test-bed. It is composed by a set of DMM-GWs and some additional BGP routers. The group of routers marked as *BGP Routers (core)* connect the DMM-GWs to the DNS server and to the CN. The DMM-GWs are marked as the *BGP Routers (access)* group to highlight the fact that they actively send BGP updates to the rest of BGP peers, while those in the core simply receive the notification and do not redistribute them. The total number of BGP routers in the testbed spans from 3 to 10 elements. The dashed blue lines simply represent the communication between the

[3] `http://openvswitch.org/`

[4] `http://osrg.github.io/ryu/`

DNS server and the DMM-GWs, but it is not made over dedicated links.

The BGP-based solution employs several implementations. The DNS server is realized with Bind9[5]. The BGP implementation used in all the nodes is from the Quagga project[6]. In addition, the DMM-GWs run a custom piece of software that performs the detection of the wireless events (attachment and detachment), the DNS queries, and it triggers the Quagga's BGP routing daemon to install/remove the local downlink route for the MN. The Quagga daemon then propagates the routing update to all the other BGP speakers, i.e., the whole BGP Routers (core) group and the remaining DMM-GWs. Like the SDN-based solution, this does not employ tunneling, but it floods the network with a routing update message for each router at every handover event.

8.1.3 Experimental tests and comparison

The objective of the experiments is to observe the behavior upon handover in all systems, and what happens to the data traffic when the MN moves from one DMM-GW to the other. For this purpose, the correspondent node generates *ping* traffic destined to the MN every 2 ms, which is below the average MN–CN RTT measured.

For each implementation, we have measured with an usual packet capturing tool[7] installed in the MN the following three events that characterize each handover :

1. The *Layer-2 handover* is measured as the interval between the first and last IEEE 802.11 control messages sent during a handover: *Deauthentication*, sent by the MN to the old AP to disconnect from it, and *Association response* received by the MN from the new AP to conclude the link establishment;
2. the *Layer-3 configuration* is the time spent since the *Deauthentication* message, to the instant when an RA message is received by the MN[8];
3. the *IP flow recovery* is measured as the interval between the last *ping* packet received or sent by the MN before the handover and the first *ping* packet received or sent after the handover.

Table 8.1 reports the mean and the standard deviation values of the results obtained for a handover in the following deployments:

i) $N = 3$ DMM-GWs in the “proxy” test-bed,
ii) $N = 3$ ERs in the SDN-based prototype,
iii) $N = 4$ BGP routers for the BGP-based platform (two BGP routers in the access and two in the core).

[5] `http://www.bind9.com/`

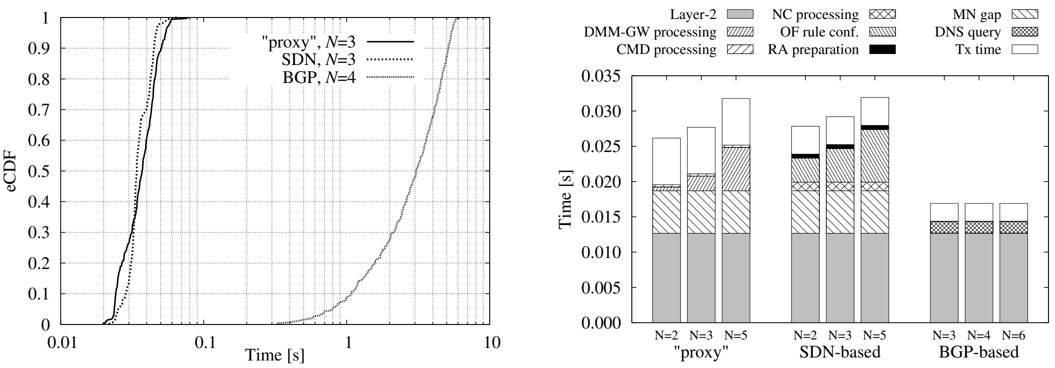
[6] `http://www.nongnu.org/quagga/`

[7] Wireshark, `http://www.wireshark.org/`

[8] The IPv6 Duplicate Address Detection is disabled since the prefix is uniquely assigned to the MN, thus it is not a necessary process.

Table 8.1: Handover latency experimental results

Type of solution		**L2 ho. (ms)**		**L3 conf. (ms)**		**IP flow rec. (ms)**	
		Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
"proxy"	$N = 3$	12.9	4.4	27.7	7.2	37.4	9.9
SDN-based	$N = 3$	12.9	4.4	29.2	4.9	35.6	6.7
Routing-based	$N = 4$	12.9	4.4	16.9	3.9	3107	1490



(a) Flow-recovery time eCDF.

(b) Layer-3 latency composition.

Figure 8.3: Results from the experimental comparison.

Note that the above choice leads to have 3 network elements involved in the handover signaling for all platforms[9]. Fig. 8.3(a) depicts the empirical CDF for the values of the *ping* recovery time in the same test scenario. Fig. 8.3(b) explores in detail the components of the Layer-3 configuration time for varying values of $N \in \{2, 3, 5\}$ in the "proxy" case, $N \in \{2, 3, 5\}$ in the SDN-based case and, $N \in \{3, 4, 6\}$ in the BGP-based case.

As it can be noticed from the results, all the three solutions take few tens of milliseconds to provide the MN with the IPv6 configuration, wherein the Layer-2 switch time is the major term. More, we observed a 5 ms gap between the instant the MN receives the IEEE 802.11 *Association response* message, and the time it sends the RS message to the DMM-GW. This gap, denoted "MN gap" in Fig. 8.3(b), is not present in the BGP-based solution, because the DMM-GW employs a dedicated detection mechanism for the Layer-2 link activation and de-activation. The BGP-based solution is thus the quickest to set up the Layer-3 configuration, also because the remaining portion of time is spent for the DNS query and the RA transmission time over the wireless link. In addition, the Layer-3 configuration time is constant for this solution regardless the number of nodes in the network. In the other two solutions, we can separate the component due to message transmission, which depends on the sum of the RTT between the MN and the DMM-GW

[9] Recall that in the BGP-based solution, if there are N elements deployed in the network, then there are $N - 1$ BGP sessions per handover, that is why the tests for the BGP-based solution envision 1 additional element compared to the other two solutions.

and the RTT between the DMM-GW and the CMD or NC, respectively for the "proxy" or the SDN-based solution. In our laboratory tests, all the nodes are close to each other, and such RTT sum is less than 5 ms. In a real deployment, with larger RTT values, the Layer-3 configuration time would tend to approximate the air time plus the distance from the central node to the farthest router involved in the signalling. The rest of components are due to processing at the network nodes and they tend to scale with the number of entities involved in the handover operations. In the "proxy" case, the heaviest burden is on the DMM-GW, because of the tunnels and routes set up, so, the largest the number of previous DMM-GWs, the longer the latency. The CMD is mainly answering to a query, so its task is accomplished much quicker. In the SDN-based case, the NC has to compute and send the rules to the ERs. In addition it has to process the RS from the MN and prepare the RA message.

The *ping* traffic recovery time, beyond the handover operations, is affected by the tasks distributed in different elements of the network, as routers and switches. For example, packets are queued at the nodes' interfaces, introducing random delays that are complex to capture. Therefore we limit our analysis on some macroscopic effects that we observed during the experiments. The most remarkable result is that, in the BGP-based solution, it takes roughly one hundred times more than in the other two solutions for the *ping* to be resumed. The reason resides in the implementation choices adopted by the Quagga developers to let the routing daemon react to changes in the routing tables. From a theoretical point of view, the new path for the *ping* packets is ready as soon as all the routers in the target path receive the BGP UPDATE messages from the DMM-GW, hence, if the UPDATE messages are sent back-to-back to the BGP peers right upon the new route is added, the interval tends to approximate the RTT with the farthest peer. However, in practice, routing protocols are not designed to react immediately to changes in the network, in order to reduce possible ping-pong effects and the consequent flood of messages. In this sense, routing protocols do not fit to high mobility scenarios, as those typical of mobile networks. In our prototype, we observed that in the Quagga BGP routing daemon it takes on average 2.997 seconds since a new route is installed until the daemon starts to distribute the update messages back-to-back. On the contrary, when a route is removed, Quagga reacts on average in 68.1 ms.

In the other two solutions, the *ping* recovery is roughly 10 ms higher than the Layer-3 configuration time for the "proxy" solution, and around 6 ms for the SDN-based one. For the latter solution, the reason is mainly due to the time required by the nodes to receive and install the rules sent by the NC, and then for the *ping* packets to flow from the egress router to the MN. In the "proxy" case, old DMM-GWs are notified by the CMD after the new DMM-GW, they update the tunnel parameters and routes to point to the new location and then they can forward packets that flow through the tunnel and finally are delivered to the MN.

8.2 DMM for mobile video caching

Besides the efforts to design a flatter network, content caching is seen as an additional valuable resource to help reducing traffic flowing in the core. Mobile Content Delivery Networks (mCDN) are expected to be key supporters tackling the avalanche of traffic, by pushing content closer to the user and therefore limiting the congestion in core links. Articles such as [104] and [105] explore the benefits respectively for 3G and 4G systems, concluding that up to two thirds of mobile traffic can be reduced.

The study in [106] surveys the current strategies for caching in mobile networks, showcasing which techniques are most suitable according to the cache locations. Following the analysis in [106], a mature object-oriented caching technique, like URL-based web caching, finds a natural application with caches co-located or placed by the PGW, being it the point where packets are no longer GTP-encapsulated, and a plain HTTP proxy server would accomplish the task. In this context, authors of [107] propose to place caches at the PGW for a centralized deployment, or at the LGW for a distributed one, and then they explore an application level content delivery optimization based on a smart scheduler.

On the contrary, caching at the RAN level requires more sophisticated caching techniques that perform data flow monitoring inspecting the packets in more depth. These methods are usually referred as Redundancy Elimination (RE) and the objective is to remove arbitrary duplicated byte patterns from the traffic flowing within a portion of the network, in this case the EPC. Interested readers might find more details in [108], where a flow monitoring tool has been deployed in a real operator's EPC, and a RE technique applied to TCP flows has been implemented.

In line with the DMM philosophy, in this section we propose the **CDN+DMM** caching solution, that exploits the flat nature of a DMM protocol by placing the CDN caches close or co-located with the DMM-GWs. We elaborate next the architectural details and then we describe the experiments conducted over the testbed implemented for the EU project MEDIEVAL (see the project's deliverables [109–111]). In our framework, we consider video traffic as the target of our study, as it is foreseen to play a dominant role in the years to come [1].

8.2.1 The CDN+DMM architecture and operations

The use of any of the DMM solutions described in the previous chapters allows the mobile user to be topologically closer to its point of connection to external IP services, as compared to current 3GPP architectures. Therefore, the user experience can be enhanced by placing content and services close the DMM-GWs, significantly reducing at the same time the load in the network core. Following this key concept, we propose the deployment of the fully distributed DMM solution seen in Section 5.3.1, integrated with a mobile

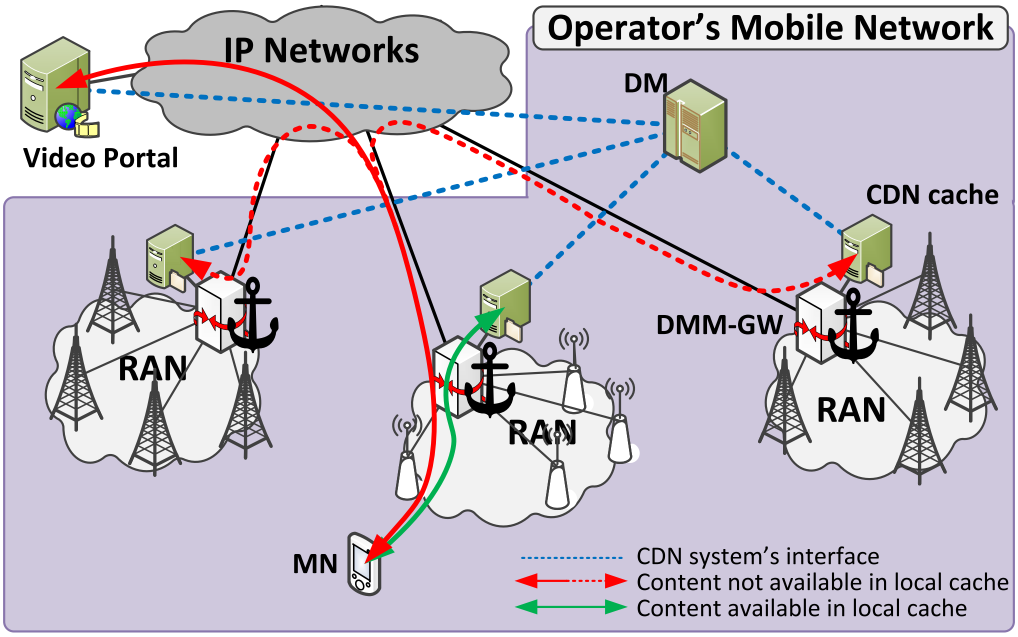

IP Networks
Operator's Mobile Network
Video Portal
DM
CDN cache
DMM-GW
RAN
RAN
RAN
MN
CDN system's interface
Content not available in local cache
Content available in local cache

Figure 8.4: Global system architecture.

CDN in which the content caches are co-located with the DMM-GWs. The architecture is illustrated in Fig. 8.4.

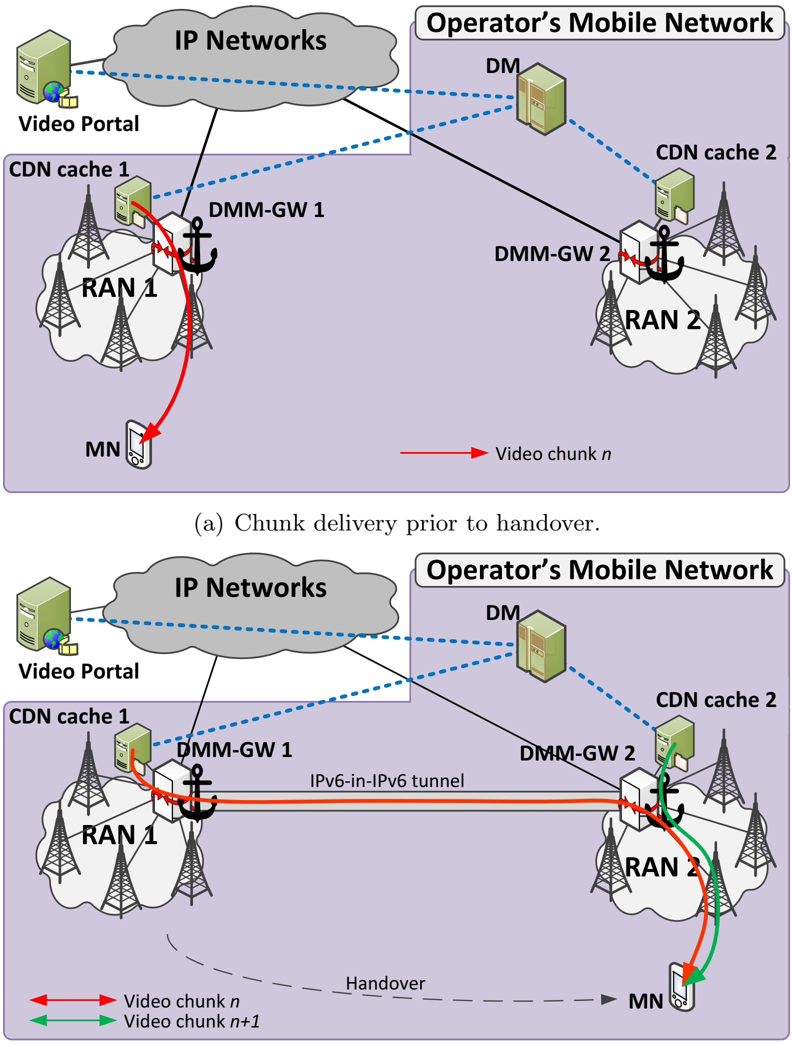
In the CDN+DMM architecture, the CDN system comprises a set of CDN nodes, i.e., the caches co-located with the DMM-GWs, and a controller, called Decision Manager (DM), that monitors and drives the activity of the CDN nodes. The DM and the CDN nodes share a signaling interface used to process the video requests and to periodically synchronize the status of the content cached at the DMM-GWs. The totality of the video content is stored at a main repository, the Video Portal, providing a web-based catalogue to choose the videos, and connected to the DM through a control interface, as the DMM-GWs. In our scenario, the main video server belongs to the mobile operator, even if located outside the mobile network, but note that it might be a third-party service. The caches at DMM-GWs store the most popular videos and host an HTTP proxy server to perform object-oriented caching based on the object's URL. Therefore, if the requested content is locally available at the cache, the video is delivered by the DMM-GW. On the contrary, the DMM-GW delegates the DM to determine the most suitable cache to serve the request. The alternative streaming source can be either another DMM-GW that possesses that file, or the main central server. In our case the suitability metric is simply the closeness to the user, but other policies might be enforced. Thereby, our CDN infrastructure delivers the service from the closest node to the MN wherein the content is available.

The streaming technique used by the CDN+DMM system is called Dynamic Adaptive Streaming over HTTP (DASH) [112], also known as MPEG-DASH. With DASH, a video file is divided into many segments (video chunks), and downloaded by the user with HTTP, after requesting the segments' URL one by one. In addition to the video chunks, a DASH video also includes a Media Presentation Description (MPD) file, containing the list of the chunks' URLs, and other ancillary information for the video client, like the bitrate, frame resolution, etc. Thus, the video client first retrieves the video's MPD from the Video Portal, and then issues HTTP requests to fetch the video segments present in the URL list. If the chunk pointed by the requested URL is is available in the local cache, then the video segment in delivered directly by the DMM-GW, otherwise the request is sent to the processing engine at the DM, that computes the best source according to the content distribution and the MN location. Then the DM instructs the DMM-GW to retrieve the object from that source, which might be the main central server or another CDN node (DMM-GW). In both cases the traffic redirection is transparent to the MN.

In case of handover, the DMM protocol ensures that ongoing communications are not interrupted by being redirected through a tunnel established between the new DMM-GW and the old one where the communication started. In the context of the CDN+DMM system, a separate HTTP session is set up per each video chunk, therefore any chunk download starts a new communication with the server. This mechanism is exploited by the CDN+DMM system to handle the handover by switching the video cache accordingly. The operations described in the following are illustrated in Fig. 8.5. In this scenario, the user is downloading chunk n, which is locally available at the DMM-GW where the MN is connected, therefore the streaming source is the DMM-GW itself (see Fig. 8.5(a)). In the middle of the download, the MN attaches to a new DMM-GW, so the remaining portion of the chunk is transmitted through the tunnel created between the old and new DMM-GWs (see Fig. 8.5(b)). When the chunk is fully received, the video player issues a request for the next chunk $n + 1$. This request appears as a new communication, so if the chunk is present at the new DMM-GW's local cache, then it is delivered directly to the user, otherwise it is fetched from another source, using the optimal link between that DMM-GW and the source. In the ideal scenario where the content accessed by the user is replicated in all the caches, then the MN always receives the content from the same DMM-GW where it is connected, handover after handover. Even if we drawn the example of Fig. 8.5 with an N-DMM protocol, this effect is the result of using any DMM solution. In the following we describe the validation of the CDN+DMM design, and we report the performance assessment on a platform where the chosen DMM protocol is the fully-distributed solution with IEEE 802.21, presented in Section 5.3.1.

8.2.2 Validation and performance assessment

The CDN+DMM network architecture has been implemented in a prototype and validated through real field experiments. The objective of the experimental assessment that follows is to carry out a proof of concept of our design, with all the system details.



(a) Chunk delivery prior to handover.

(b) Chunk delivery after handover.

Figure 8.5: System operations with user mobility.

Test-bed and implementation details. The network prototype consists in a test-bed of GNU/Linux machines running Ubuntu 10.04 deployed as illustrated in Fig. 8.6, where we highlighted the elements interconnection along with the logical interaction between the nodes functions. The prototype exhibits three control plane systems that coordinate the data delivery to moving users:

1. the Layer-2 handover management, represented in the picture by the red boxes and dashed lines;
2. the IP mobility management, indicated by orange elements;
3. the CDN system, which boxes and interfaces are colored in light blue.

The Layer-2 handover management complies with the IEEE 802.21 protocol suite, Media Independent Handover Services. MIHS are responsible to prepare, execute and

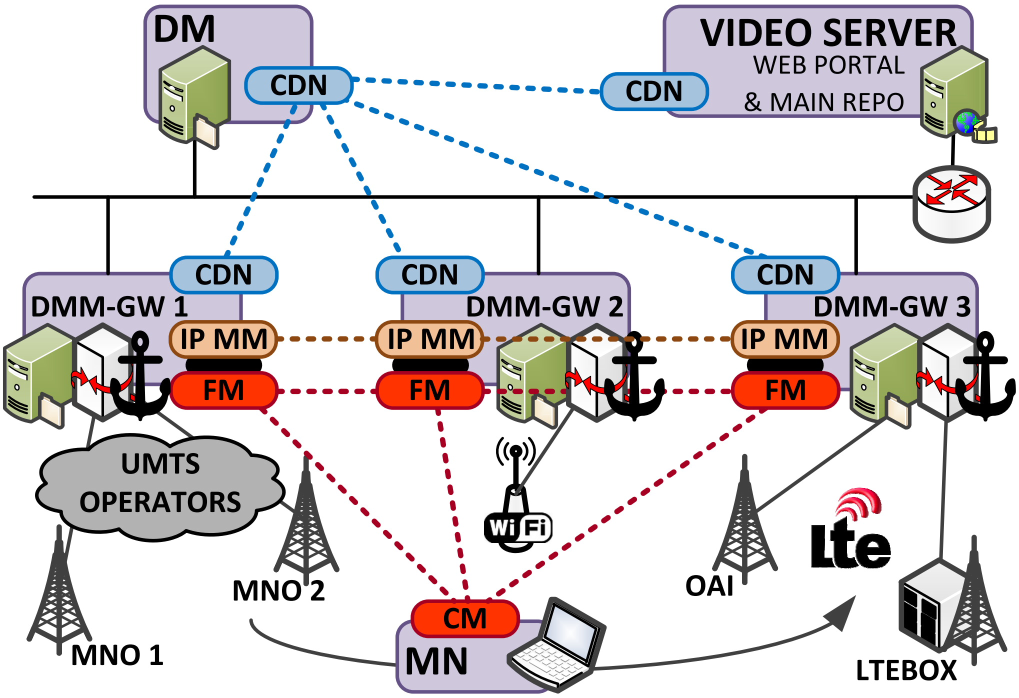


Figure 8.6: System architecture test-bed deployment.

conclude the handover except for performing the IP operations. The IEEE 802.21 implementation used is the ODTONE[10] open platform [113], installed in all the DMM-GWs and in the MN. In the MIHS framework, the MIHS-enabled functions export APIs to be used by MIHS users to execute commands and control the protocol operations. We have deployed two MIHS users: the Flow Manager (FM), installed in the DMM-GWs, and the Connection Manager (CM), running in the MN.

The IP mobility management provides the IP addresses assignment to the MNs and their reachability within the domain, and it is based on the Proxy Mobile IPv6 protocol. Its implementation has been realized modifying the code from the MAD-PMIPv6[11] project and it is installed in the DMM-GWs (IP Mobility Module - IP MM). These two systems collaborate to provide the Fully N-DMM protocol described in Section 5.3.1.

The third control mechanism deployed in the test-bed is the CDN system. Such system handles the DASH video delivery to the users and the caches population. The video service portal (i.e., the on-line video catalogue) and global repository is hosted at a remote web server (called Video Server in the picture) realized through the Apache implementation[12]. All the DMM-GWs are provided with video caches by means of the Squid[13] HTTP proxy tool. Squid intercepts the HTTP requests coming from the mobile

[10] `http://hng.av.it.pt/projects/odtone/`
[11] `http://www.odmm.net/mad-pmipv6/`
[12] `http://httpd.apache.org/`
[13] `http://www.squid-cache.org/`

users and it accommodates them if the content pointed by the URL (i.e., the video chunk) is available in the local cache, or it sends the notification to the DM. The interface between the DM and the DMM-GWs and between the DM and the video portal is realized through a custom API handled by software applications written in Perl.

Beyond the modules described above, each DMM-GW is connected to a different radio access network in the manner indicated below.

DMM-GW 1: it supplies the 3G access through tow real world UMTS networks from different commercial operators. The terminal connects alternatively to one or the other through an HSDPA USB dongle equipped with the USIM card from the corresponding operator. Then, an IPv6-over-IPv4 VPN tunnel is set up to pass through the UMTS network, and transparently connect DMM-GW1 and the MN as if they were on the same IPv6 local link.

DMM-GW 2: it offers the WiFi connectivity using a simple IEEE 802.11b/g/n network interface.

DMM-GW 3: it is responsible for the 4G, through two access modes that can be toggled. The first mode employs the Open Air Interface[14] (OAI): it is a software tool installed in both the DMM-GW and the MN that emulates over cable the LTE air interface between an eNodeB and a User Equipment (UE). The second mode adopts an Alcatel-Lucent Bell Labs LTEBOX: it consists of an embedded system hosting a commercial pico-cell eNodeB (operating in the 2.6 GHz band) connected to a vEPC, necessary for the LTE bearer setup and user credentials. The LTEBOX is connected to the DMM-GW via an Ethernet link. This method requires an LTE USB dongle plugged in the MN to establish the radio link with the eNodeB.

On the MN's side, the Connection Manager groups the three network interfaces (WiFi, 3G and 4G) into a single one. This allows hiding the use of different links to the upper network layers, by showing only one "logical" interface [114]. More, the CM performs the MIHS-related signaling with the network upon handover.

Proof of concept: the CDN+DMM system assessment. In this part, we examine the system behavior when a user watching a video roams within the domain changing access technology at each handover. The object is to carry out a proof of concept of the design described in Section 8.2.1.

In our use case scenario, the user starts accessing the data network using the 3G connectivity. He/she opens the video portal webpage in the web browser, and, then, picks his/her favourite video from the online catologue. In our test, we choose the Big Buck Bunny DASH video[15], segmented into 5 seconds chunks. During the experiments,

[14] `http://www.openairinterface.org/`

[15] `http://www-itec.uni-klu.ac.at/ftp/datasets/mmsys12/BigBuckBunny/`

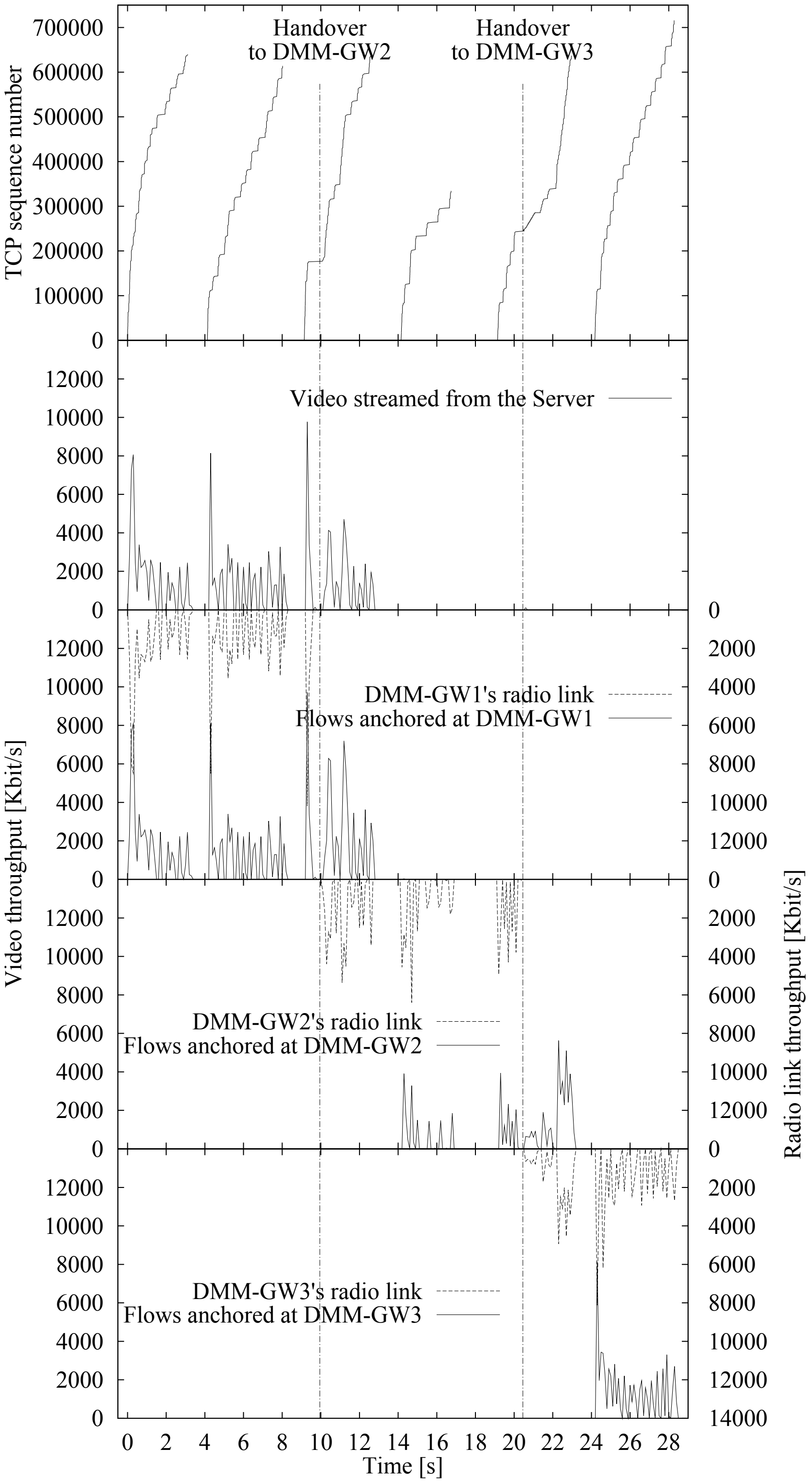


Figure 8.7: Traffic captured in the CDN+DMM system. From top to bottom: in the MN, in the Video Server, in DMM-GW1, in DMM-GW2 and in DMM-GW3

we playback the video only for the first 30 seconds, i.e., for 6 chunks only. Once the video is selected, the streaming starts for the user, being first sent the map of the all the video chunks and their corresponding URLs (i.e., the MPD file). A typical DASH client[16] proceeds then to download and playback the listed chunks one by one, with a 1 second buffer, so that the next chunk download start 1 second before the current chunk playback is over. This mechanism slows down the download rate, preventing to consume all the available bandwidth and also to download unnecessary chunks if the user is not willing to watch the whole video.

The DMM-GW cache are-filled with some video contents, in such a way that the video chunks requested by the user are available in DMM-GW2 an and DMM-GW3, but not in DMM-GW1 The MN is programmed to switch to the WiFi access around the video second 10, and then to move to the LTE link around the 20th second, so that the handovers occur in the middle of the download of chunks #3 and #5. We have illustrated in Fig. 8.7 the throughput captured in the network's nodes during the first 30 seconds of the video playback. This figure presents 5 graphs, one for each network element: from top to bottom, we have the MN, the Video Server, DMM-GW1, DMM-GW2 and DMM-GW3. The top diagram is obtained capturing at the MN's logical interface: we can observe 6 ramp-like plots, each of them is the TCP sequence number evolution for the download of the corresponding video chunk[17]. In the second subplot, we have measured the throughput at the server's outbound interface. In the remaining graphs, the solid line represents the traffic anchored at that DMM-GW, and the dashed line is the throughput measured in the DMM-GW's access link.

As long as the MN is connected to DMM-GW1 through the HSDPA link, the content source is the Video Server, because the local cache at DMM-GW1 cannot fulfil the request, and thus the DM instructs the DMM-GW to retrieve the chunks from the main repository. While being connected to DMM-GW1, the MN requests the first three chunks, therefore we can observe in the second and third graph the traffic at the Video Server and at DMM-GW1 accounting for the packets belonging to such chunks. At second 10, the activity of the DMM-GW1's radio link is interrupted by the first handover. This handoff refers to the MN switching from DMM-GW1's 3G access link to DMM-GW2's WiFi network. Nevertheless, DMM-GW1 still sends traffic for a short interval thereafter due to the mobiltity tunnel. Indeed, DMM-GW1 redirects the remaining packets of chunk #3 to DMM-GW2, that can deliver them to the MN on the wireless link. We observe this phenomenon also in the 4th subplot: DMM-GW2's radio link is active, but no traffic is coming from the direct link with the video server, nor it is generated internally. The packets indeed flow from DMM-GW1 to DMM-GW2 through the IPv6-in-IPv6 tunnel established by the IP mobility modules.

[16] We used the VLC plugin for Mozilla Firefox in Ubuntu, `http://www.videolan.org/`

[17] Note that, even if all chunks are a 5-second portion of the video, they are not equally large in KBytes.

Table 8.2: Experimental measurements

DASH Video Streaming Statistics							
Chunk length [s]		1	2	4	6	10	15
No. of chunks		597	299	150	100	60	40
Video size [B]	Chunk sum	60 254 052	58 987 593	58 344 196	58 138 660	57 509 612	57 373 964
	MPD file	51 958	26 586	13 921	9 671	6 541	4 739
	MP4 file	865	865	865	865	867	867
	Total	60 306 875	59 015 044	58 358 982	58 149 196	57 517 020	57 379 570
Data TX [B]	no tunnel	64 699 689	62 262 514	61 251 667	60 903 590	60 096 154	59 873 730
	per chunk	108 375	208 236	408 344	609 036	1 001 603	1 496 843
	with tunnel	69 464 550	67 967 699	66 837 895	66 487 790	65 679 527	65 454 149
	per chunk	116 356	227 317	445 586	664 888	1 094 659	1 636 356
Overhead [%]	no tunnel	7.28	5.50	4.96	4.74	4.48	4.35
	with tunnel	15.19	15.17	14.53	14.34	14.19	14.07
CDN Node - DM Interface Statistics							
Control packets TX		5 955	2 990	1 500	1000	600	400
per chunk		9.97	10	10	10	10	10
Control data TX [B]		570 465	286 300	143 575	95 680	57 583	38 388
per chunk		955.55	957.53	957.17	956.80	959.72	959.70
Overhead [%]		0.88	0.46	0.23	0.16	0.10	0.06

When the MN starts the new request for chunk #4, it uses the address from the LANP advertised by DMM-GW2, hence in no case the chunk request is forwarded through the tunnel. In this way, the CDN module at DMM-GW2 intercepts the request and, since the chunk is locally available, it immediately transmits the packets to the MN. Therefore, the fourth graph shows activity for the locally anchored traffic (solid line) with respcet to chunk #4, as well as for radio link (dotted link). On the contrary, the second and third graphs, respectively for the Video Server and DMM-GW1, show no traffic for the fourth video segment. Chunk #5 exhibits the same traffic activity on DMM-GW2 as the previous chunk, up to the instant when the second handover takes place. After the handoff, DMM-GW2's radio link activity stops, but still DMM-GW2 sends traffic destined to the MN. Such packets are collected by DMM-GW3 from the tunnel and delivered to destination, replicating the same reaction observed after the first handover. Finally, the sixth chunk is delivered directly by DMM-GW3.

The proposed architecture makes it possible to deploy a CDN close to the access routers of a mobile network using a simple URL-based caching method. It is obvious that the communication latency can be reduced when the content is downloaded from such CDN caches, as well as the traffic in the core links.

Video streaming measurements. In the following experiments, we measure the total amount of traffic generated by streaming a DASH video from a server to a client through the IPv6 infrastructure provided by our CDN+DMM prototype.

The video employed is the full length Big Buck Bunny DASH video (597 seconds), available to our tests in 6 different versions varying the segmentation length, i.e., chunks 1, 2, 4, 6, 10 and 15 seconds long. All versions have an average streaming bitrate around 800 Kbps. This DASH video is an MPEG-4 file, thus the MPD contains the pointers to the initialization file, namely the "MP4 file", and to the locations of the video chunks.

The streaming is measured at a DMM-GW, and we consider two streaming scenarios. In the first one, called *no tunnel*, the video is received and forwarded by the DMM-GW working as a plain IPv6 router, i.e., without encapsulating the packets. In the second scenario, called *with tunnel*, the packets are forwarded through a tunnel link to another DMM-GW before being delivered to the MN, thus, we reproduce the encapsulation[18] that would occur in case of mobility.

We compute the streaming overhead as:

$$Overhead = \frac{S_{measured} - S_{video}}{S_{video}} \tag{8.1}$$

where S_{video} is the compound video size including all files and $S_{measured}$ is the total traffic measured. The values in Table 8.2 summarize the results obtained from the experimental measurements.

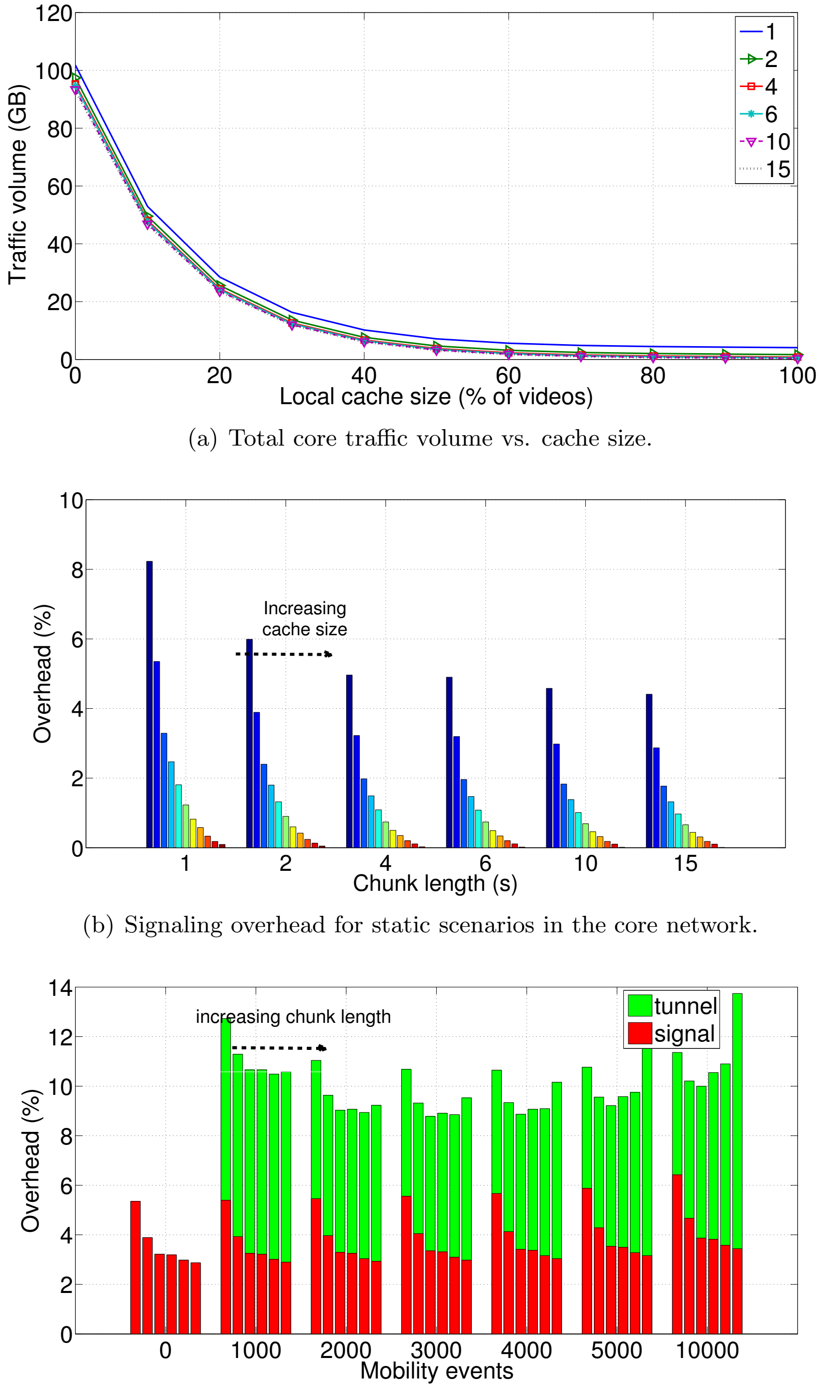
As expected, the shorter the chunks, the higher the overhead, because the video requires more HTTP and TCP sessions to be established. The streaming requires a 7.28% overhead for a video chunk length of 1 s, and 4.38% for a chunk length of 15 s. This behavior is exacerbated by the presence of a mobility tunnel, as can be seen from Table 8.2, Still, also in the *tunnel* scenario, the shorter the chunk size, the higher the overhead penalty, even if the difference is not as relevant as in the former scenario. We recall that, in current mobility protocols, like Proxy Mobile IPv6, packet encapsulation is mandatory for all IP flows, that is, for all chunks, even if the MN is static. Conversely, with our CDN+DMM solution, the packets are tunneled only after a handover, and only for the duration of the remaining portion of the chunk download. Hence, at most one chunk is tunneled per handover, and the longest chunks are penalized. For instance, with 1 second chunks, a motionless user is served by a PMIPv6 network wasting double the overhead that is required with the CDN+DMM system. With 15 seconds chunks, the overhead penalty is three times larger in PMIPv6. The bottom part of Table 8.2 describes the amount of control data transferred in the interaction between the CDN node and the DM. Such interaction serves to provide the best chunk source location, and costs less than 1 KB for each chunk.

Next, we study the traffic load in the core and access networks for a set of network settings of interest for practical implementations. We emphasize the potential reduction of network traffic among the benefits provided by the proposed framework. We evaluate the

[18] The IPv6-in-IPv6 tunnel requires 40 bytes for the extra IPv6 header.

system by considering $10\,K$ requests per video session during a period of time of 1 hour within the area of responsibility of one DMM-GW, assuming that major European operators run tens of DMM-GWs each. The average video duration is set to 258 seconds [109] with an average bitrate of 800 Kbps (as for the video tested in the experimental part). We assume that videos are played following a Poisson process such that the distribution of active video sessions is heterogeneous during the considered time interval. We further assume that videos are not played to the end (258 s), but the average video session duration is only 100 seconds. Thus, with $10\,K$ requests, a total of 1 million seconds of video is streamed in the considered network area. The length of the video chunks is selected from the following set: 1, 2, 4, 6, 10, and 15 seconds. As a result, the requests are for 1 million, $500\,K$, $250\,K$, $167\,K$, $100\,K$, and $67\,K$ chunks, respectively. This range allows the evaluation of the impact of the chunk size on the overall system performance. To study the impact of user mobility, we introduce a set of mobility levels ranging from 0 (no mobility) to $10\,K$ mobility events, i.e., on average one mobility event per requested video. We remark that the definition of mobility (and consequently of handover) refers to an MN moving to another DMM-GW, that is, changing the attachment point at the IP level. For ease of computation, we assume that the handover is on average happening in the middle of a chunk downloading process. Regarding the content distribution scheme adopted in our system, we assume that each local cache provides the most popular chunks in its own region. MPD and MP4 metadata files are always retrieved from the main server to ensure that the latest version is received. The local popularity of a video chunk is modeled following the Zipf-Mandelbrot's law [115]. This links the cache size to the number of requests than can be served by a network node. If the size of the cache is such that it can store the 10% most popular chunks, it can serve directly 65% of the users' requests. When the size of the cache is such that it can store 20% of the chunks, it can serve directly 80% of the requests. We show later that the optimal size of a cache is a trade-off between storage costs and video traffic.

We further focus our analysis on the traffic volume and signaling overhead while taking into account the mobility of users. Therefore, we count the number of messages that are required to retrieve a video either from the origin server or from the local cache. We assign to each packet type the values measured from the experiments discussed above. In our analysis we take into account the requests for MPD and MP4 files, HTTP get and response messages (the actual video data), and the signaling messages of the CDN necessary to identify the closest cache from where to retrieve the content. We do not consider the exchange of database update messages between the CDN nodes and the DM, as their size and update frequency are very dependent on the actual implementation. We split the total traffic volume in video traffic, i.e., the pure video data, and signaling overhead. We monitor the data traffic in the whole network, in particular we measure the in- and out-bound traffic volume at the joint CDN+DMM-GW node.



(a) Total core traffic volume vs. cache size.

(b) Signaling overhead for static scenarios in the core network.

(c) Signaling and tunneling overhead in the core network when the cache can store 10% of the most popular videos.

Figure 8.8: Traffic volume and overhead generated for static and mobile scenarios.

Fig. 8.8(a) shows the in- and out-bound traffic volume at the joint CDN+DMM-GW node towards the core network as a function of the percentage of chunks stored in the local cache. The general shape of the curve is a direct consequence of the video popularity distribution introduced earlier. As expected, the more the content cached in the CDN node, the less the traffic transported through the core network, and the more the traffic directly delivered from the CDN node to the user. If the cache in the CDN node is empty (left side of Fig. 8.8(a)), all data needs to be retrieved from the main video server. Providing just a few of the most popular chunks in the CDN node can already significantly reduce the traffic load in the core network. Providing further content with low popularity in the CDN node will still reduce the traffic volume in the core network, but at a lower rate. In the extreme case where all video chunks are cached locally (right side), all requests can be served by the CDN node and the traffic is only between the CDN node and the MN, i.e., on the access network side. Note that the CDN node cannot reduce the traffic volume on the access side. However, the higher the storage capacity of the CDN, the more the associated costs. With the increase of the cache size, storage costs approximately increase on a linear scale whereas the core traffic volume decreases exponentially. The network architects' goal is to find the optimal tradeoff between the two. We argue that it is worth to locally cache video chunks even if they are requested only a couple of times a day, as the storage costs trend is decreasing in future years [109].

We can further observe that the shorter the chunk length, the larger the generated overhead traffic. The overhead counts

(A) the packet overhead,
(B) the HTTP session setup for each chunk,
(C) the messages querying the DM for the optimal location of the requested chunk.

Overhead ($A+B$) is smaller for larger chunks as fewer packets are necessary to handle the network protocol operations at all layers of the IP stack. Overhead (C) obviously increases with shorter (i.e., more) chunks and smaller cache sizes, i.e., less hits for cached content. Chunks of 1 s will result in 15 times the number of `get_optimal_location` requests to the DM than chunks containing 15 s of video data. As a consequence, we argue that longer chunks reduce the traffic volume in the network. This fact is in line with the observations driven by the values of Table 8.2 and is also shown in Fig. 8.8(b), which shows the share of measured overhead in the core network for different chunk lengths when varying the amount of popular videos stored in the caches from 0 to 100% in steps of 10%. In addition, if all chunks are cached in the local CDN nodes (right-most bar of each group), nearly no overhead is measured in the core network, as only requests for the MPD and MP4 files travel through the CDN nodes.

In the following study, we add user mobility to the system. We consider the range from 0 to 10 K mobility events during the considered time period, i.e., on average one mobility event per played video. Note that a mobility event, according to our definition, is a change in the IP point of attachment, caused by handovers resulting in a change of the serving DMM-GW. If a mobility event is triggered, the DMM will forward the currently streamed chunk to the new DMM-GW to avoid the break of the ongoing HTTP session. This causes additional overhead term

(D) tunneling overhead,

as the packets of the ongoing chunk will be transported along additional network paths to the new DMM-GW, and thus they are counted as traffic at both DMM-GWs. The following chunk is then directly requested from the new DMM-GW and, thus, will not be transferred through the DMM tunnels. Fig. 8.8(c) shows the percentage of overhead for different numbers of mobility events and different chunk lengths. The group of bars at the left side (no mobility) of Fig. 8.8(c) depicts the same scenario and values as in Fig. 8.8(b) in the second bar of each group, i.e., for a cache size of 10%. We distinguish between signaling overhead ($A + B + C$) and tunneling overhead (D), as they show an opposite behavior. The signaling overhead gets smaller for larger chunk sizes as fewer chunks are requested per video, whereas the tunneling overhead increases with larger chunks due to a larger chunk being forwarded. In the static scenario (no mobility), there is no tunneling overhead. Then, for increasing user mobility, the additional overhead due to the tunneling of the ongoing chunk to the new DMM-GW outweighs the reduction of signaling overhead when large chunks are used. In addition, for short chunks, the lookup tables in the CDN nodes must hold more entries and more data is exchanged between CDN nodes and the DM to synchronize the popularity databases. Moreover, more lookup requests per time must be processed, requiring more processing power and expensive hardware. A chunk length of 4 s (Fig. 8.8(c)) represents a good trade-off between signaling and tunneling overhead, outperforming the other chunk sizes in most scenarios. Moreover, chunks of 4 s offer a good granularity for the video player to promptly adjust the video quality to changing conditions of the wireless medium.

Inter-technology handover evaluation. We next evaluate the system performance by measuring the latency experienced during a handover. In these experiments, the terminal performs a series of heterogeneous technology handovers (vertical HO). When doing so, the terminal does not release the old radio link before establishing the new one, so there is always an active radio link between the MN and the network (commonly referred to as soft-handover). This operation is realized through the IEEE 802.21 protocol. Fig. 8.9 details the handover phases and the signaling among the entities. The diagram reflects a simplified version the message sequence chart depicted in Fig. 5.9. The first step is the indication from the MN's CM to the FM residing in the current DMM-GW that the handover is imminent (message ① in Fig. 8.9). Then, the FM negotiates the radio resources with its peers in the candidates DMM-GW selected as possible targets (message ②). With this negotiation, the current DMM-GW communicates to the candidate DMM-GW(s) the addresses of the DMM-GWs that are currently anchoring the MN's flows. The

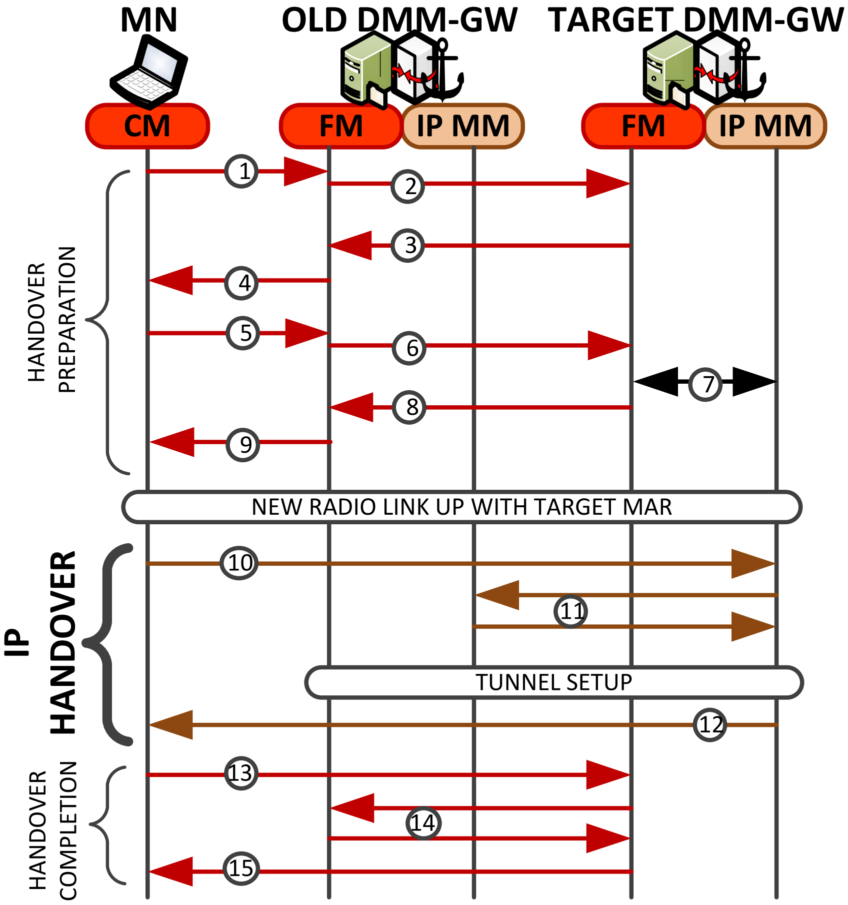


Figure 8.9: Handover phases and signaling.

candidates reply back to the FM that originated the session announcing their availability (message ③), and this FM in turn informs the CM (message ④). The CM, after checking that the target DMM-GW is available, triggers the resource preparation in the traget DMM-GW through the local FM (messages ⑤ and ⑥). The FM triggers the IP mobility module activation, interaction ⑦, providing the list of IPv6 addresses for the anchor DMM-GWs. This phase, called “Handover preparation”, ends with the acknowledgement message from the target FM (⑧) and then to the MN (message ⑨).

Once the second radio link is up, the CM switches off the old one and the MN performs the usual IPv6 Neighbor Discovery procedures in order to obtain IPv6 connectivity on the new link (message ⑩). The new DMM-GW invokes the IP MM functions to perform the IP mobility operations with the old DMM-GWs (interaction ⑪). This stage is necessary to import from the previous DMM-GWs the IP settings used for the MN in their access links and to build the IP tunnels for the redirection of old IP flows to the new location. This phase, called “IP Handover”, is accomplished when the MN is advertised the new prefix to start new sessions, along with the previous prefixes to confirm their usability for the ongoing flows (message ⑫). Next, once the operations to restore the IP

Table 8.3: Handover latency experimental results

IP Handover to	Mean [s]	Std. Dev. [s]
3G (operator 1)	0.085	0.031
3G (operator 2)	0.163	0.033
WiFi	0.003	0.001
4G (OAI)	0.010	0.003
4G (LTEBOX)	0.110	0.023

connectivity are over, there is an additional interaction based on IEEE 802.21 between the CM and the FM in the new DMM-GW to finish the transaction (message ⑬) and to release the resources on the old DMM-GW (interaction ⑭). This phase, named "Handover Completion" is concluded with the acknowledgement from the new FM to the MN (message ⑮).

During the "IP handover" phase, the terminal has already established the new radio link, but the new IP address is not configured yet, and the IP setup for old flows still points to the previous radio link (recall that the tunnel setup between the DMM-GWs and the routing update is still taking place). Thus the terminal is disconnected from the IP network during this phase. This interval is the most critical for the communications, because the MN is not able neither to send nor receive IP packets. Therefore we consider this interval as the *handover latency*, and it is the object of the measurements described below.

Table 8.3 summarizes the mean and standard deviation values obtained for the IP handover from more than 800 handovers. It should be noted that such interval is lower bounded by the Round Trip Time (RTT) between the MN and the DMM-GW in the target technology, where the router discovery procedure takes place (messages ⑩ and ⑫ in the picture). For instance, when the MN hands off to the HSDPA technology, such signaling has to traverse the whole operator's UMTS infrastructure, with a high global delay. In our experiments the two operators performed very differently, as one introduces twice the delay than the other. We observe a significant difference also when comparing the LTEBOX and the OAI platform: the former introduces a larger RTT because of the virtual EPC embedded in the system, while the latter was tuned to offer the lowest eNodeB–UE latency. WiFi is the technology that offers the best performance (the typical RTT between a WiFi AP and a station is below 5 ms).

In a real DMM deployment, the distance between the DMM-GWs involved in the handover would play a large delay contribution to the compound handover latency: the IP Handover time would be affected by the interaction ⑪, the Handover Preparation time by the exchange of messages ②③ and ⑥⑧, while the Handover completion by the interaction ⑭. Still, the most crucial remains ⑪, because the handover preparation can be timely anticipated to take into account ②③ and ⑥⑧ without having a relevant impact on the user experience, and ⑭ takes place after the flows recovery.

The MN's IP interface exposed towards the network and towards the MN's upper layers is a logical interface implemented by the logic in the CM, so the operations performed by the physical interfaces are hidden. Therefore, the IP session continuity is provided without any intervention by the MN beyond the Neighbor and Router Discovery standard operations.

8.3 Final remarks

The scope of this chapter was to provide a proof of concept of our DMM solutions based on real implementations. To the authors' knowledge, this is the only assessment available from experiments on real prototypes, as compared to the rest of research activity related to DMM that uses analytical models and simulations.

We argue that this is a relevant contribution because it permits to validate the behavior of the terminals moving within a DMM domain. Indeed, the key aspect of the DMM solutions presented in this thesis (and in other proposals) is the allocation of multiple IPv6 address to the MN at the same time, and the wise exploitation of each address to achieve and optimal data path for the applications. We have demonstrated that a standard GNU/Linux IPv6 stack implementation is able to handle multiple addresses at a time and to expose to the upper layers the most suitable one, if correctly instructed by the network.

Also, the implementation demonstrates the consistency of the signaling devised to operate in the control plane. This is once again a valuable tool to determine the completeness of the design, and it technical solidity.

Currently the ODMM project hosts an initial stage for the implementation of the C-DMM protocol, and more stable and comprehensive implementation of the "proxy" DMM solutin (called Mobility Anchor Distribution for PMIPv6 – MAD-PMIPv6).

We are also working together with other colleagues on an enhanced design and implementation of the SDN solution, aiming at making in it available in the ODMM project as well. Unfortunately, the fully distributed solution cannot be part of the ODMM project due to the proprietary implementation of the Flow Manager and the Connection Manager, which were made available by the developers only within the scope of the EU project MEDIEVAL.

Part III

Conclusions and future work

Chapter 9

Conclusions

In the last years, the commercial success of high-end portable devices like smartphones and tablets, and the proliferation of applications that make use of mobile data services have pushed operators to expand and upgrade their infrastructure, deploying state of art technologies, like the modern 4G system. This trend is foreseen to keep growing in future years, because the population of mobile broadband users is in increasing, and operators are willing to attract more subscribers, offering the best service available in the market. As a result, operators are already planning their strategies to cope with the tremendous future traffic growth, and in parallel, vendors and the research community are investigating possible research guidelines to carry out effective solutions.

One of the current research areas focuses on the the network architecture, aiming at the design of a flat system. In the past, we have observed many network architecture evolutions, aiming at integrating existing services into a common platform. For instance, the IP Multimedia Subsystem (IMS), was introduced in order to achieve a convergence of voice and multimedia services for fixed and mobile nodes. The Evolved Packet Core specifies a common packet-switched core for heterogeneous access domains, including, again, fixed and mobile nodes. Such evolution is not willing to stop, as many other pieces are going to be added to the system, and there is still room to apply optimizations in order to achieve an efficient platform. A flat mobile architecture is regarded as a target that enables an enhanced fixed-mobile convergence, able to efficiently accommodate current services and future demands.

This thesis covered the Distributed Mobility Management topic, which is a key problem when dealing with the transition from a centralized and hierarchical system to a distributed and flat one. In fact, the current mobile architecture is heavily centralized, as the mobility management is simpler in a centralized scheme: a core entity, the anchor, aggregates the traffic flows to and from several access networks, and it redirects these flows when users move from one access network to another.

The solutions proposed in this work aim at the extending two of the current centralized mobility management protocols based on IP, namely Mobile IPv6 and Proxy Mobile IPv6, according to the DMM philosophy. We have chosen to re-use existing protocols because, in terms of feasibility, it is usually more convenient to look for optimizations and upgrades rather than deploying a clean slate systems. We have proposed the following original designs:

1. One mobility solution based on a client running on the terminal;

2. Four mobility protocols that do not affect the moving host's IP stack. Three of them belong to the *partially distributed* approach, where the data plane only is distributed and the control plane is kept centralized. The remaining solution is *fully distributed*, that is, both data and control planes are untied from central entities.

3. One hybrid scheme that couples a client based solution and a network-based one.

We have analyzed the performance and scalability of all our proposals, taking into a account the signaling overhead, the packet delivery cost and the handover latency. We have compared out results with those from the centralized protocols, and we have concluded observing which scenarios take more benefits from a distributed mobility management, and which from a centralized one. In particular, we have characterized what parameters play a dominant role to determine which protocol fit bests. Mainly, the network topology and its dimensioning has a big impact on the performance: in centralized scheme both the control messages and data packets must travel through the network's access and core parts, whereas DMM enables a more flexible scenario. Then, we have stressed how traffic and mobility patterns affect the protocols' costs. DMM allows for lighter signalling when the duration of the IP sessions are short, even with high mobility, but it produces a large overhead if long flows must be maintained across many visited networks. In this latter case a classic centralized approach is preferable. Finally, the handover latency depends on the distance of the network entities involved in the signaling. DMM enables the deployment of shorter communications links, especially in the fully distributed case. In general, we have observed that in small network domains, there is no big difference between a DMM solution and its centralized counterpart, but in large deployments DMM outperforms a centralized scheme, since it allows to divide the infrastructure in smaller subsets operated by a homogeneous protocol based on IP.

Some of the network-based DMM designs proposed in this thesis have been implemented in order to prove their feasibility with real equipment. We have carried out two running prototypes operated respectively by the "proxy" partially distributed network based solution, and by the fully distributed solution combined with the IEEE 802.21 handover management protocol. Both prototypes served for public demonstrations at international conferences and exhibitions, representing the first DMM implementation effort introduced to the scientific community worldwide. The implementation codes have been made publicly available in the ODMM project, Open Platform for Distributed Mobility Management solutions. Beyond hosting the open source implementations, the ODMM website collects all the material that we have produced on the DMM topic, including multimedia archives recorded at public exhibitions.

Chapter 10

Future work

The Distributed Mobility Management working group at the IETF is currently starting to work on solutions' specifications to include in working group's drafts. Presently, only individual submissions have been discussed, and not all of them describe the protocol's details as our drafts do. However, there are still some areas that could be extended before coming at a full solution. There are some open points that are currently under discussion.

- *Application-oriented mobility management.* From the comparison of centralized and distributed mobility management protocols, we have observed that there is no a clear "winner", but rather we can identify scenarios where each candidate fits best. This fact lets imagine a hybrid deployment where DMM and CMM co-exist, switching between the two according to the specific needs. Specifically, these needs are dictated by the user's applications: those applications that can survive after an IP address change do not need mobility support at all, so a DMM approach is preferable, whereas those applications that require mobility support can be further classified in those that typically generate long flows, for which a CMM solution is preferable, and those that generates short flows, for which a DMM approach is better suited. The IETF Draft [116] illustrates some use case scenarios in this field. The research challenge in this context is twofold: on the one hand, it is necessary to identify what is the applications behavior, constrained to the typical users traffic and mobility patterns. On the other hand, once such information is available, a mechanism is needed to handle the assignment to one mobility management protocol, or how to switch between the two schemes [117].
- *Prefix coloring.* This aspect is related to the previous in the sense that an MN can be assigned multiple IPv6 prefixes, each with specific attributes, and the usage of a prefix can drive the behavior of the network, e.g., the mobility management type. For instance, we have described in Section 5.2, that after a mobility event, an MN is advertised a prefix with the preferred lifetime attribute set to a non-zero value, and one or more prefixes with a zero valued attribute. This is a simple prefix coloring method employing a binary scheme, but more sophisticated techniques can be designed. The challenge

in this context resides in how applications pick the prefix for their communications, problem related to the source address selection algorithm running in a host network stack. More on this can be found in the IETF Drafts [118, 119]

- *Multicast traffic.* The DMM requirements document states that "DMM SHOULD enable multicast solutions to be developed to avoid network inefficiency in multicast traffic delivery" [41]. The aspects to be investigated are related to mobility support for a multicast listener as well as for a multicast source, [120]. In both cases, the designer has to consider how the multicast tree should be built over a mobile infrastructure, and how to handle the multicast subscriptions when users change their attachment point to the network. The IETF Draft [121] proposes a mechanism to transfer the multicast listener context after a handover in a DMM network.
- *Flow mobility.* This problem regards "multihomed" mobile hosts, that is, those that possess more than one simultaneous active link to the network. This is the case of multi mode terminals that are able to use multiple network interfaces at the same time. For instance modern smartphones are typically equipped with a 3GPP interface (3G/4G) and a non-3GPP one (WiFi). Both the users and the network operators are interested in mechanisms that allow to transparent move the IP flows from one interface to the other without suffering service disruptions. Users are usually concerned about the costs for their data plan, so they wish to move their sessions to the WiFi network whenever it is possible, and, similarly, operators aims at strategies to perform traffic offload to reduce the potential congestion in their equipment [122,123]. IP flow mobility is already addressed for MIPv6 [124] and a standardization process is in progress for PMIPv6 [125], but yet no consistent actions have been taken within the DMM IETF working group beyond the use case analysis submitted as individual draft [126].
- *Multiple CMDs deployment.* The partially distributed approach for DMM decouples the control plane from the data plane, being the former centralized and the latter distributed. In this thesis we have designed some solutions in this category, all of them employing the Central Mobility Database (CMD), but others are available in the scientific literature, sharing the functionalities of a central node acting as database for the users' mobility sessions. Nevertheless, there are still no proposals that envision the deployment of several databases, each of them managing a smaller portion of the network, so that the DMM domain can be split in order to reduce the communication delay between such database and the mobility anchors. The research challenge is to design a signaling scheme that enables the user mobility between the DMM domains handled by two different databases, and how to perform the subsequent mobility context/profile migration.

The research items listed above reflect what are currently the main discussion topics within the DMM working group at the IETF. However, there are other DMM areas worth of attention for future research. We highlight for instance the big momentum that

Software Defined Networking (SDN) and Network Functions Virtualization (NFV) are currently gaining, and how the DMM paradigm interplays with them.

In Section 8.1.1 we already gave a brief overview of SDN, and we presented a simple DMM protocol based on SDN for the sake of the comparison with other DMM solutions. There are however other efforts in the SDN area pointing to the DMM problem space, as SDN potentially enables efficient mobility management without the intervention of centralized mobility anchors. Indeed, an SDN controller is theoretically able to handle individual IP flows and to easily redirect them to the user's location, following his/her movements, without being traversed by the flows. Unfortunately such a technique is not scalable if a controller manages all the IP flows for millions of users one by one, therefore the challenge here is related to the design of a light and scalable protocol.

NFV enables the deployment of virtual instances of network functions, so that network entities can be easily co-located or replicated independently from the underneath hardware deployment. In the test-bed platform described in Section 8.2, we used the LTEBOX by Alcatel-Lucent, an embedded system consisting in a real pico cell connected to a virtualized EPC. This is an example of the powerful and flexible tools that the NFV framework can provide. The combined application of the SDN, NFV and DMM paradigms is certainly a valuable resource to be investigated in the near future.

Appendix A

Message formats

A.1 MIPv6

The location update procedure specified in MIPv6 is called *registration* or *binding*. Two messages are defined for this procedure: *Binding Update (BU)* and *Binding Acknowledgement (BA)*.

A.1.1 Binding

A Binding Update message is generated in a modular way, appending a Mobility Header to the IP header[1].

The mobility header's presence is notified by setting the IPv6 header's "Next Header" field to the value 135. The mobility header format is defined in MIPv6 and shown in Fig. A.1: the MH field indicates the nature of the message carried in the Message Data field, and, in case of a BU message, such value is 5. Nevertheless, the mobility header is used to bear also the BA message (with MH value 6), sent by the HA back to the MN, and some other control messages useful in the registration phase and for the Route Optimization (RO) procedure. The Binding Update and Acknowledgment message format

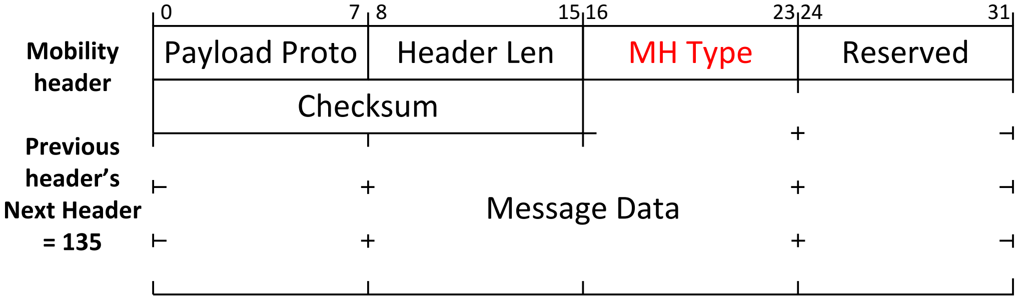
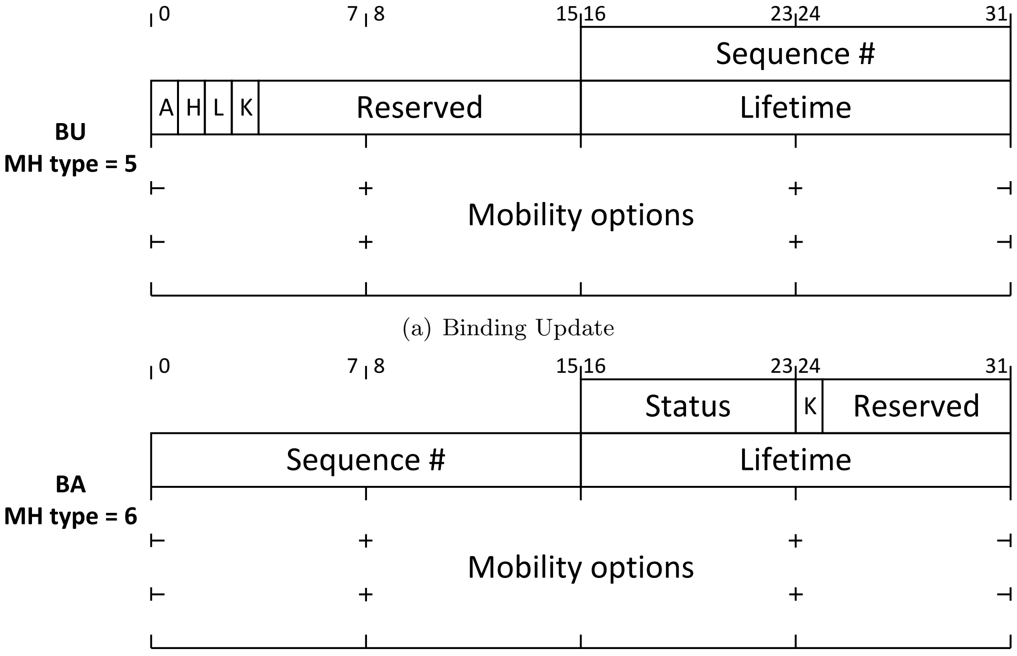


Figure A.1: Mobility Header message format.

[1] This is an improvement from MIPv4, wherein Registration Request messages are encapsulated in an UDP datagram with destination port 434.



(a) Binding Update

(b) Binding Acknowledgement

Figure A.2: MIPv6 signaling message format.

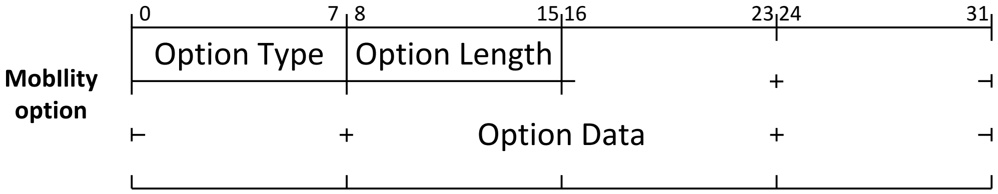
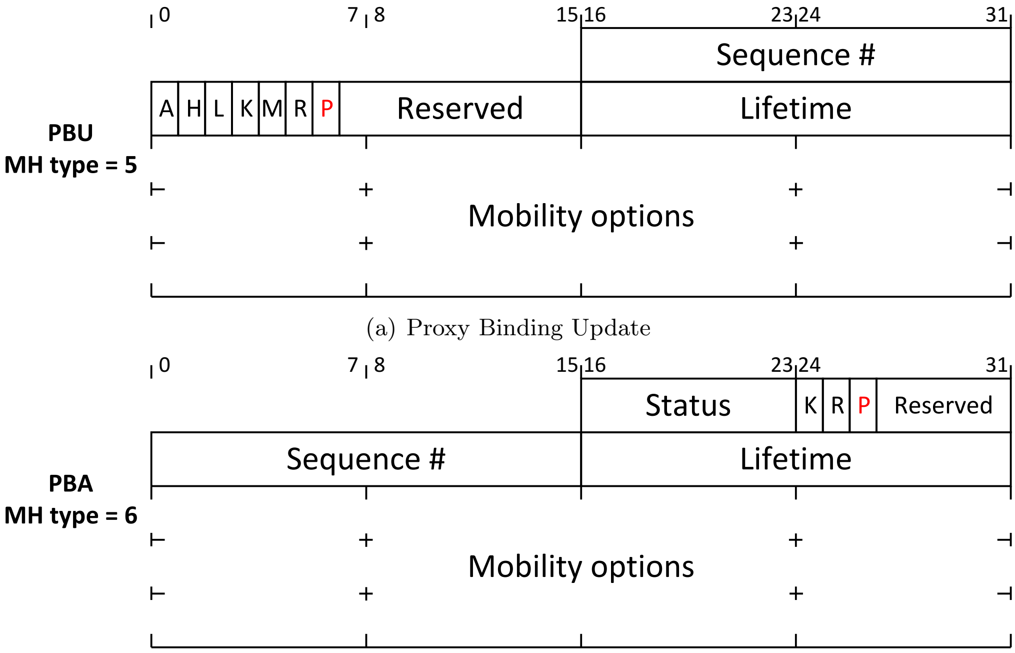


Figure A.3: Mobility Options message format.

is illustrated in Fig. A.2, 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 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 Fig. A.3). 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. Also, this modularity allows the re-usage of the mobility header and options by other IPv6 mobility protocols, without defining new messages. 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.



(a) Proxy Binding Update

(b) Proxy Binding Acknowledgement

Figure A.4: PMIPv6 signaling message format.

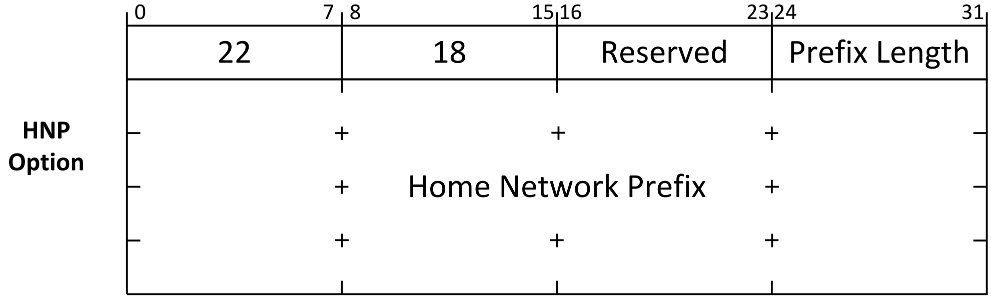


Figure A.5: Home Network Prefix mobility option format.

A.2 PMIPv6

Similarly to the procedure described for MIPv6, the registration of an MN at the LMA is called *proxy registration*. Two messages are defined for this procedure: *Proxy Binding Update (PBU)* and *Proxy Binding Acknowledgement (BA)*.

A.2.1 Proxy registration

The PBU and PBA messages are generated in a similar way as the BU and BA messages described before, i.e., a mobility header of the format illustrated in Fig. A.1 conveys the byte sequence as fomratted for the BU and BA messages, except for an additional P (proxy) flag set to 1 (see Fig. A.4). Moreover, extra information are included in the form of mobility options: Fig. A.5 depicts the format of the Home Network Prefix

(HNP) mobility option.

A.3 N-DMM

The Network-based DMM protocols propose modifications and extensions to the PMIPv6 Binding Cache Entry of an MN to store extra parameters. In addition, the partially distributed N-DMM solutions require the specification of two additional mobility options to be appended to Proxy Binding Update and Proxy Binding Acknowledgement messages. These options are the *Serving DMM-GW mobility option* and the *Previous DMM-GW mobility option*, which message format is described below along with the new BCE structure.

A.3.1 Binding Cache Entry

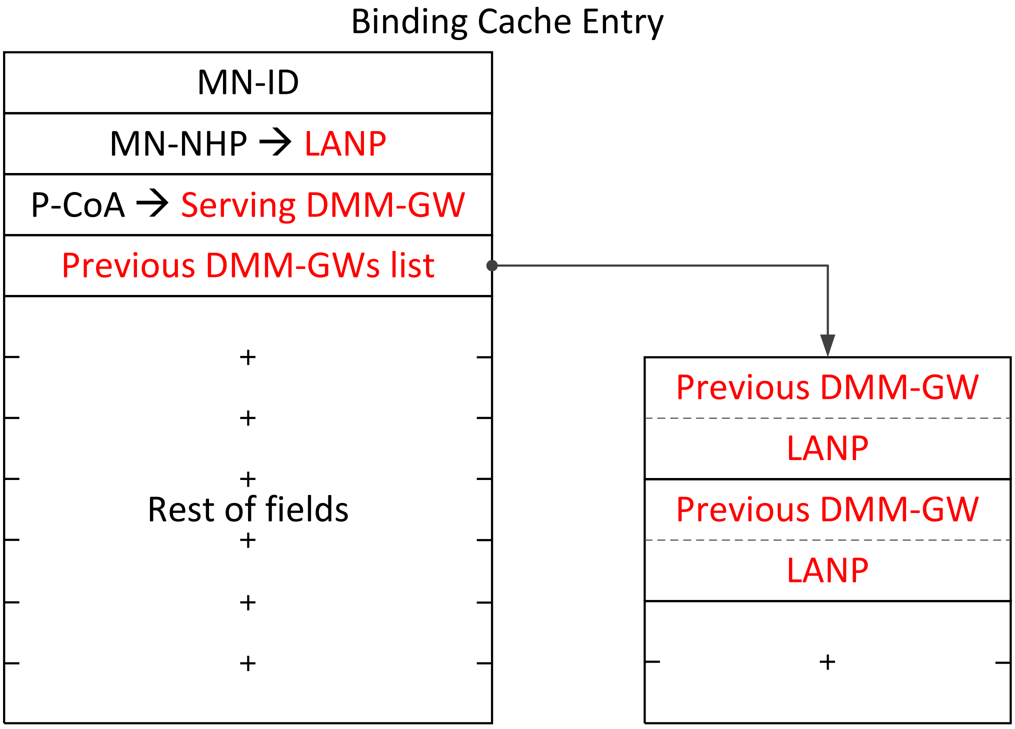


Figure A.6: Binding Cache Entry structure.

The N-DMM BCE of an MN renames the MN-HNP and P-CoA fields respectively into *LANP* and *Serving DMM-GW*. In addition, a new field is added, called *Previous DMM-GWs list*, containing a pointer to a list of previous anchors of the MN. Each element of the list contains the IPv6 address of a previous DMM-GW and the LANP assigned by that DMM-GW to the MN. The BCE structure is illustrated in Fig. A.6.

A.3.2 Serving DMM-GW mobility option

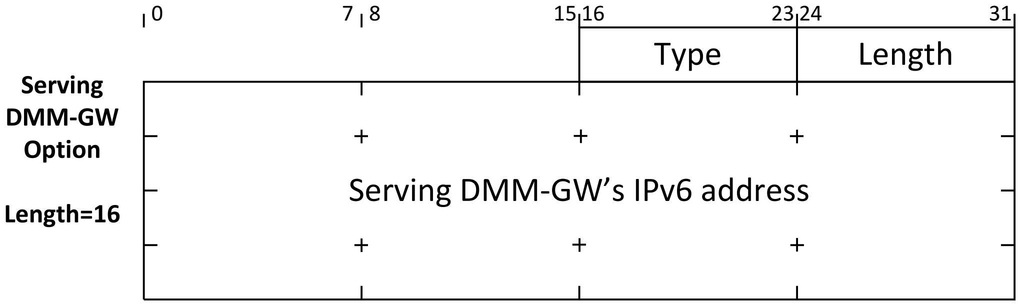


Figure A.7: Serving DMM-GW mobility option format.

The *Serving DMM-GW mobility option* is appended to the Proxy Binding Update message, sent by the CMD to a previous DMM-GW to indicate the new MN point of attachment. The length field must contain the value 16. The option format is illustrated in Fig. A.7.

A.3.3 Previous DMM-GW mobility option

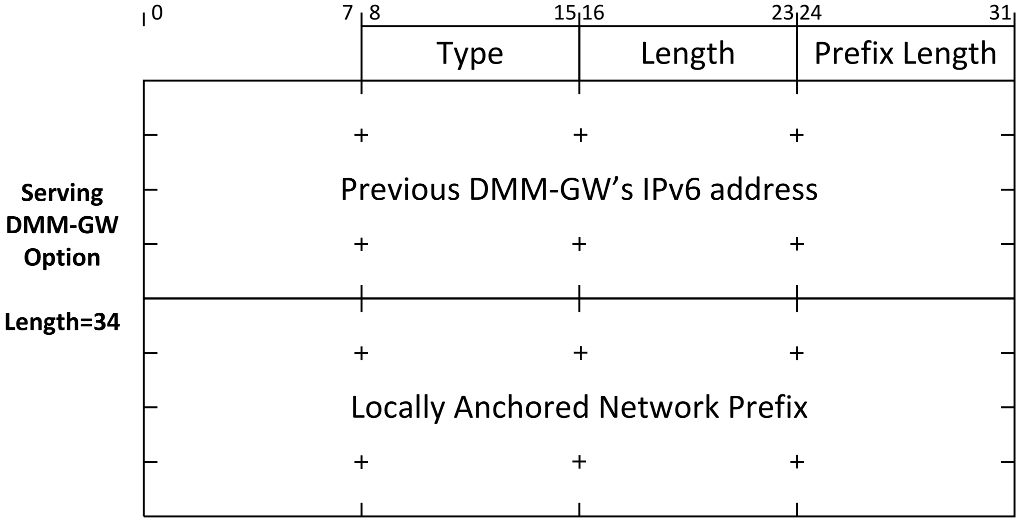


Figure A.8: Previous DMM-GW mobility option format.

The *Previous DMM-GW mobility option* is appended to the Proxy Binding Acknowledgement message, sent by the CMD to the new serving DMM-GW to indicate the address of an old DMM-GW for the MN and the prefix it assigned. There may be more instances of the Previous DMM-GW mobility option in a PBU message. The length field must contain the value 34. The option format is illustrated in Fig. A.8.

Appendix B

IEEE 802.21 and PMIPv6: fully network-controlled mobility

As already mentioned in Section 2.3, MIH Services do not provide mobility management at Layer-3 or upper levels, therefore such a protocol has to be integrated with the IEEE 802.21 suite to guarantee service continuity at the application level. Since most of today's network applications run over an IP transport network, a natural choice can be an IP mobility protocol, e.g., adopting MIPv6 or PMIPv6. This latter is indeed the protocol chosen for the use case scenario describe below. Note that what follows is one of the possible deployment choices that can be selected among those suggested by the IEEE 802.21 standard.

B.1 System design

The architecture model is drawn in Fig. B.1. Besides the PMIPv6 entities (LMA and MAG), we emphasize there the Handover Controller (HOC) network entity. It is an omniscient node that retains the complete vision of the access network, i.e., it is aware of all the connected MNs, all the PoAs and all the MAGs deployed in a certain area, together with the binding relationships existing between such elements. Thus, the HOC knows to which access networks PoAs and MAGs belong to: such database is statically configured in the HOC.

Almost all elements in the network, MAGs, MNs and HOC, contain an MIHF. Each of the network entities employs the MIHF functionality in a different way by providing an MIH User as a control module interacting with the MIHF. The MIHF and the MIH User within the HOC provide the intelligence through which the HOC manages and coordinates the handover procedure for the MNs. It interacts with the rest of the MIH enabled entities in the network by receiving MIH Events and sending MIH Commands. The HOC is the MN's PoS, and it communicates to the MIHF in the terminal through

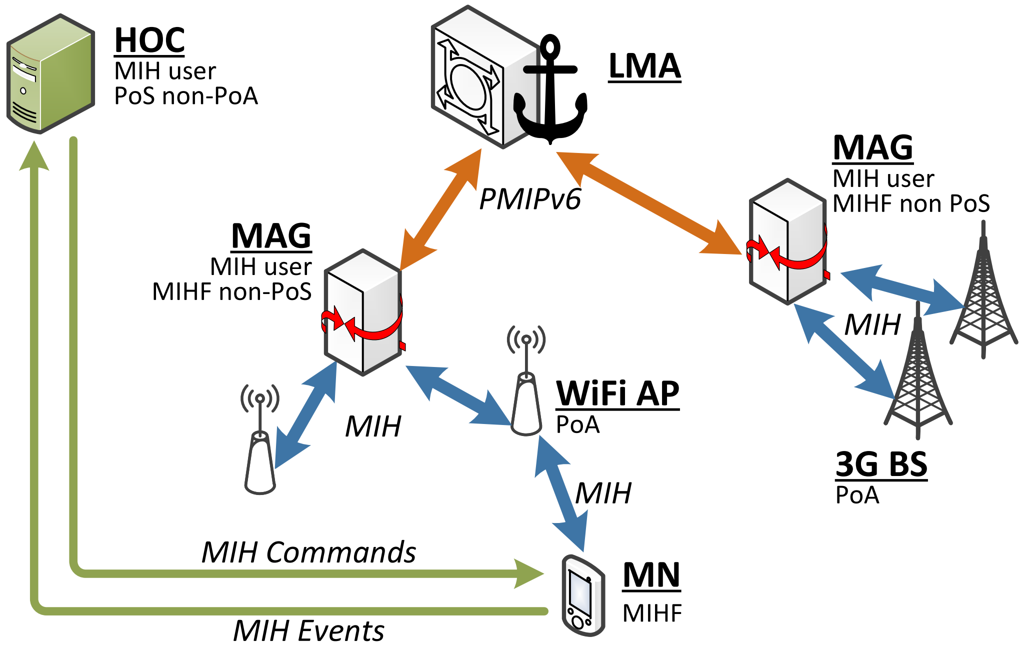


Figure B.1: IEEE 802.21 and PMIPv6 architecture model.

the terminal's active interface, The MIH User located in the terminal is in charge of executing the commands received from the HOC. Although being a PoS, the HOC does not include any PoA for the MN (PoS non-PoA), which conversely are connected to the MAGs. MAGs run a MIHF as well, but such MIHF does not exchange MIH message with the MN (MIHF non-PoS). However, it is responsible to trigger the PMIPv6 procedure when it receives the corresponding link layer event (MIH_Link_Up) from one of its PoAs exposed to the MNs.

As IEEE 802.21 provides a very flexible set of services, the reporting of the status of radio conditions and other operations are performed in an asynchronous manner based on exchanging MIH messages between the MIH entities across the domain. Next we detail the protocol behavior within the PMIPv6 scenario.

B.1.1 Discovery and Bootstrapping

Bootstrapping in this context is regarded as the *power up* procedure and consists of *i)* powering on a network interface, *ii)* selection of the best PoA based on the signal strength, *iii)* network attachment and IP address configuration, and *iv)* MIHF registration. Regarding *iv)*, before a control communication channel is setup between two MIH peers (typically MN and PoS), the MIH user in the MN should start a discovery process, that can be processed either at Layer-2 or at Layer-3. IEEE 802.21 standard describes procedures for Layer-2 based discovery while here the approach suggested in [127] is followed, hence executing it either via DNS [128], or via DHCP [129]. No assumptions are made

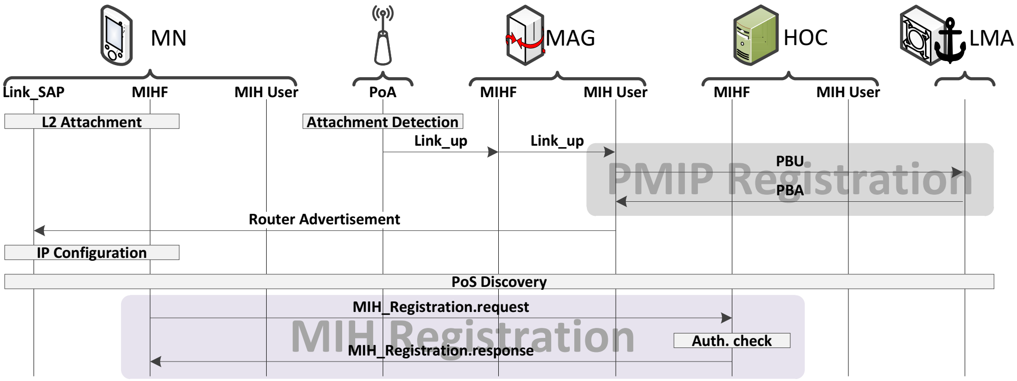


Figure B.2: MIH bootstrap signaling

on the timing of such discovery procedure (it could happen while access authentication is performed). The address of the PoS is therefore used during the bootstrapping phase for MIHF registration between the HOC and the MN.

Figure B.2 describes the whole bootstrap signaling flow operation. When the node powers up its interface, the terminal starts the Layer-2 attachment procedure, which includes also the authentication of the MN in the access network and can be processed through Layer-2 credentials provided by the MN to the PoA, that are used to query the AAA infrastructure. Upon completion of Layer-2 attachment procedure, the PoA sends a Link_up message to the MIHF@MAG that forwards it to the MIH-User@MAG. This message triggers the PMIPv6 registration through the corresponding signaling (PBU/PBA handshake, see Section 2.1.2). At the end of the PMIPv6 procedure, the MN receives a Router Advertisement necessary for the IP configuration, including the IPv6 default route.

As soon as the MN has obtained a valid default route, a MIH peer discovery procedure takes place to retrieve the HOC's address (e.g., via DNS). Subsequently, the MIHF@MN initiates the MIH Registration procedure sending an MIH_Register.request to the MIHF@HOC entity. The HOC shall check the MN MIHF_ID, allocate its MIH registration entry and finally reply with a MIH_Register.response message. In case of successful MIH registration, after the MIH_Register.response, the MIHF@HOC also sends a MIH_Configure_Link message to the MIHF@MN, to configure various parameters related to the active link. One of such parameters is the threshold used to detect the worsening of the active link radio conditions.

The MIH_Configure_Link concludes the Bootstrap procedure, as the MN has IP connectivity and can roam across the domain without experiencing any session discontinuity, taking advantage of the MIH functionalities offered by the MIHF@HOC which it has registered to.

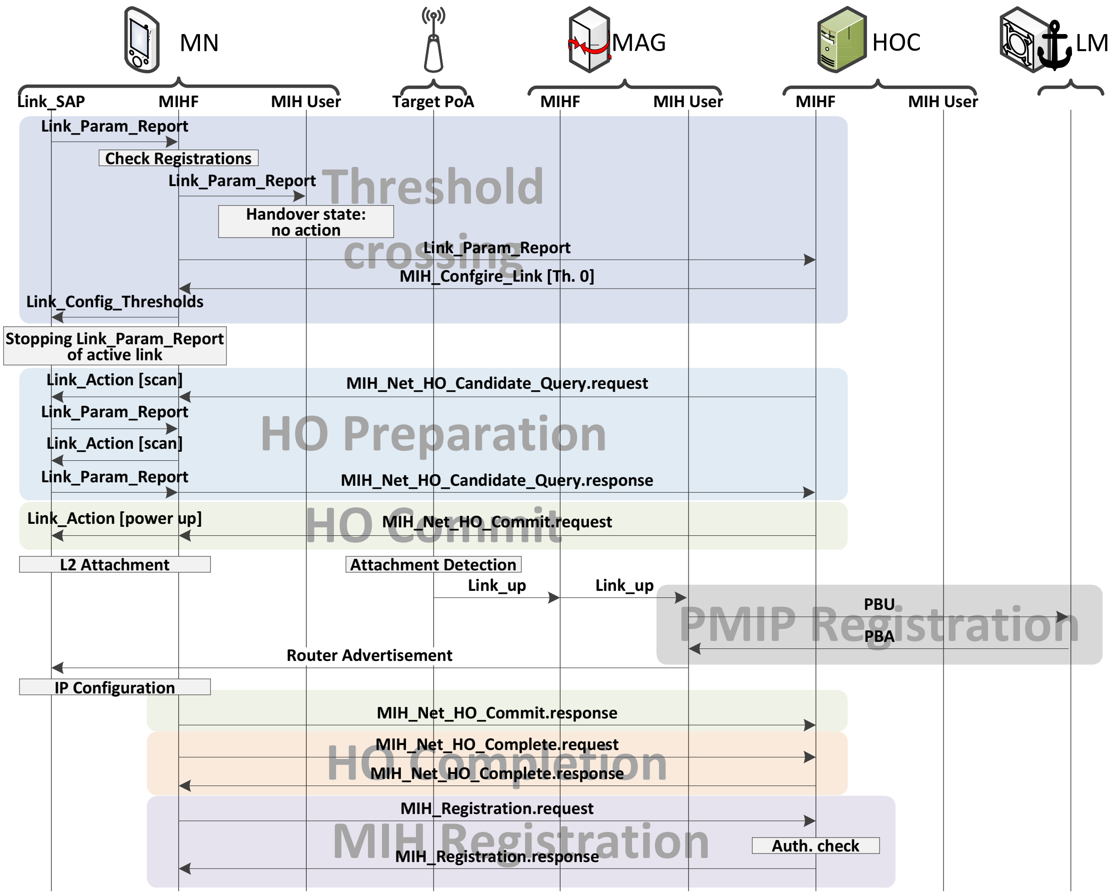


Figure B.3: MIH proactive handover signaling

B.1.2 Handover Procedure

The handover (HO) procedure is shown in Figure B.3. The MN periodically triggers measurement reports about the active radio link conditions. Upon crossing the previously configured threshold, the MIHF@MN sends a MIH_Link_Parameter_Report to the MIHF@HOC informing about the event, and consequently, the MIHF@HOC initiates the handover preparation by replying with a MIH_Net_HO_Candidate_Query.request message. When the MN receives the Candidate Query, a series of Link_Action.request are sent from the MIHF@MN to the Link_SAP to scan the radio conditions for the different candidate target PoAs. The lower layers (device technology drivers) collect the required information and reply back to the MIHF@MN, which generates and sends to the HOC a MIH_Net_HO_Candidate_Query.response with the list of found PoAs, ordered by the best received signal strength.

When the MIHF@HOC receives the response, it can correlate the received data with other information it has regarding the available access networks in the domain. Based on different criteria and on the intelligence implemented in the MIH-User@HOC, the MIHF@HOC sends a MIH_Net_HO_Commit.request message to the MN (different parameters can be sent taking into account policies for resource allocation, user profile, application requirements). Upon receiving the Commit.request, the MIHF@MN triggers a Link_action.request primitive towards the Link_SAP with link action set to Power_up. The Link_SAP then translates the primitive in the specific technology one in order to perform the attachment to the target link.

As the presence of a new attached MN is detected by the new PoA, a MIH_Link_up message is sent to the corresponding MAG, and next the normal PMIPv6 signaling is triggered. After the IP configuration is completed, the MN can send the MIH_Net_HO_Commit.response message to the HOC to inform it about the new attachment result. Additionally, the MIHF@MN sends a MIH_MN_HO_Complete.request message to ensure that on the network side the procedure is also complete. Upon receiving the message the MIHF@HOC can update its cache with the new location of the handed-over MN. After the MIH_MN_HO_Complete.response is sent back to the MIHF@MN, a new registration is triggered between the MN and the HOC. The registration is used, in this specific case, to simply renew the old expiration time and data, therefore it could be theoretically skipped or postponed. It should be noted that in case of changing the HOC, a discovery and registration would be needed in order to proceed with any further new communication. After the registration is complete and the MN receives from the HOC the settings for the thresholds by means of the MIH_Configure_Link_message, the MIHF@MN state turns to idle and the terminal is ready to restart this procedure once again and report measurements periodically.

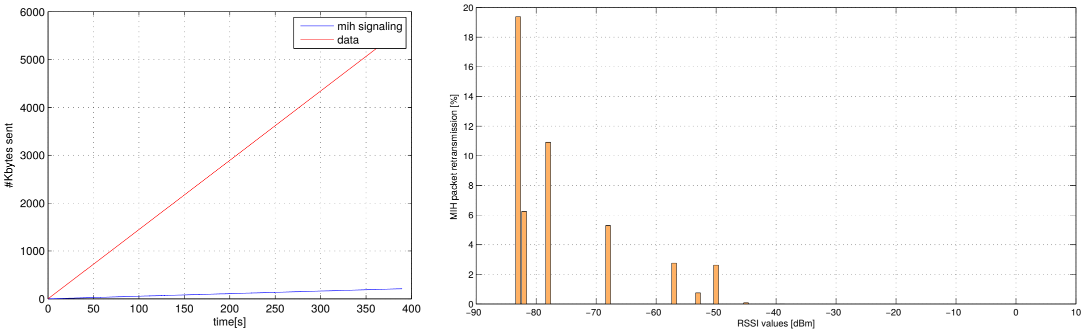
It is worth noting that through the use of the IEEE 802.21 standard and the coordination between the different MIHFs, it is possible to execute a make-before-break handover.

B.2 Results from an implementation study case

The proposed architecture has been implemented and set up in a test-bed for an IPv6 network comprising three MAGs, an LMA, and an MN (the nodes are PCs running Linux kernel 2.6.16, and all the MIH functions/signaling was implemented in C). Due to testbed limitations in our scenario we only built a IEEE 802.11g wireless access, thus replicating a WiFi-to-WiFi intra-technology handover. To emulate large network delays the Imunes[1] platform has been configured between the MAGs and LMA to artificially increase/decrease the end-to-end RTT between the MN and the LMA. The LMA further implements the HOC controller for network controlled handovers.

The proposed platform exploits two different protocols to achieve a cross-layer mobility management: the goal of both is to minimize the over-the-air signaling (thus reducing the associated bandwidth consumption), as well the handover latency (thus reducing the

[1] http://old.tel.fer.hr/imunes/



(a) Ratio between a VoIP stream and the MIH signaling over IEEE 802.11g

(b) ReTX vs RSSI

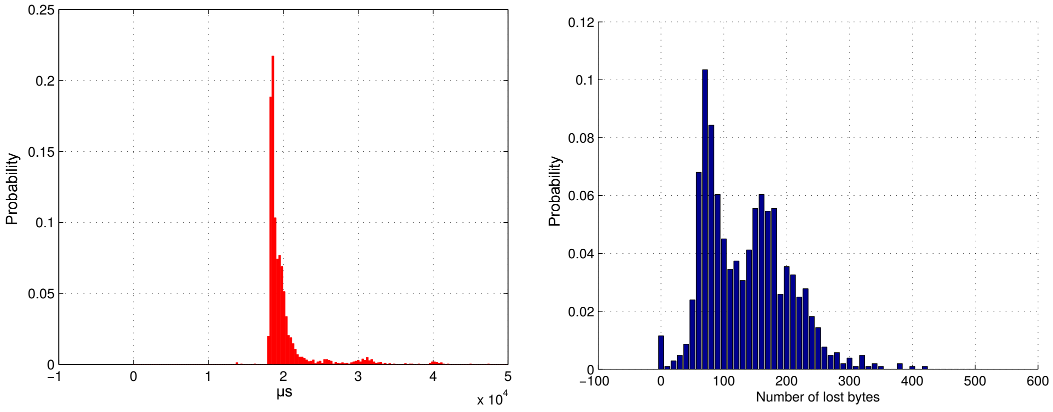
Figure B.4: Experimental results for the signaling load.

associated packet loss). What follows is the measurements performed and their results for the signalling load and the handover latency.

Since the PMIPv6 protocol runs only between network components, the signaling study focuses on the MIH messages sent through the wireless interface between the MN and the network. To prove that our system provides similar results in a real scenario with respect to the simulations reported in [130], we measured the required signaling to perform handovers between the three MAGs. Fig. B.4(a) represents more than 200 handovers performed in 400 seconds. We captured both the signaling data and the VoIP stream running between the MN and a correspondent node in the network and plotted the cumulative function over time. The different slope between the MIH and VoIP lines shows how little the MIH traffic costs over time. This experimental outcome matches the simulated results in [130].

It is worth noting that packet loss over the wireless medium is often a reason for performance decay in telecommunications systems, thus a packet retransmission mechanism should be implemented. To fulfill this requirement we followed the approach suggested in [127], that, instead of using TCP reliability to carry the MIH messages, leverages on the MIH built-in request/response scheme over UDP datagrams. The system has been tested in different radio environments spanning from very good conditions to very poor link quality. Fig. B.4(b) shows the relationship between the control message retransmissions and the RSSI. It is important to notice that in the tests we did, most of the retransmissions were in the form of bursts generated by the same packet retransmitted several times (up to 4 or 5). Those tests were repeated several times and produced slightly different results, but the trend was constant and proved the importance of the acknowledgement mechanism, effectively increasing the efficiency of MIH operations. The plot supports the considerations given above showing where retransmissions take place.

Seamless mobility is one of the requirements to provide real time services in next generation networks. Handover latency is here regarded as the time required to switch



(a) Latency distribution

(b) Packet loss

Figure B.5: Experimental results for the handover over a VoIP stream.

link as well as to configure the link for IP communication. To evaluate handoff latency we introduced timestamps in the code executed in MIHF@MN process and we logged the operations flow. We can see from the collected data shown in Fig. B.5(a), that the average handover delay is about 20 ms, no handover takes less than 15 ms and very few more than 35 ms. The main contribution is given by the time required by the physical layer to attach to the new link, since we deployed the PMIPv6 nodes such to have a delay between LMA and MAGs of 1 ms.

The time required to switch link has an impact in the time the terminal is not reachable and packets cannot be delivered. To evaluate the packet loss during handovers we employed a standard VoIP stream (100 packets/s, 100 Bytes/packet), and we calculated the amount of received data every 100 ms. In Fig. B.5(b) we can see the distribution of packet loss for the VoIP data stream. We argue an average of 300 bytes lost per handover is an acceptable result for real time multimedia communications.

In order to simulate a real world scenario we employed Imunes to increase routing delays between the MAGs and the LMA. Thus we emulated the case of an MN handing over from a link with a small RTT to links with one and two orders of magnitude larger RTT. We configured the Imunes machine with the following parameters:

- RTT LMA ⇔ MAG1: 1 ms (same value of previous tests)
- RTT LMA ⇔ MAG2: 20 ms
- RTT LMA ⇔ MAG3: 200 ms

The results obtained in this test nicely match the handover delay results, since what we expected was a longer latency mainly affected by the increased RTT due to the PBU/PBA exchange. In the tests with 20 ms and 200 ms RTT, we got a set values distributed around 40 ms and 220 ms respectively, that corresponds to the values already obtained for the 1 ms RTT case, plus artificial RTT, underlining the impact of a very large RTT on the overall handover latency.

Given the nature of our design, it should be noted that the RTT affects only the attachment to the new link. In fact the optimal threshold configuration for handover initiation allows the HOC to timely start the handover preparation phase while the MN is still attached to the old link. After the MN switches to the new link, the Link_up message triggers the PMIPv6 registration which is affected by the RTT (registration requires an handshake between MAG and LMA). Since the HOC knows the status of the network, it is desirable to consider the RTT as parameter when performing target selection, aiming at minimizing the effect of moving from one link with low RTT to a link with a larger one.

References

[1] CISCO, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018,” in *White Paper*, February 2014.

[2] F. Giust, A. De La Oliva, and C. J. Bernardos, “Flat Access and Mobility Architecture: an IPv6 Distributed Client Mobility Management solution,” in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, April 2011, pp. 361–366.

[3] C. J. Bernardos, A. De La Oliva, and F. Giust, “An IPv6 Distributed Client Mobility Management approach using existing mechanisms,” Internet-Draft (work in progress), draft-bernardos-dmm-cmip-01.txt, IETF, January 2014.

[4] F. Giust, A. De La Oliva, C. J. Bernardos, and R. Costa, “A network-based localized mobility solution for Distributed Mobility Management,” in *Wireless Personal Multimedia Communications (WPMC), 2011 14th IEEE International Symposium on*, October 2011, pp. 1–5.

[5] C. J. Bernardos, A. De La Oliva, and F. Giust, “A PMIPv6-based solution for Distributed Mobility Management,” Internet-Draft (work in progress), draft-bernardos-dmm-pmip-03.txt, IETF, January 2014.

[6] F. Giust, C. J. Bernardos, S. Figueiredo, P. Neves, and T. Melia, “A hybrid MIPv6 and PMIPv6 distributed mobility management: The MEDIEVAL approach,” in *Computers and Communications (ISCC), 2011 IEEE Symposium on*, June 2011, pp. 25–30.

[7] F. Giust, C. J. Bernardos, and A. De La Oliva, “HDMM: Deploying client and network-based Distributed Mobility Management. A hybrid approach,” *Telecommunication Systems, Springer*, 2015, to appear.

[8] ——, “Analytic Evaluation and Experimental Validation of a Network-Based IPv6 Distributed Mobility Management Solution,” *Mobile Computing, IEEE Transactions on*, vol. 13, no. 11, pp. 2484–2497, November 2014.

[9] F. Giust, A. De La Oliva, and C. J. Bernardos, “Mobility management in next generation mobile networks,” in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 14th IEEE International Symposium and Workshops on a*, June 2013, pp. 1–3.

[10] F. Giust, L. Cominardi, and C. J. Bernardos, “Distributed Mobility Management for future 5G networks: overview and analysis of existing approaches,” *Communications Magazine, IEEE*, vol. 53, pp. 142–149, January 2015.

[11] L. Cominardi, F. Giust, C. J. Bernardos, and A. De La Oliva, "Distributed mobility management solutions for next mobile network architectures," *under submission.*

[12] D. Munaretto, F. Giust, G. Kunzmann, and M. Zorzi, "Performance analysis of dynamic adaptive video streaming over mobile content delivery networks," in *Communications (ICC), 2014 IEEE International Conference on*, June 2014, pp. 1053–1058.

[13] F. Giust, G. Kunzmann, D. Munaretto, C. J. Bernardos, and B. Sayadi, "Caching in flat mobile networks: design and experimental analysis," *Vehicular Technology Conference Spring (VTC 2015-Spring), 2015 81st IEEE*, April 2015, to appear.

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

[15] C. Perkins, "IP Mobility Support for IPv4," RFC 3344, IETF, August 2002.

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

[17] C. Perkins, D. Johnson, and J. Arkko, "Mobility Support in IPv6," RFC 6275, IETF, July 2011.

[18] C. Perkins, "IP Encapsulation within IP," RFC 2003, IETF, October 1996.

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

[20] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," RFC 4862, IETF, September 2007.

[21] J. Arkko, V. Devarapalli, and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents," RFC 3776, IETF, June 2004.

[22] V. Devarapalli and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture," RFC 4877, IETF, April 2007.

[23] S. Kent, "IP Encapsulating Security Payload (ESP)," RFC 4303, IETF, December 2005.

[24] V. Manral, "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)," RFC 4835, IETF, April 2007.

[25] P. Nikander, J. Arkko, T. Aura, G. Montenegro, and E. Nordmark, "Mobile IP Version 6 Route Optimization Security Design Background," RFC 4225, IETF, December 2005.

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

[27] LAN/MAN Committee of the IEEE Computer Society, "Media Independent Handover Services," IEEE, Standards for Local and Metropolitan Area 802.21, 2008.

[28] 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, September 2010.

[29] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865, IETF, June 2000.

[30] V. Fajardo, J. Arkko, J. Loughney, and G. Zorn, "Diameter Base Protocol," RFC 6733, IETF, October 2012.

[31] 3GPP, "Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C)," 3GPP, TS 29.274, September 2011.

[32] M. Olsson, S. Rommer, C. Mulligan, S. Sultana, and L. Frid, *SAE and the Evolved Packet Core: Driving the mobile broadband revolution.* Academic Press, 2009.

[33] P. Lescuyer and T. Lucidarme, *Evolved Packet System (EPS): The LTE and SAE Evolution of 3G UMTS.* Wiley, 2008.

[34] A. De La Oliva, A. Banchs, I. Soto, T. Melia, and A. Vidal, "An overview of IEEE 802.21: Media-Independent Handover Services," *Wireless Communications, IEEE*, vol. 15, no. 4, pp. 96–103, August 2008.

[35] 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 & Mobile Summit (FuNeMS), 2011*, 2011, pp. 1–8.

[36] E. Demaria and L. Marchetti, "Dimensioning considerations for distributed mobility architecture," Internet-Draft (work in progress), draft-demaria-dmm-dimensioning-considerations-00.txt, IETF, March 2012.

[37] P. Bertin, S. Bonjour, and J. Bonnin, "Distributed or Centralized Mobility?" in *Global Communications Conference (GLOBECOM), 2009 IEEE*, November 2009, pp. 1–6.

[38] P. Bertin, S. Bonjour, and J.-M. Bonnin, "A Distributed Dynamic Mobility Management Scheme Designed for Flat IP Architectures," in *New Technologies, Mobility and Security (NTMS), 2008 IFIP International Conference on*, November 2008, pp. 1–5.

[39] M. Fischer, F.-U. Andersen, A. Kopsel, G. Schafer, and M. Schlager, "A Distributed IP Mobility Approach for 3G SAE," in *Personal, Indoor and Mobile Radio Communications (PIMRC), 2008 19th IEEE International Symposium on*, September 2008, pp. 1–6.

[40] H. A. Chan, H. Yokota, J. Xie, P. Seite, and D. Liu, "Distributed and Dynamic Mobility Management in Mobile Internet: Current Approaches and Issues," *Journal of Communications, Academy Publisher*, vol. 6, no. 1, pp. 4–15, 2011.

[41] Chan, H. A. (Ed.), "Requirements for Distributed Mobility Management," RFC 7333, IETF, August 2014.

[42] D. Liu, J. C. Zúñiga, P. Seite, H. A. Chan, and C. J. Bernardos, "Distributed Mobility Management: Current Practices and Gap Analysis," RFC 7429, IETF, January 2015.

[43] J. C. Zúñiga, C. J. Bernardos, A. De La Oliva, R. Costa, and A. Reznik, "Distributed Mobility Management: A Standards Landscape," *Communications Magazine, IEEE*, vol. 51, March 2013.

[44] 3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access," 3GPP, TS 23.401, September 2011.

[45] ——, "Local IP Access and Selected IP Traffic Offload (LIPA-SIPTO)," 3GPP, TR 23.829, October 2011.

[46] ——, “LIPA Mobility and SIPTO at the Local Network,” 3GPP, TR 23.859, April 2013.

[47] ——, “Study on co-ordinated Packet data network GateWay (PGW) Change for Selected IP Traffic Offload (CSIPTO),” 3GPP, TR 22.828, June 2014.

[48] ETSI, “Network Functions Virtualisation (NFV); Use Cases,” GS 001, October 2013.

[49] W. Hahn, “3GPP Evolved Packet Core support for distributed mobility anchors: Control enhancements for GW relocation,” in *ITS Telecommunications (ITST), 2011 11th IEEE International Conference on*, 2011, pp. 264–267.

[50] ——, “Flat 3GPP Evolved Packet Core,” in *Wireless Personal Multimedia Communications (WPMC), 2011 14th IEEE International Symposium on*, October 2011, pp. 1–5.

[51] C. J. Bernardos, J. C. Zúñiga, and A. Reznik, “Towards Flat and Distributed Mobility Management: a 3GPP Evolved Network Design,” in *Communications (ICC), 2012 IEEE International Conference on*, June 2012, pp. 6855–6861.

[52] Y.-H. Kim, Y.-H. Han, M. Kim, Y. S. Park, S. J. Moon, J. H. Lee, and D. K. Choi, “Distributed PDN gateway support for scalable LTE/EPC networks,” in *Consumer Communications and Networking Conference (CCNC), 2014 11th IEEE*, Jan 2014, pp. 139–144.

[53] M. Liu, X. Guo, A. Zhou, S. Wang, Z. Li, and E. Dutkiewicz, “Low latency IP mobility management: protocol and analysis,” *EURASIP Journal on Wireless Communications and Networking, Springer*, vol. 2011, no. 1, pp. 1–16, 2011.

[54] R. Wakikawa, G. Valadon, and J. Murai, “Migrating home agents towards internet-scale mobility deployments,” in *emerging Networking EXperiments and Technologies (CoNEXT), 2006 ACM International Conference on*, December 2006.

[55] H. Ali-Ahmad, M. Ouzzif, P. Bertin, and X. Lagrange, “Distributed dynamic mobile IPv6: Design and evaluation,” in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, April 2013, pp. 2166–2171.

[56] H. A. Chan, “Proxy Mobile IP with distributed mobility anchors,” in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, December 2010, pp. 16–20.

[57] P. P. Ernest, H. A. Chan, and O. E. Falowo, “Distributed mobility management scheme with mobility routing function at the gateways,” in *Global Communications Conference (GLOBECOM), 2012 IEEE*, December 2012, pp. 5254–5259.

[58] P. P. Ernest, H. A. Chan, J. Xie, and O. E. Falowo, “Mobility management with distributed mobility routing functions,” *Telecommunication Systems, Springer*, 2015, to appear.

[59] P. P. Ernest, H. A. Chan, O. E. Falowo, L. A. Magagula, and S. Cespedes, “Network-based distributed mobility management for network mobility,” in *Consumer Communications and Networking Conference (CCNC), 2014 11th IEEE*, 2014, pp. 417–425.

[60] P. P. Ernest and H. A. Chan, “Enhanced handover support and routing path optimization with distributed mobility management in flattened wireless networks,” in *Wireless Personal Multimedia Communications (WPMC), 2011 14th IEEE International Symposium on*, October 2011, pp. 1–5.

[61] K. Xue, L. Li, P. Hong, and P. McCann, "Context Transfer for Multicast support in Distributed Mobility Management (DMM)," Internet-Draft (work in progress), draft-vonhugo-multimob-dmm-context-02, IETF, March 2013.

[62] L. Yi, H. Zhou, D. Huang, and H. Zhang, "D-PMIPv6: A distributed mobility management scheme supported by data and control plane separation," *Mathematical and Computer Modelling, Elsevier*, vol. 58, no. 5, pp. 1415–1426, 2013.

[63] H. Jung, M. Gohar, J. Kim, and S. Koh, "Distributed mobility control in Proxy Mobile IPv6 networks," *Communications, IEICE Transactions on*, vol. 94, no. 8, pp. 2216–2224, 2011.

[64] M. Boc, A. Petrescu, and C. Janneteau, "Anchor-based routing optimization extension for Proxy Mobile IPv6 in flat architectures," in *Wireless Personal Multimedia Communications (WPMC), 2011 14th IEEE International Symposium on*, October 2011, pp. 1–5.

[65] P. Bertin, S. Bonjour, and J.-M. Bonnin, "An Evaluation of Dynamic Mobility Anchoring," in *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 70th IEEE*, September 2009, pp. 1–5.

[66] K. Munir, X. Lagrange, P. Bertin, K. Guillouard, and M. Ouzzif, "Performance analysis of mobility management architectures in cellular networks," *Telecommunication Systems, Springer*, 2015, to appear.

[67] P. Louin and P. Bertin, "Network and host based distributed mobility," in *Wireless Personal Multimedia Communications (WPMC), 2011 14th IEEE International Symposium on*, October 2011, pp. 1–5.

[68] P. Seite and P. Bertin, "Distributed Mobility Anchoring," Internet-Draft (work in progress), draft-seite-dmm-dma-07.txt, IETF, February 2014.

[69] H. Ali-Ahmad, M. Ouzzif, P. Bertin, and X. Lagrange, "Comparative performance analysis on dynamic mobility anchoring and proxy mobile IPv6," in *Wireless Personal Multimedia Communications (WPMC), 2012 15th IEEE International Symposium on*, September 2012, pp. 653–657.

[70] ——, "Performance analysis on network-based distributed mobility management," *Wireless Personal Communications, Springer*, vol. 74, no. 4, pp. 1245–1263, 2014.

[71] ——, "Distributed Mobility Management: Approaches and analysis," in *Communications Workshops (ICC), 2013 IEEE International Conference on*, June 2013, pp. 1297–1302.

[72] L. Yi, H. Zhou, D. Huang, and H. Zhang, "Performance analysis for distributed mobility management schemes based on flow duration," in *GLOBECOM Workshops (GC Wkshps), 2012 IEEE*, December 2012, pp. 1068–1072.

[73] C.-S. Li, F. Lin, and H.-C. Chao, "Routing optimization over network mobility with distributed home agents as the cross layer consideration," *Telecommunication Systems, Springer*, vol. 42, no. 1-2, pp. 63–76, 2009.

[74] T.-X. Do and Y. Kim, "Distributed network mobility management," in *Advanced Technologies for Communications (ATC), 2012 IEEE International Conference on*, October 2012, pp. 319–322.

[75] R. Farha, K. Khavari, N. Abji, and A. Leon-Garcia, "Peer-to-Peer mobility management for all-IP networks," in *Communications (ICC), 2006 IEEE International Conference on*, June 2006, pp. 1946–1952.

[76] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 149–160, August 2001.

[77] Y. Zhai, Y. Wang, I. You, J. Yuan, Y. Ren, and X. Shan, "A DHT and MDP-based Mobility Management Scheme for Large-Scale Mobile Internet," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, April 2011, pp. 379–384.

[78] L. Yu, Z. Zhijun, L. Tao, and T. Hui, "Distributed Mobility Management Based on Flat Network Architecture," in *Wireless Internet Conference (WICON), 2010 The 5th ICST*, March 2010, pp. 1–6.

[79] P. McCann, "Design of a flat wireless Internet Service Provider network," in *Wireless Personal Multimedia Communications (WPMC), 2011 14th IEEE International Symposium on*, October 2011, pp. 1–5.

[80] ——, "Authentication and Mobility Management in a Flat Architecture," Internet-Draft (work in progress), draft-mccann-dmm-flatarch-00.txt, IETF, March 2012.

[81] K.-H. Lee, H.-W. Lee, W. Ryu, and Y.-H. Han, "A scalable network-based mobility management framework in heterogeneous IP-based networks," *Telecommunication Systems, Springer*, vol. 52, no. 4, pp. 1989–2002, 2013.

[82] V. Kafle, Y. Kobari, and M. Inoue, "A Distributed Mobility Management scheme for future networks," in *Kaleidoscope 2011: The Fully Networked Human? - Innovations for Future Networks and Services (K-2011), Proceedings of ITU*, December 2011, pp. 1–7.

[83] A. Gurtov, M. Komu, and R. Moskowitz, "Host identity protocol: identifier/locator split for host mobility and multihoming," *The Internet Protocol Journal*, vol. 12, no. 1, pp. 27–32, 2009.

[84] A. García-Martínez, M. Bagnulo, and I. Van Beijnum, "The SHIM6 architecture for IPv6 multihoming," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 152–157, 2010.

[85] H. Zhang, F. Qiu, H. Zhou, X. Li, and F. Song, "A Distributed Mobility Management Solution in LISP networks," Internet-Draft (work in progress), draft-zhang-dmm-lisp-00.txt, IETF, September 2012.

[86] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "The Locator/ID Separation Protocol (LISP)," RFC 6830, IETF, January 2013.

[87] C. Bernardos, A. De La Oliva, P. Serrano, A. Banchs, L. M. Contreras, J. Hao, and J. C. Zúñiga, "An architecture for software defined wireless networking," *Wireless Communications, IEEE*, vol. 21, no. 3, pp. 52–61, June 2014.

[88] L. Valtulina, M. Karimzadeh Motallebi Azar, G. Karagiannis, G. J. Heijenk, and A. Pras, "Performance Evaluation of a SDN/OpenFlow-Based Distributed Mobility Management (DMM) Approach in Virtualized LTE Systems," in *GLOBECOM Workshops (GC Wkshps), 2014 IEEE*, December 2014, pp. 105–110.

[89] T. Aura, "Cryptographically Generated Addresses (CGA)," RFC 3972, IETF, March 2005.

[90] J. Laganier, "Authorizing Mobile IPv6 Binding Update with Cryptographically Generated Addresses," Internet-Draft (work in progress), draft-laganier-mext-cga-01.txt, IETF, October 2010.

[91] Crawford, M. and Haberman, B. , "IPv6 Node Information Queries," RFC 4620, IETF, August 2006.

[92] A. De La Oliva, I. Soto, M. Calderón, C. J. Bernardos, and M. I. Sánchez, "The costs and benefits of combining different IP mobility standards," *Computer Standards & Interfaces, Elsevier*, vol. 35, no. 2, pp. 205–217, 2013.

[93] H. Soliman, "Mobile IPv6 Support for Dual Stack Hosts and Routers," RFC 5555, IETF, June 2009.

[94] F. Baumann and I. Niemegeers, "An evaluation of location management procedures," in *Universal Personal Communications (ICUPC), 1994 3rd IEEE International Conference on*, September 1994, pp. 359–364.

[95] I. F. Akyildiz, J. S. M. Ho, and Y.-B. Lin, "Movement-based location update and selective paging for PCS networks," *Networking, IEEE/ACM Transactions on*, vol. 4, no. 4, pp. 629–638, August 1996.

[96] Y.-B. Lin, "Reducing location update cost in a PCS network," *Networking, IEEE/ACM Transactions on*, vol. 5, no. 1, pp. 25–33, February 1997.

[97] Y. Fang, "Movement-based mobility management and trade off analysis for wireless mobile networks," *Computers, IEEE Transactions on*, vol. 52, no. 6, pp. 791–803, 2003.

[98] S. Pack and Y. Choi, "A Study on Performance of Hierarchical Mobile IPv6 in IP-Based Cellular Networks," *Communications, IEICE Transactions on*, vol. 87, no. 3, pp. 462–469, March 2004.

[99] C. Makaya and S. Pierre, "An Analytical Framework for Performance Evaluation of IPv6-Based mobility Management Protocols," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 3, pp. 972–983, March 2008.

[100] I. Akyildiz and W. Wang, "A dynamic location management scheme for next-generation multitier PCS systems," *Wireless Communications, IEEE Transactions on*, vol. 1, no. 1, pp. 178–189, January 2002.

[101] S. Krishnan, R. Koodli, P. Loureiro, Q. Wu, and A. Dutta, "Localized Routing for Proxy Mobile IPv6," RFC 6705, IETF, September 2012.

[102] N. Moore, "Optimistic Duplicate Address Detection (DAD) for IPv6," RFC 4429, IETF, April 2006.

[103] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, IETF, January 2006.

[104] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, S. Sen, and O. Spatscheck, "To Cache or not to Cache: The 3G case," *Internet Computing, IEEE*, vol. 15, no. 2, pp. 27–34, 2011.

[105] B. A. Ramanan, L. M. Drabeck, M. Haner, N. Nithi, T. E. Klein, and C. Sawkar, "Cacheability Analysis of HTTP traffic in an Operational LTE Network," in *Wireless Telecommunications Symposium (WTS), 2013 IEEE*, April 2013, pp. 1–8.

[106] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *Communications Magazine, IEEE*, vol. 52, no. 2, pp. 131–139, 2014.

[107] F. Z. Yousaf, M. Liebsch, A. Maeder, and S. Schmid, "Mobile CDN Enhancements for QoE-Improved Content Delivery in Mobile Operator Networks," *Network, IEEE*, vol. 27, no. 2, pp. 14–21, 2013.

[108] S. Woo, E. Jeong, S. Park, J. Lee, S. Ihm, and K. Park, "Comparison of caching strategies in modern cellular backhaul networks," in *Mobile systems, applications, and services (MobiSys), 2013 11th ACM International Conference on*, June 2013, pp. 319–332.

[109] MEDIEVAL, "Advanced CDN mechanisms for video streaming," Deliverable D5.3, June 2012. [Online]. Available: http://www.ict-medieval.eu/resources/Medieval_D5.3_final_revised.pdf

[110] ——, "Final testbed status report," Deliverable D6.2, June 2013. [Online]. Available: http://ict-medieval.eu/resources/Medieval_D6.2.pdf

[111] ——, "Second Period Testing Report," Deliverable D6.4, July 2013. [Online]. Available: http://ict-medieval.eu/resources/MEDIEVAL_D6.4_Final.pdf

[112] ISO/IEC, "Dynamic adaptive streaming over HTTP (DASH)," SC 29 23009-1, April 2012.

[113] D. Corujo, C. Guimaraes, B. Santos, and R. Aguiar, "Using an open-source IEEE 802.21 implementation for network-based localized mobility management," *Communications Magazine, IEEE*, vol. 49, no. 9, pp. 114–123, September 2011.

[114] T. Melia and S. Gundavelli, "Logical Interface Support for multi-mode IP Hosts," Internet-Draft (work in progress), draft-ietf-netext-logical-interface-support-10.txt, IETF, September 2014.

[115] A. E. Essaili, E. Steinbach, D. Munaretto, S. Thakolsri, and W. Kellerer, "QoE-driven resource optimization for user generated video content in next generation mobile networks," in *Image Processing (ICIP), 2011 18th IEEE International Conference on.* IEEE, 2011, pp. 913–916.

[116] H. Ali-Ahmad, D. Moses, H. Moustafa, P. Seite, and T. Condexia, "Mobility Anchor Selection in DMM: Use-case Scenarios," Internet-Draft (work in progress), draft-aliahmad-dmm-anchor-selection-01.txt, IETF, July 2013.

[117] A. Yegin, J. Park, K. Kweon, and J. Lee, "Terminal-centric distribution and orchestration of IP mobility for 5G networks," *Communications Magazine, IEEE*, vol. 52, no. 11, pp. 86–92, November 2014.

[118] S. Bhandari, G. Halwasia, S. Gundavelli, H. Deng, L. Thiebaut, J. Korhonen, and I. Farrer, "DHCPv6 class based prefix," Internet-Draft (work in progress), draft-korhonen-dmm-prefix-properties-03.txt, IETF, October 2012.

[119] J. Korhonen, B. Patil, S. Gundavelli, S. P., and D. Liu, “IPv6 Prefix Mobility Management Properties,” Internet-Draft (work in progress), draft-bhandari-dhc-class-based-prefix-05.txt, IETF, July 2013.

[120] S. Figueiredo, S. Jeon, and R. L. Aguiar, “IP Multicast Use Cases and Analysis over Distributed Mobility Management,” Internet-Draft (work in progress), draft-sfigueiredo-multimob-use-case-dmm-03.txt, IETF, October 2012.

[121] D. von Hugo, H. Asaeda, and P. Seite, “IP Multicast Use Cases and Analysis over Distributed Mobility Management,” Internet-Draft (work in progress), draft-sfigueiredo-multimob-use-case-dmm-03.txt, IETF, October 2012.

[122] A. De La Oliva, C. J. Bernardos, M. Calderon, T. Melia, and J. C. Zúñiga, “IP flow mobility: smart traffic offload for future wireless networks,” *Communications Magazine, IEEE*, vol. 49, no. 10, pp. 124–132, October 2011.

[123] 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 Communications, Springer*, vol. 61, no. 4, pp. 603–627, 2011.

[124] G. Tsirtsis, H. Soliman, N. Montavont, G. Giaretta, and K. Kuladinithi, “Flow Bindings in Mobile IPv6 and Network Mobility (NEMO) Basic Support,” RFC 6089, IETF, January 2011.

[125] C. J. Bernardos (Ed.), “Proxy Mobile IPv6 Extensions to Support Flow Mobility,” Internet-Draft (work in progress), draft-ietf-netext-pmipv6-flowmob-11.txt, IETF, July 2014.

[126] K. Sun and Y. Kim, “Use case analysis for supporting flow mobility in DMM,” Internet-Draft (work in progress), draft-sun-dmm-use-case-analysis-flowmob-dmm-03.txt, IETF, October 2014.

[127] T. Melia, G. Bajko, S. Das, N. Golmie, and J. C. Zúñiga, “IEEE 802.21 Mobility Services Framework Design (MSFD),” RFC 5677, IETF, December 2009.

[128] P. Mockapetris, “Domain Names - Implementation and Specification,” RFC 1035, IETF, November 1987.

[129] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6),” RFC 3315, IETF, July 2003.

[130] T. Melia, A. De La Oliva, A. Vidal, I. Soto, D. Corujo, and R. Aguiar, “Toward IP converged heterogeneous mobility: A network controlled approach,” *Computer Networks, Elsevier*, vol. 51, no. 17, pp. 4849–4866, 2007.