UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

Analysis, Design and Experimental Evaluation of Connectivity Management in Heterogeneous Wireless Environments

Autor: Maria Isabel Sanchez Bueno
Director: Dr. Antonio de la Oliva, Universidad Carlos III

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

Leganés (Madrid), 2015

UNIVERSIDAD CARLOS III DE MADRID

Ph.D Thesis

Analysis, Design and Experimental Evaluation of Connectivity Management in Heterogeneous Wireless Environments

Author: Maria Isabel Sanchez Bueno
Advisor: Dr. Antonio de la Oliva, University Carlos III

DEPARTMENT OF TELEMATIC ENGINEERING

Leganés (Madrid), 2015

Analysis, Design and Experimental Evaluation of Connectivity Management in Wireless Networks in the Presence of Heterogeneous Access Technologies

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

Prepared by
Maria Isabel Sanchez Bueno, IMDEA Networks Institute, University Carlos III

Under the advice of
Dr. Antonio de la Oliva, University Carlos III

Departamento de Ingeniería Telemática, Universidad Carlos III de Madrid

---

Date: April 2015

This work has been supported by IMDEA Networks Institute.

TESIS DOCTORAL

Analysis, Design and Experimental Evaluation of Connectivity Management in Heterogeneous Wireless Environments

Autor: Maria Isabel Sanchez Bueno
Director: Dr. Antonio de la Oliva, Universidad Carlos III

Firma del tribunal calificador:

Firma:

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de 2015

Abstract

The future of network communications is mobile as many more users demand for ubiquitous connectivity. Wireless has become the primary access technology or even the only one, leading to an explosion in traffic demand. This challenges network providers to manage and configure new requirements without incrementing costs in the same amount. In addition to the growth in the use of mobile devices, there is a need to operate simultaneously different access technologies. As well, the great diversity of applications and the capabilities of mobile terminals makes possible for us to live in a hyper-connected world and offers new scenarios. This heterogeneity poses great challenges that need to be addressed to offer better performance and seamless experience to the final user. We need to orchestrate solutions to increase flexibility and empower interoperability.

Connectivity management is handled from different angles. In the network stack, mobility is more easily handled by IP mobility protocols, since IP is the common layer between the different access technologies and the application diversity. From the end-user perspective, the connection manager is in charge of handling connectivity issues in mobile devices, but it is an unstandardized entity so its performance is heavily implementation-dependent.

In this thesis we explore connectivity management from different angles. We study mobility protocols as they are part of our proposed solutions. In most of the cases we include an experimental evaluation of performance with 3G and IEEE 802.11 as the main technologies. We consider heterogeneous scenarios, with several access technologies where mobile devices have also several network interfaces. We evaluate how connectivity is handled as well as its influence in a handover. Based on the analysis of real traces from a cellular network, we confirm the suitability of more efficient mobility management.

Moreover, we propose and evaluate three different solutions for providing mobility support in three different heterogeneous scenarios. We perform an experimental evaluation of a vehicular route optimization for network mobility, reporting on the challenges and lessons learned in such a complicated networking environment. We propose an architecture for supporting mobility and enhance handover in a passive optical network deployment. In addition, we design and deploy a mechanism for mobility management based on software-defined networking.

Contents

List of Figures

List of Tables

List of Acronyms

ANDSF	Access Network Discovery and Selection Function
AR	Access Router
AP	Access Point
BA	Binding ACK
BC	Binding Cache
BR	Binding Refresh
BU	Binding Update
CLC	CROWD Local Controller
CGA	Cryptographically Generated Addresses
CN	Correspondent Node
CoA	Care of Address
CoRE	Care of Route Error
CoRT	Care of Route Test
CoRTI	Care of Route Test Init
COTS	Commercial-Off-The-Self
CRC	CROWD Regional Controller
CROWD	Connectivity management for eneRgy Optimised Wireless Dense networks
DAD	Duplicate Address Detection
DAS	Distributed Antenna Systems
DBA	Dynamic Bandwidth Allocation
DMM	Distributed Mobility Management
DMM-GW	Distributed Mobility Management Gateway
DNS	Domain Name Server
DHCP	Dynamic Host Configuration Protocol
DSRC	Dedicated Short Range Communications
EPON	Ethernet Passive Optical Network
FA	Foreign Agent
GPS	Global Positioning System
HA	Home Agent
HMIPv6	Hierarchical MIPv6

HoA Home Address
HoAA Home Address Advertisement
HoRT Home Route Test
HTTP Hypertext Transfer Protocol
IETF Internet Engineering Task Force
IoT Internet of Things
IP Internet Protocol
IPsec Internet Protocol Secure
LMA Local Mobility Anchor
LMD Localized Mobility Domain
M2M Machine to Machine
MAAR Mobility Anchor and Access Router
MAG Mobile Access Gateway
MANET Mobile Ad-hoc Network
MAP Mesh Access Point
MIH Media Independent Handover
MIP Mobile IP
MIPv6 Mobile IPv6
MN Mobile Node
MNP Mobile Network Prefix
MNPBU Mobile Network Prefix Binding Update
MR Mobile Router
MRHA Mobile Router Home Agent bidirectional tunnel
NEMO B.S. Network Mobility Basic Support
ODL Open Daylight
ONF Open Networking Foundation
OVS Open Virtual Switch
OVSDB Open Virtual Switch DataBase
P2P Peer to peer
PAN Personal Area Network
PBA Proxy Binding Acknowledgment
PBU Proxy Binding Update
PMIPv6 Proxy Mobile IPv6
PoA Point of Attachment
PON Passive Optical Network
PoS Point of Service
RA Router Advertisement
RAN Radio Access Network
RS Router Solicitation

RSA	Rivest, Shamir, Adleman
RSSI	Received Signal Strength Indicator
RTSP	Real Time Streaming Protocol
RTT	Round Trip Time
SDN	Software-Defined Network
SEND	Secure Neighbor Discovery Protocol
SLAAC	Stateless Address Autoconfiguration
SSH	Secure SHell
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TELNET	TELetype NETwork
TFTP	Trivial File Transfer Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Transmission System
UPS	Uninterruptible Power Supply
USB	Universal Serial Bus
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad hoc Network
VARON	Vehicular Ad hoc Route Optimization for NEMO
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
VSN	Vehicular Sensor Network
WAVE	Wireless Access in Vehicular Environments
WLAN	Wireless Local Area Network
WOBAN	Wireless Optical Broadband Access Network

Part I

Introduction and State of the Art

Chapter 1

Introduction

1.1 Motivation

Mobile data traffic demand is growing exponentially and this trend is expected to continue in the near future. With the development of more powerful mobile end-user terminals and the diversification of mobile devices the global mobile traffic generated is expected to increase nearly tenfold, according to [3]. Hence the necessity for the upcoming deployment of 5G networks, with the ambitious goals of increasing capacity by 1,000, providing 10 Gbps data rates while diminishing network latency and gaining energy efficiency [4]. Challenged by this explosion of traffic demand, mobile network operators are looking for efficient solutions, fostering the appearance of heterogeneous deployments that make use of diverse access network technologies.

To cope with the growing demand for connectivity, operators have also turned to the densification of the Radio Access Network (RAN) [5] by e.g., deploying small cells (micro, pico, femto cells) or Distributed Antenna Systems (DAS), increasing coverage as needed. By delivering service to the users in smaller areas, operators aim at improving spectral efficiency and alleviating the congestion due to management of a greater amount of connections. However, there is the challenge to deploy a higher number of Access Points, with the subsequent costs in terms of money and resources. In addition, denser deployments will translate into mobile users performing handover more frequently and still will demand to guarantee seamless connectivity and quality service delivery.

We are also witnessing a diversification not only in the access technologies, but also in the devices that get connected. Mobile phones have been the type of device with the largest growth so far, but it is expected to have 26 billion devices connected by 2020 [6] including phones, tablets, PCs, laptops, consumer electronics and other devices conforming the Internet of Things (IoT). Machine to Machine (M2M) communications are expected to keep growing, along with the raise of new scenarios: e.g., vehicular networks, smart cities, drones or connected homes.

Consequently, together with the significant increase in mobile traffic, the word to define connectivity management, in our days and in the near future, is heterogeneity. Connectivity management covers a broad field of research, in which we can find diversity at several levels:

- **User terminal**: End user enjoys a wide offer of mobile or portable devices which possibly are equipped with different network interfaces. The multiple combinations of hardware, operating systems and applications add complexity to the design of standard connectivity management solutions, which need to be carefully analyzed to provide the best user experience depending on the terminal characteristics.

- **Access technology**: The different families of wireless access technologies allow operators to diversify their offer, and users to choose according to the services provided. However, interoperability cannot be always guaranteed, the choice for the best network connection is hard to fulfill and the end user cannot benefit from the wide variety and the perks of this heterogeneity.

- **IP Mobility management**: The widely-used IP protocol has been extended with different flavors for mobility management, providing host-based solutions (Mobile IP [7]), network-based solutions (Proxy Mobile IPv6 [8]), network mobility protocols (Network Mobility Basic Support protocol [9]), distributed (Distributed Mobility Management, under discussion in the IETF[1]) or hierarchical deployments (Hierarchical Mobile IPv6 [10]). However, their different characteristics make each of the solutions more suitable for different scenarios and the requirements and implications derived from the usage of one or another protocol need to be evaluated, since there is not a unique solution that manages mobility efficiently in all scenarios.

- **Applications**: The market penetration of smart devices and the growing trend towards personalized and customizable services result in a plethora of applications. Each one of these applications handle user mobility and network connectivity in its own way.

This heterogeneity makes a unique solution not feasible nor appropriate to handle connectivity management. In this thesis, we make questions on the requirements, the characteristics and the needs of connectivity management from different points of view, which include the network, the mobility protocols, the characterization of mobile traffic and, ultimately, the end user. In this way, we can figure out the potential (combination of) solutions to enhance connectivity management in an efficient way to satisfy every player. We evaluate connectivity management in different wireless networking scenarios, including vehicular adhoc networks, software defined networking or wireless optical

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

broadband access networks. Our evaluation is, as much as possible, experimental and we mainly work with 3G (UMTS)[2] and IEEE 802.11 [11] as access technologies to test connectivity on Commercial-Off-The-Self (COTS) devices.

1.2 Thesis Contributions

In this thesis we study connectivity management in different scenarios. In our view, mobility management cannot be performed in a single way as there are heterogeneous environments where a unique solution cannot perform optimally in all of them. Instead, mobility management needs to be able to adapt to the requirements of the specific scenario and user demands. Following this motivation, we approach several scenarios to enhance connectivity management, e.g. vehicular networks, wireless-optical broadband access networks or software defined networks, and we also evaluate connectivity management in mobile terminals, wireless local area networks and IP mobility protocols. To the possible extent, our assessments are experimental.

The contributions of this thesis are summarized as follows. First, we study the performance of several IP mobility protocols and their combinations, providing an experimental comparison and measurements of handover delay. There are different approaches, designed to cover specific gaps on the initial IP specification or address specific issues, e.g., scalability, mobility of complete networks or routing optimizations. However, these protocols should not be considered in isolation, but complementing each other, since they tackle different situations.

Second, we apply some of these IP mobility protocols to three different networking scenarios: *i)* vehicular networks, *ii)* wireless-optical broadband access networks, and *iii)* software defined networking integrated with distributed mobility mechanisms. Vehicular networking is one of the most demanding scenario for connectivity management. Vehicular Ad hoc Network (VANET)s need connectivity management solutions tailored to their inherent characteristics, different to any other networking scenario. We base on VARON [12], a route optimization mechanism for network mobility (NEMO B.S.) protocol [9], to build an experimental prototype. Vehicular Ad hoc Route Optimization for NEMO (VARON) was already validated via extensive simulations, but we aimed at exploring the issues that would arise in real world VANET communications. Our experience let us improve the route optimization protocol described in [12], providing solutions to the problems we faced in the experimental phase. We believe our work can give very helpful insights for other researchers in the area, beyond improving the performance of our particular solution. Wireless Optical Broadband Access Networks (WOBANs) combine the performance of optical networks with the convenience of wireless technologies. Motivated by the increasing interest in the deployment of small cells, we take one step further and

[2] `http://www.3gpp.org/technologies/keywords-acronyms/103-umts`

propose an architecture to support mobility management and seamless handover by combining Ethernet Passive Optical Networks (EPON) with PMIPv6 [8] and IEEE 802.21 Media Independent Handover (MIH) [13] functionality. The combination of the EPON operation, PMIPv6 and MIH procedures enables seamless handover and reduces packet loss. Moved by the current trend towards virtualization and software defined networking, we design and experimentally validate a network-based mobility solution, which benefits of the flexibility and the scalability properties of Distributed Mobility Management (DMM) and Software-Defined Network (SDN). Our solution has been designed in the scope of the FP7 ICT Connectivity management for eneRgy Optimised Wireless Dense networks (CROWD) project and it is transparent to the mobile terminal, although we consider a host mobility solution too. We implement our mobility management mechanism in IEEE 802.11 networks, and demonstrate the promising features of our solution—in terms of mobility support, data path re-configuration, and transparency to the mobile devices—using OpenFlow protocol [14] to control access points and network switches. However, our approach is designed so that it can encompass other technologies (e.g., LTE) and heterogeneous access networks.

Third, we also evaluate the mechanisms provided by IEEE 802.11 standard family that can contribute to improving mobility and connectivity management. Along the experimental work developed for this thesis, we found IEEE 802.11 to have a major impact on the performance of mobility management procedures, being responsible for most of the interruption of a connection during handover. Therefore, in Chapter 7 we study some of the amendments to the IEEE 802.11 standard approved within the last years and part of the present version of the standard. Namely, we focus on 802.11k [15], 802.11r [16] and 802.11v [17] because of their potential to improve mobility and connectivity management. The slight impact of these amendments has called our attention on their feasibility and motivated us to evaluate their impact on the research community, as well as to look for practical implementations or commercial hardware including support for their operation. In particular, we focus on these amendments because there has been a reasonable time from their official approval, long enough to evaluate their impact.

Fourth, we focus on the performance of the connection manager running in mobile devices and we perform a thorough experimental analysis of connectivity management in the three main families of operating systems (Android, iOS and Windows Phone). This analysis covers handover performance and application survival to handover. As we work with mobile devices that support cellular and WiFi connectivity, we evaluate the possibility of using the WiFi interface for offloading the cellular connection. In VARON, the route optimization for VANETs that we implement, we leverage multi-interface connectivity management. In addition, mobile devices are commonly equipped with different network interfaces. One of the ways to take advantage of these capabilities is to considered WiFi as an alternative for 3G offloading, which can be used to just refer to the handover

of all IP flows from one interface (3G), to a secondary one (WiFi). The opportunities that this technology enables are maximized when a fine-grained flow selection is allowed. For example, an operator might prefer not to offload VoIP flows, due to the inherited difficulties in providing QoS guarantees on an unmanaged WiFi access, while video traffic might be always offloaded to a technology providing higher bandwidth. Mobile terminals are evolving towards intelligent, more powerful devices. Therefore, we can take advantage of this improvement in their capabilities to enhance an entity that we find of the utmost importance, although it is often disregarded: the *Connection Manager*. The smartphone operating system provides a set of mobility-related functions from which an application can benefit in case it decides to handle mobility. These functionalities depend on the operating system and include connectivity events such as network up or down events, and commands exposed to the application layer to extract information on connection availability. Usually the terminal connectivity is handled by the connection manager, in charge of deciding which is the best connection for the terminal in a specific moment, and the application has to deal with those decisions. However, there is not much documentation publicly available about the behavior of the connection manager.

Finally, we perform an analysis and characterization of mobile data traffic from a real network operator with actual data from their network, thanks to a collaboration through the FP7 CROWD project. We apply this characterization to confirm the suitability of distributed mobility management to current and future network deployments. We extract the user location information from the control plane and match it with the user data traffic. We parse and group the data to collect the main characteristics of the mobile traffic and apply it to mobility management. We compare and analyze the cost of applying distributed mobility management mechanisms in the operator's network, based on the real characteristics of mobile data traffic, such as, flow duration and frequency of handover, among others.

In a nutshell, the work in this thesis is distributed among *i)* the study of IP mobility protocols, applied to different scenarios, namely VANETs, WOBANs and SDN, *ii)* the study of connectivity management in mobile terminals and IEEE 802.11 networks and *iii)* the study of mobile data traffic to confirm the suitability of IP mobility protocols. The work in this thesis covers:

1.3 Thesis Overview

This thesis is structured in four parts. In the remaining of Part I, Chapter 2 reviews the state of the art and related work of the topics covered in this thesis. Part II studies mobility management and introduces the three scenarios we have evaluated. In Chapter 3 we study different combinations of mobility protocols, their associated overhead associated and indicate the scenario for each of them. We include as well an experimental

performance evaluation. Chapter 4 describes the process to make VARON operative in a real vehicular platform, from the initial implementation in a lab-controlled environment to the tests in the experimental prototype. We also define improvements for the original protocol design. In Chapter 5 we design an integrated mobility architecture for a wireless optical broadband access network, which takes advantages of the similarities in the EPON and PMIPv6 network architectures. In Chapter 6 we apply software-defined networking concepts to mobility management and present an architecture for a network-based and a host-based solution, both developed under the scope of the FP7 ICT CROWD project. Part III focus on connectivity management in devices with different network interfaces, namely WiFi and cellular interfaces. We address the impact of the WiFi connection in the management of mobility as we have experienced in our work that a significant part of the interruption during a handover is due to WiFi procedures. In Chapter 7 we study some of the amendments to the IEEE 802.11 standard approved within the last years and part of the present version of the standard. Namely, we focus on 802.11k [15], 802.11r [16] and 802.11v [17] because of their potential to improve mobility and connectivity management. In Chapter 8 we measure the impact on energy consumption derived from the simultaneous use of the cellular and WiFi interfaces for offloading. We evaluate this mechanism through several measurements, comparing the energy consumed by each interface on its different transmission states, proving that flow mobility allows a longer battery lifetime. In Chapter 9, we take the smartphone as a black box and analyze the current state of the mobility support at the connection manager in different terminals, providing a functional view of the differences between the major operating systems and the different improvements that can be done to optimize the mobile user experience. We assess and evaluate the default network connectivity, the intra- and inter-technology handover and application performance during handover for smartphones running the three main mobile operating system families, i.e., Android, iOS and Windows Phone. We advocate for the empowering of the connection manager as a tool to enhance connectivity management as a whole. In Chapter 10 we characterize mobile data traffic from a major network operator in Turkey. We extract information about mobility in their network and study the benefits of applying distributed mobility concepts given the observed characteristics. In Part IV we extract the main conclusions of our work and introduce future lines of work.

1.3.1 Summary of publications

This thesis covers contributions from the following publications:

- "On providing mobility management in WOBANs: Integration with PMIPv6 and MIH". M. Isabel Sanchez, Manuel Uruea, Antonio de la Oliva, Jose Alberto Hernandez, Carlos J. Bernardos. IEEE Communications Magazine, Vol. 51, Issue 10, October 2013 pp. 172-181. ISSN: 0163-6804 DOI: 10.1109/MCOM.2013.6619581.

- "The costs and benefits of combining different IP mobility standards". Antonio de la Oliva, Ignacio Soto, Maria Calderon, Carlos J. Bernardos, M. Isabel Sanchez. Computer Standards & Interfaces Vol. 35, Issue 2 (February 2013), pp. 205-217. Available online: 9th Sept. 2012. DOI: 10.1016/j.csi.2012.08.003.

- "On the implementation, deployment and evaluation of a networking protocol for VANETs: the VARON case". M. Isabel Sanchez, Marco Gramaglia, Carlos J. Bernardos, Antonio de la Oliva, Maria Calderon. Elsevier Ad Hoc Networks, Vol. 19, August 2014 pp. 9-27. Available online: 13th February 2014. DOI: 10.1016/j.adhoc.2014.02.001.

- "Experimental Analysis of Connectivity Management in Mobile Operating Systems" M. Isabel Sanchez, Antonio de la Oliva, Carlos J. Bernardos. Accepted for publication in Elsevier Computer Networks.

- "On IEEE 802.11k/r/v Amendments: Do They Have a Real Impact?" M. Isabel Sanchez, A. Boukerche. Accepted for publication in IEEE Wireless Communications Magazine.

- "Energy consumption savings with 3G offload". M. Isabel Sanchez, Carlos J. Bernardos, Antonio de la Oliva, Pablo Serrano. First International Workshop on Cloud Technologies and Energy Efficiency in Mobile Communication Networks (CLEEN 2013) (VTC2013-Fall Workshops), Sept. 2013, Las Vegas, NV, USA.

- "Mobility Management: Deployment and Adaptability Aspects Through Mobile Data Traffic Analysis". M. Isabel Sanchez, Engin Zeydan, Antonio de la Oliva, A. Serdar Tan, Utku Yabas and Carlos J. Bernardos. Submitted to Computer Communications Special Issue on Mobile Traffic Analytics.

Additionally, the following works have been published during the development of this thesis:

- "NS-3-based Real-time emulation of LTE Testbed using LabVIEW platform for Software Defined Networking (SDN) in CROWD Project". Rohit Gupta, Bjoern Bachmann, Russell Ford, Sundeep Rangan, Nikhil Kundargi, Amal Ekbal, Karamvir Rathi, Maria Isabel Sanchez Bueno, Antonio de la Oliva and Arianna Morelli. Workshop on ns-3 (WNS3), May 13th - 14th, Barcelona (Spain).

- "Tackling the increased density of 5G networks; the CROWD approach". M. Isabel Sanchez, Arash Asadi, Martin Drxler, Rohit Gupta, Vincenzo Mancuso, Arianna Morelli, Antonio de la Oliva and Vincenzo Sciancalepore. 5GARCH - VTC Spring, 11th May 2015, Glasgow (United Kingdom).

- "Modeling and Analysis of Opportunistic Routing in Multi-hop Wireless Networks". A. Darehshoorzade, M. Isabel Sanchez, A. Boukerche. The 22nd IEEE International Symposium on Modeling Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2014), September 2014, Paris, France.
- "The Playground of Wireless Dense Networks of the Future". Claudio Cicconetti, Arianna Morelli, M. Isabel Sanchez, Antonio de la Oliva, Vincenzo Mancuso, Martin Draexler, Rohit Gupta, Laurent Roullet, Hassan Ahmad. Accepted as poster, Future Networks and Mobile Summit, July 2013, Lisbon, Portugal.

Chapter 2

Related work

2.1 Introduction

This Chapter introduces the operation of the protocols and the main technologies related to the work developed in this thesis. This serves as background on vehicular ad hoc networks, software defined networking applied to mobility management, connectivity management in devices with several network interfaces, technologies aimed to facilitate handover, as IEEE 802.21, Ethernet passive optical networks and wireless optical broadband access networks. All these topics are addressed in this thesis, so we provide a compilation of the related work in the literature, identifying the gaps and the differences between our work and previous works.

2.2 IP Mobility Protocols

Although mobility can be managed at different layers of the protocol stack, the network layer is one of the most plausible candidates for several reasons. First of all, the movement of the user originates mainly changes in the localization and the routing, impacting directly the IP configuration. Furthermore, if the IP layer assumes the mobility management, the application layer remains agnostic of user mobility and potential changes of mobility management protocols. In addition, it enables the support for heterogeneous access networks. However, IP protocol was not originally designed to support mobility, therefore, other protocols have been designed to extend IP functionality for mobility management. This is the case of the IP mobility protocols such as Mobile IPv6 (MIPv6), Proxy Mobile IPv6 (PMIPv6) and the Network Mobility Basic Support protocol (NEMO B.S.), that we review hereafter.

2.2.1 Basic concepts about mobility

The role of IP addresses is twofold: they are used both as identifiers and locators and that makes difficult to support mobility. Mobility management decouples these two roles, as a mobile node keeps being reachable at the same IP address, regardless of its point of attachment to the network. In its Home Network, the mobile node has a *permanent* IP address, called Home Address (HoA) and when it changes its point of attachment the Mobile Node (MN) uses an additional *temporary address*, the Care-of Address (CoA), which is topologically correct in the visited network.

In host-based solutions for mobility management, the mobile node itself is responsible of the signaling and routing updates involved with the movement. This requires modifying the networking stack of the mobile terminal but increases flexibility. On the contrary, network-based solutions prevent the mobile node from the burden of mobility management and make the movement across the different networks transparent to the mobile terminal. In addition, we can also classify mobility as global – the mobile terminal is reachable from anywhere in the Internet – or localized – the mobility management operates in a delimited area, denoted as Localized Mobility Domain (LMD).

2.2.2 Host-based mobility: Mobile IPv6

Mobile IPv6 [18] introduces the entity of the Home Agent (HA), which keeps track of the movement of the mobile node out of its Home Network and binds both Home Address (HoA) and Care of Address (CoA) addresses when the mobile node is roaming, as shown in Figure 2.1. Upon attaching to a visited network and configuring a CoA, the mobile terminal notifies its HA by sending a Binding Update (BU) message. The HA updates the location of the mobile node in its Binding Cache (BC) and acknowledges the new configuration by the transmission of a Binding Acknowledge (BA) message. Then, the traffic to the mobile node will be addressed to its HoA, and after reaching its Home Network, it will be tunneled from the HA to the mobile node in its new location.

One of the main problems of this protocol is the so-called *triangular routing*, as illustrated in Figure 2.2. Despite the existence of a more efficient route, the traffic from a Correspondent Node (CN) to the MN is routed to the Home Network, where the HA encapsulates in a tunnel towards the MN. Analogously, the traffic from the MN to the CN has to be routed through the tunnel to the HA, which de-encapsulates and forwards it. In order to solve this problem, the specification defines a route optimization that enables the CN to route the traffic directly to the CoA of an MN without involving the HA by making use of the IPv6 Routing Header option. If the CN has a recent entry in its binding cache for the MN it can send packets directly to that address. Otherwise, the CN will not know the current CoA of the mobile node, so it will send the traffic to its HoA and the HA will forward it to the MN. Then, the MN will inform the CN of its current CoA.

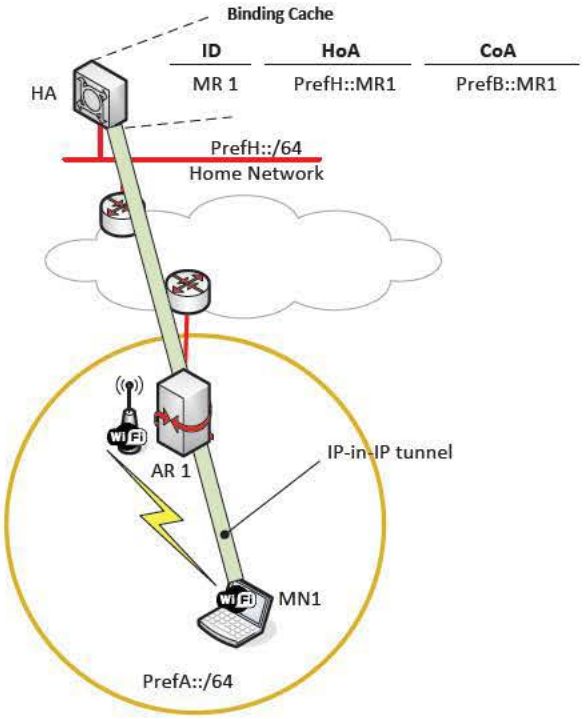


Figure 2.1: MIPv6 operation.

When the MN performs a handover to another network, it has to inform not only the HA but also the CN to update their bindings.

The latency introduced by signaling correspondent nodes can increase arbitrarily depending on their location. To decrease this delay, Hierarchical MIPv6 (HMIPv6) [19] proposes localized mobility management by a Mobility Anchor Point (MAP) that handles the movement of the MN inside the domain, hiding it to the HA and the CNs. Additionally, mobility between separate domains is handled by MIPv6. FMIPv6 [20] aims at reducing the interruption during handover by using link layer triggers to anticipate the movement of the MN. Authors in [21] propose a combination of HMIPv6 and FMIPv6, similar to the one considered by IETF [22]. Their simulation results in an 802.11 network suggest that although the improvement by HMIPv6 and FMIPv6 in the handover latency, depending on the traffic source and the load in the network, these variations can perform worse than MIPv6, due to the increase in signaling overhead. Simulation experiments in [23] show the difference in handover delay when the mobile node roams from the home network to a visited network or between two visited networks, illustrating in this way the need for the route optimization. In their simulations the delay between two visited networks is lower because the distance to the HA from the MN is higher than to the CN. There are also experimental works, not based on simulation, as [24] that

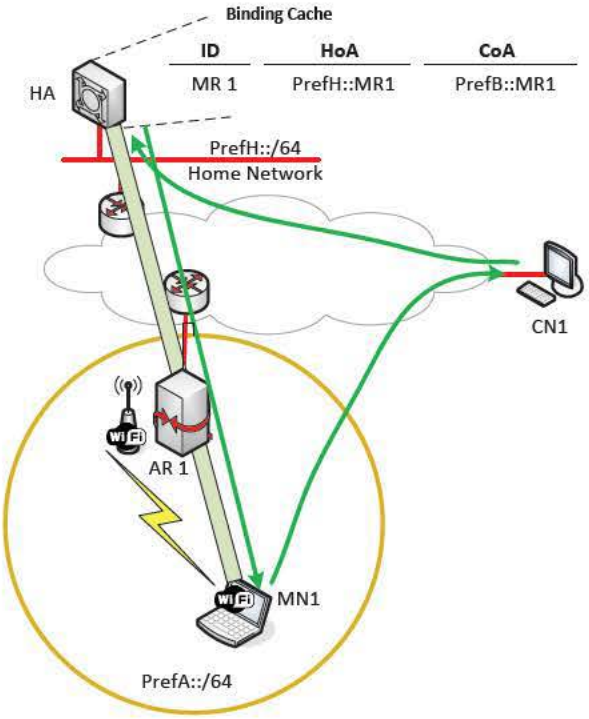


Figure 2.2: Triangular routing problem in MIPv6.

measures handover latency with MIPv6 in an 802.11b WLAN at both layer 2 and layer 3. They mention also HMIPv6 and fast handover variations, but do not include them in the experimental measurements. They consider scenarios with a single and multiple users at different data rates, but they conclude that for the most realistic scenario the contribution of the layer 2 to the total handover latency is very significant, reaching up to 8 s. As the operation of MIPv6 can only start after the interruption at layer 2, they consider introducing link-layer triggers to decrease the handover latency.

2.2.3 Network-based Mobility: Proxy Mobile IPv6

Proxy Mobile IPv6 (PMIPv6) [8] is a mobility management protocol that allows legacy mobile terminals to perform handover operations across heterogeneous networks, without their involvement in the management of their own IP mobility signaling. Proxy Mobile IPv6 (PMIPv6) provides mobility support within a localized area, called a Localized Mobility Domain (LMD). In this domain, the mobile node keeps the same IPv6 address although it changes its point of attachment to the network. For its operation, PMIPv6 defines the entities of Local Mobility Anchor (LMA) and Mobile Access Gateway (MAG). The MAG is usually the access router for the mobile node and it performs the signaling on behalf of the mobile nodes that attach to it. There are several MAGs in an LMD. The

LMA maintains the state regarding the location of the MN in the LMD and an IPv6-in-IPv6 tunnel with every MAG to forward the data traffic of their MNs. Figure2.3 shows an example of PMIPv6 operation. The LMD where PMIPv6 provides mobility support comprises two MAGs, and one LMA. When an MN first arrives at the LMD, it attaches to an Access Point and sends a Router Solicitation (RS) message requesting an IPv6 prefix. This message is received by the MAG, which asks the LMA for an IPv6 prefix for the MN through a Proxy Binding Update (PBU) message. Next, the LMA replies to the MAG with a newly assigned IPv6 prefix for the MN through a Proxy Binding Acknowledgment (PBA) message and stores the mapping in its local lookup table, the Binding Cache (BC). Then, the MAG forwards the IPv6 prefix to the MN through a Router Advertisement (RA) message. Finally, the LMA uses the existing IPv6-in-IPv6 tunnel with the MAG (or creates a new one if there is none) for the data traffic exchanged by the MN with the network. The process of handover is illustrated in Figure 2.4. When the MN moves to the coverage area of a second MAG, the process is repeated, but this time the LMA finds an existing entry in its Binding Cache for that MN, and therefore replies to the MAG with the same IPv6 prefix that the MN was using previously, updating the record for the MN and diverting its traffic to the new MAG tunnel. Thanks to the fact that the MAGs show the same layer-2 and IPv6 link local addresses to the MNs, these do not detect any layer-3 change while moving within the LMD.

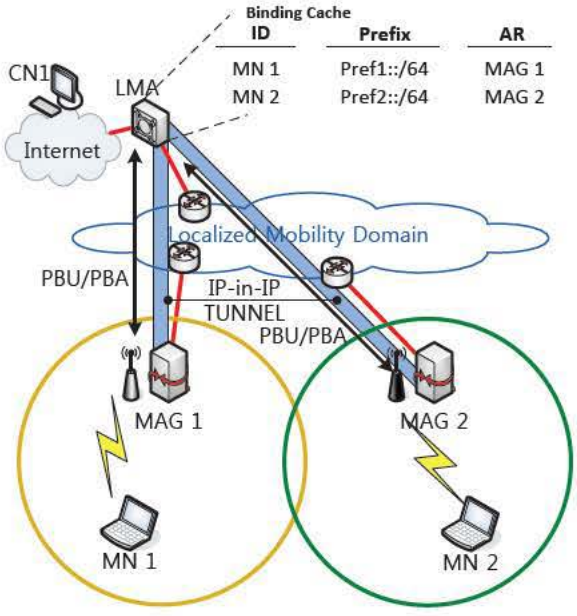


Figure 2.3: Proxy Mobile IPv6 (PMIPv6) scenario.

PMIPv6 outperforms MIPv6 in terms of handover latency in the comparative analysis performed in [25], where authors also suggest the convenience of a combination of the

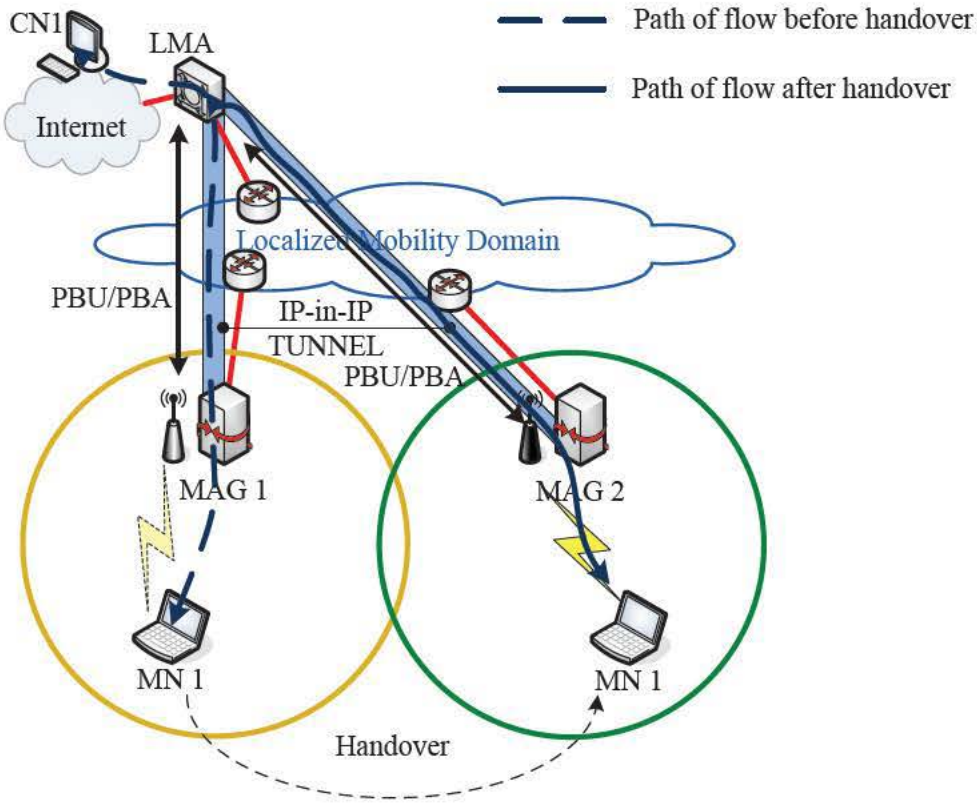


Figure 2.4: Handover scenario in PMIPv6.

mobility protocols rather than their isolated operation. A study of handover performance is done in [26] comparing MIPv6, PMIPv6, HMIPv6, FMIPv6 and FPMIPv6, which is a combination of PMIPv6 and link layer triggers to anticipate handover. The numerical analysis confirms that the use of link layer information in FMIPv6 and FPMIPv6 help considerably to reduce the handover latency, followed by PMIPv6 and the worst results are achieved by MIPv6. They also consider the link conditions and the impact of the Duplicate Address Detection (DAD) procedure in the duration of the interruption. We will provide an extensive analysis and comparison of these aspects in our study of mobility protocols and their combination, including an experimental evaluation in Chapter 3.

2.2.4 Network Mobility

[1] The Network Mobility Basic Support (NEMO B.S.) protocol [9] was proposed by the IETF to enable mobility of complete networks. This, for example, allows a set of devices deployed in a vehicle (e.g., a car, bus or train) to benefit from Internet connectivity. To

[1] http://www.ietf.org/

do so, the NEMO Basic Support protocol extends the basic end-host mobility solution, MIPv6, to provide mobility management to complete networks. In this solution, a mobile network (known also as *network that moves* – NEMO[2]) is defined as a network whose point of attachment to the Internet varies with time. A specialized device, called the Mobile Router (MR), connects the NEMO to the Internet and manages mobility on behalf of the mobile network. It is assumed that the NEMO has a home network, connected to the Internet, where it resides when it is not moving. Since the NEMO is part of the home network, the mobile network nodes (MNNs) use IP addresses that belong to one or more address blocks assigned to the home network: the mobile network prefixes (MNPs). These addresses remain assigned to the NEMO even when it is away from home.[3] When the NEMO is connected to a visited network, the MR acquires an address from the visited network, the care-of address, and sends a Binding Update (BU) message to the HA that adds the CoA of that Mobile Router to the Binding Cache (BC). The HA notifies the success of the operation with a Binding ACK (BA) message. Thus, when the NEMO is attached to another network, packets addressed to the mobile network nodes will still be routed to the home network, and redirected by the home agent to the current location of the MR in a bidirectional tunnel (see Figure 2.5).

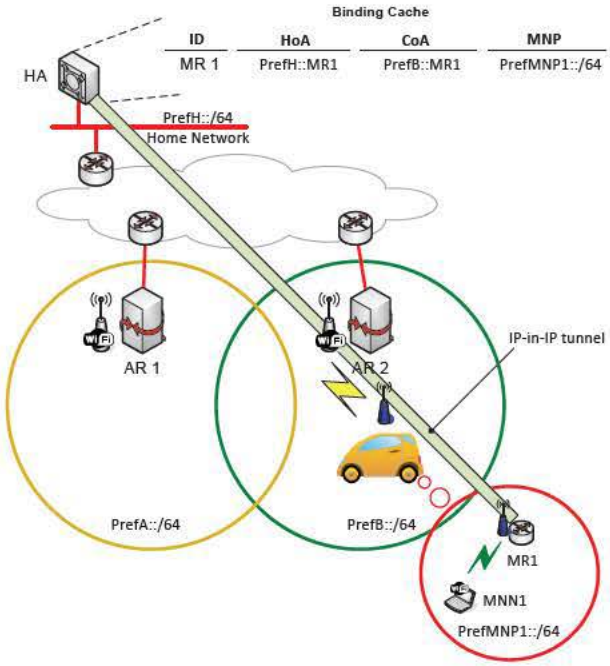


Figure 2.5: NEMO Basic Support protocol operation overview.

[2]NEMO can mean NEtwork MObility or NEtwork that MOves according to the context.
[3]These addresses only have topological meaning when the NEMO is at home.

2.2.5 NEMO-Enabled Localized Mobility support (N-PMIPv6)

N-PMIPv6 [27] fully integrates mobile networks with Proxy Mobile IPv6. N-PMIPv6 is not a standard, but we include this protocol in our analysis because it is an interesting extension to PMIPv6 specially designed for communications in public transportation systems. The basic idea is to extend a localized mobility domain to include mobile networks as well, so a user terminal is not only able to roam between fixed gateways (i.e., MAGs that do not move, as in conventional PMIPv6), but also between fixed and mobile gateways (called mMAGs, which are also able to roam within the domain), without changing the IPv6 addresses they are using (see Figure 2.6). A moving gateway (i.e., an mMAG) behaves as a mobile node from the viewpoint of fixed gateways, since moving gateways move between different fixed gateways while keeping the same IP address. Besides, a moving gateway behaves as a regular gateway from the perspective of mobile nodes, and extends the localized domain by providing attached terminals with IPv6 prefixes of the domain, and by forwarding their packets through the localized mobility anchor (i.e., the LMA). An additional bi-directional tunnel between the moving gateway and the localized mobility anchor is used to hide the network topology, and avoid changing the particular prefix assigned to a terminal while roaming within the same domain. The target scenarios are public transportation systems, in which fixed MAGs are deployed in stations and moving MAGs in vehicles (buses, trains, for example).

There have been later versions of these combination of protocols, as N-NEMO [28] and P-NEMO [29]. We focused in the one we have described above as this is the one that we consider in our analysis of the combination of mobility protocols in Chapter 3.

2.2.6 Distributed Mobility Management

Traditionally, IP mobility management relies on a centralized mobility anchor, such as the Home Agent in Mobile IPv6 (MIPv6) or the Local Mobility Anchor (LMA) in PMIPv6, being that central entity the one that manages all the bindings. However, such a centralized architecture may encounter scalability and performance issues as the number of mobile nodes and the volume of data traffic increases [30] that motivate the need for Distributed Mobility Management (DMM) approaches. Some of these problems are the low scalability due to needing new mobility anchors as the number of mobile nodes and data traffic increases, per node mobility support that increases congestion on mobility anchors due to mobility support for all flows, single point of failure due to many mobile nodes connection to single mobility anchors [31], and non-optimal routes which may result in longer delays and excessive loads in the core network. In order to define a distributed mobility management mechanism that can adapt the rigid previously existing solutions, IETF chartered the DMM working group in 2012 [32]. The major difference with respect to traditional IP mobility management is that DMM distributes the mobility

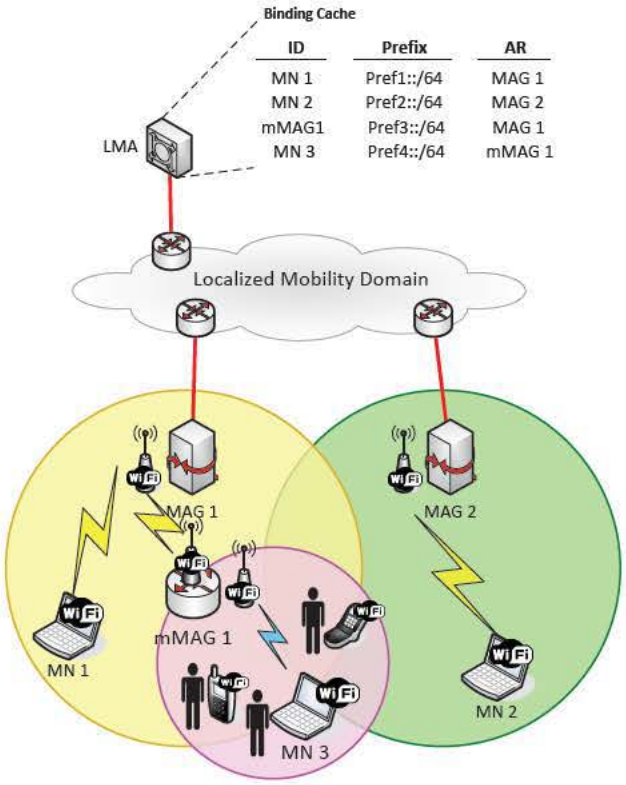


Figure 2.6: N-PMIPv6 protocol operation overview.

anchoring at the edge of the access network, effectively flattening the network by removing hierarchies. To do that, there are three different approaches being considered [33]: host-based, network-based or routing based approaches.

The host-based approach adapts MIPv6 by deploying multiple Home Agents in the access network. However, it requires modifying the mobile terminal to keep track of the different addresses and perform the signaling as it configures a new IPv6 address in every visited network. Note also that the tunnels as the MN moves are established between the former HA and the MN itself.

In our work we have mainly consider the network-based DMM approach. PMIPv6-based DMM defines a new entity, the Mobility Anchor and Access Router (MAAR), which absorbs the roles of the Mobile Access Gateway (MAG) and LMA in PMIPv6 to place the mobility management closer to the edge of the network. Figure 2.7 shows the handover process in this approach. When the mobile node changes its point of attachment to the network, the ongoing sessions keep anchored to the previous MAAR while the new sessions will be managed by the anchor in the target network. Data traffic is tunneled between both anchors and forwarded to the mobile node, which can deregister from the previous

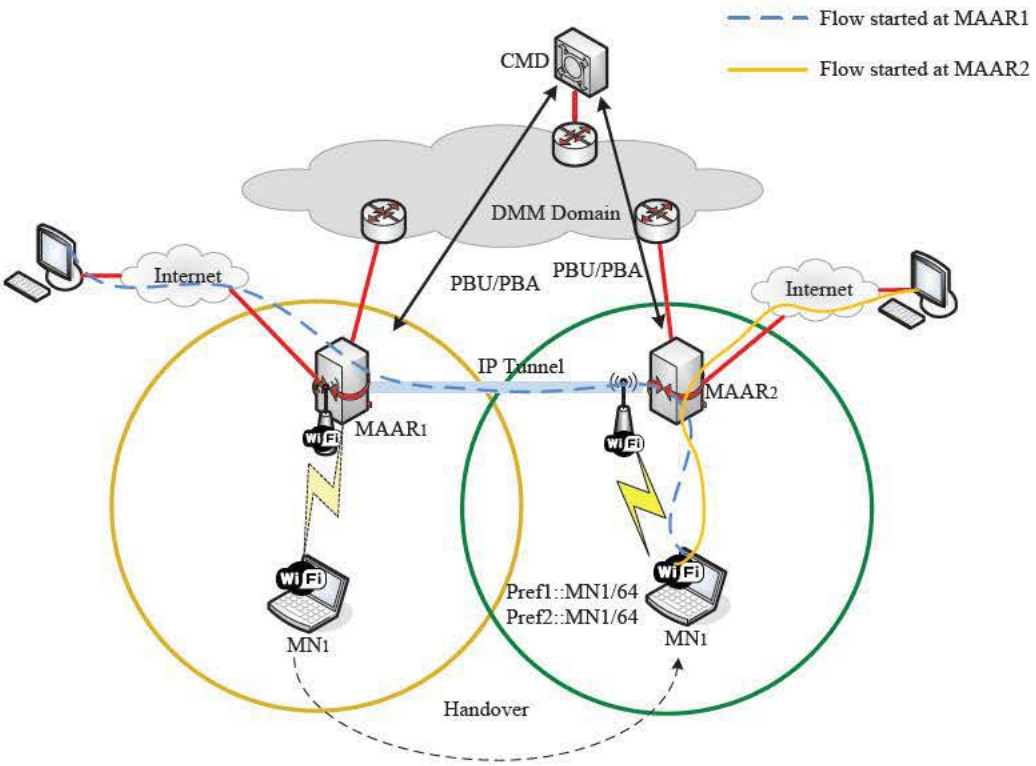


Figure 2.7: DMM operation.

anchor once the ongoing sessions are terminated. Complexity increases as the mobile node performs more handovers, but assuming that most of the sessions are relatively short, most of the data traffic is routed optimally without tunneling, so in the general case DMM results in a more efficient and scalable approach for mobility management. One of the key aspects of DMM is that the flows originated by the mobile terminal once it is connected to a new anchor are not tunneled, but only the flows performing handovers require the overhead of tunneling through the network. For networks with low mobility characteristics, this is a key difference, since most of flows will not require differentiated handling from standard IP routing, reducing overhead.

Routing-based DMM makes use of a routing protocol within the domain to support the mobility, instead of tunneling as PMIPv6 or MIPv6 do. The access router, as the MN moves within the domain, keep sending routing updates when it discovers the new address of the MN. However, routing-based solutions incur more signaling overhead which may cause scalability issues and its performance depends on the convergence of the routing protocol.

The flattened network architecture turns DMM a strong candidate for the efficient delivery of video traffic. PMIPv6-based, MIPv6-based and routing-based DMM approaches

are compared in [34] in terms of overhead, handover latency and packet delay to analyze their suitability for efficient video transmission. They conclude that PMIPv6-based approaches are more suitable for real-time interactive applications due to the lower handover delay. On the other hand, MIPv6-based approaches are suitable for applications with more flexible requirements about delay while routing-based approaches are indicated for low mobility scenarios, due to the high signaling overhead. From a practical implementation aspect, [35] demonstrates the first practical evaluation results of distributed mobility management on an implementation of a Linux-based prototype.

2.3 Vehicular Ad Hoc Networks

As a part of this thesis, we implement and run an experimental evaluation of a vehicular communication protocol. In order to do so, we have selected a particular solution we are familiar with, VARON, published in [12]. This section provides some background information on vehicular communications, for the reader to better understand our experimental work on VARON. This is required to follow the explanation on the problems faced during the prototyping phase, as well as the solutions we designed to tackle them, which will be presented in Chapter 4. As it has been a trending topic for some time, we also present the related work on implementations of real vehicular testbeds.

2.3.1 IP vehicular communications

The are two main types of communication in a vehicular scenario: between a vehicle and the infrastructure (V2I) or vehicle-to-vehicle (V2V). In the near future several devices within a vehicle, such as internal sensors, on-board computers or infotainment back-seat boards will likely benefit from having Internet connectivity, and we have to consider also external devices carried by passengers, such as laptops or smartphones. In this vehicle-to-Internet scenario, it is commonly assumed that a specialized node,the Mobile Router, provides external connectivity to these devices in the vehicle, which form a mobile network. A mobile router is in charge of providing connectivity to the intra-vehicle network, also managing transparently its mobility, that is, without any additional requirements to the attached devices. It is notable the applicability of NEMO B.S. protocol to this scenario. These mobile routers are also expected to have multiple access technologies available, so they can take advantage from this heterogeneity to forward the traffic through the most appropriate interface (e.g., 3G/LTE, WLAN).

Besides Internet access, there are several applications which involve V2V communications, such as semi-automatic driving or gaming in platooned vehicles. In this case, the role of ad-hoc networks is even more clear, as they naturally enable V2V communications without involving any external infrastructure.

It is commonly assumed that the mobile router deployed in each vehicle has at least three network interfaces: one *ingress* interface to communicate with the nodes inside the vehicle that belong to the mobile network (e.g., WLAN, Ethernet), one or more *egress* interfaces to connect to the Internet (e.g., 3G/LTE), and an additional ad-hoc interface (e.g., WLAN) to communicate with neighboring cars and to set up multi-hop ad-hoc networks. Another common assumption is that vehicles can always communicate with other vehicles through the Internet. In addition they may communicate directly if a multi-hop route can be set up in the VANET. In our prototype, presented in Chapter 4, the access through the Internet is provided by the NEMO Basic Support protocol and the multi-hop ad-hoc route in the VANET is set up by VARON, the route optimization protocol under evaluation.

2.3.2 Vehicular Ad Hoc Networks: VANETs

Vehicular networking is a scenario especially demanding in terms of mobility due to their dynamic environment, both in terms of network topology and link conditions. Vehicular Ad-hoc Networks are a particular kind of mobile ad-hoc network, characterized by high mobility of nodes, short-lived links and unstructured nature. VANETs are very dynamic and lack a pre-established topology, operating in a fully distributed way without control nor monitoring from a centralized entity. This makes more complex to define solutions for vehicular protocols, as there are many different variables defining the nature of VANETs. They can be very dense, as in an urban area, or very sparse, as in a remote highway, and these conditions can vary in a few minutes. Moreover, the protocols designed for VANETs have to provide communications with reliability and deal with different sources of interference.

These particular characteristics of VANETs have fostered the publication of routing protocols tailored for them. These protocols adapt the mechanisms in ad-hoc networking to the vehicular environment and the mobility patterns of vehicles. In this section, we review the main mechanisms commonly present in VANET protocols and match them to their equivalent in VARON, the protocol that we evaluate. Some of these mechanisms were not included in the original definition of VARON, but added as found necessary during our experimental process, as described in Section 4.3.2. The main procedures commonly found in VANET solutions are identified next:

- **Self-organization and discovery**. Vehicular communications occur in a highly dynamic and variable environment with no entity in charge of the network management. To deal with the unstructured nature of VANETs it is essential to include a mechanism for node discovery, so that vehicles are aware of the presence of neighbors. In addition, vehicles have to be able to distribute information among a group. The scope of this group varies depending on the protocol and the scenario: it can

be a message to be distributed only to nodes playing a special role in the network, such as cluster heads, or to every single node in a given area in the case of geo-routing protocols. Most of the solutions designed for VANETs include a flooding or broadcast mechanism by which nodes can announce their presence, position, speed or direction among other information. These messages are sent periodically in order to effectively deal with the very dynamic nature of these networks. Depending on the density of the network, the distribution of these messages can originate a so-called broadcast storm, if the transmission is very frequent. On the other hand, if the transmission interval is too long, the discovery may be less effective, establishing a trade-off between signaling overload and effectiveness of the mechanism.

- **Reliable signaling**. After performing the node discovery, multi-hop ad-hoc protocols may use signaling messages to set up paths between selected nodes. This signaling may have different scopes, may be multicast or unicast and can be sent at different rates. Reliable delivery is critical as signaling often involves creating or updating the protocol state. VANET solutions might make use of confirmations or acknowledgments to make an involved network node aware of potential failures, and avoid in this way state inconsistency in the network.

- **Use of cellular communication**. A cellular communication channel is sometimes present in the VANET scenario as a reliable always-on connection, providing backup to the unstable ad-hoc network. Many solutions transfer critical or time-constrained information via the cellular access network. Interestingly, this cellular connection has been one of the main causes of failure in our experiments, having considerable packet losses and very variable delay. Therefore, we claim that this common assumption on the reliability of the cellular connection cannot be taken for granted in real world scenarios.

- **Mechanisms to deal with link quality variability**. The existence of a symmetric communications channel is also often assumed when designing a VANET protocol. However, VANETs conditions are very dynamic, and the reception of a message from a node cannot translate into a subsequent successfully transmission on the way back. Moreover, the use of multicast and unicast signaling messages, which may be sent at different rates in the WLAN, impose different transmission distance ranges. In addition, in the real world, the communication range and the link quality are affected by many different and variable factors, such as vehicles' speed, taller vehicles passing by or just the weather conditions. Link conditions impact strongly the lifetime of a multi-hop route and therefore, it becomes essential to monitor link status in order to maintain the communications quality for as much time as possible.

2.4 VARON: a VANET solution for NEMO route optimization

Vehicular Ad-hoc Route Optimization for NEMO (VARON) is a solution originally designed in [12]. As part of the work developed in this thesis we have introduced some modifications to the original design, based on the key knowledge acquired by implementing a real prototype, as it will be further described in Section 4.3.1.

VARON enables the optimization of vehicle-to-vehicle communications in a secure way by combining a network mobility approach – that supports vehicle-to-Internet communications via a cellular interface – and a vehicular ad-hoc approach – used when a multi-hop network becomes available (i.e., communication takes place between vehicles that are close enough to communicate in a VANET formed by the mobile routers deployed within those vehicles, and perhaps within other vehicles in their surroundings). We do not consider intra-vehicle communications, as we manage the mobile network as a whole, represented by the entity of the mobile router. VARON route optimization process takes place as follows:

1. Self-organization and discovery of reachable networks. Each mobile router needs to find out which other MRs are available within the VANET, that is, which mobile network prefixes (MNPs) are reachable through its ad-hoc interface. To that purpose, every MR periodically broadcasts a message called Home Address Advertisement (HoAA), which contains its home address and an associated lifetime. These messages are announced through the ad-hoc interface using a hop-limited flooding, so every MR becomes aware of the MNPs that can be reached through the VANET. This mechanism makes a node visible to other nodes potentially interested in establishing a direct communication through the VANET. In our case, in order to avoid missing optimization opportunities and not to flood the network with signaling, these messages are sent at a configurable interval, which was set to 10 seconds in our experiments[4].

2. Reliable signaling for the creation and validation of a secure ad-hoc route. The ad-hoc routing protocol should at least provide the same security level than today's Internet communication. The mechanism used by VARON to set up and maintain a secure ad-hoc route is based on [36], modified and extended to fulfill the requirements of a network mobility-based vehicular scenario:

    - First, once a mobile router has identified an optimization opportunity, this MR (called originator MR) has to trigger the ad-hoc route setup by sending to its one-hop neighbors a Care-of Route Test Init (CoRTI) message. This

[4]Note that in our experiments we tried several configurations, achieving best performance for the 10s interval.

message is re-broadcast by intermediate routers using a limited flooding until the message reaches its final destination, which is the mobile router handling the target prefix (called target MR).

- Second, the target MR generates a reply message called Care-of Route Test (CoRT) message, and unicasts it back to the MR that triggered the procedure. Note that the route used to deliver the CoRT message is learned by the intermediate mobile routers during the limited flooding of the CoRTI message, and that the reverse route is learned while delivering the CoRT message.
- Third, this new bi-directional ad-hoc route (called care-of route) cannot be used yet to forward packets between the mobile networks managed by the originator and target MRs, as authorization from the MRs handling those prefixes needs to be verified. This requires exchanging two additional messages, called Home Route Test (HoRT), using the default NEMO route through the 3G interface (called home route), and two final messages, called Mobile Network Prefix Binding Update (MNPBU), through the VANET.

VARON signaling is secured using cryptographic mechanisms, as extensively explained in [12]. Figure 2.8 shows a simplified example of the signaling in the route optimization process.

In VARON, these signaling messages define the state machine of the route optimization process, which needs to be properly completed. The nodes involved have to monitor the current state at every moment, in order to ensure consistency at both ends of the communication. Note that some of the signaling messages are transmitted through the cellular communication channel, which proved not to be very reliable and sometimes introduced a considerable delay. A simple but effective way to perform this monitoring is the use of timeouts to control that these messages are received within a certain interval. In order to avoid wasting time trying to optimize a route when the link conditions are not favorable, VARON checks the quality of the link during the initial signaling stage and abort the optimization in case the quality is not good enough. In this way, we deal with the link quality variability that may cause a failure in the optimization process or the creation of routes that can be used for a very short period of time (which makes them practically useless).

If an ad-hoc route becomes invalid because it expires or it is broken, and traffic is received through this route, a Care-of Route Error (CoRE) message is sent (and forwarded) by each MR in the path to the originator MR. For example, let us consider the scenario in Figure 2.8a where MR A and MR B have an ongoing communication using a multi-hop route in the VANET (MR X and MR Y are intermediate hops). If MR Y is forwarding data from MR A to MR B, and detects that the link to the next hop in the path (MR B itself) is broken, then MR Y sends a CoRE message towards the source (MR A). This

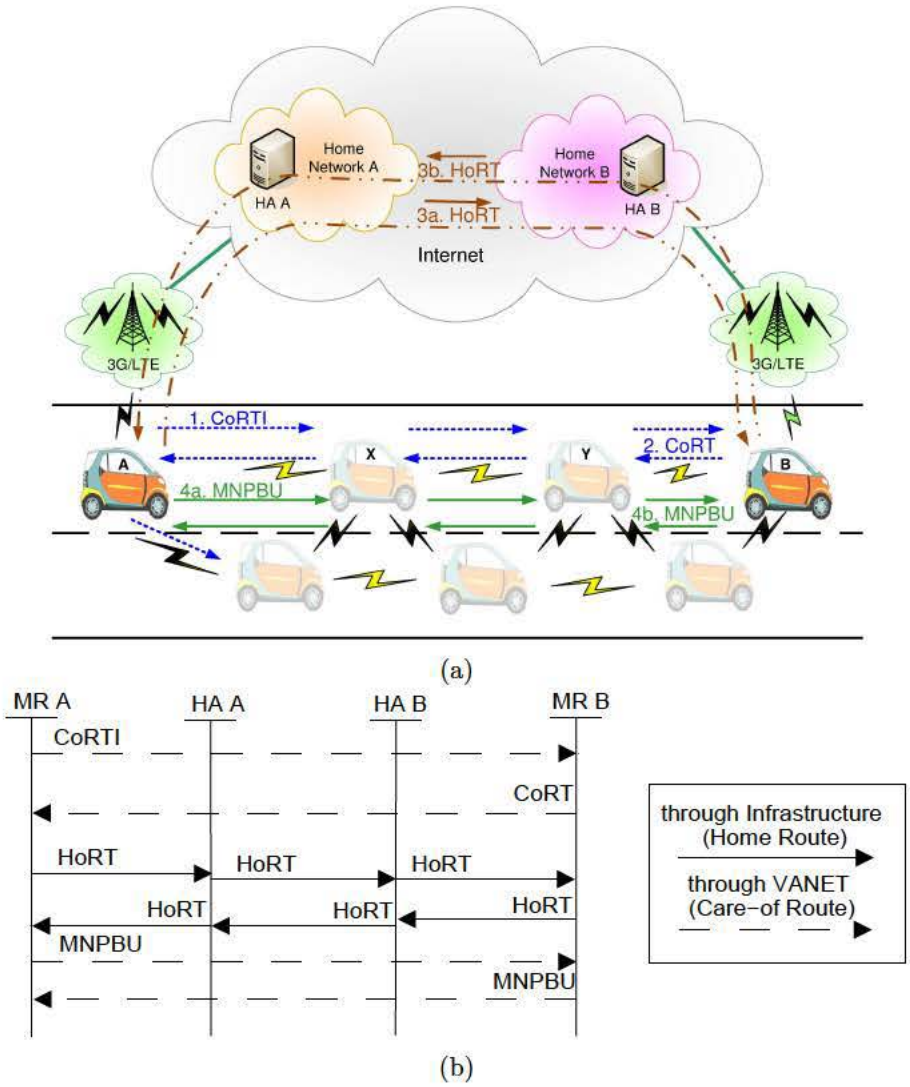


Figure 2.8: VARON signaling.

message is received by MR X and forwarded to MR A, who notifies MR B the withdrawal of the care-of route with a CoRE message sent through the Internet connection. Then, MR A and MR B switch back to the home route, though they may start a new route discovery procedure to set up an optimized care-of route within the VANET.

Due to the link quality variability commonly found in VANETs and the need to deal with several sources of interference, it is very important to continuously monitor the link quality. As soon as a quality degradation is detected, the optimized route is withdrawn and all the communications take place via the cellular interface again. A significant part of our experiments focused on the configuration of the most adequate link quality thresholds, both to establish a multi-hop route in the VANET and to withdraw this route, falling back to the default Internet connection. The design of these thresholds is extremely important in order to avoid short-lived routes, which additionally incur useless signaling overhead

in the network, that might also disturb other users' communications.

Some of the aforementioned mechanisms are included in our implementation as a result of the experimental learning process, as we will explain in more detail in Section 4.3.2.

2.4.1 Experimental deployments for vehicular networking

Research on vehicular communications has extensively addressed many different aspects, from routing protocol design to location privacy or peer-to-peer file sharing. So far, only a minority of the existing research includes experimental results, due to the considerable challenges posed by real-life experimentation.

In order to minimize the costs required to deploy a stable experimental platform, some renowned institutions have opted for deploying their own vehicular testbed. This is the case of the Campus Vehicular testbed at UCLA or *C-Vet* [37], a comprehensive testbed deployment open to external researchers too, the *VanLan* testbed at Microsoft campus in Redmond [38] formed of eleven APs distributed all over the campus and two vans equipped with mobile nodes, or *DieselNet* and *UMass DOME* testbed at Amherst[5] [39]. The deployment of a stable vehicular testbed makes possible to run experiments frequently, to evaluate and compare networking protocols as well as to design models for mobility and traffic patterns validation, among many other applications. However, these platforms focus on single-hop V2I communication scenarios, not looking into multi-hop V2V/V2I communications, such as the ones considered by VARON.

A more moderate and frequent approach present in the literature is the use of a wireless network infrastructure already deployed, and roam around in order to assess your own vehicular networking protocol or put network performance to the test. In this way, researchers may extract significant insights from real-world experiments at a lower implementation cost. For example, Deshpande et al. [40], benefit from a metro-scale WiFi deployment provided by an ISP to download files during test drivings of different length; and Giannoulis et al. [41] focus on the evaluation of channel quality and design an AP quality scoring mechanism to evaluate WiFi performance in the vehicular environment, thanks to the urban wireless mesh network deployed by the TFA (Technology For All) in Houston in cooperation with Rice university[6].

Following this approach, most of the existing works in the literature are centered on delivering data to or from moving vehicles by means of WiFi APs opportunistically accessed along their way. *Cabernet* [42] has been deployed in 10 taxis in Boston and presents a transport protocol to avoid the shortcomings of TCP when dealing with 802.11 networks and a scanning mechanism to reduce delay in the wireless association process. Note that in VARON, ongoing communications are not delayed by the latency in the wireless association, as data is being transferred by the cellular connection. Note also

[5] http://prisms.cs.umass.edu/dome/
[6] http://tfa.rice.edu/

that authors in [42] implement timeout optimizations to avoid losses. Similarly, VARON switches to using the WLAN route only if the network is reliable. In relation to that, several works study the most appropriate handoff technique and try to predict WiFi connectivity to avoid losses in the data transfers from or to a vehicle, but very few tackle the issue of vehicle to vehicle communications. Authors in [43] measure packet delivery ratio and packet inter-arrival time between vehicles that travel together in a 960-km long test drive. However, their measurements base on the transmission of beacons and their successful reception, whereas we look for a more general solution. *ViFi* [44], tested in VanLan and DieselNet, modifies the wireless driver to evaluate different handoff strategies in an ad-hoc network and implements a relaying mechanism with a main AP (namely the anchor) and several auxiliary APs to forward data to its final destination. However, this mechanism can only operate in a deployment where all the access points are working on the same channel and requires vehicles to send beacons at regular intervals. In addition, they evaluate different handover strategies, based on RSSI, beacon reception ratio and performance history. These last two mechanisms cannot be adopted by VARON, since in a wireless ad-hoc network not all the nodes are sending beacons nor is feasible to maintain a history record of performance. Based on that, we concluded that RSSI monitoring was the only possible choice.

Authors in [45] also play with the 3G-WiFi interaction and present *Wiffler*, which postpones transmission of delay-tolerant data to the availability of a stable WiFi connection and switches back to 3G if the packet cannot be transmitted fast. This approach uses WiFi as an auxiliary tool for improving 3G transmission and avoids switching to 802.11 unless it can provide quality communications. However, the offload of the cellular connection translates into delaying transmission of non-critical applications, which may not satisfy the user. VARON switches from 3G connection to a wireless multi-hop ad-hoc route between neighboring vehicles, taking advantage of the locality of communication end-points. Moreover, a potential enhancement considered for future work is to offload some flows, instead of all the traffic, from one route to the other, taking into account traffic characteristics and mobility patterns. In a similar way, authors in [46] present a network stack, *CafNet*, which lets the application decide which data to send when a WiFi connection becomes available, instead of transmitting "outdated" information buffered by the link layer when there is a connectivity change. Contrarily, VARON does not defer or discard any transmission, but it modifies the routing when the conditions in a wireless multi-hop path are favorable.

In summary, there are recurrent issues in vehicular experimental works: *i)* the connection establishment latency and the association time in 802.11 networks, *ii)* the choice of a proper handover strategy, and *iii)* the data transfer methodology, as the network dynamics involve frequent disconnections and lossy links. In order to enhance communications and avoid these problems, most proposals design predictive methods, based on caching or

keeping history of the signal strength, link quality or performance associated to an access point in a certain location and rely on familiarity of routes and paths [38], [44], [45], [47]. VARON aims at optimizing inter-vehicle communications. Therefore, keeping track of the signal quality at a certain geographical point is meaningless, since the vehicles may not roam around that point in the future and even in that case, there is no guarantee that the wireless link between them would keep the same conditions.

2.5 Multi-interface Connectivity Management

Connectivity management in multi-interface devices mainly follow two models, widely known as *weak host* and *strong host* models. The weak host model will accept any packet destined to one of its IP addresses, regardless of the interface where the packet is received. On the contrary, the strong host model will only accept the packet if the destination address matches that one of the interface which received it. Different operating systems decide to implement one or the other. For instance, Linux implements the weak host model, whereas Windows Vista and Windows 7 default to the strong host, although weak host model behavior is configurable. Such implementation decisions affect the performance of the devices, especially when different access technologies are available.

As stated in [48], a node with multiple interfaces faces several challenges that single-interfaced nodes do not experience, such as source and destination address selection or routing when several interfaces are active simultaneously. If the simultaneous usage of different interfaces is supported, an important design choice is the behavior when an application is using one interface and a better connection becomes available at another one. Open sessions can be transferred to the new network connection or remain in the current interface. The connection manager has to deal with these decisions, but as the behavior and architecture of the connection manager is not standardized, its performance depends on the implementation of the specific OS or platform. There are three main approaches for the design of the connection manager [49]: *i)* a centralized approach that bases the interface selection on info provided or programmed into the application or user input *ii)* configuring settings per application or *iii)* taking decisions according to system configuration (routing tables, manual configuration, default interface).

As we will see in our analysis of the connection manager in smartphones in Chapter 9, these devices use Stateless Address Autoconfiguration (SLAAC) to configure their IPv6 address, usually from their hardware addresses. However, these addresses compromise privacy of the mobile user, so they generate also a random temporary address [50] to be used in outgoing communications. Nevertheless, the use of temporary addresses do not replace SLAAC addresses. Therefore, a mechanism has been proposed [51] to generate stable interface identifiers to configure SLAAC addresses (or other configuration mechanisms as DHCPv6) that change as the IP prefix changes, without compromising the user

privacy and being stable for each network interface in a subnet.

2.6 Connection management in smartphones

Table 2.1: Comparison with related work on smartphone networking and Connection Manager

Ref.	System	Scope	Main features studied	Contributions	Main conclusions
[52]	Android Windows Mobile	Logging app	User interactions, application use, network traffic, energy drain	Evaluate the diversity range across users and time and their impact on network and energy	Patterns on app usage and user interaction time; apply diversity in usage to predict energy drain
[53]	Android	Logging app	Application popularity and usage patterns	Application usage models, app-launching optimization, personalized optimization framework for task manager	User experience can be improved by knowing usage patterns and context-aware resource management
[54]	iOS Android	Speedtest app	Cellular and WiFi network performance	Temporal and geographical analysis, aggregate performance	Similar throughput performance but, iOS higher latency
[55]	Android	Logging app	Cell, WiFi usage, phone usage	Analysis on application, phone, WiFi and cellular traffic usage	On average, WiFi traffic is 30% of the total data consumption
[56]	Android	Logging app	Traffic volume in 3G and WiFi networks	Aggregated and per-user analysis	Total traffic via WiFi much larger than via 3G, most by a small number of users
[57]	Smartphones Laptops	Wireless traffic captured by a gateway router in campus network	Network traffic, TCP impact, comparison to laptops, app layer parameters	Study network performance and traffic, mainly focusing on TCP-related parameters	Akamai and Google servers are heavily used. Different receive window advertised by iOS and Android, but similar performance
[58]	iOS Android Windows Mobile	Customized app: *3GTest*	TCP performance, RTT, DNS lookup time as metrics for app and network performance	Measurement tool and methodology for app performance comparison, 3G network performance for various operators	Smartphones are often the web browsing performance bottleneck, rather than the network
[59] [60]	Windows PC	Customized connectivity management	Vertical handover, session continuity and handover decision triggers	Connection manager to detect changes between WLAN and WWAN, virtual connectivity manager	Reduce number of handovers by keeping active connection instead of switching to WWAN when RSSI is below a threshold
[61]	Android	Application for connectivity management	WiFi offloading, simultaneous usage of cellular and WiFi interfaces	App for WiFi offloading and content aggregation (*Enhanced Android Connection Manager*)	Different use cases for WiFi offloading: content aggregation, SIM authentication or flow optimization and segregation
[62]	Android iOS	Server logs, sniffed wireless traffic and media player source code	Video streaming performance comparison.	Thorough comparison of iOS and Android clients behavior on streaming	Media players follow different content request and buffer management approaches. Redundant traffic downloaded by iOS

A significant part of the previous works in the literature analyze the energy consumption of smartphones, however, we focus on the network stack and how a multi-interfaced smartphone manages its network connections. We have compiled in Table 2.1 the previous works that address network performance in smartphones for a better comparison. Network connectivity has been addressed, but mostly in terms of application usage and traffic patterns. For instance, [52] conducted a thorough study of application popularity and usage, characterizing the patterns followed by different demographic groups of users and the traffic generated. Their study confirms the high diversity in smartphone usage, leading to the conclusion that the tools in use may provide acceptable performance in average, but it could be considerably enhanced by some specific knowledge on applications performance and usage. Similarly, [53] also uses a logging application installed in the smartphone of a group of users and presents a personalized optimization for Android smartphones, based on application usage patterns per user, showing that the default task manager can be enhanced to improve user experience. We argue that a similar approach to these two can also be extended to network connectivity management.

The use of mobile devices equipped with several network interfaces motivates the performance study in [54], which characterizes consistency and compare the WiFi and the cellular accesses worldwide in terms of download/upload speeds and latency. The

first promising application of this diversity in network connectivity is WiFi offloading. However, [55] shows that in spite of the dense WiFi deployment, cellular data consumption is still dominating and analyses the reasons behind that fact. A similar study is conducted in [56], but in this case, the authors conclude that the percentage of offloaded traffic is not negligible, being mostly exchanged at home APs. The differences between these two studies may lie on the geographical differences in their datasets. Yet again, it is proven that a unique solution for connectivity management cannot perform optimally, advocating for a kind of customizable solution per user or based on usage or mobility patterns. Chen et al. in [57] evaluate network performance of handheld devices by monitoring the traffic captured at a university campus. They also confirm the predominant presence of TCP and HTTP flows in the traces analyzed and focus their analysis on parameters such as slow start, the advertised receive window and characteristics related to the TCP flows. However, our analysis is centered on the network connectivity management and the performance in case of a handover. While the study in [57] is restricted to a WiFi connection, Huang et al. [58] evaluate network and application performance over 2G and 3G cellular accesses. They measure UDP and TCP throughput before examining the performance of two widely used applications, as web browsing and video streaming, by comparing them with different combinations of smartphone and network operator.

Devices equipped with multiple interfaces typically rely on the entity of the connection manager mostly for the interface selection, and then provide networking information to the applications. In the literature we can find alternative designs for the connection manager. Zhang et al. [59] study mobility management between WLAN and WWAN and propose an architecture that relies on a connection manager and a virtual connectivity manager, which integrates end-to-end information to be used to optimize the handover. Their connection manager includes RSSI monitoring and network availability detection modules. This architecture is experimentally tested in [60], achieving promising results such as 2.1 seconds interruption from WLAN to cellular and seamless handover from cellular to WLAN, being able to select the best AP to associate with (in terms of available bandwidth).

To the best of our knowledge, only [61] tackles the shortcomings of the current Android Connection Manager by developing an application that enhances and extends its functionality. Their application plays with the possibilities offered by the usage of multiple interfaces and – by enabling simultaneous usage of cellular and WLAN interfaces – adds support for WiFi offloading, flow segregation and content aggregation. However, they do not provide an assessment on the Connection Manager itself neither analyze the overall behavior under different network scenarios and various conditions. Besides, [62] compares iOS and Android behavior in streaming, finding out that Android and iOS media players request data from the server differently and they also have different buffer management policies. As of the time of writing we are the first ones to provide a thorough analysis

of the network management supported by an experimental evaluation of the inter- and intra-technology handover under a wide variety of configurations. In addition, we examine the network attachment executed by each device and we analyze the behavior of several applications in case of a handover, and how they can handle the change in the global connectivity of the terminal – most of the times unsuccessfully. The performance of an application does not only depend on the ability of the developer to handle network connectivity, but also the accessibility and flexibility offered by the operating system and the exposed API. Above all, we compare the three most popular families of operating systems worldwide[7] and state the differences and similarities among the connection manager of an Android, iPhone and Windows Phone 8 devices, establishing the guidelines for further improvements in this unexplored feature.

2.7 Inter-technology handover: IEEE 802.21

The IEEE 802.21 Media Independent Handover (MIH) Services [13], [63] standard defines a common interface to allow the optimization of handovers between heterogeneous IEEE 802 systems, as well as between IEEE 802 and cellular systems. This is achieved by adding a technology-independent function – the Media Independent Handover Function (MIHF) – that intermediates between upper and lower layers and improves the communication between different entities, either locally (in the mobile node) or remotely (as network functions). Each MIHF has a set of users, mainly mobility protocols, which use MIHF to get information related to handover and manage the handover process. Upper and lower layers communicate with the MIHF through a set of primitives grouped by 802.21 in several service access points (SAP): *(i)* the MIH_SAP provides the interface for the higher-layer users with the MIHF; *(ii)* the MIH_LINK_SAP is the interface between the MIHF and lower layers; and, *(iii)* the MIH_NET_SAP is the interface that communicates remote MIHF entities. Figure 2.9 represents the MIH architecture. As we can see, the MIH standard defines different roles for the network entities according to the relationship between the network-based MIHFs and the MN: these are the Point of Service (PoS) and Point of Attachment (PoA). The PoS identifies a network-based MIHF that talks directly with an MN, while the PoA corresponds to the network-side end-point of a layer-2 link with the MN. Note that a MN can have several PoSs, as it can exchange messages with more than one network entity.

In addition, 802.21 defines three main mobility services:

- The Media Independent Event Service (MIES) provides event classification and reporting of dynamic changes in link characteristics, status and quality. When an event is generated, it is delivered to all entities that subscribed to that event. Events are useful to detect new links or determine when a handover is possible.

[7] `http://www.businessinsider.com/android-is-utterly-dominant-in-europe-2013-7`

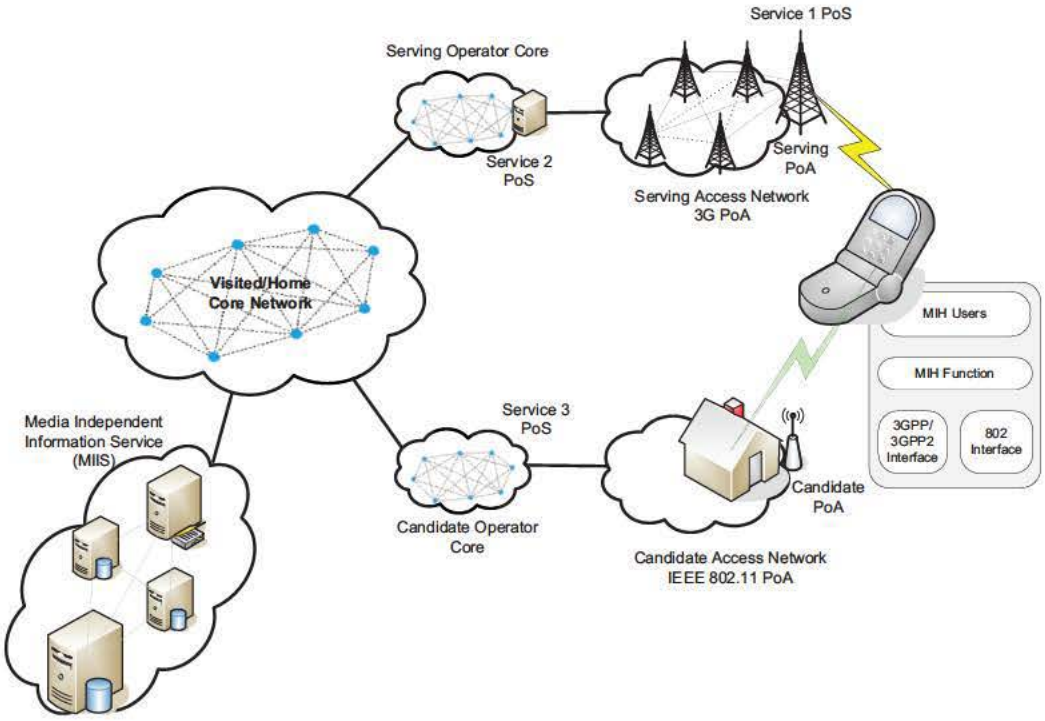


Figure 2.9: MIH architecture.

- The Media Independent Command Service (MICS) enables MIH clients to manage and control the link behavior related to handover and mobility. MIH users can send commands to lower layers, locally or remotely, through the MIHF. These commands can be used for instance to configure the lower layers or to initiate a handover.

- The Media Independent Information Service (MIIS) provides details about the characteristics and services provided by the serving and surrounding networks. This service allows the MIHF in a node to get information about heterogeneous networks in the area that facilitates handover.

2.8 Software Defined Networking

SDN is a paradigm shift in network management and configuration, as it makes directly programmable network control, decoupling it from the data forwarding function. As a result, SDN eases control tasks for network administrators in a flexible, dynamic and adaptable architecture with high programmability. SDN has raised in popularity, it has been backed by vendors and attracted the attention of the research community, as it eases testing protocols and networking algorithms.

OpenFlow [14] is an open standard created at Standford University and now being developed by the Open Networking Foundation (ONF)[8] to interact between the control and data forwarding layers in an SDN. The OpenFlow switch specification is currently at its version 1.5.1 [1], and functionalities have been progressively added from the first version. An OpenFlow-enabled switch delegates the control plane to an external software program, called a controller, and the controller and the switch communicate via OpenFlow protocol. Figure 2.10 shows the structure of an OpenFlow-enabled switch, which is mainly

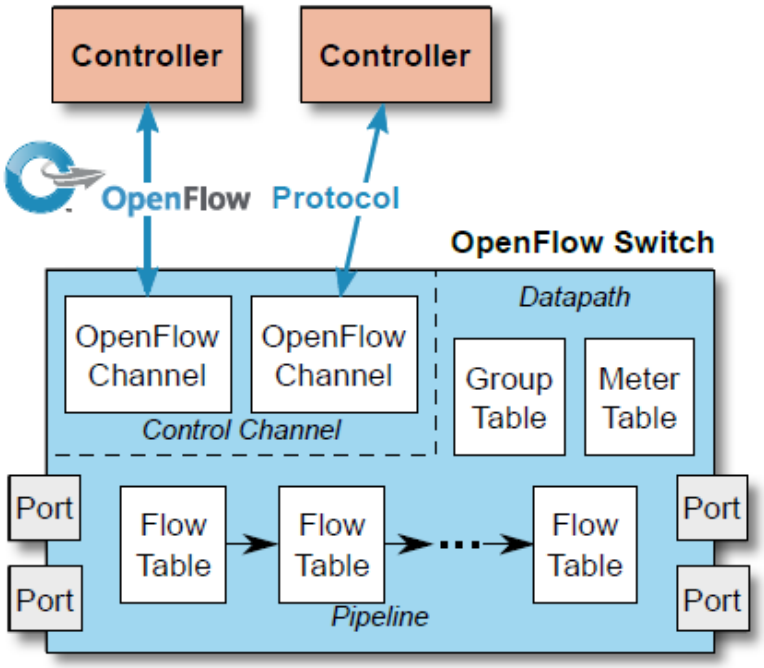


Figure 2.10: OpenFlow-enabled switch [1].

composed of a set of flow tables, a group table and one or more OpenFlow channels to a controller. The flow tables contain a set of entries with the instructions to follow in case a packet matches their respective conditions. If the switch receives a packet that matches an entry in a flow table, it applies to the packet the corresponding forwarding instructions. When the received packet has no matching entries, the switch forwards it to the controller, which decides how to handle the packet. This decision can be to drop the packet or to add or modify a flow entry in the switch to forward that kind of packets. A packet can be matched against different packet header fields or the ingress port, for instance. The flexibility provided lets a network node behave as a router, a switch, a firewall or implement other functionality depending on the rules configured by the controller.

The SDN architecture is structured in three main layers, as illustrated in Figure 2.11.

[8] https://www.opennetworking.org/index.php

The controller manages the central layer and expose a Northbound interface to applications and services in the upper layer; and *ii)* a Southbound interface to the network devices. OpenFlow protocol lies in the Southbound interface, as it communicates the controller and the network devices. Applications use the controller to obtain information about the network, perform network analytics or orchestrate new rules.

OpenFlow is the most popular southbound protocol, but it is not the only way that a controller has to communicate with the network devices. Open Virtual Switch DataBase (OVSDB) [64] is also well-know as it was developed as part of Open vSwitch project, an open source virtual switch to run in Linux-based environments. OpenFlow and OVSDB can be used jointly by the controller of a virtual switch (the former to set up the flow tables and the latter to configure the Open Virtual Switch (OVS) itself) and vendors are starting to support OVSDB in their switch platforms. OpenFlow has become the protocol of reference and it has been integrated into a number of SDN frameworks with wider scope and objectives, including Network Function Virtualization (NFV) (e.g., OpenNaaS[9], Project Floodlight[10], OpenDaylight[11]). The flexibility of OpenFlow has led to the implementation of numerous controllers—e.g., NOX, POX, Ryu, FloodLight—to interact with OpenFlow switches, developed in a wide range of programming languages and supported in many platforms. Some of the controllers are kept open, others are proprietary solutions as many vendors (e.g. Cisco, HP, Juniper, VMWare) have interest in developing SDN solutions and integrating them into their commercial products.

The heterogeneity in wireless access networks and the wide range of offered services force operators to carry significant volumes of traffic with different properties. SDN offers network operators flexibility to handle that traffic from different technologies and eases configuration, even controlling different parts of the network with different policies or applying rules temporarily. As examples of the use of SDN in wireless networks, the centralized control can be applied to avoiding inter-cell interference and the abstraction offered to coordinate offloading among different wireless technologies [2].

2.8.1 Software Defined Networking for mobility management

In Chapter 6 we provide a solution for SDN-based mobility management. In the following we overview related works in the literature comparing them to our solution.

The flexibility and programmability of SDN architectures have contributed to the proliferation of several large deployments designed by leading research institutions. Among the first deployments we find B4 [65], which is Google's SDN-based wide area network (WAN) to interconnect its data centers around the world. For B4, an extensive analysis is available on how to manage the routing and traffic engineering through OpenFlow and

[9] `http://www.opennaas.org/`
[10] `http://www.projectfloodlight.org/`
[11] `http://www.opendaylight.org/`

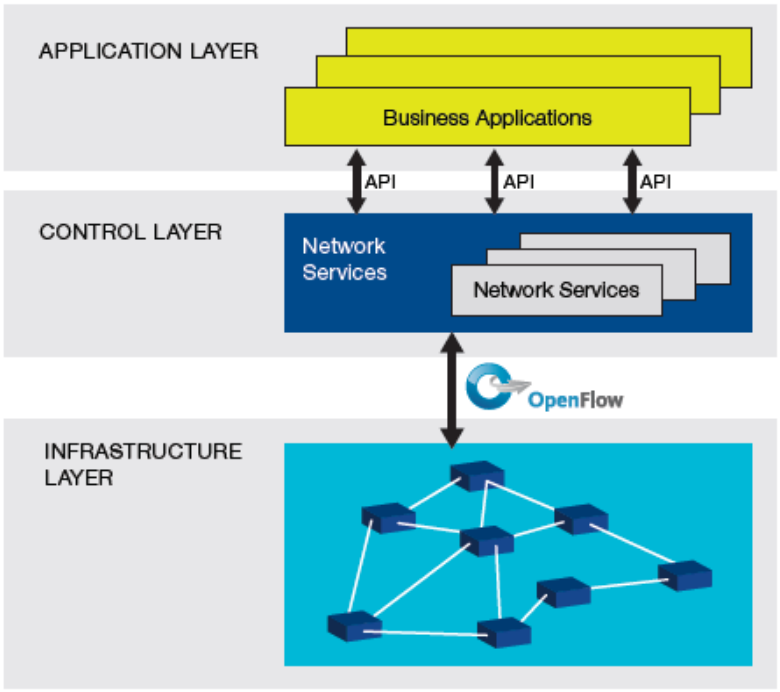


Figure 2.11: SDN architecture [2].

the designers of B4 provide interesting insights on design, performance, scalability and failure resilience of their solution. Despite being a large scale implementation, providing mobility is not within the objectives of B4.

With regard to wireless access and mobility management, one of the most remarkable SDN deployments applied to wireless networking is *OpenRoads* [66] (also known as *OpenFlow Wireless*) developed at Stanford University, open to researchers for running their algorithms concurrently by means of virtualization. OpenRoads incorporates different wireless technologies, namely WiFi and WiMAX, and one of its early proofs of concept was based on providing mobility across multiple technologies [67]. In addition, the performance of OpenRoads has been demonstrated by means of an n-casting transmission solution [68]. All these approaches are based on the same principle as our layer-2 mobility approach, reconfiguring the data-path, although they do not consider IP mobility or the design of a scalable architecture as we do. All the tools used by OpenRoads are open source, so as to make the infrastructure reproducible by other research groups in their own networks. Likewise, our implementation shows the flexibility of current SDN software tools available as open source and is built upon commercial-off-the-shelf devices. At the moment we only focus on IEEE 802.11 access points, but we are planning to extend our testbed to include heterogeneous access technologies in the short term.

A full software-defined mobile network (SDMN) is defined in MobileFlow [69], and its authors provide a comparison to the current Evolved Packet Core (EPC) architecture. A

prototype implementation is also proposed in [69], based on the concept of the MobileFlow Forwarding Engine (MFFE), which encompasses all the user plane protocols and functions, and the MobileFlow Controller (MFC), which is a logically centralized entity that configures dynamically the MFFEs (i.e., the data plane). Although MobileFlow is OpenFlow-based, MFFEs must also support operations that are not carried out at the switch-level, as layer-3 tunneling, for instance. Mobility management can be supported as the controller can update forwarding rules according to the tunnel encapsulation or decapsulation requirements. This approach is also followed in our implementation, where we can set the tunneling and forwarding rules above link layer and the controller updates the forwarding rules in the OpenFlow domain. A different approach presented in [70] proposes to move the EPC to the cloud by means of virtualization and implementing GTP extensions for OpenFlow for mobility management. The mobility solutions proposed by both works are based on the same mobility concepts currently used in cellular networks, hence inheriting the scalability issues commented in Section 2.2.6.

An architecture to support flow-based routing in wireless mesh networks by means of OpenFlow is proposed in [71], which has been evaluated with a mobility management implementation that focuses on network-initiated handovers triggered by IEEE 802.21 MIH Events. This implementation includes an OpenFlow controller and a Monitoring and Controlling Server (MCS) that decides to which Mesh Access Point (MAP) the mobile node should connect and updates the forwarding rules. During the handover, the controller configures temporary routes that forward the traffic to the two MAPs involved and will be removed when the handover is complete. The minimum outage due to handover that they achieve is on average 200 ms and their results also confirm that the main contribution to this delay is due to the association to the new MAP. A similar approach is followed in [72], but in this case with in-band signaling and relying on a centralized controller for data rules and a local distributed controller that takes care of the control rules. In our architecture, we also propose two hierarchy levels for the controller deployment, namely the local controller (CLC) and the regional controller (CRC) with the aim of managing the mobility between different domains. However, we go beyond and provide mechanisms for IP mobility continuity while roaming to different domains.

2.9 Mobile data traffic analysis for mobility management

Cellular networks have evolved into extremely complex systems, where performance and behavior depend on the interaction of a multitude of logical modules. Any analysis performed over these systems faces scalability challenges due to the number of nodes and amount of traffic served. An increase in both magnitudes is expected due to the growing traffic demands and the so-called RAN densification. In this scenario, operators struggle to monitor and analyze their networks through a plethora of vendor specific

probes and management systems, providing information which is difficult to aggregate and analyze. The lack of tools for the design and optimization of next generation networks, carrying several orders of magnitude more traffic and serving a wider set of potential clients (including machine and humans) is a challenge, requiring novel techniques that are able to provide trends, relations and design guidelines for the deployment of new systems. A promising trend in this area is the use of Big Data techniques to gather information on the behavior of the network, analyzing and inferring knowledge out of the myriad of data flows transported by the network [73]. It is a common practice to monitor the traffic flowing through the network to evaluate network performance or to look for delay or usage patterns [74]. With the increase of data traffic and the raise of more powerful processing techniques and capabilities, this traffic monitoring has turn to Big Data techniques to analyze the traffic [75]. In Chapter 10 we aim at going one step beyond and provide some insights on the usage of resources for mobility management with actual data from a real operator's network.

Big data and data analytics are recently emerging to facilitate the development of new analytics applications, and to leverage the mobile operators understanding and exploitation of data which is constantly flowing through their networks infrastructures. In a more general point of view, Big Data platforms are utilized (namely Hadoop [76]) for the exploitation of data analytics by telecom operators. Examples range from anomaly detection for IT infrastructure security and resiliency [77], network coverage analysis [78] or proactive caching for 5G [79], [80] to social network analysis for consumer behavior modeling [81].

However, none of the studies described above have demonstrated the deployment and adaptability aspects of DMM-based solutions inside a real mobile operator by exploiting a comprehensive mobile data analysis. Therefore, it is clear that in order to embrace new technologies for long term alternatives to current centralized cellular infrastructures, recently proposed DMM solutions need to be further investigated in the context of applicability and adaptability for mobile operators. Based on this observation, in Chapter 10 we focus on the validation of DMM performance using real operator's network data by exploiting big data techniques for the analysis. To the best of our knowledge, the comparison of distributed mobility management with current centralized mobility management approaches with large amount of mobile data usage is not available in the literature. Using our analysis, we extract interesting conclusions by bringing real world considerations about the applicability of a distributed mobility management solution inside mobile operator's network.

2.10 Ethernet Passive Optical Network and WOBANs

A Passive Optical Network (PON) is a passive point-to-multipoint (PtMP) optical access network following a tree topology, as shown in Figure 5.1. The leaf nodes, called Optical Network Units(ONUs), are connected to the root node, referred to as the Optical Line Terminal (OLT) via a passive splitter/combiner that needs no power supply or configuration. The role of such passive device is two-fold: First, it splits the signal coming from the OLT into, typically, 32 or 64 copies for the ONUs in the downstream direction; and second, it combines the signals generated by the ONUs into a single one in the upstream direction. Thus, the PON operates as a broadcast-and-select network in the downstream direction, since the data sourced at the OLT is replicated by the passive splitter/combiner and delivered at all ONUs.

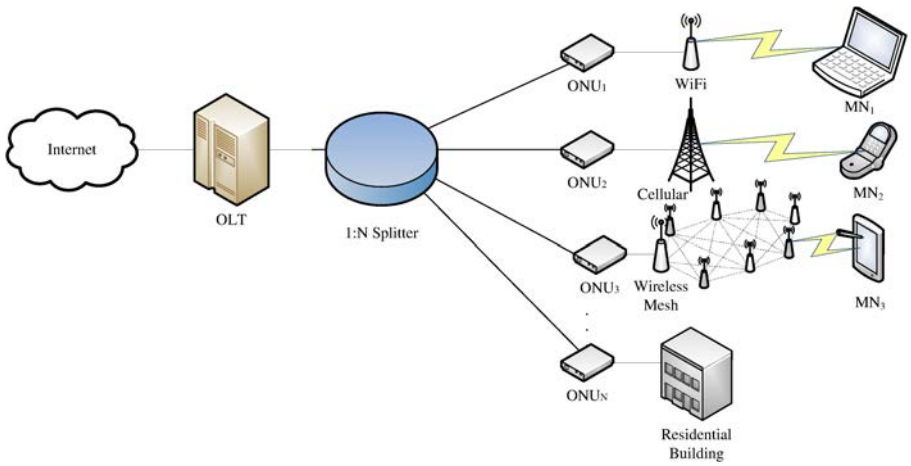


Figure 2.12: Ethernet Passive Optical Network (EPON) architecture

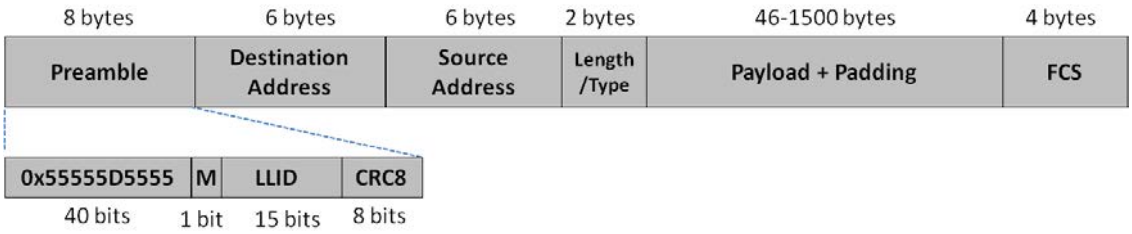


Figure 2.13: IEEE 802.3ah EPON frame format.

An Ethernet PON (EPON) is a type of PON employing IEEE 802.3/Ethernet frames, which contain a small EPON-specific header in its Preamble (see Figure 2.13). The Logical Link Identifier (LLID) field specifies the recipient ONU(s) of the EPON frame, and the Mode bit specifies whether the LLID is unicast (M=0) or broadcast/multicast (M=1). A unicast LLID is assigned to each ONU by the OLT. Hence, upon the reception of an EPON frame, every ONU must check the LLID field to filter out all the frames intended

for other ONUs.

On the other hand, the upstream wavelength is shared by all ONUs on a Time Division Multiplexing (TDM) basis, so a channel access arbitration mechanism must be defined to avoid collisions at the passive splitter/combiner. In light of this, the IEEE 802.3ah standard also defines the Multi-Point Control Protocol (MPCP), where the OLT schedules transmission windows to the ONUs after a clock synchronization process. A number of Dynamic Bandwidth Allocation (DBA) algorithms have already been defined in the literature, being Interleaved Polling with Adaptive Cycle Time (IPACT) [82] the most popular one.

The research community has proposed the combination of optical and wireless technologies to provide anywhere-anytime broadband access networks meeting the bandwidth requirements of the next-generation applications. The term WOBAN, which stands for Wireless-Optical Broadband Access Networks, has been used in [83] to refer to a Passive Optical Network (PON) whose termination points are attached to wireless (either WiFi- or WiMAX-based) or cellular access technologies.

WOBANs are seen as attractive broadband access networks since they combine the benefits of PONs:

- high bandwidth capacity, typically 1 - 10 Gbit/s,
- network based on a passive infrastructure that eases Operation, Administration and Management (OAM),
- Capital Expenditure (CAPEX) savings since many users share the optical fiber connection to the Central Office,
- simple interoperability for the case of Ethernet-based PONs,

together with the advantages of mature wireless or cellular technologies:

- radio coverage of hundreds of meters for WiFi, and several kilometers for cellular technologies,
- tens to hundreds of Mbit/s for WiFi, and tens of Mbit/s for cellular technologies,
- wide popularity and availability of wireless devices among end-users.

Essentially, a WOBAN is a PON whose terminating points, the ONUs, are attached to one or more heterogeneous wireless Access Points (WiFi, WiMAX, cellular or other). However, before such a hybrid broadband access network becomes a reality, it is required to study certain aspects of the mobility management of users moving between wireless Access Point (AP)s attached to the PON. These, and other related challenges are outlined in [84]. Indeed, previous studies have already addressed a number of key aspects to achieve real optical-wireless integration, i.e. those related to keeping Quality of Service (QoS)

over the EPON and WiMAX [85] [86], effective routing strategies considering capacity and delay aspects [87] or from an energy efficiency point of view [88]. However, very few works have considered mobility in WOBANs. For instance, [89] proposes a new, non-standard, mobility mechanism that requires special hardware at the ONUs and PON splitter, as well as support from the end-user mobile node. The authors in [90] also propose to integrate PMIPv6 over EPONs defining two possible architectures, but it requires additional changes to provide mobility support. In Chapter 5 we provide an architecture design for a Wireless Optical Broadband Access Network (WOBAN) that includes mobility support and enhanced handover thanks to the integration with PMIPv6 and 802.21 MIH protocols.

Part II

Mobility Management in Heterogeneous Scenarios

Chapter 3

Combination of Mobility Protocols

3.1 Introduction

Several IP mobility protocols have been standardized in the recent years to provide support for a specific functionality and requiring the operation of particular network nodes. Whereas each IP mobility protocol provides a solution for a specific feature, some scenarios could benefit from a combination of several solutions. Therefore we have studied and compared the performance of some of these protocol combinations in terms of signaling overhead and handover latency, including an experimental evaluation. This evaluation will help to understand the impact of implementing each of the mobility protocols, as well as their combinations. Although the current trends indicate that there is not a unique solution for the mobility scenario, our analysis highlights the needs to alleviate the costs and improving performance.

3.2 Combining IP Mobility Protocols

The protocols for mobility support described in Chapter 2 are designed to be used independently. However there are circumstances in which two or more of them can be combined. In most cases, the combination is the result of individual actions of the different actors –users, operators– involved in the scenario, each of them deploying a solution to fulfill their own requirements. For example a client-based solution is set up by a user requiring global mobility, but then, the user's MN visits a network where the operator has deployed a network-based solution to provide mobility support to its visiting nodes. On the other hand, the combination can also be planned to get together different functionalities, for example network mobility and host mobility. The basic combinations do not require modifying the individual protocols. Although they are used together, they

are not aware of each other and they do not have explicit mechanisms to cooperate, so there is no increased complexity because there is no new functionality implemented in the involved nodes. We next describe and analyze different combinations of IP mobility protocols, explaining the motivation for each combination, the functionality resulting from that combination, and the additional complexity, if any, that each of the particular combinations brings.

3.2.1 MIPv6+PMIPv6

A mobile node uses MIPv6 to obtain global mobility support (i.e., it can roam to any visited network while keeping global reachability and session continuity). On the other hand, an operator deploys PMIPv6 to offer local mobility support within the domain without requiring any support from the user terminals. In this scenario, a MIPv6 node may visit the PMIPv6 domain.

The operation of MIPv6 in the mobile node when visiting a PMIPv6 access network is the same as when visiting any other foreign network: initially, after attaching to the domain, the mobile node gets an IP address to be used as its care-of address, and registers it in its global mobility agent (i.e., the home agent), to bind this temporal address to its permanent address (i.e., home address). Since the IP address used in the PMIPv6 domain remains the same while roaming within this domain, movements are transparent to the mobility management software in the user terminal (i.e., MIPv6). Furthermore, the terminal can also move to an access network outside the domain while keeping ongoing sessions. This is done by the terminal getting another CoA from the new access network and using MIPv6 to keep its home agent updated with its new location.

In this combination of IP mobility protocols, no explicit cooperation among the protocols or extra complexity is required, so each of the mobility protocols functions as usual, not even being aware of their simultaneous operation.

3.2.2 NEMO B.S.+PMIPv6 (+MIPv6)

A mobile router uses NEMO B.S. to obtain global mobility support for itself and the network behind it. A node inside the mobile network can be a regular IP node without mobility support, if it is not going to move away from the mobile network. It can also be a node using MIPv6 to have global mobility support by itself, i.e., to be able to leave the mobile network and roam to other networks. In addition, an operator deploys PMIPv6 to offer local mobility support enabling local roaming without requiring any support from visiting nodes (hosts or routers).

A particularly relevant use case scenario for this example is the provision of Internet connectivity in public transportation systems (e.g., buses) where users benefit from seamless access using mobility unaware devices, while the network mobility support takes care

of managing the mobility on behalf of the terminals. Some of the access networks the mobile network may visit could also provide PMIPv6 support. In the situation where NEMO B.S. and PMIPv6 protocols are combined, the mobile router gets a care-of address when it enters the localized domain, and registers this address in its home agent, binding the mobile network prefixes managed by the mobile router (that is, the IPv6 prefixes used inside the mobile network) to its current location (i.e., its care-of address). Since this new acquired IPv6 address is provided by the PMIPv6 domain, it does not change while the mobile network roams within the localized domain, and therefore its movements are transparent to NEMO B.S. software in the mobile router. Moreover, the mobile network is able to roam not only within the localized domain but also outside the domain, thanks to the NEMO B.S. operation that provides global mobility support. A user terminal in the mobile network attached to the mobile router will not be able to leave the mobile network without breaking its ongoing sessions unless this terminal has MIPv6 support itself.

As in the MIPv6+PMIPv6 case, in this combination of IP mobility protocols no explicit cooperation or extra complexity is required, so each of the mobility protocols operates as usual, not even being aware of their simultaneous operation.

3.2.3 MIPv6+N-PMIPv6

This combination is very similar to the case of (MIPv6+PMIPv6). A mobile node uses MIPv6 to obtain global mobility support. In addition, an operator deploys N-PMIPv6 to offer local mobility support enabling local roaming without requiring any support from user terminals. With N-PMIPv6 this local mobility domain is composed of fix and moving access gateways. In this scenario, a MIPv6 terminal may visit the N-PMIPv6 domain.

In this scenario a user terminal can move both within a localized domain without changing its care-of address and can also leave the domain without breaking any ongoing communications, by acquiring a new care-of address from the new access network and using MIPv6. The difference with the first combination is that the localized domain integrates both fixed gateways (MAGs) and moving gateways (mMAGs), so that a user terminal is able to roam between fixed and mobile access infrastructure within the domain without requiring any IP mobility support in the terminal, thanks to the use of N-PMIPv6 protocol. Whenever the terminal changes its location within the domain, the new access gateway, fixed or mobile, will update the terminal's location in the LMA.

An example of this scenario could also be a public transportation system, where mobility unaware devices would not only get Internet access while moving (e.g., in a bus or train) or while waiting at the station platforms, but also while roaming between fixed and mobile access infrastructure (e.g., getting on or off the bus). Additionally, the use of MIPv6 would also enable a mobile node to roam outside the localized domain, for example, when leaving the public transportation environment.

As in the previous cases, in this combination of IP mobility protocols, no explicit cooperation among the involved protocols or extra complexity is required, so each of the mobility protocols functions as usual, not even being aware of their simultaneous operation.

3.2.4 NEMO B.S.+N-PMIPv6 (+MIPv6)

In this combination, as in the previous one, an operator deploys N-PMIPv6 to offer local mobility support within the domain without requiring any support from the user terminal. But, in addition, the operator also deploys NEMO B.S. mobile router capabilities in the moving gateways, which enable the corresponding mobile networks to be able to move outside the localized domain while keeping ongoing sessions. This could be a common configuration if the mobile network needs to move out of a domain (e.g., a bus leaves the N-PMIPv6 localized domain deployed in a city and connects to another network operator). In this combination, thanks to the use of the N-PMIPv6 protocol, the localized domain integrates both fixed gateways (MAGs) and moving gateways (mMAGs), so that a user terminal is able to roam between fixed and mobile access infrastructure within the domain without changing its IP address. The terminal can also be connected to a mMAG that moves outside the localized domain and, thanks to the use of NEMO B.S. functionality, this movement will be transparent to terminals in the mobile network, i.e., they will not need to change their IP addresses. The terminal can also use MIPv6 to obtain global mobility, i.e., to be able to roam outside the access infrastructure provided by the operator through N-PMIPv6 and the mobile networks created by using NEMO B.S.

The most efficient way of deploying this scenario is by co-locating the the home agent of the mobile router and the local mobility anchor of the moving gateway in the same node, so they share the range of addresses to be used (i.e., the mobile network prefixes of the NEMO are part of the IPv6 address space of the localized domain and, therefore, they are topologically anchored at the LMA). With this configuration, the localized domain becomes also the home network of the global mobility support (i.e., home domain). Therefore, when the mobile network is at the home domain, packets addressed to a user terminal attached to this mobile network are forwarded as in the N-PMIPv6 simple case, through the LMA. This means that when the mobile network is away from its home domain, a bi-directional tunnel is created between the mobile router – after obtaining a new care of address from the visited network – and its home agent, used to forward all the traffic from or to terminals connected to the mobile network. Note that in case the mobile network moves out of its home domain, the mobile router (also with moving gateway functionality) cannot act anymore as a moving gateway, either because the visited domain is not an N-PMIPv6 localized domain or because the moving gateway lacks the appropriate security associations with the localized mobility anchor of the visited domain.

When the mobile network is not at its home domain, a user terminal moving away from the mobile network would need to change its IP address, thus breaking ongoing sessions unless the mobile node has its own MIPv6 support.

In this combination a node in the network has to combine LMA (N-PMIPv6) and HA (NEMO B.S.) functionality. Additionally, the moving access gateways have to combine mMAG (N-PMIPv6) and MR (NEMO B.S.) functionality. [91] documents the issues that might arise from the interactions between PMIPv6 and MIPv6 when the LMA and the HA are co-located, being some of their recommendations applicable to the NEMO B.S. + N-PMIPv6 combination addressed here. The implementations of N-PMIPv6 and NEMO B.S. can work independently (actually, [91] recommends to avoid the LMA and HA entities sharing their binding cache). Nevertheless we have to guarantee the compatibility of the addressing assigned by both protocols to the same nodes. This can be done by using static pre-assignments of IP prefixes to be used by each mMAG/MR. In the mMAG/MR node, mobile router functionality can be triggered for example by changes in the used IP prefix in the outgoing interface, and moving MAG functionality can be triggered by detecting the advertisement of an IP prefix in the outgoing interface that belongs to the mMAG's home network. Other means of triggering the protocols are also possible, such as using hints from the authentication mechanism in the access network. When the mMAG/MR enters a visited network away from its home domain, it has to register the IP prefixes used inside the mobile network in the HA. When the mMAG/MR enters the home domain it has to register itself in the LMA and also it has to register the identities of the nodes attached to the mMAG. In the LMA/HA, each implementation processes its own signaling and behaves accordingly, without affecting the other one. We could make some optimizations by enabling cooperation between both protocols. For example, in the LMA/HA both implementations could share a database with information about prefixes and the identities of nodes using them. The database would be updated dynamically by both implementations. Therefore, for making this combination work, we need to combine the implementation of different protocols in the same nodes (LMA/HA and mMAG/MR) and the corresponding configuration. This means some added complexity in both the LMA/HA and the mMAG/MR. But the added complexity is not much compared with the independent implementation of the mMAG and the MR functionalities, and the operator gains the ability to offer transparent connectivity service to nodes roaming in its domain or connected to its mobile networks even when they move to other domains, and that without depending on functionality or configuration in the mobile nodes themselves. The possible use of MIPv6 in a mobile node to achieve global mobility by itself is independent of the N-PMIPv6+MIPv6 solution, both work unaware of each other, so there is not added complexity in this case.

3.3 Performance Analysis

In this section we present the results of an analytic performance evaluation of combining different IP mobility solutions, in terms of protocol overhead and handover latency.

3.3.1 Overhead

We analyze the overhead in terms of the control headers introduced for the tunnels configured by each combination. In this analysis we focus not only on the amount of added packet overhead, but also on which segments of the network suffer from this extra overhead, as the impact is more important if the additional control information appears on wireless segments of the network. In order to simplify this exercise, we limit the analysis of the MIPv6 overhead to the case where Route Optimization (RO) is enabled. The difference between using RO mode or bi-directional tunneling (BT) mode is basically the following: in BT mode, the overhead is higher (40 bytes, instead of 24), but it only appears between the mobile terminal and its home agent, while in route optimized mode the overhead is present along the complete path between the mobile terminal and its correspondent node.

3.3.1.1 MIPv6 + PMIPv6

In this case, 24 bytes of additional overhead are added in the whole path between the mobile node and the correspondent node, due to the use of Mobile IPv6 (in RO mode), plus an IPv6 tunnel (40 bytes) between the localized mobility anchor and the MAG where the mobile node is attached to, due to the use of Proxy Mobile IPv6. It is important to note that, out of the overall overhead, only the 24 bytes added by Mobile IPv6 are present in the wireless access.

3.3.1.2 NEMO B.S. + PMIPv6 (+ MIPv6)

Two different tunnels are involved to enable the communications of the mobile network: one between the mobile router and its home agent, due to the use of NEMO B.S., and another one between the LMA and the MAG serving the mobile router, due to the use of Proxy Mobile IPv6. Thus, there are up to 80 additional bytes of overhead in some wired segments of the path (when both tunnels are present), and up to 40 bytes in the wireless access due to the use of NEMO B.S., though not in the last wireless hop between the user terminal (i.e., the MNN) and its MR. Note that this last wireless hop is where the effect on battery consumption and bandwidth waste is likely to be more significant. A third overhead component (24 bytes in route optimized mode) is required if the terminal attached to the mobile network is itself a MIPv6 mobile node outside its home network.

3.3.1.3 MIPv6 + N-PMIPv6

In this case a mobile node is attached to a moving gateway, and three overhead components are required: *i)* 24 bytes between the mobile and the correspondent node due to the use of Mobile IPv6 in route optimized mode, *ii)* an IPv6 tunnel (40 bytes) between the LMA and the fixed gateway where the moving gateway is attached to, and *iii)* a second tunnel between the LMA and the moving gateway.

3.3.1.4 NEMO B.S. + N-PMIPv6 (+ MIPv6)

When a mobile network is attached to a moving MAG (which is at its home N-PMIPv6 domain), and assuming a deployment scenario in which the LMA and the HA of the mobile router are co-located, two IPv6 tunnels are required: one between the localized mobility anchor and the fixed gateway serving the moving gateway, and a second one between the localized mobility anchor and the moving gateway. If the user terminal that is getting access through the mobile network is a mobile node running Mobile IPv6 (which is outside its home network), an additional 24-byte overhead component is required due to the use of Mobile IPv6 in route optimized mode.

Figure 3.1 shows the overhead of all the analyzed combinations over the different network segments. Depending on the combination under consideration, we can have up to three extra headers in the wired access network backhaul, up to two in the wireless access backhaul (i.e., between the fixed access network and the moving MAG/mobile router), and up to one in the last wireless hop to the terminal.

In order to understand the effect of the mobility overhead with user traffic, we have taken data from a real access network deployment offering Internet access during a conference (ACM CoNEXT 2008 [92]). The average packet size for UDP or TCP traffic is 710 bytes (including all headers). For the case of three additional overhead components (104 bytes), the extra headers account for a waste of 14.6% of the bandwidth. If two extra headers are required, the waste is between 9% and 11.26% (for the cases of 64 and 80 bytes of overhead, respectively). Finally, if only one overhead component is needed, the bandwidth waste is between 3.38% and 5.6% (for the cases of 24 and 40 bytes). Nevertheless, note that these figures just represent a mixture of user traffic – composed mostly of HTTP data – in a conference. In mobile scenarios the overhead penalty will tend to be worse, for example with the expected increase of VoIP traffic. It is worth highlighting that the extra headers – in addition to the bandwidth waste – also involve the extra energy consumption required to transmit them, which is significant in wireless environments. Moreover, it is commonly argued that the problem is not so important in the access network backhaul, because it is usually wired and bandwidth is not severely limited, but wireless multi-hop access networks are becoming increasingly popular, which weakens this reasoning.

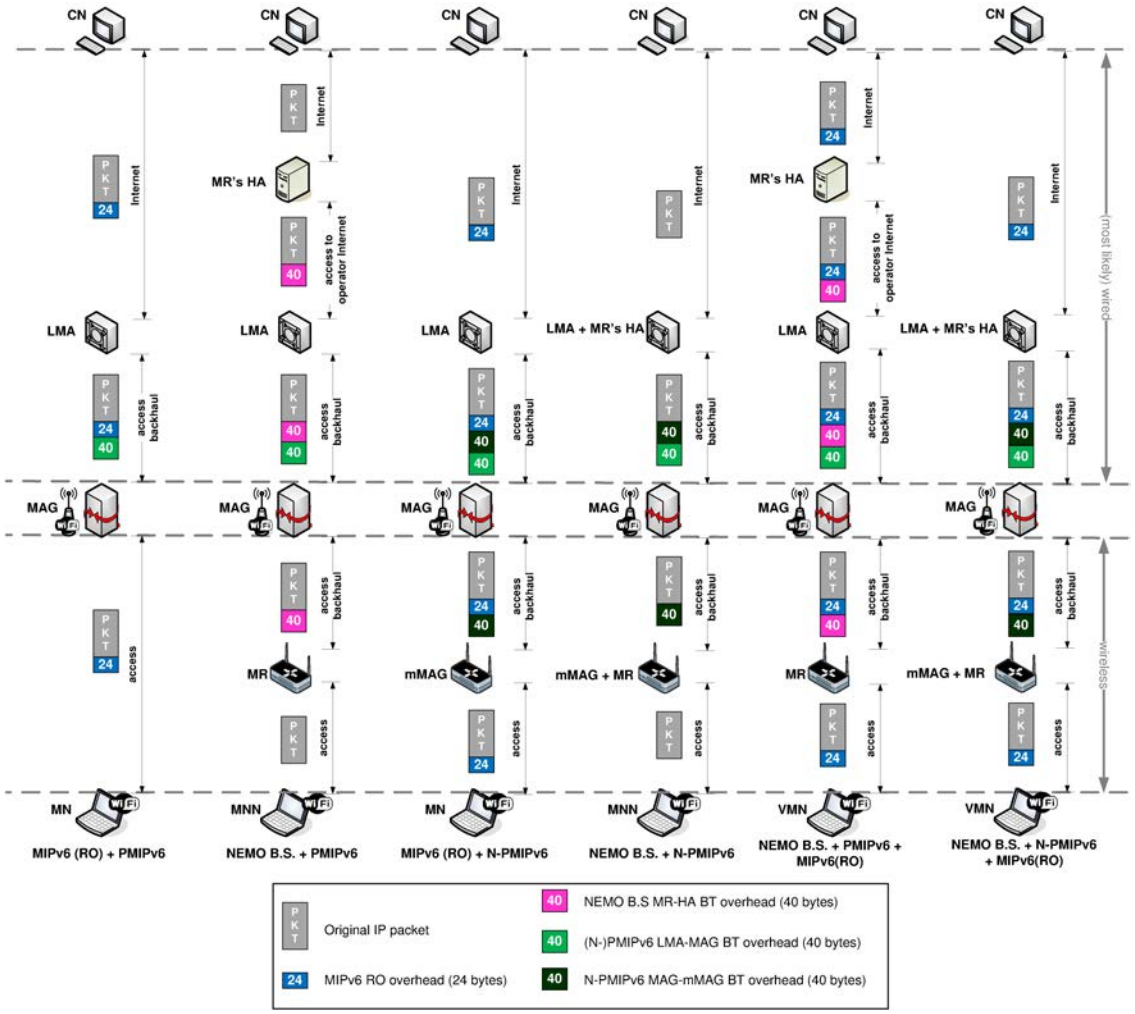


Figure 3.1: Overhead of the different combinations

3.3.2 Handover latency of IP mobility protocols

The handover delay of mobility protocols can be expressed as a combination of independent factors such as the layer-2 handover time, the movement detection time, the IP configuration time, and the specific mobility signaling delay. We briefly describe each of these factors:

1. Layer-2 handover time (T_{L2}^{ho}). It is defined as the time required by the layer-2 technology to perform a handover (i.e., disconnecting from its current point of attachment and connecting to a new one). In the case of an IEEE 802.11-based layer-2 technology, this time usually involves the channel scanning for candidate APs, plus the time required for re-association. As presented in [93], this delay can be modeled with a Beta probability distribution function. In [94] it is shown how its mean value can be reduced up to 50 ms by appropriately selecting the number of channels being scanned. For the handover delay analysis we conduct later, we take this last

number as the mean value of the Beta distribution, while the standard deviation of the resulting distribution is of 3.66 ms.

2. Movement detection time (T_{MD}). This delay corresponds to the time required by the terminal to detect that it has moved to a different layer-3 point of attachment. This detection can be performed by functionality at the IP layer or assisted by layer-2 mechanisms. If we focus on IPv6 mechanisms, movement detection can be done in different ways. The most simple (and the most widely supported) consists in using Router Advertisement (RA) messages. An access router periodically multicasts unsolicited RA messages. Typically, the time interval between these advertisements follows a uniform distribution: whenever a router advertisement is sent from an access router, a timer is set to a uniformly distributed random value [95] between the configured *MinRtrAdvInterval* (R_m) and *MaxRtrAdvInterval* (R_M). Although IPv6-based movement detection mechanisms are well known and supported, new optimized mechanisms to detect the connection of a terminal to a new point of attachment have been and are still being developed. That is the case of a mechanism known as link layer triggers, of which IEEE 802.21 Media Independent Event Service is a good example. This technology enables lower layers to notify the occurrence of a certain event, such as attachment or disconnection, to higher layers, e.g., the mobility management protocol. We consider the use of layer-2 assisted movement detection since it is the mechanism introducing the smallest delay, almost negligible.

3. IP configuration time (T_{IP}). This is the time required by the IP stack to configure a new IP address and update the forwarding table. This time depends on the hardware and operating system since this operation is generally performed by the kernel. It should be noted that this delay is not always present in a handover event, as the network-based mobility protocols ensure that the IP address, as well as the default router, of the moving terminal remain the same while roaming within the domain. On the other hand, if client-based mobility is used, the MN needs to configure a new IP address if the former one is no longer valid, and signal it to the anchoring point, e.g., in MIPv6 the MN needs to configure a new care-of address and send a Binding Update message to its HA. In the rest of our analysis we assume that the IP configuration time is very short (negligible), since the operations of configuring an address and updating the routing in the IP stack should not require a long time in modern computers or hand-helds. It is also worth noticing that we assume the use of IPv6 SLAAC mechanisms. The use of a different IP address configuration mechanism (e.g., DHCP) is likely to incur in higher delays.

4. Signaling delay ($T_{BU/BA}$ or $T_{PBU/PBA}$). This is the time required to update the HA or LMA, respectively. It highly depends on the distance between the entities participating in the mobility management: the terminal/gateway/mobile router on

the one side and the localized mobility anchor/home agent on the other side. In order to model this behavior we use measurements taken from the PingER (Ping end-to-end reporting) project[1]. We take the average value of the reported values for 3 types of scenarios characterized by the distance between the communication peers. In particular, we take a "local" delay characterized by an average value of 5.37 ms, a "regional" delay of 18.32 ms and a "continental" delay of 138.79 ms. To characterize the delay in an Internet path, these values are used as the mean of a Weibull distribution with variance provided by Hurst parameters of 0.8, 0.65 and 0.5 for local, regional and continental delays respectively [96] [97].

After explaining the different independent factors that are common to the handover time for all mobility solutions, we focus on understanding the handover delay for the different IP mobility management protocols (MIPv6, NEMO B.S., PMIPv6 and N-PMIPv6).

1. *MIPv6/NEMO B.S.* The delay incurred by a MIPv6 terminal performing a handover (in bi-directional tunnel mode) or by a mobile router, can be expressed as:

$$T(MIPv6\ BT/NEMO) = T_{L2}^{ho} + T_{MD} + T_{IP} + RTT(MN/MR, HA), \qquad (3.1)$$

where RTT(MN/MR,HA) represents the round trip time between the mobile node or mobile router, and the corresponding home agent.

In case route optimization mode is used in MIPv6, we assume the mobile node is performing optimistic registration [98], which means that the route optimization related signaling is performed in parallel with the registration signaling with the HA. For this case, an additional $RTT(CN, HA)$ component should be added to the delay shown in Eq. (3.1).

2. *PMIPv6.* If PMIPv6 is used to manage the mobility of the user terminal, the handover delay can be expressed as:

$$T(PMIPv6) = T_{L2}^{ho} + T_{MD} + RTT(MAG, LMA) \qquad (3.2)$$

In this case, in addition to the time required for the layer-2 handover and movement detection, we also add the signaling delay between the gateway and the localized mobility anchor. In case the MN is entering the localized domain for the first time, an additional T_{IP} component should be added to the delay shown in Eq. (3.2).

[1] `http://www-iepm.slac.stanford.edu/pinger/`

Table 3.1: Delay of the combination of different mobility solutions

Entity Moving		MIPv6+ PMIPv6	NEMO B.S.+ PMIPv6	MIPv6+ N-PMIPv6	NEMO B.S.+ N-PMIPv6	NEMO B.S.+ PMIPv6+MIPv6	NEMO B.S.+ N-PMIPv6+MIPv6
MN	within LMD	T(PMIPv6)	N/A	T(PMIPv6)	T(PMIPv6)	*MR* → *MAG*: T(MIPv6+PMIPv6) *MAG* → *MR*: T(MIPv6)	T(PMIPv6)
	to LMD	T(MIPv6+PMIPv6)	N/A	T(MIPv6+PMIPv6)	N/A	T(MIPv6)	T(MIPv6+PMIPv6)
MR	within LMD	N/A	T(PMIPv6)	T(PMIPv6)	T(PMIPv6)	T(PMIPv6)	T(PMIPv6)
mMAG	to LMD	N/A	T(NEMO+PMIPv6)	NA	T(PMIPv6)	T(NEMO+PMIPv6)	T(PMIPv6)

3. *N-PMIPv6.* The difference in terms of handover delay between the case of using N-PMIPv6 and the regular Proxy Mobile IPv6 case is just the additional hop between the moving MAG and the fixed MAG that has to be traversed. We consider this difference negligible in the next calculations.

3.3.3 Handover Latency for the Combinations of IP Mobility Protocols

We analyze next the resulting handover delay for the different combinations of mobility protocols when the mobile node or the mobile router/moving gateway moves. This is done for the cases when the handover is performed within the Localized Mobility Domain (LMD), or entering the LMD. In all the following situations we assume the visited domain supports PMIPv6 or its extension N-PMIPv6.

Table 3.1 summarizes the resulting handover delays for the different mobility combinations.

3.3.3.1 MIPv6+PMIPv6

This combination corresponds to a user terminal with MIPv6 support that hands off to a localized domain. In this case there are two possible mobility scenarios: *i)* The terminal moves within a localized domain, hence the mobility of the terminal is handled within the domain using PMIPv6, so the handover delay is equal to the PMIPv6 case, or *ii)* the terminal enters a localized domain, then the terminal handles itself using MIPv6, performing a handover to the localized domain. In this case the use of both mobility solutions is done in a sequential way: the terminal first configures an address that belongs to the prefix obtained through the operation of PMIPv6, which is then used as its care-of address by the MIPv6 protocol running on the terminal. The handover delay of this approach, when bidirectional tunnel mode is used, can be expressed as:

$$\begin{aligned} T(MIPv6+PMIPv6) &= T_{L2}^{ho} + T_{MD} + T_{IP} + RTT(MAG, LMA) \\ &+ RTT(MN, HA) \\ &= T_{L2}^{ho} + T_{MD} + T_{IP} + RTT(MAG, LMA) \\ &+ RTT(MN, LMA) + RTT(LMA, HA) \\ &\simeq T_{L2}^{ho} + 2 * RTT(MAG, LMA) \\ &+ RTT(LMA, HA). \end{aligned} \tag{3.3}$$

We separate the RTT between the MN and the HA in two parts: the RTT between the MN and the LMA, and the RTT between the LMA and the HA. This is because when the MN is in a PMIPv6 domain, its traffic, including the MIPv6 signaling, has to go through the LMA. Separating the RTT in two parts allows us to consider the influence of the distance between the LMA and the HA. Additionally, the RTT between the MN and the LMA is equal to the RTT between the MAG and the LMA plus the delay in the hop MAG-MN that we consider negligible.

In the case of optimistic route optimization mode, an additional $RTT(CN, HA)$ component should be added to the delay shown in Eq. (3.3).

3.3.3.2 NEMO B.S.+PMIPv6

In this combination, instead of a user terminal running MIPv6, the entity moving is a mobile router running the NEMO B.S. protocol. As in the previous section, two situations may arise: *i)* the mobile router moves within a localized mobility domain, hence the mobility of the router within the domain is handled by using PMIPv6, or *ii)* the mobile router enters a localized domain, so the router handles the macro-mobility (using the NEMO B.S. protocol). This case is similar to the MIPv6+PMIPv6 without route optimization, being Eq. (3.3) also applicable in this scenario (considering a mobile router instead of a mobile node).

$$\begin{aligned} T(NEMO+PMIPv6) &= T_{L2}^{ho} + T_{MD} + T_{IP} + RTT(MAG, LMA) \\ &+ RTT(MR, LMA) + RTT(LMA, HA) \\ &\simeq T_{L2}^{ho} + 2 * RTT(MAG, LMA) \\ &+ RTT(LMA, HA). \end{aligned} \tag{3.4}$$

3.3.3.3 MIPv6+N-PMIPv6

This scenario considers a user terminal with MIPv6 support that may move to an LMD where gateways are able to move (mMAGs), hence there are two mobile entities:

the mobile node and the moving gateway. In case the terminal attaches to a moving gateway, as shown in the delay explanation for the N-PMIPv6 solution (see Section 3.3.2), the difference in delay between the PMIPv6 and N-PMIPv6 cases is due to an additional hop in the local network. Taking this fact into account, the combined solution of MIPv6+N-PMIPv6 slightly increases the delay by $RTT(mMAG, MAG)$ – which we consider negligible – compared to MIPv6 + PMIPv6.

If the entity moving is a moving gateway, it can move either within the localized mobility domain or from outside to the domain.

N-PMIPv6 protocol allows the moving gateway to roam within the domain (using PMIPv6 protocol). Hence, in the case the moving gateway roams within the domain, the handover delay is equal to the one obtained in PMIPv6. The case where the moving gateway moves to a different localized mobility domain is not considered here, since there is no mobile router functionality in the moving gateway and, therefore, mobility cannot be granted.

3.3.3.4 NEMO B.S.+N-PMIPv6

This combination considers user terminals without MIPv6 support and mMAGs that also incorporate NEMO B.S. functionality, so they can hand off outside the LMD. As in the previous section, in this scenario two entities are able to move, the user terminal and the moving gateway. In case the terminal moves, it can attach to another access router belonging to the same domain and the mobility is handled by N-PMIPv6.

The case where the terminal hands off from outside to the domain is not applicable since the terminal does not have mobility support in this case and, hence, it cannot hand off from outside the localized mobility domain. If the entity moving is the moving gateway, it can move *i)* within the domain, using the N-PMIPv6 protocol that allows the moving gateway to roam within the domain, or *ii)* to the domain from the outside (in this case the moving gateway is outside the domain and it performs a handover to it). As the moving gateway is part of the domain, the prefix used by its mobile router functionality belongs to the localized domain. Therefore it only requires performing a PMIPv6 registration in order to hand off to the domain.

3.3.3.5 NEMO B.S.+PMIPv6+MIPv6

This scenario encompasses a user terminal supporting MIPv6, and a localized mobility domain where there is an Mobile Router (MR) attached. The terminal can move within the domain or can move to it from outside. In case the terminal is attached to the domain, it can be anchored to a gateway or to the MR. For the scenario where the terminal is attached to the mobile router and hands off to a gateway, the terminal uses both MIPv6 and PMIPv6 to reconnect to the domain, i.e., it is equivalent to the MIPv6+PMIPv6

scenario. On the other hand, if the terminal is attached to a gateway and hands off to the mobile router, it must use MIPv6 to regain connectivity, since the address provided by the MR does not belong to the domain, but to the home network of the router.

The case where the terminal is moving to the domain from the outside is equivalent to the MIPv6+PMIPv6 case. In this mobility combination scenario the mobile router can also move, being this case equivalent to the NEMO B.S.+PMIPv6 scenario.

3.3.3.6 NEMO B.S.+N-PMIPv6+MIPv6

This scenario considers a terminal with MIPv6 support and a localized mobility domain where there are mMAGs with mobile router functionality. This scenario can be analyzed as a combination of the previous scenarios. The case where the terminal moves to the domain, is equivalent to the combination MIPv6+N-PMIPv6, while the case where the mobile router moves is equivalent to the NEMO B.S.+N-PMIPv6 combination.

3.3.4 Delay performance analysis

Figure 3.2 gives some insight about the impact of the combination of different mobility solutions on the delay experienced during handovers. It represents, for different mobility solutions and different topologies, the percentage of handovers with a delay below a 150 ms threshold. This value is a reasonable disruption time for most applications, assuming the use of some buffering function to minimize packet loss during the interruption caused by the handover in the communication [94]. The first two bars on the left of the figure are the percentage of handovers whose delay is below 150 ms for NEMO B.S. or MIPv6 in bi-directional tunneling mode (BT), depending on the distance between the Home Agent and the Mobile Node/Router. Using global mobility the MN/MR can move everywhere. Additionally, the visiting network may provide local mobility support. This allows better efficiency in handovers inside the local domain (see the bars referring to PMIPv6 in Figure 3.2) but at the cost of a longer handover delay to move into the local domain, as shown in the bars referring to the MIPv6 BT/NEMO B.S.+PMIPv6 combination. This cost increases with the distance between the MAG and the LMA. For example, a MIPv6 terminal that is not using Route Optimization and it is at regional distance from its HA has a probability close to 95% of having a delay below 150 ms when executing a handover to an access network without local mobility support; but when the access network has local mobility support the probability is reduced to the 55-90% range depending on the distance between the MAG and the LMA in the local domain and the distance between the LMA and the HA. It is also worth noticing that the effect of the delay MAG-LMA is greater than the effect of the delay LMA-HA. This is because the path MAG-LMA is traversed both for the PMIPv6 and for the MIPv6 signaling while the path LMA-HA is only traversed by the MIPv6 signaling – see Eq. (3.3). Figure 3.2 also shows the

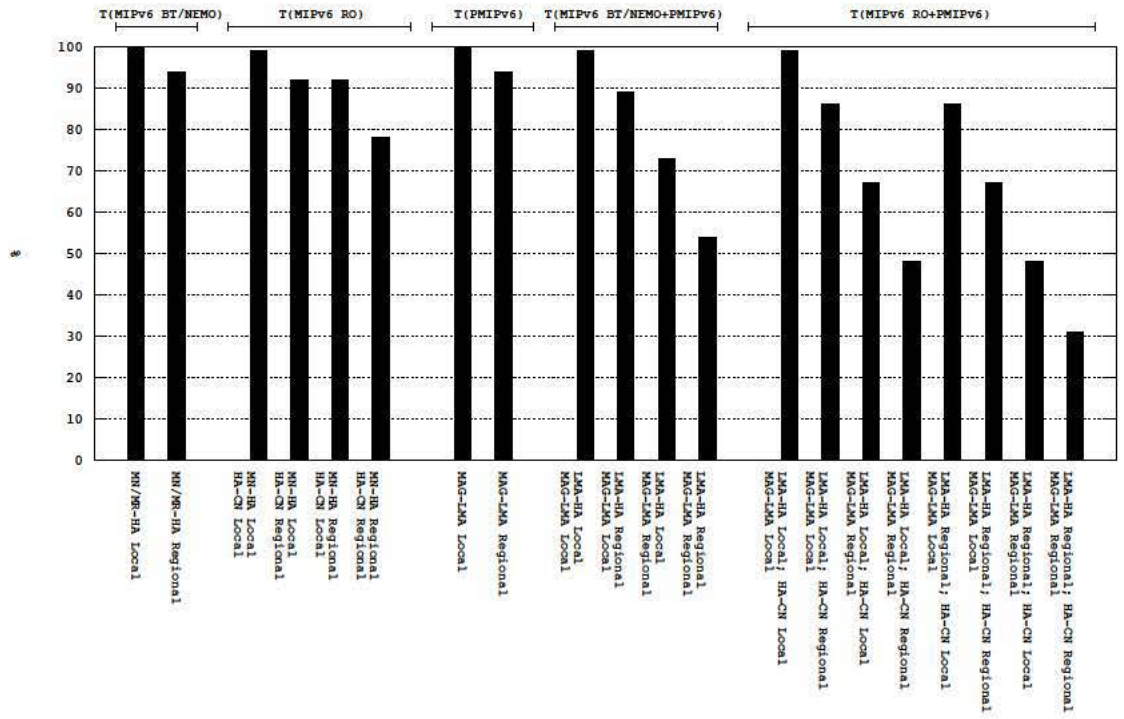


Figure 3.2: Percentage of handovers below 150 ms

performance of the handovers of MIPv6 terminals with Route Optimization, and of the handovers of MIPv6 terminals with Route Optimization moving to a PMIPv6 domain.

To fully understand the importance of these results we need to consider two aspects. First, when evaluating handover performance we need to focus on the longest handover that the terminal can suffer in any type of handover, because that will determine the performance and the needed mechanisms (e.g., buffering) to avoid interruptions in the terminals communications. The second aspect is the frequency of the handovers: if some type of handover was very unusual, we could accept having worse performance in that type of handover. The frequency of the type of handovers depends on the scenario, but we argue that the trend in mobile communications networks is towards very dynamic mobility scenarios in which nodes will change of access network very frequently according to the access network availability and the terminal requirements, so probably every type of handover will become usual.

3.4 Experimental analysis

We next complement the findings of our analytic study, by performing an experimental evaluation using Linux-based implementations of IP mobility protocols and conducting experiments under different scenarios.

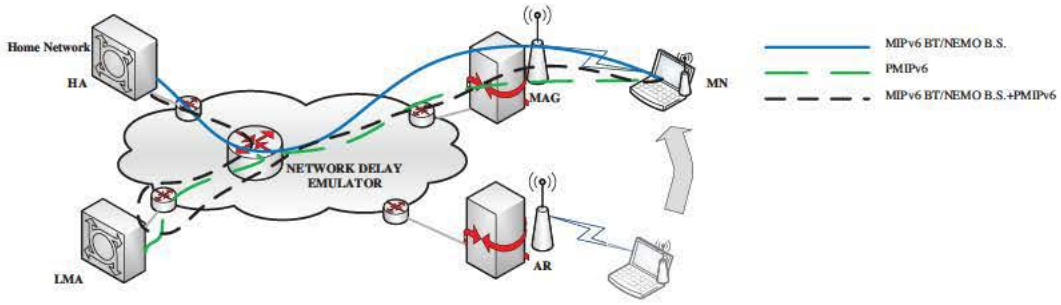


Figure 3.3: Testbed description

3.4.1 Testbed description

In order to experimentally evaluate the behavior of some of the combinations of mobility protocols considered, we have designed and deployed a testbed covering the scenarios shown in Figure 3.3. The role of the HA, MR/MN, LMA, MAG and AR are played by Linux boxes. The MR/MN, MAG and AR are equipped with Atheros wireless cards using the `ath5k` driver.[2] We have used an in-house software implementation of the MIPv6 BT/NEMO B.S. protocol developed under the framework of the POSEIDON project[3] and the OAI PMIPv6 implementation.[4]

The main purpose of these tests is to confirm the findings of our analytic analysis in terms of handover delay. In order to do so, we have measured the handover time, splitting it into several steps: *i)* layer-2 handover delay (T_{L2}^{ho}), *ii)* movement detection delay (T_{MD}), *iii)* IP configuration time (T_{IP}) and *iv)* mobility signaling delay ($T_{BU/BA}$ and $T_{PBU/PBA}$) of the mobility protocol.

The measurements of the different handover steps have been taken by monitoring each of the network interfaces of all the mobility entities involved and time-stamping the transmission and reception of each control message. To emulate different distances between the entities participating in the mobility management, we have introduced an additional router, capable of adding a variable delay using `netem`[5] (Network delay emulator in Figure 3.3). In order to extract statistically reliable figures, each test has been repeated between 20 and 40 times.

The time for performing a layer-2 handover (T_{L2}^{ho}) has been measured as the time elapsed between the moment that the wireless client starts trying to associate with a new access point and the moment this association is effectively completed. The movement detection delay (T_{MD}) includes the time required for the layer-2 event notifying of the new link layer connection. The reception of this event at the IP layer triggers the MN/MR to

[2] `http://linuxwireless.org/en/users/Drivers/ath5k`
[3] `http://enjambre.it.uc3m.es/~poseidon/`
[4] OpenAir Interface PMIPv6: `http://www.openairinterface.org/components/page1103.en.htm`
[5] Network Emulator: `http://www.linuxfoundation.org/collaborate/workgroups/networking/netem`

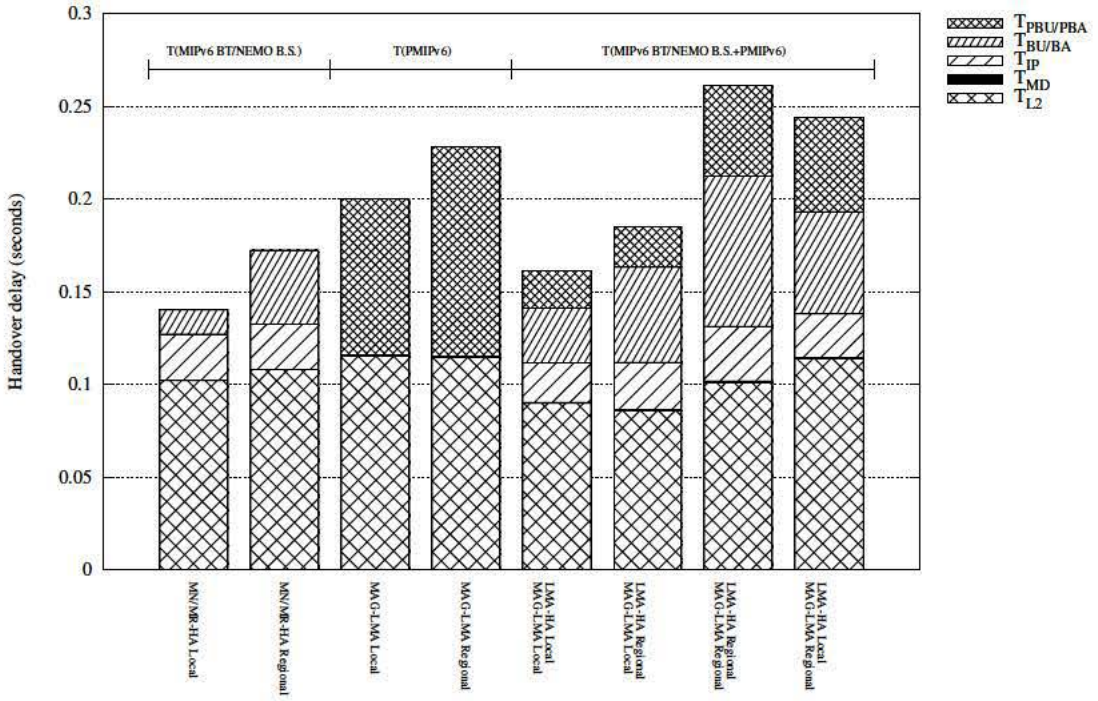


Figure 3.4: Experimental delay analysis

send a Router Solicitation message. As explained in the Section 3.3.2, the IP configuration time (T_{IP}) is only present on the client-based (e.g., MIPv6 or NEMO B.S.) handovers. In network-based solutions (e.g., PMIPv6), this time is only present when the node attaches for the first time to the localized domain, since roaming within the domain does not require a change in the IP address assigned to the terminal nor its forwarding state. The mobility signaling delay ($T_{BU/BA}$ and $T_{PBU/PBA}$) is inherent to each of the mobility solutions. For MIPv6 BT/NEMO B.S., this time is measured as the time elapsed between the transmission of the BU message and the reception of the BA at the mobile node. For PMIPv6 this time does not only include the PBU-PBA exchange between the MAG and the LMA, but also the time required by the MAG to send a Router Advertisement conveying the IP prefix assigned to the mobile node/router in the domain. Finally, the combination of MIPv6 and PMIPv6 incurs the longest total signaling time, as it comprises the time since the transmission of the PBU by the MAG to the reception of the BA by the mobile node.

3.4.2 Experimental Results and Evaluation

Figure 3.4 shows the different components of the total handover delay obtained from our experimental tests. These tests confirm our initial findings about the performance

of the analyzed combinations, and bring some additional interesting conclusions as well. It is remarkable that the main contribution to the overall handover time is the layer-2 handover delay. Our measured average layer-2 handover delay is about 100 ms, although we should highlight that this layer-2 handover has been performed without any further optimization, and therefore a lower delay could be obtained by, for example, appropriately configuring the channels to scan and fine tuning the layer-2 mechanisms involved in the association process.

The movement detection time is confirmed to be negligible, with an average measured time of less than a millisecond. On the other hand, the experimentation has shown that the process of configuration and management of the IP addresses and routes is very time consuming. While these procedures are not present for the case of PMIPv6, the IP configuration time becomes relevant in the case of MIPv6 BT/NEMO B.S. and the combination MIPv6 BT/NEMO B.S+PMIPv6, as the mobile node/router cannot send the Binding Update message until the egress interface is configured properly with the IP addresses assigned by the access router. In fact, our experiments show that the configuration of an IP address in the terminal requires an average of 20 ms,[6] which seems to be an implementation-specific aspect that can be improved.

The signaling time of the mobility protocols strongly depends on the distance between the entities involved in the mobility management (i.e., MAG and LMA, or HA and MR and their combinations). When the distance is local, practically all the handovers require less than 150 ms to be completed. When the delay is regional, the percentage of handovers requiring less than 150 ms decreases. The fact of combining protocols also has an impact, being the distance between the MAG and LMA the most significant factor. A longer delay between these two elements impacts significantly the overall performance of the combination, as reflected by the comparison of the *LMA-HA Local MAG-LMA Local* with the *LMA-HA Local MAG-LMA Regional* cases in Figure 3.4.

If we look at the mobility signaling delay ($T_{PBU/PBA}$ and $T_{BU/BA}$), it is worthwhile mentioning that besides the delay due to the RTT between the involved mobility entities, there is also a non-negligible delay caused by the processing time of the software implementation of the mobility protocols. Note that the implementations used in our experiments are research software, with code not optimized for performance.

This results in a measured signaling time noticeably higher in our experiments than the one considered in our theoretical analysis. One particularly unfortunate case is the PMIPv6 signaling delay resulting from the used implementation, as it requires the LMA to remove the tunnel that was being used before doing any further processing of a received PBU. This is a bug of the implementation that introduces a considerable additional delay.[7]

[6]Note that Duplicate Address Detection (DAD) was disabled in our tests. Had it been enabled, it would have added an average of 1 second to the IP configuration time.

[7]This bug has already been reported to the software developers.

If we compare the handover delays of PMIPv6 and MIPv6 BT/NEMO B.S.+PMIPv6 in Figure 3.4, we can see that the PMIPv6 signaling delay $T_{PBU/PBA}$ of the latter is considerably lower because, in that case, the MN/MR just arrives to the localized domain, so no previous tunnel has to be removed. Based on this, although we have included the results of the PMIPv6 case for completeness, we do not consider them representative of the real performance of the protocol.

If we compare the results of the handover delay of our theoretical analysis in Section 3.3.4 with the experimental results here, we can see that measured values from the conducted tests are higher than the ones resulting from the analytic evaluation. This is due mainly to higher layer-2 handover delays and some processing times not considered in the theoretical analysis (most are implementation-specific, therefore not invalidating our analysis). Nevertheless, our general finding that combining IP mobility protocols tends to increase the overall handover delay is confirmed, as can be seen by comparing the MIPv6 BT/NEMO B.S. results with the MIPv6 BT/NEMO B.S.+PMIPv6 results in Figure 3.4.

3.5 Summary

We have identified, described and analyzed different combinations of IP mobility protocols standardized by the IETF and the functionality that they provide. An important result of the analysis is that, although the combination is needed to be able to obtain a mix of the properties of the different protocols, it also involves a cost, both from the points of view of additional overhead and of handover delay.

Regarding handover delay, combining different mobility solutions has an impact on the performance, even when some of the steps required by each of the mobility protocols – such as movement detection – are shared, resulting in a longer handover interruption. We have also showed that this performance penalty can be significant in certain cases.

The experimental results presented in the previous section raise the need to develop mechanisms to alleviate the costs of combining different mobility solutions, while keeping the advantages brought by the combination. We cannot expect a single mobility solution with all the functionalities to be universally adopted. Instead, we advocate for solutions that include in their design the flexibility to activate and deactivate the use of their supported mobility functionalities (as already supported in a very limited way by the latest 3GPP specifications [99]). For example, it would be interesting to design a mechanism so that the mobile nodes and the network can negotiate about the mobility support functions required in a particular situation, choosing a particular set of functions from those available as required. This implies placing a very active role in the mobile nodes regarding the use of mobility solutions. This is in contradiction with part of the motivation for the network-based mobility approach that has been favored by some operators. However, in fact, the heterogeneity of access networks and the current trend towards environments

with several options for network access from different operators make unavoidable to place more responsibility in the mobile nodes regarding the mobility solutions to use, at least if they want to enjoy an optimized performance adapted to their needs but compatible with network requirements.

We also give an insight of how future mobility protocols can be designed, or previous ones adapted, to facilitate the combination of different mobility protocols. The key properties required are: flexibility to allow the activation and deactivation of mobility functionalities according to terminal and network requirements, the existence of a system so the network and the mobile nodes can exchange the information needed to support that activation and deactivation, and the provision of a default mobility support service for legacy terminals. There is also the possibility of pushing mobility anchors to the network edges, following a DMM approach to decrease mobility signaling delays as the mobility anchors are closer to the MN.

Chapter 4

Connectivity Management in VANETs

4.1 Introduction

Vehicular communications are no longer just a research topic. Governments, car manufacturers and telecommunication players have been working towards the definition of a communication architecture to improve road safety and efficiency that benefits from vehicles having communication capabilities. Among the candidate architectures that could provide connectivity in a vehicular environment, vehicular ad-hoc networks (VANETs) are probably the most popular ones, due to their decentralized nature which supports unmanaged operation without infrastructure involvement.

Over the last decade, considerable effort has been devoted to the analysis of technical solutions that could be applied to the vehicular scenario. This effort has mainly targeted road safety and traffic efficiency services, although Internet alike applications have also been considered recently, given their importance in terms of users' demand. As one very simple example of the impact of vehicular communications in research, a search for the keywords *vehicular communications* in *Google Scholar* returns more than 275k results.[1] However, most of this existing research has not been experimentally validated because of its high cost and the complexity required to deploy and maintain a vehicular testbed.

A VANET-based mechanism published in 2008, called VARON [12], proposes a route optimization for vehicular ad-hoc networks. VARON was validated via extensive simulations and the results proved not only the feasibility of the solution, but also showed that VARON is able to provide interesting performance gains. Many other research works show the same kind of positive feedback in a simulation-only validation, without exploring the issues that potentially arise in a real world experimentation. We believe in system research based on real life implementations, and this drove us to go all the way down to

[1] This result dates from January 2014.

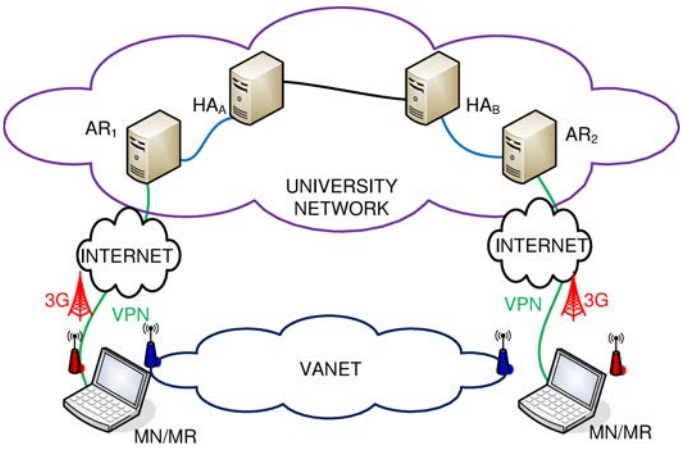


Figure 4.1: Complete network and VARON prototype scenario.

prototyping and testing our solution. We soon found out that many of the problems we faced during the process and the solutions we adopted can also give very helpful insights for other researchers in the area, so we do not focus just on the performance of our particular solution. Therefore, we report on the whole process of taking a conceptual solution, implementing it, first in a closed lab environment, and then deploying it on a real-life testbed.

4.2 From simulation to the lab: Experiments in a controlled environment

In this section we report about the implementation and deployment of a lab-based VARON prototype, as well as the testbed built for its validation and performance assessment.

4.2.1 Network scenario

In order to experimentally validate VARON in the laboratory, the network scenario presented in Figure 4.1 was deployed. During this phase (lab tests), all the network elements were physically deployed in a laboratory at the University Carlos III of Madrid. The implementation of each entity and the description of the main elements that are part of the prototype are described next:

- The MR is the key element in the VARON prototype. In the initial phase of the experiments, this entity was implemented in an Asus WL-500g Premium router, a low cost device equipped with a 266 MHz CPU and 32 MB of RAM. In spite of its limited capabilities, it was initially selected[2] because of its low cost, acceptable

[2]We will show that this decision proved to be wrong.

Table 4.1: HW platform for the different nodes in the network

Device		CPU	RAM	OS version
MR	**Router**	266MHz	32MB	OpenWRT (Kamikaze)
	Laptop	Dual core 1.33GHz	2GB	Ubuntu Linux
Home Agent		Dual core 1.66GHz	2GB	Ubuntu Linux
Access Router		Dual core 1.66GHz	2GB	Ubuntu Linux
MNN		Dual core 1.66GHz	2GB	Ubuntu Linux

performance, the possibility of customizing its firmware according to our needs and, most importantly, the flexibility in terms of potential number of network interfaces. The device is equipped with two USB ports that can be used to increase the number of network interfaces, by connecting, e.g., a 3G USB dongle. Being able to deploy a heterogeneous network is essential for the functionality that VARON aims to offer. In addition, we replaced the original miniPCI wireless network card with one using an Atheros family chipset, which also supports the 802.11a mode.

- Each vehicle behaves as a mobile network managed by its mobile router, and implements the Network Mobility Basic Support (NEMO B.S.) protocol. This requires the deployment of home agents (implemented in two PCs) at their respective home networks, and access routers at each of the visited networks. In a first stage, all the tests were performed indoor in the laboratory and WLAN was used as the technology to connect to the infrastructure (instead of 3G). In a second stage, tests were performed outdoor and 3G was used as access technology. The access routers were physically located at the laboratory and, since we used a commercial 3G IPv4-only network, we set up a Virtual Private Network (VPN)[3] over the public Internet to offer IPv6 connectivity between the access routers and the mobile routers.

- A USB 3G dongle (Huawei E1752C) was used as the additional network interface providing the mobile network access to the infrastructure in order to reach its home network. The MR used the 3G connectivity to reach an IPv6 access router, via a VPN tunnel that hides the traversal of the public Internet and emulates the direct connection (i.e., one-hop distance) between mobile router and access router.

- A netbook was attached as a mobile network node (MNN) to each mobile router. Each of these nodes acquires an IPv6 address belonging to the mobile network prefix (MNP) managed by the MR it is attached to.

[3]OpenVPN: Open source VPN, available at `http://openvpn.net/`

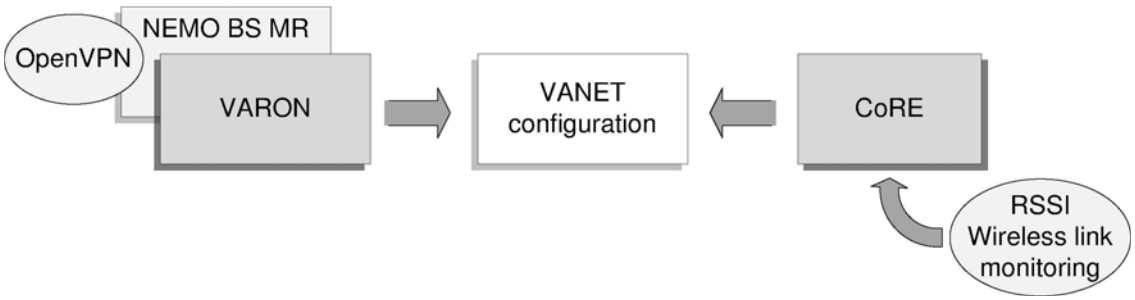


Figure 4.2: Software modules implemented in the mobile router.

4.2.2 Software implementation

This section describes the different software modules of the VARON implementation. The key entity of the prototype is the mobile router, which implements the modules shown in Figure 4.2.

The `NEMO BS MR` module performs the tasks of a mobile router according to the NEMO B.S. protocol: it performs the signaling exchange with the home agent, and configures the bidirectional tunnel used as default data path (home route), keeping the reachability of the mobile network while moving.

The `VARON` module performs all the signaling defined by the protocol, taking care of setting up a care-of route and then establishing a tunnel between the mobile routers involved in the VARON optimization. Once this process is completed, traffic flowing between the mobile nodes managed by the MRs is forwarded through the care-of route, instead of using the default route to the home network through the Internet, with the subsequent delay reduction and bandwidth gain. Both, `VARON` and `NEMO BS MR` modules were developed in C.

Given the dynamic environment of vehicular networks, care-of routes may not last for a long time. The original VARON design specified the use of Care-of Route Error (CoRE) messages to signal when a care-of route should not be used anymore. However, the mechanisms that could trigger CoRE signaling were not fully specified in [12]. This is actually one of the aspects of the VARON design that we wanted to analyze in more detail and improve as part of this experimental work. The `CoRE` module is in charge of monitoring the wireless links used by active care-of routes, in order to avoid packet losses due to insufficient link quality. The `ath5k` wireless driver[4] used in our prototype provides Received Signal Strength Indication (RSSI) measurements of every node within communication range, which can be used as a metric to detect link quality degradation with the next hop of an optimized (care-of) route. VARON original design proposed monitoring layer-2 acknowledgments as a possible mechanism to assess link quality but we discarded this mechanism because in our experiments it did not detect quickly enough

[4] `http://wireless.kernel.org/en/users/Drivers/ath5k/`

Table 4.2: Time needed for establishing VARON optimized route

Care-of route length (hops)	Time [ms]
1	549.6±26.5
2	853.5±61.7
3	1166.5±76.4
4	1434.96±47.7
5	1759.8±63.5
6	2055.2±85.1
7	2377.5 ±97.6
8	2639.4±74.8

the link quality oscillations. We adopted an approach used by other works found in the literature (see Chapter 2): to take the RSSI as an indicator of link quality, although its very oscillating nature makes its use very challenging.

VARON internally sets an RSSI threshold: values below this threshold indicate that the communications are likely to fail. VARON monitors the RSSI using a sample-averaging algorithm to avoid frequent switching between an optimized care-of route and the default home route. The sample-averaging algorithm implemented in our prototype is referred to as *Weighted Mean of 3 Samples* (WM3S) [100], that weights the last three RSSI samples according to the following formula:

$$y[n] = \alpha \cdot x[n-2] + \beta \cdot x[n-1] + \gamma \cdot x[n] \tag{4.1}$$

The three weights[5] are distributed in order to weight more the most recent sample. The RSSI sampling frequency is a configurable parameter to be able to modify it according to the test running and the kind of traffic.

When the value of $y[n]$ resulting from (4.1) computed by a mobile router goes below the configured threshold, a CoRE message is sent to the other endpoint of the VARON tunnel through the Internet. The transmission and the reception of this message by an MR automatically withdraw the optimized route, falling back to the default route over the Internet.

4.2.3 Validation in a controlled environment

The first validation and experimental analysis of VARON was conducted indoor, inside a networking lab of our university. These initial indoor tests focused on assessing the feasibility of the protocol and validating the prototype, as well as analyzing the impact of the length of the care-of route on the performance in a non-mobile scenario.

The first tests involved only two mobile routers, using also IEEE 802.11 as wireless technology to connect to the fixed infrastructure (instead of 3G), increasing the number

[5]In the final VARON experimental evaluation, reported in Section 4.3.3, we used the following values: $\alpha = 0.1$, $\beta = 0.3$ and $\gamma = 0.6$.

of MRs subsequently until reaching a maximum of 9 nodes (i.e., 8 hops). All the nodes were located inside the same room, so there was direct radio connectivity among them. In order to emulate a multi-hop scenario, `ip6tables`[6] was used to limit the reachability of the nodes at the IP layer.

Table 4.2 shows the time needed to complete the establishment of the care-of route for different number of hops. Most of that time is spent by the mobile router performing cryptographic operations (using a key size of 512 bits), essential for security enforcement but very time consuming for a device with limited capabilities such as the nodes we used in these tests. Note that during the optimized route setup time, the data connection is never interrupted, as traffic is forwarded using the home route, always available.

4.3 From the lab into reality: On-road experimentation

In the previous sections we have reported on the initial evaluation efforts of VARON, which started with an experimental validation in a controlled environment, allowing us to check the correctness of the protocol and serving as a proof of concept of VARON. In many vehicular research works, the experimental analysis stops here, i.e., solutions do not *leave* the simulator or a controlled-environment laboratory. We aimed to go beyond this, by proving the feasibility of the mechanisms proposed by VARON under realistic conditions. This section describes the deployment of the vehicular network prototype, the experiments conducted, as well as the main results and the lessons learned from this experience.

4.3.1 Scenario and testbed deployment

Starting from the VARON prototype evaluated in the laboratory, we followed an incremental approach to improve and further develop the prototype while validating it under real life conditions.[7]

Commercial off-the-shelf (COTS) devices are very convenient in terms of flexibility, cost and size, but they present limitations due to its processing capabilities and storage capacity, which affect the performance of the prototype. For instance, as experienced during our trials, the computing load needed for handling simultaneously the several interfaces, the VPN connection and the cryptographic operations added significant delay in the process. Moreover, the addition of the 3G USB interface noticeably lowered the performance of the COTS device. This led us to upgrade the hardware of mobile routers, wireless cards and antennae. We replaced the router in the field trials phase with a laptop

[6] `ip6tables` is a Linux kernel tool that let the user examine and configure the tables of IPv6 packet filter rules.

[7] Some of the first changes aimed to experiment with real vehicles were first tested in our lab, as for example the use of 3G and the VPN setup. This saved us from spending valuable time in the field.

Figure 4.3: A snapshot of the experimental set up.

capable of easily handling all the processes that run in parallel in the mobile router. The capabilities of the hardware finally used are comparable (or even lower) to the ones of the future in-vehicle deployed communication devices. The laptop was equipped with an 802.11a/b/g Ubiquiti SRC wireless card with an MMCX plug for an external antenna at the outdoor testing phase. At the operating frequency and the transmission data rate, the sensitivity of the wireless card is $-94dBm \pm 1dB$.[8] Additionally, the communication range of the WLAN link has been enlarged by attaching an omnidirectional antenna working in the range of 5 GHz.

The first round of experiments involved two vehicles, and was aimed at validating the VARON route optimization and evaluating the feasibility of inter-technology handovers. On a second round, we added another vehicle in order to assess the multi-hop route optimization and estimate the suitability of the multi-hop mechanism.

Each of the vehicles (see Figure 4.3) was fully equipped with a mobile router (a Linux-based laptop) installed on the car's roof, next to the antenna, while the mobile network node (a Linux-based netbook), connected to the MR by Ethernet, is placed inside the car. Both devices are powered by means of an AC/DC adapter, although the battery in the netbooks can easily last longer than our experiments.

The selection of a proper scenario for the test drives is an important decision. We chose the Leganes scientific cluster[9] because of two main reasons: *i)* it presented a relatively low traffic load (to prevent us from obstructing other vehicles), and *ii)* it had a regular street design, with roundabouts and straight stretches, which is a key point for the repeatability of the experiments.

From the very beginning, the design of the experiments took into account repeatability as one critical requirement. Figure 4.4 shows the starting position of each vehicle and the

[8] `http://dl.ubnt.com/src_datasheet.pdf`
[9] `http://www.leganestecnologico.es/`

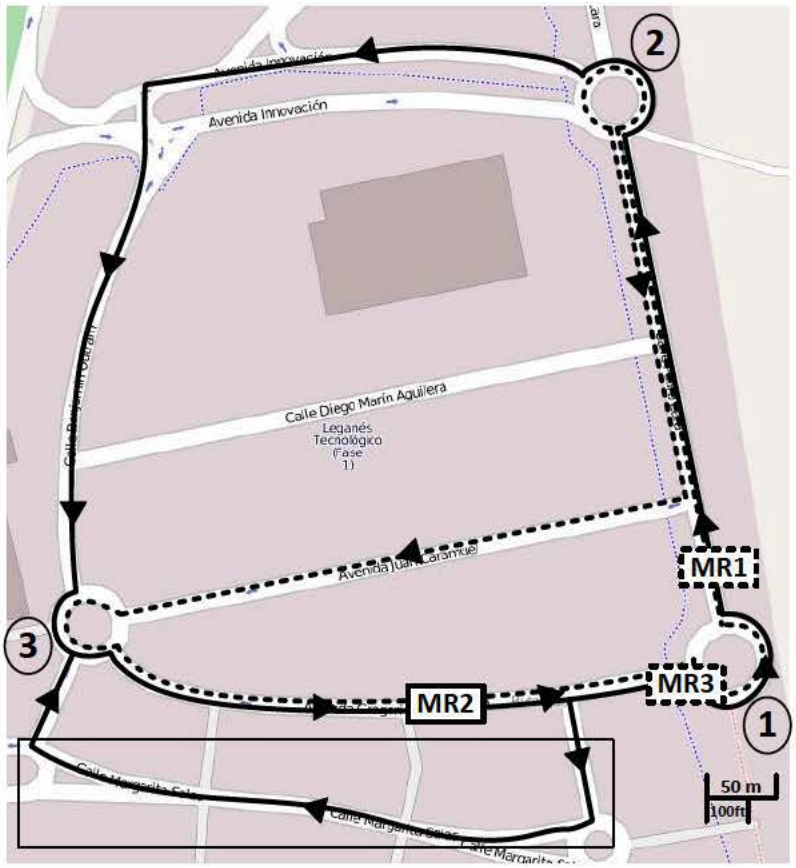


Figure 4.4: Test itinerary followed in the Leganes scientific cluster.

itinerary they followed in the 2-node and 3-node tests.[10] Each testing round consisted of the following steps:

1. The initial position of the vehicles was such that no direct MR-to-MR connectivity is possible.

2. One vehicle (from now on referred to as MR2) started getting closer to the other vehicle, MR1.

3. When they were close enough to enable WLAN communication, they started the VARON route optimization signaling, which resulted in setting up a care-of route if the process ends successfully. Otherwise, the optimization process was aborted, waiting for a new optimization opportunity. In the case of the 3-vehicle scenario, MR2 initially followed the path enclosed in a box at the bottom of Figure 4.4. When the link between MR2 and MR3 was broken, the route fell back to the 3G connection, until MR2, which continued approaching by the same initial street, was close enough to enable WLAN communication again.

[10] In the 3-node test, MR1 and MR3 followed the same path. MR2 also followed the path enclosed by the box at the bottom in the 3-node scenario.

4. Vehicles moved forward together until reaching an intersection (roundabout 2), where they took separate paths, compromising wireless link quality and thus forcing a handover back to the cellular connection at some point in time, triggered by a CoRE message.

5. Vehicles continued on their corresponding designated paths, which led them to meet each other at roundabout 3, enabling the establishment of a care-of route again.

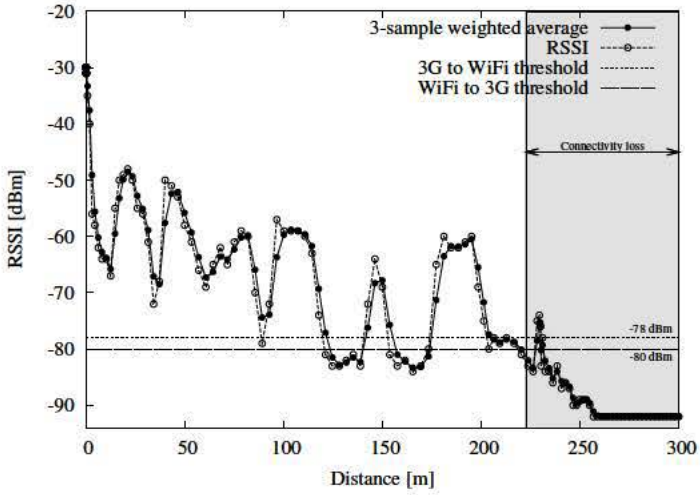
6. They continued towards their initial positions, inducing a second handover to the cellular default route.

Therefore, a complete round accounted for two inter-technology handovers from the cellular network to the WLAN and another two from the WLAN to the cellular network. This allowed VARON to recover from an optimization and to confirm that both MRs are able to start a new route optimization, if required. Note that to trigger the inter-technology handover the vehicles just followed their paths, changing the distance or introducing obstacles between them, allowing for the link conditions to change. Each test was repeated a minimum of 20 times, both for the single-hop (2-node) and two-hop (3-node) scenarios. The average speed of the vehicles during these tests was 50 km/h.

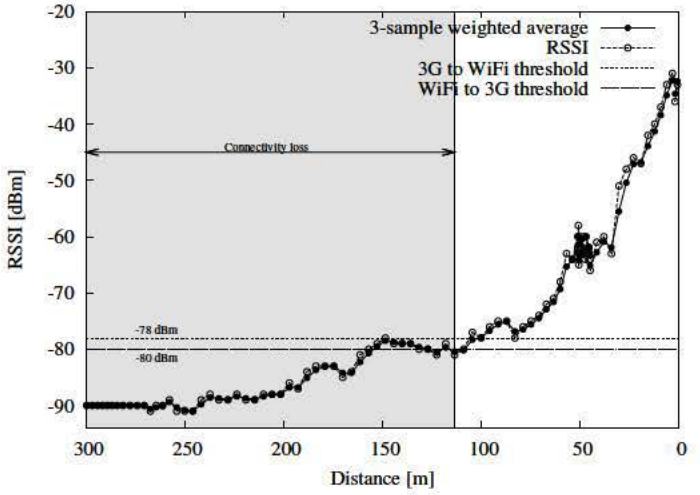
To detect when to switch from the care-of route to the home route, VARON considers the link quality metric provided by the RSSI. As mentioned before, the use of RSSI for this purpose is challenging, due to the dynamism of VANETs. In order to fine tune our algorithm, we first analyzed how the WLAN RSSI and the connectivity varied with the distance between nodes. We selected the thresholds for triggering the handover from the WLAN to 3G based on these results, which are shown in Figure 4.5. The gray areas denote lack of WLAN connectivity due to packet losses and errors. Then, we need to anticipate those losses by sending a CoRE message to the two communication ends, as explained in [12] and in Section 2.4. Obtained results provide some interesting insights: *i)* the degradation of the link quality due to the increasing distance between nodes is evident, but *ii)* distance is not the only determinant factor. For instance, in light of the differences observed in the measurements when the nodes are moving farther or closer to each other, we can claim the importance of their relative positions, the location of the antennae (i.e., their alignment depends on the actual shape of the car) and the multi-path reception.

4.3.2 Early testing and implementation feedback

As we will describe next, to fully deploy VARON in a real scenario we had to adjust and improve the original VARON design in order to tackle the different issues we found in the process, which relate to the common VANET mechanisms identified in Chapter 2.



(a)



(b)

Figure 4.5: Experimental RSSI degradation due to distance between nodes.

4.3.2.1 Effectiveness of the limited flooding

VARON implements a limited flooding mechanism for announcing the presence of a mobile network prefix in the VANET. Every mobile router is aware of the prefixes that are available within a configured advertisement scope and can decide whether to start a route optimization process. These announcements, named HoAA or Home Address Announcement, are broadcast and forwarded by every node until the hop limit counter allows so (this defines the advertisement scope). Before forwarding it, every mobile router waits for a uniformly distributed random time, in order to reduce the collision probability and increase the effectiveness of the mechanism. In addition, in order to refresh the information, these announcements are transmitted periodically. Therefore, the transmission interval needs to be chosen carefully. The reception of these messages trigger the route

optimization process, so on the one hand, a long interval would reduce the number of opportunities as the presence of some mobile networks may go unnoticed. On the other hand, the interval cannot be set too short neither, not only to avoid a broadcast storm, but also to keep the wireless links between nodes stable. In our prototype this transmission interval is configurable, to allow quick changes in the field if necessary, and we conducted tests with a value of 5 and 10 seconds. Experiments with a value of 5 seconds showed a lower discovery rate, there were more errors in the reception of the messages and if two vehicles were approaching from a zone where they could not reach each other, there was not enough time for the wireless link to provide favorable conditions. The experiments with a 10 s interval showed a higher success ratio, as there were less collisions, the wireless link between the nodes involved had enough time to stabilize and discovery happened within the time that the two vehicles were likely to hear each other. The use of transmission intervals longer than 10 seconds would be inefficient, as the network conditions could dramatically change between transmissions, leading to much less stable VANET connections.

4.3.2.2 Use of timers to ensure optimization state consistency

The wireless medium in the VANET may fail during the route optimization process, as the radio link quality depends on many factors and does not remain exactly the same under any given conditions. As a consequence, VARON messages may be lost or delayed during the care-of route setup. In order to ensure consistency in the VARON state machine and avoid stale states, a modification of the original VARON design was introduced, consisting of the *use of timers to deprecate an ongoing optimization attempt.*

In order to make more clear the use of timers, we consider next a VARON specific example. With VARON, a mobile router may complete all the optimization steps on its side, while the other end does not, for example because the last signaling message to complete the process (an MNPBU) is not received. It is not possible to know whether the last MNPBU was lost or was not even sent (because the first MNPBU is the one actually lost). Therefore, a timer is set when an originator mobile router sends the first MNPBU. If no reply-MNPBU is received before this timeout expires, then the originator mobile router sends a CoRE message to the target MR, to ensure that the state is consistent (no optimization is in place) at both mobile routers.

4.3.2.3 Is the cellular connectivity that reliable? The need for the CoRE ACK

Cellular connectivity is often assumed to be reliable and to offer full coverage. However, this assumption proved out not to be valid according to our experience in the field. Measured 3G connectivity was variable and unstable, delivering data in bursts very often.

Additionally, the available bandwidth was very much dependent on the location and, in general, lower than claimed by mobile operators, exhibiting also unacceptable and variable delays. By default, VARON routes through the cellular network some signaling messages, such as the CoRE message that withdraws the use of an optimized (care-of) route when it does not meet the minimum required quality. However, our on-field experiments showed that 3G connectivity was not as reliable as we would expect from a commercial service, and that sometimes this small signaling message (CoRE) is delayed for a long time or even lost.

This finding made us introduce another modification to the original VARON design: the need for *acknowledging CoRE messages*. When a mobile router receives a CoRE, it has to send back a CoRE ACK message. If the acknowledgment is not received within a pre-configured time window, a new CoRE message is sent, to ensure that the other mobile router receives it, and therefore prevent it from using the care-of route anymore. Note that without the CoRE ACK, if the original CoRE message is lost, the resulting routes between the two mobile routers involved would end up being asymmetric – one would use the default home route, while the other one would still use the optimized, but likely non-working, care-of route.

4.3.2.4 Avoiding ping-pong optimization effects and short-lived care-of routes: the use of radio link quality thresholds

In the original VARON design, an optimization attempt was initiated as soon as a mobile router detected that an optimization to a prefix used by an ongoing communication was possible. This basically meant that a CoRTI message was sent upon reception of a HoAA matching a prefix with which there was already an ongoing traffic exchange. However, our experiments showed that it was possible that this HoAA message was received under bad link quality conditions that could not guarantee a successful completion of the route optimization process (for instance, due to the distance between vehicles or the asymmetry of the wireless links). Note also that HoAA messages are broadcast, which has a twofold effect: first, there are no link layer re-transmission mechanisms in place to ensure their reception and secondly, the Modulation and Coding Scheme (MCS) has to be set to the lowest of the Basic Service Set, which limits the distance range covered by the transmission.

Another modification to the original VARON design was introduced to deal with this issue and improve the rate of successful route optimizations: the use of *radio link quality thresholds*, based on the RSSI of the different received signaling messages. A mobile router would not send a CoRTI message if the signal strength towards the transmitter of the corresponding HoAA is not higher than a pre-configured threshold. Analogously, a mobile router would not send a CoRT message if the signal strength towards the transmitter of the corresponding CoRTI message is below this threshold.

Early field test showed that some route optimizations had a very short duration, meaning that although they were successfully established, a CoRE message was sent soon after due to poor link conditions. Our experiments showed that if a certain average link quality is not met during the optimization procedure, then the signal quality is very likely to quickly drop below the threshold, triggering the transmission of a CoRE message, and therefore withdrawing the optimization. In order to further reduce the likeliness of setting up a short-lived optimized route, we introduced the following mechanism. The mobile router keeps monitoring the RSSI from the transmission of the CoRTI message, which initiates an optimization attempt, until the moment of sending the MNPBU, which completes it. If the average signal strength measured in all the signaling messages received from the mobile router used to reach the target MR is not above the previously defined threshold, the route optimization process is aborted.

4.3.3 Experimental results

In the previous section we have described different improvements to the original VARON design triggered by our implementation experience. We next focus on the actual evaluation of the VARON protocol performance, by presenting the main results and measurements collected during our experiments. While doing so, and although we refer to a particular solution (VARON), the obtained results can also be used to evaluate the suitability of some of the most common networking mechanisms for VANETs, such as multi-hop routing, message signaling flooding or access technology heterogeneity.

Table 4.3 summarizes the results for the most representative experiments. A first experiment characterizes the VARON signaling delay, which is the time required to send and process all the VARON protocol messages, set up the required tunnels and forwarding, and configure a care-of route. We performed this experiment both in the lab and outdoors, using 3G connectivity and different hardware choices for the mobile router: the Asus WL500g Premium router and a regular laptop. The results show that the limited resources of the COTS devices impact the route establishment delay, which is considerably shorter when the laptop is used.

We highlight that the time required to set up a care-of route does not directly impact the user experience, as traffic is routed normally via the default (home) route while the VARON optimization signaling is taking place. On the other hand, the time required to withdraw an optimized route does have an impact, as the quality of a link, part of the optimized path, might become so poor that the path is unusable, and therefore it is critical to anticipate this event and switch to the default route. We measured this "withdrawal time" as the time elapsed since the mobile router detects a link quality degradation triggering the transmission of a CoRE message, until the moment a CoRE ACK is received. Note that since the CoRE and CoRE ACK messages are sent via the 3G interface, the care-of route withdrawal latency is mainly caused by the delay over the 3G

Table 4.3: Experimental results

Parameter	Care-of route length (hops)	
	1	**2**
VARON signaling delay [ms] Indoor, COTS router as MR	908 ±91	1450 ±199
VARON signaling delay [ms] Outdoor, COTS router as MR	1210 ±590	1750 ±850
VARON signaling delay [ms] Outdoor, laptop as MR	550 ±74	514 ±39
Care-of route withdrawal time [ms]	524±233	
Successful optimization attempts	14.53%	10.84%
Discarded HoAA	82.90%	55.96%
Aborted optimization attempts	2.56%	13.64%
VARON efficiency	79.3%	60.7%

network, independently of the number of hops in the care-of route. Figure 4.6 shows the box-and-whisker plot for these measurements, representing minimum, maximum, median and first and third quantiles for the care-of route set up and withdrawal latencies, for both the single-hop and two-hop scenarios.

Another relevant performance metric is the efficiency of VARON, defined as the amount of time that a vehicle takes advantage of VANET communication instead of using the default cellular connection. Connectivity in the VANET making use of IEEE 802.11 technologies depends on the speed of the vehicles, their relative positions or their path, among other variables. In order to provide a measurement that is not biased by these factors, we set up IP background traffic over pre-configured routes through the VANET between the two mobile routers involved in the optimization. Note that the static pre-configured routing uses an addressing space independent of the one used by VARON, to avoid any impact on the experiments. We measured the time that an optimized route is in use over the total amount of time that connectivity in the VANET is possible. This experiment also served to confirm the effectiveness of the RSSI threshold that triggers the switch from the home route to the care-of route. Our tests reflect that the route established by VARON took advantage of the VANET connectivity on average on a 79.3% out the total time with VANET connectivity available for the single-hop scenario, and on a 60.7% for the two-hop scenario.

Table 4.3 also shows the number of successfully completed VARON optimizations, which is quite low because many received HoAAs were simply discarded due to its low associated RSSI (note the high percentage of discarded HoAAs). An optimization process may not be completed because the radio link conditions are not good enough, not only at the reception of a HoAA, but also during the whole optimization procedure. It is remark-

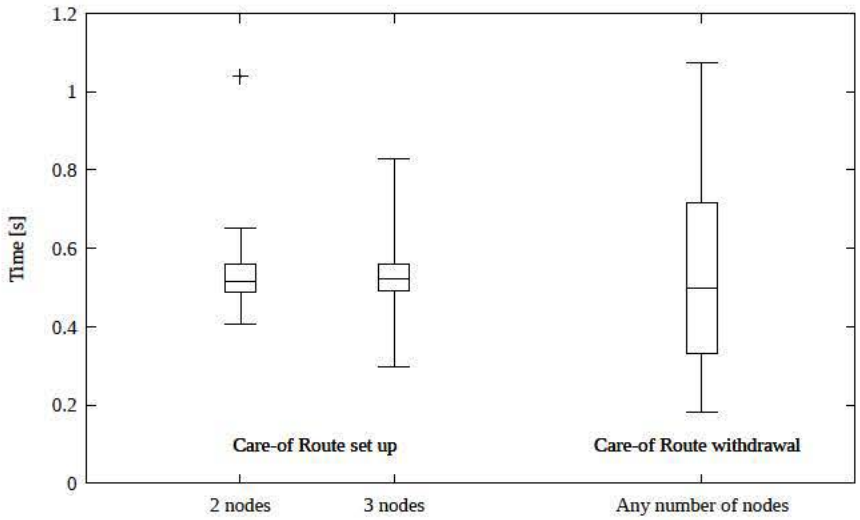


Figure 4.6: Care-of route set up and withdrawal latencies.

able that even with this low percentage of completed optimization attempts, VARON achieved a quite high efficiency (around 60-70%). This is actually possible thanks to the use of the RSSI thresholds (not devised in the original design), which avoids attempting to establish a care-of route on a poor-quality wireless (multi-hop) link. This improvement also contributes to saving useful resources in terms of signaling overhead and energy consumption.

In order to better illustrate how VARON signaling benefits from the use of RSSI thresholds to provide a seamless user experience, Figure 4.7 provides a snapshot of a complete VARON experiment. The sequence number of user generated traffic is shown in the figure, together with the RSSI measured by the originator mobile router. At the beginning (left side of the figure), the communication between two mobile networks is using the home (default) route via the 3G network. At some point in time (around $t = 20s$), a mobile router decides that a route optimization was possible, based on the reception of a HoAA message with good RSSI. This triggered the VARON signaling sequence on the mobile router, which was successfully completed with an MNPBU message (as shown in the bottom close-up, this process took around $0.5s$). There was no perceptible interruption in the IP packet flow, as the traffic was being forwarded via the home route until the care-of route has been established. The optimized care-of route was used for more than $70s$, until the measured RSSI dropped below the configured threshold, triggering the CoRE - CoRE ACK signaling, which made the mobile routers revert to the home default route. This procedure took also less than $0.5s$, as shown in the upper close-up in the figure. Like in the previous care-of route setup procedure, there was no perceptible impact on the IP packet flow.

This experimental performance assessment shows that VARON provides interesting advantages by enabling the use of the VANET instead of the more expensive, and not so

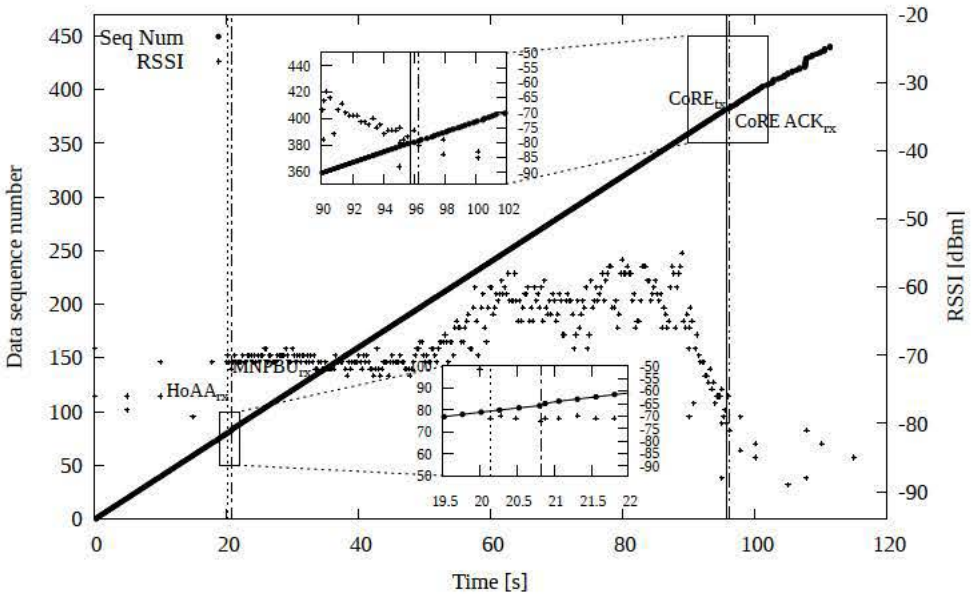


Figure 4.7: VARON route optimization signaling and data sequence number.

reliable some times, cellular network. It was critical, though, to revise our initial design of the protocol, based on our findings during the initial development and testing of VARON.

4.4 Summary of lessons learned

This section enumerates the most important lessons learned from the implementation, deployment and evaluation of VARON. The exercise of bringing a vehicular communication protocol into reality has enriched our knowledge, directing our attention to very specific issues that may go unnoticed otherwise. We think that these findings are very relevant for the design of successful vehicular networking solutions. We categorize the lessons learned into two groups: those related to the actual deployment of a testbed and a vehicular prototype, and those specifically related to the vehicular networking protocol.

4.4.1 Testbed deployment and vehicular prototype

4.4.1.1 When and where tests should be performed

The location and time for the tests are key elements in the deployment of a real vehicular prototype. The location has to be selected taking into account the safety of the people conducting the tests and the people on the road. As repeatability of the tests is critical in order to achieve statistically meaningful results, the test vehicles may not follow exactly the same behavior of a regular vehicle. Similarly, the time of the day has to be carefully chosen, so the impact of undesired external conditions is minimized. Some of these conditions might not be under the control of the people doing the experiments,

as for example the quality of the 3G connectivity, which in our case experienced non negligible variations throughout the day.

4.4.1.2 Selection of the proper hardware

COTS devices are often chosen to deploy test prototypes because they are conveniently handy and inexpensive. However, due to their limited capabilities, they may not be appropriate to every experimental deployment. In our case, COTS devices turned out to be very convenient for the development and preliminary testing phases, but not for the tests on the road (see Section 4.3.1). Moreover, while running tests on the field, it is essential to be able to debug and reset the software and hardware easily, since resources are limited and time on the field is a very valuable resource. Therefore, it is advisable not to try to buy cheap hardware, but rather select the equipment that better resembles the final product, while providing facilities for testing and debugging.

4.4.1.3 WLAN driver specifics

The design of link layer mechanisms has to take into account the specific hardware and WLAN driver behavior. For instance, the different wireless drivers show different results under the same testing conditions. Node discovery and synchronization in ad-hoc mode are specially critical. In our experiments, we adapted our implementation to two different wireless driver implementations, namely *madwifi* and *ath5k* as we noticed that the support of ad-hoc mode is quite different from one with respect to the other.

4.4.1.4 Power supply

When performing field trials, a long-live and reliable power source might not be available. This has to be taken into account when selecting the hardware, as not all the possible solutions are equally convenient. In our tests, we used both battery-powered Uninterruptible Power Supply (UPS) units and AC/DC inverters plugged to the car to provide power to the nodes. The use of netbooks, which usually present high battery life, is also very convenient.

4.4.2 Vehicular networking protocol

4.4.2.1 Multi-hop vs single-hop in the vehicular environment

Vehicular communication protocols may use single- or multi-hop communications. This actually makes a significant difference, because in addition to the obvious routing considerations, there are practical implications. For example, vehicular protocols quite often rely on broadcast or flooding mechanisms, which do not perform in practice as expected from simulation results (our experiments show low delivery rates of broadcast

packets). Special care should be taken when designing a vehicular communication protocol to ensure that signaling messages meant to traverse a multi-hop network actually reach their destination, and if that is not the case, that there are mechanisms in place to detect it and react accordingly.

Additionally, the implementation of the wireless ad-hoc mode is typically much less developed and debugged as compared to the infrastructure one, which is more widely used. Besides, the differences in ad-hoc mode support and capabilities among the different available hardware and drivers are more disparate than for the case of infrastructure support.

4.4.2.2 Higher dynamism

Many vehicular protocols re-use concepts and solutions from other multi-hop ad-hoc communication scenarios. As already highlighted, the vehicular environment is very dynamic and suffers from severe radio conditions. This requires a more careful design of the VANET protocols, especially in terms of robustness and redundancy of the signaling. As an example, the reachability of a certain node via a multi-hop VANET route at a given moment does not guarantee that this connectivity will still be there some time after it was tested. Therefore, additional mechanisms should be designed to continuously evaluate this connectivity and detect potential disruptions or situations that indicate a high probability of imminent disconnection.

4.4.2.3 Cellular networks performance

3G networks are commonly assumed to provide always-on connectivity. Even some proposals make use of 3G networks for the delivery of critical safety messages, as compared to the use of the VANET. However, our experiments revealed that 3G may not always be as reliable and stable as expected, but the opposite: measured delay showed great variability, bandwidth fluctuated and was often low and there were non-negligible packet loses. This behavior of course depends on the mobile operator and the location, but we believe that roads are particularly prone to suffer from this because operators do not dimension their networks with the goal of providing full 3G data coverage in roads yet. This is likely to change in the future, once mobile data access from vehicles becomes more popular and also when the new cellular technologies, such as LTE, get deployed.

4.5 Summary

Vehicular communications have been extensively researched in the past, with a plethora of different solutions being proposed. However, there is still a lack of experimentation and deployment experience, with some remarkable exceptions of large scale

testbed efforts.

With this work, we have tried to fulfill two ambitious goals: *i)* to experimentally validate and evaluate a vehicular communication protocol, VARON, which was initially proposed and extensively simulated in 2008; and *ii)* to report on the many insights and lessons that we have learned throughout the process of fully implementing VARON and deploying it in a real scenario. The validation and experimentation analysis of VARON was conducted incrementally, starting from experiments performed in a controlled laboratory environment, and then moving to tests performed with up to 3 vehicles. While this is a reduced number of vehicles, we argue that the obtained results, and more importantly, the knowledge acquired from prototyping VARON in this scenario, would also apply to larger testbeds, as the issues we encountered have to be faced by each communicating vehicle individually. In terms of performance, VARON has shown to be feasible, enabling the use of opportunistically set-up VANET routes between two communicating vehicles, that would have to make use of a cellular network connection otherwise.

While prototyping and experimenting VARON in real vehicles, we had to face and tackle several challenging issues, resulting in some modifications and enhancements to the original VARON design.

Chapter 5

Integrating Optical Broadband Networks and Mobility

5.1 Introduction

In this Chapter we propose an integrated mobility architecture for converged optical-wireless architectures, particularly for WOBANs, based on PMIPv6 and the handover optimizations enabled by the use of the IEEE 802.21 MIH Services. Essentially, the architecture locates the MAGs defined in PMIPv6 at the leaf nodes of the PON, that is, the Optical Network Units (ONUs), and the LMA at the root, the Optical Line Terminal (OLT). This architecture facilitates the optical-wireless integration for several reasons: First, the OLT-LMA module is able to combine traffic statistics of the ONU-MAGs and mobility information of the end MNs connected to every MAG. This might be useful to redefine the Dynamic Bandwidth Allocation (DBA) strategies based on the MNs attached to the ONUs, rather than on the ONUs solely. Second, the mobility procedures of PMIPv6 may be optimized thanks to the single-IP-hop, point-to-multipoint topology of the PON, which provides multicast services at no extra bandwidth cost. And third, the OLT-LMA leverages the static location of the ONUs to infer possible destinations of Mobile Nodes, and to design an effective procedure to reduce packet loss during the handover process. The use of MIH in the PMIPv6-WOBAN architecture proposed is twofold: On the one hand, the link-layer event support provided by the MIES is employed to trigger certain actions at the MAG, for example sending a Proxy Binding Update upon MN attachment to a connected Access Point. On the other hand, the handover optimization signaling is used to piggyback information that enables the WOBAN mobility optimizations described in this Chapter.

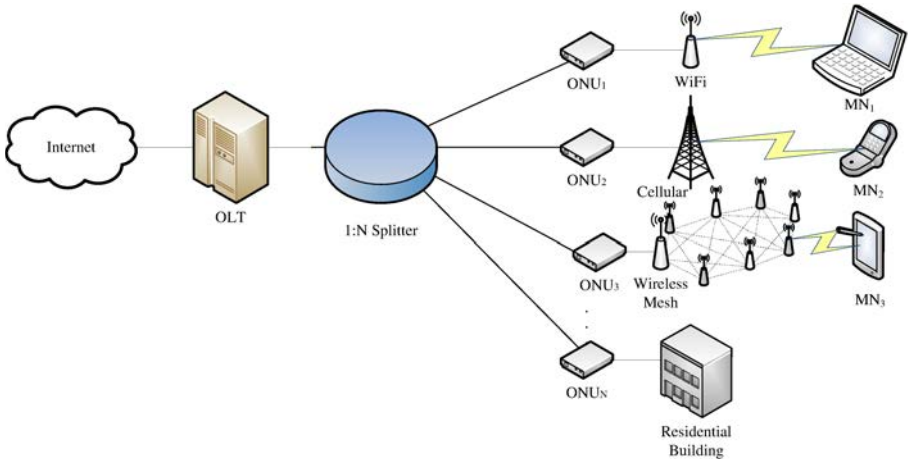


Figure 5.1: Ethernet Passive Optical Network (EPON) architecture

5.2 Integrated PMIPv6-WOBAN architecture

Figure 5.2 (Top) shows the proposed PMIPv6-WOBAN integrated architecture, together with the signaling (left boxes) and data (right boxes) paths. For simplicity, we show three ONUs, two of them connected to a single WiFi Access Point, and another one connected to three Access Points. Note that the LMD comprises the whole radio coverage of the five APs of the WOBAN.

In the following, we consider that the PON is terminated either on single wireless Access Points or any other wireless layer-2 cloud (for instance, wireless mesh topologies), as long as these behave as a single IP hop. It is worth emphasizing that the PMIPv6-WOBAN architecture requires that any wireless access network connected to an ONU behaves as a single layer-2 domain. In such a case, there is no difference between an isolated AP connected to the MAG or a group of APs forming a mesh, as long as they operate as a single IP hop. The interested reader is referred to the IEEE 802.11s amendment, which details how to deploy a layer-2 wireless mesh.

An EPON exhibits a hierarchical architecture where the OLT dynamically assigns time-slots to the ONUs following a DBA algorithm, for instance IPACT. This structure resembles the one of a PMIPv6-enabled LMD, in which the LMA anchors the IP addresses assigned to the mobile nodes that are attached to the MAGs, which are likewise controlled by the LMA. Based on this observation, we propose an integrated PMIPv6-WOBAN architecture, where the LMA is collocated with the OLT, and the MAGs are collocated with the ONUs (one MAG per ONU). From an MIH functionality perspective, we consider that the PoS resides at the OLT, while all ONUs are also IEEE 802.21 capable, that is, they implement an MIHF acting as non-PoS MIH entities.

It should be noted that, although WiFi has been assumed as the access technology throughout the previous description of the proposed PMIPv6-WOBAN architecture, other

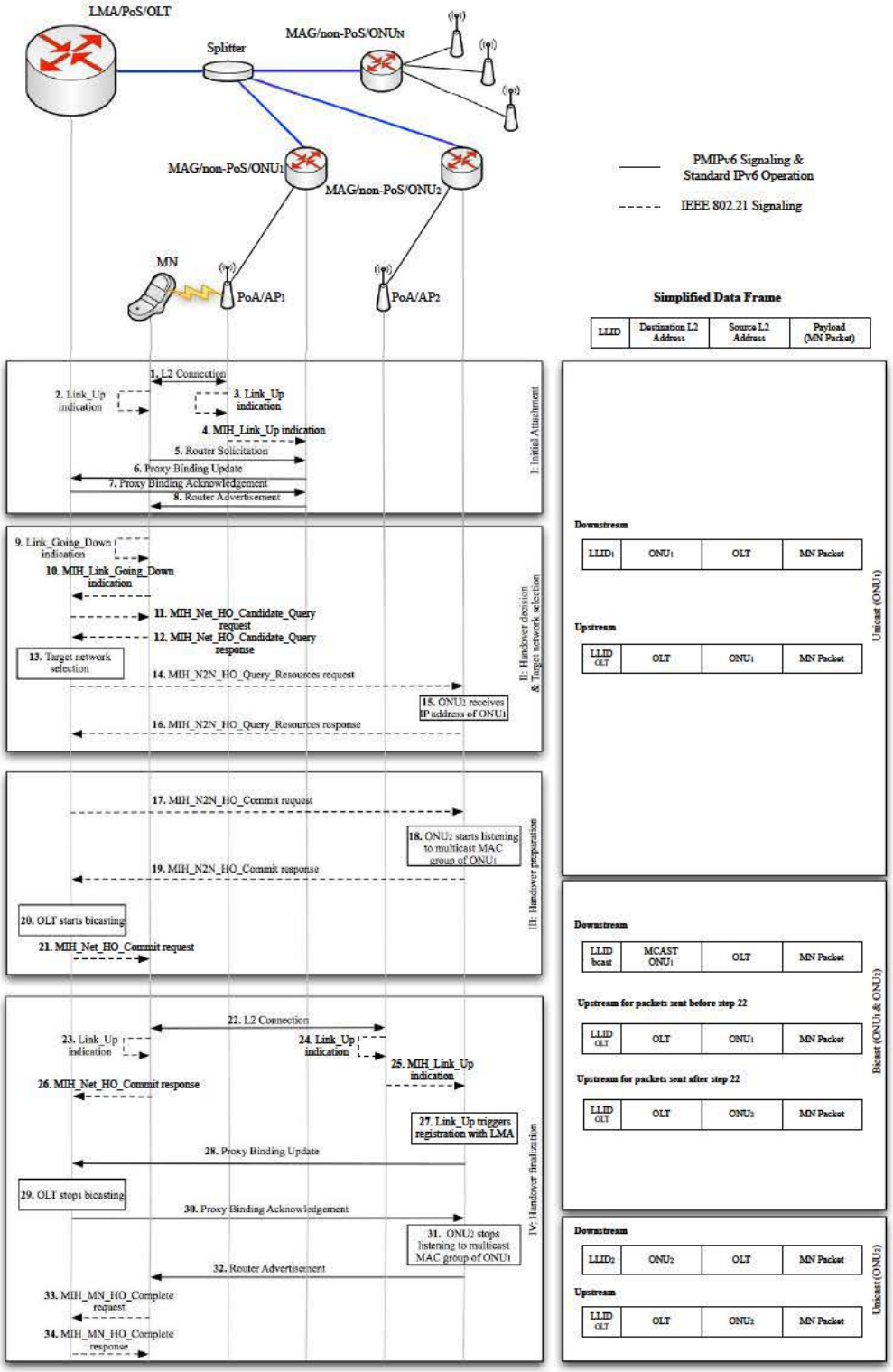


Figure 5.2: End-user mobility inside PMIPv6-WOBAN. Signaling on the left boxes, and simplified data frames on the right hand side

technologies are also supported, as long as the mobile node supports IPv6 and IEEE 802.21. Both protocols are defined to operate over different wireless technologies, such as WiMAX and 3GPP ones, among others. Since ONU-MAG devices behave as IPv6 routers, their interfaces may employ completely different layer-2 technologies, including wireless ones. If the wireless attachment point is not co-located with the ONU but it is a remote device connected by Ethernet to the ONU-MAG, then the wireless technology should support IEEE 802.1D bridging capabilities (so wireless frames could be sent back and forth as Ethernet frames, as well as to support intra-ONU mobility) or the communication between the MAG and the MN should be able to behave as a point-to-point link as stated by PMIPv6. As an example, interworking with 3GPP cellular networks is enabled by the use of PMIPv6, which is one of the solutions adopted by the 3GPP Evolved Packet System (EPS) to provide inter-access mobility support between 3GPP and non-3GPP access networks.

The proposed integrated PMIPv6-WOBAN architecture enables a set of optimizations that enhance the operation of each protocol with the intrinsic characteristics of the other. Next we present these optimizations and describe the architecture by detailing the different mobility-related operations.

5.2.1 Initialization and Mobile Node attachment

The first box of Figure 5.2 describes the MIH events (dashed arrows) and PMIPv6 messages (solid arrows) exchanged between the mobility management modules when a new MN first arrives at the WOBAN. These steps are:

(1-5) The MN initially attaches to an AP following standard technology-dependent procedures (in this case following the association procedure defined by IEEE 802.11). The finalization of the layer-2 attachment generates a `Link_Up` event at the MN[1], which sends a `Router Solicitation` message that reaches the access router – i.e. the MAG. In parallel a `Link_Up` event is also generated at the PoA (AP_1), which is the corresponding end point of the layer-2 connection. This event is propagated through an `MIH_Link_Up indication` to the MAG, triggering the registration to the LMA.

(6) This message triggers the MAG at the ONU to send a `Proxy Binding Update` to the LMA at the OLT, requesting for an IPv6 prefix for this new MN.

(7) The LMA looks up its Binding Cache and creates a new entry for this new MN. The LMA delegates an IPv6 prefix to the MN, conveyed back to the MAG in a `Proxy Binding Acknowledgement` message.

[1]Following the IEEE 802.21, the `Link_Up` event is generated by the wireless driver when a layer-2 connection is established on that particular link interface.

(8) The MAG then sends a `Router Advertisement` to inform the MN about the assigned prefix, allowing the MN to configure a valid IPv6 address using standard Stateless Address Autoconfiguration (SLAAC) techniques.

After this process is completed, the MN has network connectivity, and its traffic should be encapsulated on an IPv6-in-IPv6 tunnel from/to the MAG to/from the LMA. Such a tunnel between the MAG and the LMA yields us to the first optimization proposed in the integrated PMIPv6-WOBAN architecture.

Optimization no. 1: No need for IPv6 tunneling. In this case, there is no need for an IP-in-IP tunnel between the MAG and the LMA since the connection is one-hop distant. Essentially, all traffic sent by the OLT is received by all ONUs, which just filter out EPON frames destined to other nodes based on the LLID and destination MAC address. In the uplink direction, the same reasoning applies since the traffic sent by a given ONU arrives only at the OLT (never at the other ONUs). This benefit leverages the point-to-multipoint topology of the EPON and applies to the communication between the LMA-MAGs, thus reducing the overhead of the communication.

5.2.2 Handover operation in the integrated architecture

Now, consider the previous MN moves to a new AP, which triggers a handover in the mobile network. Such an action may occur due to mobility reasons (the MN is moving away of the radio coverage of the AP) or due to network reasons (the current ONU/AP is overloaded and the network decides to move some users to a neighboring ONU/AP). Then, the IEEE 802.21 MIH framework is used to enhance the handoff performance by making it proactive (the so-called *make-before-break* approach). The following procedure assumes that the handover decision and target selection is performed by the OLT-LMA, since it has full knowledge about the status and available resources of the WOBAN. Thus:

(9,10) At some point in time, the link layer at the MN detects poor signal level (or any other suitable metric), triggering a `Link_Going_Down` event at the MN, which indicates an imminent loss of radio coverage. This message is propagated to the network entity in charge of the mobility management of the MN (e.g., the PoS), in this case the OLT-LMA. Note that this message follows the path $MN \rightarrow AP \rightarrow ONU \rightarrow OLT$, encapsulated as a layer-3 packet (as defined in RFC 5164).

(11,12) Then, the OLT-LMA suggests a list of suitable PoAs to the MN with the `MIH_Net_HO_Candidate_Query request` message. This is an optimized list since the OLT-LMA contains both the network load status of the PON (collected via MPCP) and the mobility management information about the number of users attached to each AP (thanks to the MAGs). Thus, the OLT-LMA suggests which APs/channels are worth scanning, hence reducing the handover operation delay. The MN

indicates its preferences to the OLT-LMA using an `MIH_Net_HO_Candidate_Query response` message.

(13) *Optimization no. 2: Optimal Target Network Selection.* The OLT-LMA has all the information regarding the traffic load of each ONU and the geographical location of every AP. Hence, the OLT-LMA can make a decision about the best ONU/AP to handover to.

(14-16) Once the most suitable AP is chosen, the OLT-LMA queries the new MAG for its suitability in hosting the moving MN, via the `MIH_N2N_HO_Query_Resources request` primitive. Through this message, the OLT-LMA is also able to inform the target ONU (ONU_2) of the IP address of the ONU currently serving the MN (ONU_1). This information is later used to optimize the handover (see step 18). Under the assumption that the target MAG accepts the MN, the new MAG would then reply to the LMA with an `MIH_N2N_HO_Query_Resources response` message.

The next set of steps corresponds to the handover preparation procedure:

(17) Next, the LMA informs the target MAG (ONU_2) about the imminent handover. The MN will be notified about the target MAG in step 21, once the handover preparation is ready.

(18-21) *Optimization no. 3: Data bicasting during handover.* At this moment, the LMA and the target MAG (ONU_2) know that the MN handoff is imminent. In order to avoid packet loss during the handover process, the OLT sends all the MN traffic to both old and new ONUs (bicasting). To enable this optimization, each PMIPv6-WOBAN ONU has an associated multicast MAC address (which may be pre-configured or directly derived from its IP address), and thus receive all the broadcast EPON frames destined to that address, or to any other multicast MAC address they are listening to. Then, it is just necessary that ONU_2 joins the layer-2 multicast group of ONU_1 (step 18), and the OLT starts sending the MN traffic to the broadcast LLID and the ONU_1 multicast MAC address (step 20). This way, the MN traffic is received by ONU_1 and ONU_2 only, since the other ONUs filter these EPON frames out because of the destination multicast MAC address (Figure 5.2, second box of the simplified data frames). Therefore such multicast-based bicasting does not consume any additional bandwidth capacity of the EPON. The bicasting preparation starts when the target MAG (ONU_2) receives the `MIH_N2N_HO_Commit request` from the LMA, and before it replies back with an `MIH_N2N_HO_Commit response` (remark that the IP address of ONU_1 was sent to ONU_2 in message 14). Finally, once the data bicasting process has started, the LMA informs the MN that it is now able to perform the actual layer-2 handover to the target AP (message 21).

(22-32) The following procedure is very similar to the one explained for messages 1 to 8. Once the layer-2 connection to the new AP is established, the MN and AP_2 generates a `Link_Up` event, which is used by MAG_2 to trigger the sending of a `Proxy Binding Update` message (step 28). Since the PMIPv6 handover is now complete, the OLT can stop the bicasting and send the MN's traffic directly to the new serving MAG (ONU_2), which also stops listening to the multicast MAC address of ONU_1 (Figure 5.2, bottom right box). Note that the MN keeps using the same IPv6 address, despite the change of MAG/AP since it is provided with the same prefix used in the previous attachment by means of a `Router Advertisement` (step 32). Thus, in both up/downstream directions, PMIPv6 hides all mobility management details to the MN.

(33,34) Finally, once the handover is complete, the MN notifies the PoS (OLT) through the `MIH_MN_HO_Complete` messages.

Finally, when an MN moves between two APs of the same ONU-MAG, the handover procedure is much simpler. Essentially, the handover may be performed at layer 2 without the need for any PMIPv6 signaling. Also, data bicasting is not necessary since this type of handover involves a single ONU-MAG.

5.3 Discussion of performance optimizations

The three mobility optimizations proposed in the previous section have a clear impact on the performance operation of the PMIPv6-WOBAN, as noted from the following sections.

5.3.1 Bandwidth waste reduction

Thanks to the first optimization, no IPv6-in-IPv6 tunnel is required between the LMA and the MAGs, which saves 40 bytes per packet (due to the extra IPv6 tunnel header). This accounts for a 5.6% of bandwidth waste in realistic wireless access scenarios [92] (with an average packet size of 710 bytes), and could even rise to 46.5% for voice over IP calls where traffic is encoded with the Internet low bit rate codec (iLBC) used by Skype. This accounts for an important amount of bandwidth savings.

5.3.2 Fast handover procedure

Real tests conducted with an in-house PMIPv6 implementation show an average handoff interruption time of 950 ms (without layer-2 triggers). Most of this delay can be reduced thanks to the use of MIH link-layer triggers (to send `Proxy Binding Update` messages), together with the fact that the OLT-LMA tells the MNs which channels to

scan for the handover process (second optimization). In-house measurement studies conducted with a modified version of the `ath9k` driver[2] that scans only one channel and includes other layer-2 attachment refinements have shown average handover delays of 68.58 ± 0.76 ms, which is much smaller than the original one-second handover delay of a PMIPv6 architecture without MIH and optimal channel selection. This optimized delay value is given in Table 5.1 and shall be used in the validation section 5.3.5.

5.3.3 Reduced packet loss

The bicasting (third) optimization proposed in the integrated architecture, where the OLT-LMA multicasts the traffic of a moving MN to both the old and the new ONU-MAGs, allows minimizing packet loss during the handover process. This optimization does not consume any extra bandwidth, since the MN's data traffic is not duplicated but just transmitted once through the PON. This sharply contrasts with other standardized handover-optimization mechanisms that have a suboptimal routing in WOBAN scenarios. For instance, Fast Handovers for Proxy Mobile IPv6 (FPMIPv6) [101] is based on redirecting the MN traffic during a handover by means of a direct IPv6-in-IPv6 tunnel between the old and new MAGs. However this solution does not fit well in a WOBAN scenario because all packets sent between two MAGs (i.e. ONUs) need to go through the OLT. Therefore, during an FPMIPv6 handover, the MN downstream traffic would have to be sent first from the LMA to the old MAG ($OLT \rightarrow ONU_1$), which in turn would encapsulate it in an IPv6-in-IPv6 tunnel and send it to the new MAG ($ONU_1 \rightarrow OLT \rightarrow ONU_2$). This is clearly suboptimal for a PON scenario.

5.3.4 Handover estimated delay

Table 5.1: Values used on the theoretical and simulation analysis

Delay	Analytical model	Comment	Values
T_{WLAN}: MN $\leftrightarrow$ AP	Gaussian ($mean \pm std$)	WLAN delay, value from in-house experimentation [102]	1.18 ± 0.474 ms
$T_{AP \leftrightarrow ONU}$: AP $\leftrightarrow$ ONU	82μs/KByte	Serialization delay in a 100 Mbps Ethernet connection	0.12 ms
$T_{ONU \rightarrow OLT}$: ONU $\rightarrow$ OLT	$1.5\frac{NT_g}{1-\rho}$	Worst case upstream PON TDMA [103]	5 ms
$T_{OLT \rightarrow ONU}$: OLT $\rightarrow$ ONU	5μs/km + 8.2μs/KByte	Worst case downstream PON (propagation and serialization delay)	0.112 ms
$T_{L2_{ho}}$: layer-2 Connection (Wireless Link)	Gaussian ($mean \pm std$)	Value from in-house experimentation [102]	68.52 ± 0.76 ms
T_{LMA}: Processing time at LMA	Gaussian ($mean \pm std$)	Value from in-house experimentation [102]	0.6 ± 0.2 ms

Table 5.1 shows an estimate[3] of the different delays involved in the handover decision, preparation and finalization. Following this Table,[4], the total worst-case delays in each block of Figure 5.2 can be estimated as:

[2] See `http://linuxwireless.org/en/users/Drivers/ath9k`

[3] The PMIPv6 implementation used in our experiments can be found in `http://www.openairinterface.org/openairinterface-proxy-mobile-ipv6-oai-pmipv6`

[4] In Table 5.1 N refers to the number of ONUs (typically 32 or 64), $T_g = 1.5\mu s$ refers to the guard time, and ρ is the total traffic load in the PON.

i) Initial attachment: $T_{L2_{ho}} + T_{AP\leftrightarrow ONU} + T_{ONU\rightarrow OLT} + T_{LMA} + T_{OLT\rightarrow ONU} + T_{AP\leftrightarrow ONU} + T_{WLAN} \simeq 75.6$ ms.

ii) Handover decision: $(T_{WLAN}+T_{AP\leftrightarrow ONU}+T_{ONU\rightarrow OLT})+(T_{OLT\rightarrow ONU}+T_{AP\leftrightarrow ONU}+T_{WLAN})+(T_{WLAN}+T_{AP\leftrightarrow ONU}+T_{ONU\rightarrow OLT})+T_{OLT\rightarrow ONU}+T_{ONU\rightarrow OLT} \simeq 19.14$ ms.

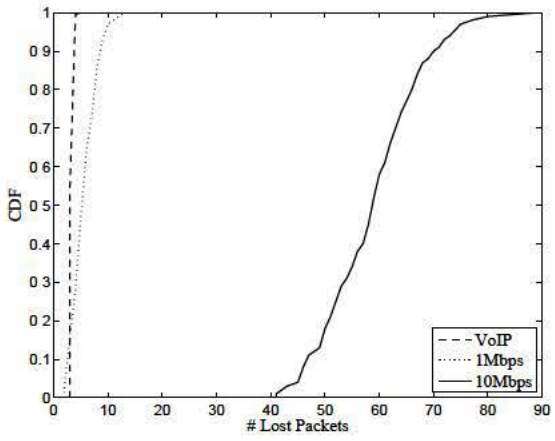
iii) Handover preparation: $T_{OLT\rightarrow ONU} + T_{ONU\rightarrow OLT} + (T_{OLT\rightarrow ONU} + T_{AP\leftrightarrow ONU} + T_{WLAN}) \simeq 6.54$ ms.

iv) Handover finalization: $T_{L2_{ho}}+T_{AP\leftrightarrow ONU}+T_{ONU\rightarrow OLT}+T_{OLT\rightarrow ONU}+(T_{AP\leftrightarrow ONU}+T_{WLAN}) + (T_{WLAN} + T_{AP\leftrightarrow ONU} + T_{ONU\rightarrow OLT}) + (T_{OLT\rightarrow ONU} + T_{AP\leftrightarrow ONU} + T_{WLAN}) \simeq 82.78$ ms

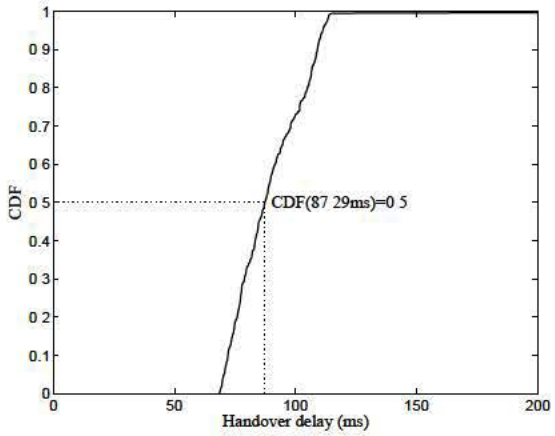
Without the bicasting optimization, data loss may occur from the beginning of step 22 (layer-2 connection) to the end of step 30 (Proxy Binding Acknowledgment), since only at this time the target ONU-MAG may forward data to the incoming MN. Essentially, the bicasting flow allows the target ONU-MAG to buffer the MN's data until step 30.

5.3.5 Validation of the proposed architecture based on simulation

This section further evaluates the performance improvements achievable by the optimizations proposed before. To do so, we have extended the event-based special-purpose simulator developed in [104] with EPON and PMIPv6 modules. The simulator estimates the service disruption time caused during the handover process of an MN that moves between two APs connected to different ONUs. In our scenario, we consider a PON with $N = 32$ ONUs employing IPACT, where the distance between a given ONU and the OLT is 5 km (i.e. 25 μs one-way propagation delay). Each of the ONUs is connected to an IEEE 802.11g AP via Fast Ethernet (we consider a negligible propagation delay between AP and ONU and a serialization delay of $82\mu s/KByte$). To simulate the EPON and specifically the IPACT algorithm, the simulator uses a hybrid packet analysis based approach. On one hand, each packet exchanged between the OLT and the studied MNs in downlink is simulated on a packet by packet basis. On the other hand, uplink traffic follows an analytical model, used to compute the average queuing delay of the packets sent by the MNs. For the validation of our proposal, we consider that the EPON operates at medium load levels (30-40% [105]). Following this, we assume a total offered load in the upstream channel of 30% (i.e. $\rho = 0.3$), thus producing an average cycle time of $E(T_{cycle}) = \frac{NT_g}{(1-\rho)} = 0.23\ ms$ for a guard time of $T_g = 5\ \mu s$. In IPACT, the cycle time denotes the time elapsed between the beginning of two consecutive transmission windows for the same ONU in the upstream channel. Hence, we have approximated the average queuing delay of a packet arrival at a given ONU as $\frac{3}{2}E(T_{cycle})$, as noted in [103].



(a)



(b)

Figure 5.3: Simulation results: (a) CDF of the number of lost packets without optimization and (b) CDF of the optimized handover delay

Regarding the PMIPv6 implementation, we extended the OLT with basic PMIPv6 functionality by developing a version of the Binding Cache to store the mapping between Proxy Care-of Address and corresponding IP address of the ONU (MAG), extra functionality required to parse the Proxy Binding Update messages and create the Proxy Binding Acknowledgement was also added. A secondary module, performs a mapping between the ONU's IP and MAC addresses, hence removing the need for an IP tunnel as required by the standard PMIPv6 specification. This module also implements the bicasting functionality by replacing the destination unicast MAC address with the multicast one when necessary (after appropriate signaling is received). In the simulation, the OLT sends data to an MN, which at a random time, initiates a handover between two ONUs. The IEEE 802.21 signaling is implemented as successive message exchanges, but we only consider the actual size of the MIHF messages, not their real content. The impact of this signaling on the simulation is, as expected, an extra delay between the starting of the handover

process and the subsequent layer-2 detachment.

To validate the proposal, we have considered three different traffic sources: *i)* the case when the OLT transmits Poisson traffic to the MN at an average bitrate of 1 and 10 Mbps (packet size of 1500 bytes) which considers the case of aggregated traffic with low and high data rate profile respectively; and *ii)* the case when the MN receives a Skype-like VoIP communication (iLBC codec, 50 packet/sec, 116 bytes/packet). We further assume a total offered load in the downstream direction of 30% (same as upstream, used to compute the average queuing time). We simulated each step during the handover process with the parameters explained in Table 5.1, whose values are supported by real testbed scenarios (see related references in the table), and we measured the number of lost packets without any of the proposed optimizations (data bicasting, no need for IPv6 tunneling and optimal target selection) in order to quantify the performance gains in terms of packet loss and connectivity disruption time. The only optimization that impacts the packet loss is the data bicasting, which prevents it completely because the data addressed to the MN is being bicasted and buffered before its actual movement. In the simulation we consider that the handover process delay is reduced to the value of $T_{L2_{ho}}$ shown in Table 5.1.

Figure 5.3 shows the empirical cumulative distribution function (CDF) of the number of lost packets in a WOBAN as the one proposed, without the data bicasting optimization. As shown, this optimization brings important packet loss savings for the three traffic profiles. For example, in the VoIP traffic profile, the number of lost packets shows a median of four packets (Figure 5.3a), which implies about 80 ms of conversation disruption. Similar benefits are shown for the other two traffic profiles, since the data bicasting optimization implies no packet loss during the handover. Finally, Figure 5.3b presents the handover duration CDF. As previously explained, the handover time for all cases is similar, since we assume an optimized layer-2 delay in all scenarios. As shown in Figure 5.3b, the average handover duration corresponds to 87.29 ms, which is in line with the values computed theoretically in section 5.3.4.

5.4 Summary

Hybrid WOBANs are a promising technology to provide high-speed wireless access to end-users. We take one step further in the integration of wireless-optical technologies by proposing an integrated PMIPv6-WOBAN architecture that simplifies the mobility management of MNs. This architecture maps the PMIPv6 framework and IEEE 802.21 MIH Services into the hierarchical structure of the WOBAN's passive optical network by collocating the local mobility anchor with the optical line terminal, and the mobile access gateways with the optical network units, which controls a set of heterogeneous wireless Access Points (e.g., WiFi and cellular). Such a tight integration enables the optimized use of resources, since the centralized OLT-LMA node is able to combine traffic

statistics and user mobility information collected from the ONU-MAGs, and thus may initiate handovers of MNs due to ONU or AP overload. Moreover, the proposed architecture includes a number of optimizations that leverage the particular characteristics of a WOBAN. For instance, the single-hop, point-to-multipoint topology of an Ethernet PON avoids the overhead of maintaining tunnels between the LMA and its MAGs; and enables the use of multicast EPON bicasting during handoffs to prevent from packet loss, therefore providing a seamless handover experience. We argue that the complexity of the proposed architecture is moderate since OLT/ONU devices are already full-fledged IP routers, hence adding LMA/MAG functionality would require a minor software update. Regarding the optimizations considered, only the "Optimized Target Selection" may incur in high complexity in case it is integrated with the OLT scheduler, since then a smart scheduler, providing higher bandwidth to the ONUs with a higher number of attached users may be implemented, requiring access to the Dynamic Bandwidth Allocation table at the OLT.

Finally, this PMIPv6-WOBAN architecture supports other access technologies different from WiFi in the LMD, and therefore a multi-interfaced MN would be able to roam between different technologies or make use of them simultaneously (e.g., to perform traffic offloading, flow mobility or multi-link aggregation).

Chapter 6

Software Defined Networking and Distributed Mobility Management

6.1 Introduction

Software Defined Networking (SDN) has implied a paradigm shift in network management. Due to the increasing heterogeneity of wireless access technologies and the densification of the access networks, SDN appears as a flexible solution, keeping costs under control. In this Chapter, we provide an SDN network architecture for mobility management based on the concepts of PMIPv6 and DMM and we also include a host-based mobility solution. Our architecture has been designed in the scope of the FP7 ICT CROWD project.

6.2 SDN Architecture for DMM

In this section we provide the detailed description of our SDN-based architecture for distributed mobility management. We propose a network-based mobility solution, which is transparent to the mobile terminal and inherits the flexibility and scalability typical of SDN. Our mobility management architecture relies on two main entities, the Local Controller (CLC) and the Regional Controller (CRC), and provides mobility at two differentiated levels. In particular, we present first the mechanisms enabling fast layer-2 mobility within a domain (the so-called *district*); afterward, we present the SDN-based DMM solution designed for the inter-district (regional), layer-3 mobility. It is worth highlighting that our two-tier hierarchical approach to the controller deployment follows the guidelines of the ONF, hence it is already aligned with the latest standardization efforts in the area.

We define a district as an isolated single-technology domain, similar to the localized mobility domains of PMIPv6. A district is composed of a dense deployment of APs con-

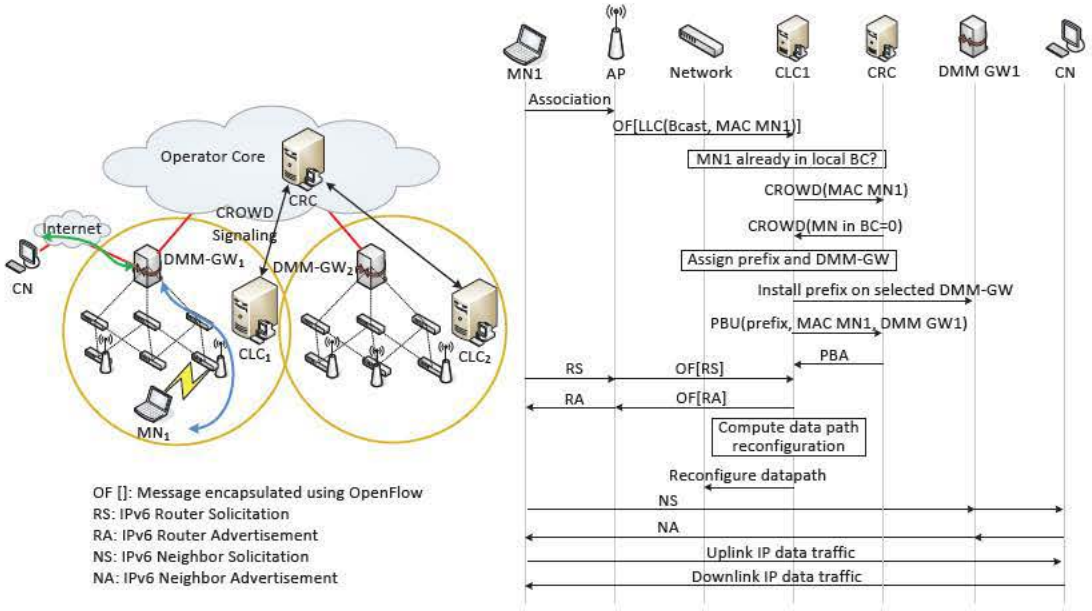


Figure 6.1: Initial attachment procedure.

nected together by a switched network of SDN-enabled interconnection nodes (OpenFlow switches). A district includes, at least, one Distributed Mobility Management Gateway (DMM-GW) that connects the district to the Internet and plays a central role in the inter-district mobility solution. In Figure 6.1 we present an example of the complete architecture with two districts. In this case, the districts are composed of IEEE 802.11 Access Points, an OpenFlow-capable backhaul connecting the APs to the DMM-GW and a local controller (CLC). In addition, the regional controller (CRC) is located at the operator core network and coordinates the attachment of a mobile terminal to a district as well as the inter-district mobility.

Upon attachment of the mobile terminal (MN1 in Figure 6.1), standard APs (bridged to a wired network) generate a Logical Link Control (LLC) message that serves as the mechanism to update the forwarding tables in the switched network. As all the network is OpenFlow capable, this LLC message is encapsulated in an OpenFlow message and sent to CLC_1. The LLC message contains the MAC address of the terminal and the MAC address of the AP. In the case the mobile terminal has not been previously attached to the district controlled by CLC_1, it does not have any previous entry of the terminal on its Binding Cache (BC).[1] In order to check if the node is already registered in a previous district, CLC_1 will contact the CRC. Let us assume in this first example that the terminal has not been attached previously to any AP in the domain controlled by the CRC, so CLC_1 assigns a new IPv6 prefix and a DMM-GW to the terminal, stores this information

[1]Inherited from PMIPv6, this table stores bindings between terminals and points of attachment in our architecture.

on its BC and notifies the CRC about the assigned prefix, DMM-GW and MN identifier (e.g., MAC address). In this way, the CRC is able to keep track of every MN attachment. After successfully attaching to a new network, the standard procedure for the terminal is to send a Router Solicitation (RS) to configure its IP address using IPv6 SLAAC. As in the case of the LLC message, the OpenFlow-enabled network encapsulates the RS message and sends it to CLC_1. The CLC answers the RS with a RA message, providing the prefix and default router, e.g. DMM-GW_1, selected before. Hence, through the mediation of the CLC hijacking the RA functionality of the network, we are able to control the IP level attachment of the terminal within the region. Note that, although we assume a MN sends a RS message every time it attaches to a new point of attachment, in our solution we can trigger the RA message even if the RS is not sent, which is a very useful feature in case the MN does not completely follow the standard (most Linux boxes do not send RS messages unless the interface goes down). After this initial configuration, CLC_1 is able to compute the required matching rules and data path modifications to forward the terminal's packets to DMM-GW_1. These modifications are configured into the network through the OpenFlow protocol, requiring several message exchanges among CLC_1 and the different switches conforming the path between the terminal and DMM-GW_1. Once the data path is configured, packets originated at the terminal with the DMM-GW as layer-2 destination are transparently forwarded at layer-2. For the layer-3 stack of the terminal the path to the DMM-GW is a single hop. Finally, after performing a Neighbor Discovery procedure, the mobile terminal is able to exchange packets with any CN through DMM-GW_1.

6.2.1 Intra-district Mobility

The intra-district handover is illustrated in Figure 6.2, where the mobile terminal MN1 attaches to a second AP within the same district. In this case the only required change is to modify the data path, so packets are forwarded between the new AP and DMM-GW_1. The flow diagram depicted in Figure 6.2 represents the procedure. The new AP, upon attachment of MN1, generates an LLC message. Upon reception of this message, CLC_1 is able to notice the movement of the terminal looking up in its BC the MAC address of the AP the terminal was connected to.[2] With this information, CLC_1 is able to compute the required modifications to the data path for the terminal's packets. Accordingly, the OpenFlow configuration in the district is modified, and the terminal's packets will flow from the new AP to the old DMM-GW, i.e., DMM-GW_1 (we assess the handover-related delays in Section 6.4).

[2]Note that using the MAC address represents just a practical example for terminal identification, although any other kind of terminal identifier could be used instead.

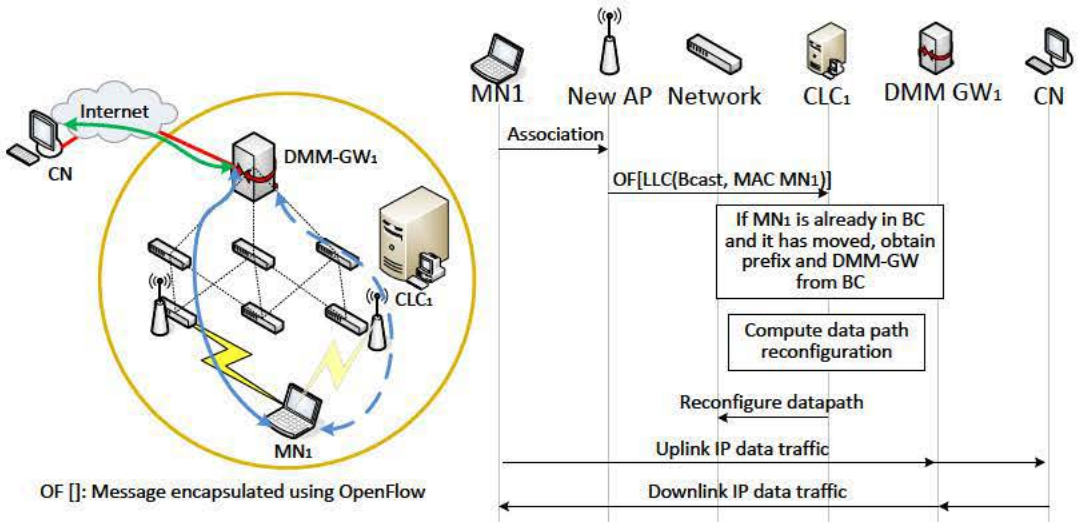


Figure 6.2: Intra-district handover.

6.2.2 Inter-district Mobility

The handover between two districts is presented in Figure 6.3. In this case the CRC orchestrates the joint operation of the two CLCs. Basically, the CRC keeps the list of DMM-GWs to consider in a handover and coordinates the operation. The configuration of the IP layer on the DMM-GWs and the tunnel set up between them is handled locally by the CLCs in each district. Continuing with the example presented above, the inter-district handover happens when the MN attaches to an access point in a different district. Upon this attachment, the CLC (CLC_2) checks if the node is registered on its internal BC. This is the first time the terminal attaches to this district, so CLC_2 checks with the CRC previous registrations of this mobile terminal in the region. In this case, the CRC has information regarding the terminal, and transmits it to CLC_2, which decides the DMM-GW to be used within this district (DMM-GW_2) and informs the CRC of this decision. The CRC stores this information on its local BC for future reference. Then, several procedures take place in parallel. First, the CRC informs CLC_1 of the new location of MN1. With this information, CLC_1 configures DMM-GW_1 with an IP-in-IP tunnel connection to DMM-GW_2 and changes the routes at DMM-GW_1 so that the prefix used by the terminal is routed through the tunnel. In parallel, CLC_2 configures the new prefix in DMM-GW_2 and sets up the IP-in-IP tunnel towards DMM-GW_1. Once the tunnel is established, the data path in the new network is configured as explained in the previous cases. When the procedure is complete, packets from the core network to MN1 are forwarded first to DMM-GW_1, which tunnels them to DMM-GW_2, as depicted in Figure 6.3. After reaching DMM-GW_2, the OpenFlow data path forwards the packets to the appropriate location of MN1 within the district.

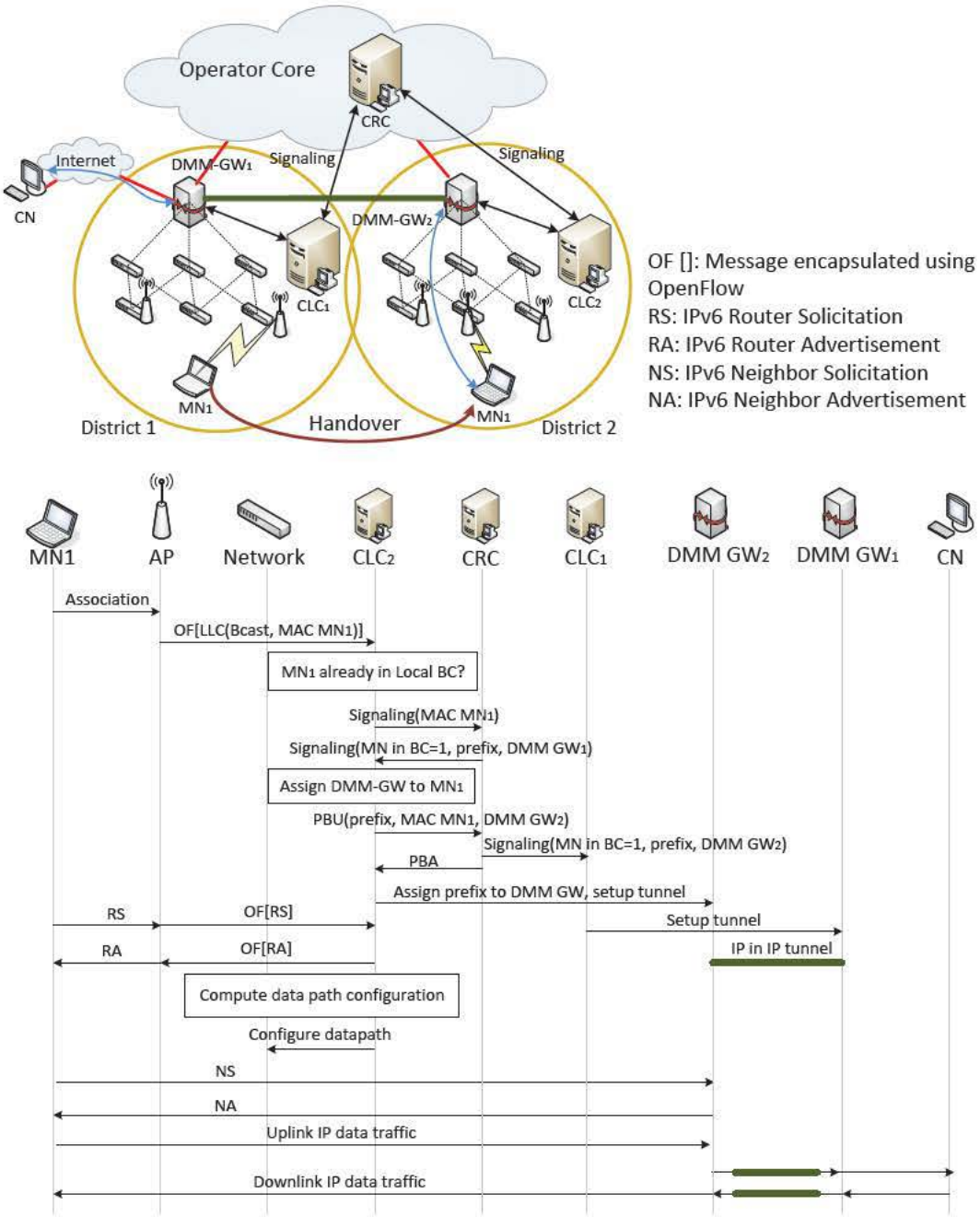


Figure 6.3: Inter-district handover.

6.3 Integration of host-based mobility management

We have designed a mechanism to ensure session continuity also when the MN moves out of the CROWD domain. This mechanism is based on MIPv6 and involves the MN in its own mobility management. When the mobile node leaves a CROWD district connecting to a new visited network and configures a new care-of address, the MN contacts the

CRC. Similarly to MIPv6 operation, the MN uses a BU message to notify its new address to the CRC, which in this case acts as a HA. Then, the CRC gives the IP address of the former DMM-GW to the MN and notifies the CLC in the last district visited by the MN to update the forwarding in the district. The CROWD Local Controller (CLC) will command the correspondent DMM-GW to establish a tunnel with the MN in its new location. This operation is similar to the DMM approach, but the tunnel is established with the MN itself, instead of with a new Access Router (AR) or DMM-GW. Figure 6.4 shows the handover process and the frame exchange for our host-mobility DMM design. The

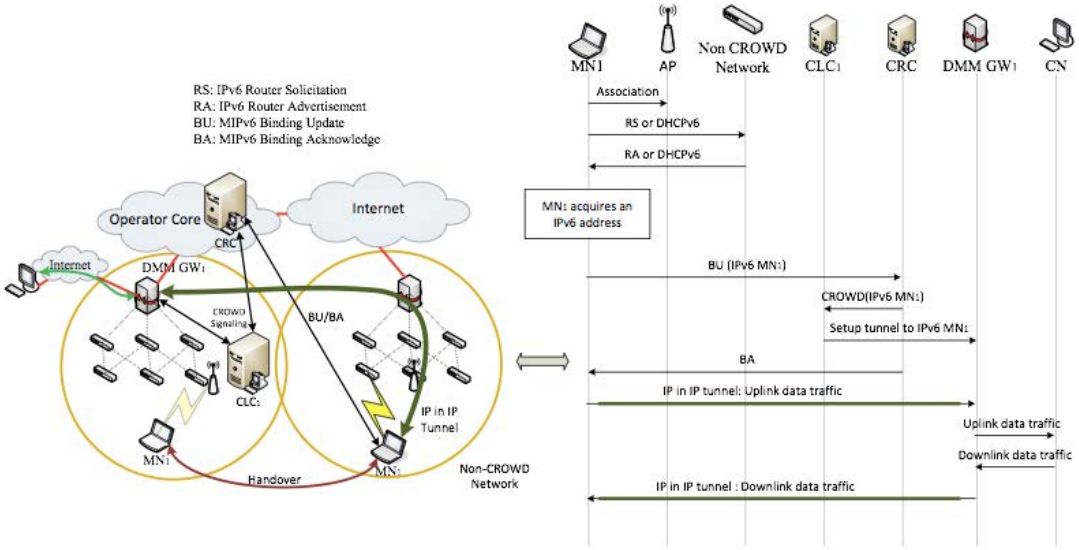


Figure 6.4: Host-based DMM approach.

drawback of this mechanism is requiring the modification of the mobile node. However, involving the mobile node gives more flexibility and the MN can connect to any network, even if it does not implement mobility support. Note that the access technology of the visited network can be different from the technology used in the district where the MN was previously attached to. In addition, this solution is another example of the usefulness of the combination of mobility protocols.

6.4 Implementation and performance evaluation

From Figure 6.1, where we present the architecture of our solution, we can identify the main architectural elements, splitting our testbed into three kinds of entities: *i)* the regional (CROWD Regional Controller (CRC)) and local (CLC) SDN controllers manage the forwarding rules in the OpenFlow-enabled backhaul and handle the mobility of the MN; *ii)* the OpenFlow switches, which include, both wired backhaul and wireless access network elements because the wireless interfaces in the APs are controlled as in

Table 6.1: Main HW and SW characteristics of the testbed

Entity	Device (OS, kernel version)	OF-enabled
Switched backhaul	Linksys WRT54GL (OpenWrt Backfire, 2.6.32)	✓
IEEE 802.11 APs	Alix 2d3 (Ubuntu 12.04, 3.2.0)	✓
DMM-GW		X
Regional and local controllers	Desktop (Debian 7.4, 3.2.0)	Ryu controller

normal OpenFlow switches; and *iii)* the DMM Gateway, which is the only entity not related to OpenFlow and acts as the gateway in the district. Table 6.1 gathers the main characteristics of these elements.

In our testbed we have two 802.11-OpenFlow enabled APs, an OpenFlow switch, and a DMM-GW in each district. Additionally, each district is managed by its own local controller, and CLCs are coordinated by a CRC. The network deployment is showed in Figure 6.5.

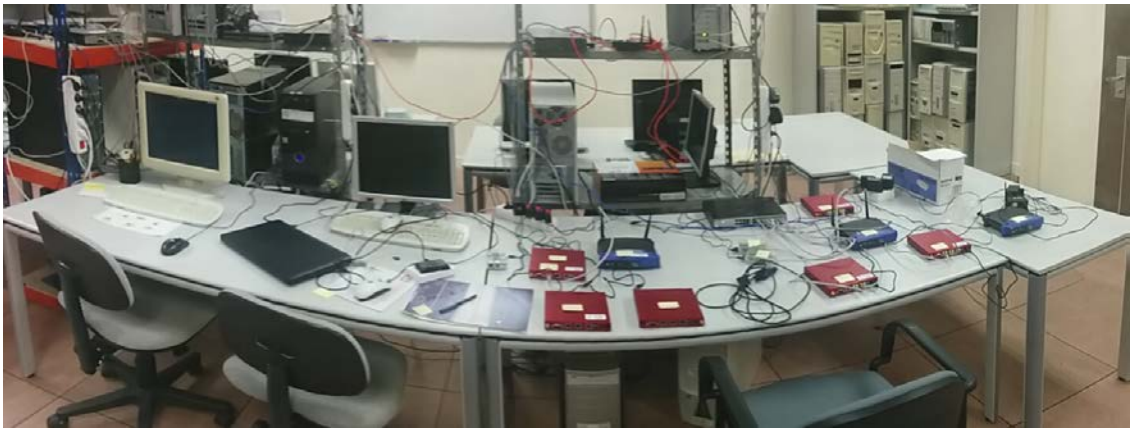


Figure 6.5: Experimental deployment

The OpenFlow signaling is distributed between the switches and the controllers in an outband connection. The wireless access is part of the OpenFlow-enabled network, so the connection between MN and DMM-GW is a single-hop connection at the network (IP) layer. Since the solution works at layer-3, it is necessary to define new APIs to control the IP layer configuration of the terminal's session anchor point (DMM-GW). Specifically, we need two new APIs: one to convey mobile node information to the CRC, (e.g. the IPv6 prefix and DMM-GW assigned to it), and a second one to configure the IP layer of the DMM-GW, the prefixes reachable through the interfaces and to setup an IP-in-IP tunnel. We have implemented our local and regional controllers using the Ryu framework,[3] which provides a comprehensive API that eases the network management.

To characterize the performance of our prototype we run 30 repetitions of every ex-

[3] `http://osrg.github.io/ryu/index.html`

periment. Figure 6.6 shows the CDF of the delay due to the intra- and inter-district handovers, measured as the interruption in data traffic, which also includes the link layer association. The intra-district case shows a lower latency, because the MN is already known in the district and therefore a simple change in the point of attachment to the network is needed in that case. The inter-district handover implies changes at layer-3 level, including the configuration of the IP-in-IP tunnel between the two DMM-GWs involved. However, the delay is very similar to the intra-district handover, because the controller runs the configuration of the gateways and the establishment of the OpenFlow datapath in parallel.

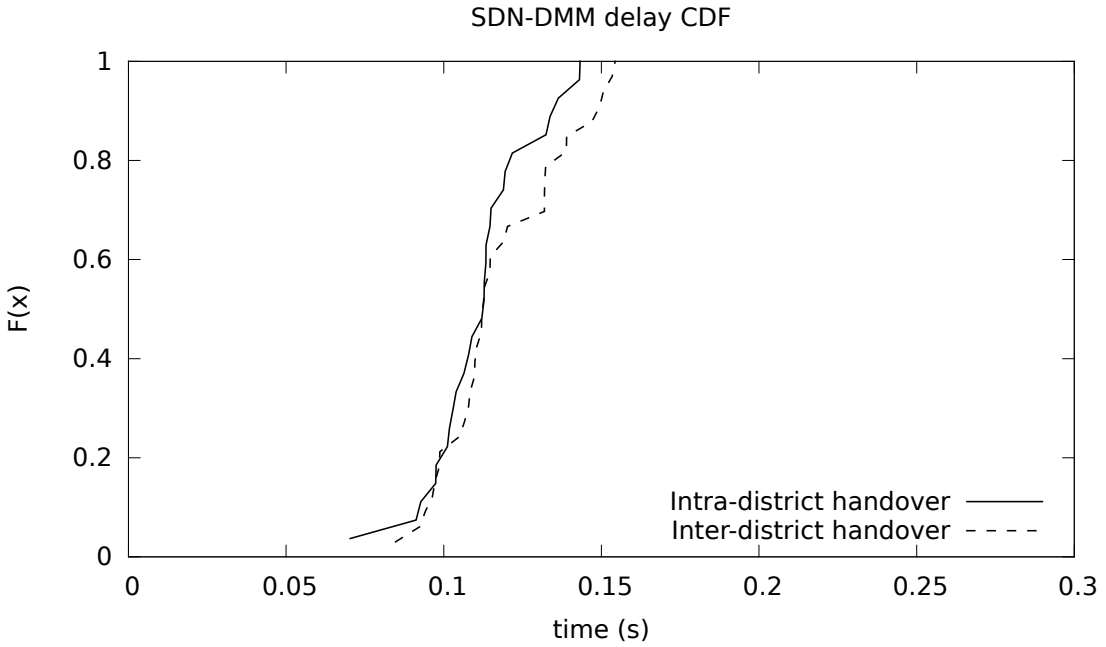


Figure 6.6: CDF of the different handover delays

Following with the evaluation of the performance of our SDN solution, we measure the throughput we can achieve for TCP traffic, using the *iperf* traffic generator. We compare the case the MN is attached to one of the districts and the case that it has performed a handover and its traffic is tunneled between the current and former DMM-GWs. The average throughput is 8.6 *Mb/s* and 8.1 *Mb/s* respectively, confirming the impact of this tunneling is very low (the IP-in-IP tunnel just adds 40 bytes of headers). It is noticeable that the maximum throughput available in the SDN framework is quite limited for the moment. This is a well-known open issue and very heavily influenced by the use of the Linksys WRT54GL router as OpenFlow-switch.[4] Note that the signaling due to OpenFlow does not impact the data traffic performance, as we use outband signaling within the SDN elements, and the data transmission from the MN is only delayed at its first packet, when

[4] `http://archive.openflow.org/wk/index.php/Pantou_:_OpenFlow_1.0_for_OpenWRT#Performance`

the matching rules are firstly installed in the OpenFlow switches.

To evaluate the impact of the SDN operations, we measure the delay introduced by the operations carried out by the controller in our SDN-based DMM solution. Processing the first packet generated upon the attachment of the MN takes, on average, 21.6 ms. On the other side, processing the first data packets, which trigger the data path configuration, takes on average 12.3 ms for uplink and 9 ms for downlink. From our experience, the choice of the device running the controller has a significant impact on the performance of the prototype, especially in the case of handover. Still we continue working to optimize and reduce the signaling and (re)configuration delays.

Modifying the switching to provide mobility support in the districts may not be the most intuitive solution. However, in this way we provide a worst-case boundary and in addition, we can take advantage of a multi-purpose implementation. Our aim is to enable DMM support and be able to improve performance in the network by having different networking features implemented and running in parallel.

6.5 Summary

We have designed a mechanism for mobility management based on SDN principles. It is envisioned to coordinate connectivity in a dense deployment of access points or base stations, in an area populated by single-technology domains called districts. Our SDN-enabled proposal offers several advantages over traditional layer-2 mobility mechanisms for what intra-district handovers presents. Through the control of the prefix delegation by the controller, the MN can be attached to any of the DMM-GWs in the district, or even to several of them at the same time. This allows a new degree of freedom, since the network can balance the load at the different exchange points to the Internet. In addition, the control of the data path, operated by the controller, enables the network to provide fast mobility of the terminal. The data path reconfiguration scales better with the number of switches than standard spanning tree approaches, whose reaction time can be measured in seconds for large networks [106]. Moreover, the algorithm used for controlling the data path can be arbitrarily complicated, allowing complex traffic engineering operations. The data path forwarding decision engine can consider load balancing metrics, or even complex mechanisms minimizing the number of nodes requiring changes in their forwarding tables. In addition, we could employ different forwarding decision engines for different nodes or traffic classes, prioritizing specific metrics, such as delay or available bandwidth along the path (leveraging the OpenFlow monitoring capabilities). Lastly, standard mobile nodes, such as Linux boxes, do not send RS messages unless the interface goes down, hence they do not typically do it when performing a handover. This forces developers to add the transmission of this RS message or specific attachment-detection traps to the access points. This is the case of PMIPv6, for example. By leveraging the SDN approach,

our architecture does not require the transmission of the RS message, hence making the deployment easier than standard PMIPv6 approaches. It is worth highlighting that in our design we prioritize flexibility over reconfiguration delay. Such an intra-domain mobility enables the control of all possible parameters of the routing of packets, selecting the path according to any arbitrary metric and, as such, it exhibits the worst possible delay (for SDN solutions) at installing the path, because all the elements in the path are reconfigured. Other approaches oblivious to the actual internal behavior of the network can also be applied to our solution, such as the use of pre-configured VLAN-based paths connecting access points to DMM-GWs, hence improving the reconfiguration delay but decreasing the flexibility of the network.

As for the inter-district handover, the key advantage of this procedure over non-SDN DMM approaches is the flexibility on selecting the most appropriate DMM-GW to be used for each terminal. Since the CLC is able to decide and attach the prefix used by the terminal to any of the DMM-GWs available at the district, with our approach, the target network is able to implement any load balancing or complex algorithm to decide the best DMM-GW to use. In addition, the data path forwarding within the district is decoupled from the layer-3 mobility. In order to minimize the security implications of this approach, our design enforces that the only entity talking with the DMM-GWs belonging to a certain district is the local controller in that district. In such scenario it is possible to think of specific security associations between the CLC and DMM-GWs of the district. Finally, although we have not implemented or included this functionality in the current design, it is worth highlighting that the same design can be used to provide IPv4 DMM capabilities to the network by using the Address Resolution Protocol (ARP) instead of IPv6 Neighbor Discovery protocol.

Finally, another of the benefits of using an SDN approach for the control of the user mobility is the choice of different mobility protocols per user or even per flow, if needed. This is performed by just discriminating the flows of a certain user and installing the appropriate behavior in the network according to the mobility policies for the user. In this way, we can provide specific features in a per flow basis, e.g., some flows will use DMM, while others PMIPv6 or even MIPv6, depending on the traffic class or the user subscription.

Part III

Multi-interface Devices and Connectivity Management

Chapter 7

Support for enhanced connectivity management in IEEE 802.11 networks

7.1 Introduction

The penetration rate of IEEE 802.11 wireless technology has increased ever since 1997, when the first version of the standard was approved. The evolution of portable devices has led to high-profile mobile devices, e.g. tablets, smartphones, laptops, etc. that support several network interfaces. This, joined to the increasing user demand for anywhere-anytime connectivity have also contributed to the high impact of this technology. Proof of this success is that nowadays we are surrounded by 802.11 Access Points (AP) that offer wireless connectivity almost everywhere, from office buildings, hotels or restaurants to transportation systems.

Throughout the research conducting to this thesis, we have witnessed that 802.11 technology has a major impact on mobility management procedures. As seen in Chapter 3, most of the interruption during a handover is due to the layer two reconnection. Due to its importance for the optimization of handover and higher layers mobility management we looked into the amendments to the standard that can potentially contribute to mobility management.

The 802.11 family has been growing continuously and in this Chapter we focus on three amendments to IEEE 802.11 standard approved within the recent years which are part of the present version of the standard, namely 802.11k [15], 802.11r [16] and 802.11v [17]. 802.11k addresses the necessity of taking radio measurements to monitor the wireless links and network status; 802.11v enables the management of the wireless network, covering several aspects of configuration and data exchange; and 802.11r enables fast transitions between two Basic Service Sets (BSSs) within the same Extended Service Set (ESS). The

BSS is the basic building block in the WLAN, and a set of interconnected BSSs forms an ESS, all grouped under the same ESS identifier (ESSID).

We have chosen to study these amendments because even though they address relevant issues with interesting insights for improving performance of WiFi networks, specially being applied to resource management and change of AP, they have been overlooked by most players involved in the research and technology deployment. The slight impact of these amendments has called our attention on their feasibility and motivated us to evaluate their impact on the research community, as well as to look for practical implementations or commercial hardware including support for their operation. In particular, we focus on these amendments because there has been a reasonable time from their official approval, long enough to evaluate their impact.

7.2 Historical perspective

Although 802.11 is a widely deployed technology, stable and mature, it is continuously evolving [107]. The first implementations of the initial version of the standard easily identified compatibility issues among devices from different vendors, which have been solved by the certification provided by the WiFi Alliance, which made 802.11 technology commonly known as WiFi. Therefore, in the present all WiFi-certified devices are able to inter-operate, they are fully compatible and the configuration of the connection can take place almost transparently to the final user. This transparency favors the user unawareness of the changes in the standard. This is very reasonable, because regular users just care about having wireless connectivity with acceptable performance. That is why for instance, one of the latest amendments, 802.11n, has become popular, known by the general user and widely supported in commercial products, because it increases the available bandwidth considerably over its predecessors, but the main changes in the physical or MAC layers go unnoticed for the majority of WiFi users. Likewise, manufacturers do not contribute to advertise most of the numerous amendments, beyond the ones that enhance the performance experienced by the user, as those are the facts that make their product attractive to their clients. In this way, recent amendments 802.11ac and 802.11ad are growing in popularity because of the high throughput they intend to offer. However, other amendments as 802.11aa, which includes significant improvements for multimedia traffic, have been almost unnoticed.

The IEEE 802.11 standard has gradually progressed over the years, aiming to adapt its specification to the evolution of wireless networking and traffic characteristics. Proof of that are the revised versions approved in 1999, 2007 and 2012, which integrate amendments developed within the years in between. All these amendments address issues of very different nature. For instance, IEEE 802.11-2007 [108] includes the amendment 802.11e with MAC enhancements for including QoS support, 802.11g that increases data rate in

Table 7.1: Summary of the Radio Resource Measurements defined by 802.11k

Measurement	Reporting policy	Information reported	Comments
Beacon	Request-Report	List of APs the STA can receive from in the specified channels	Passive (Information comes from Beacons), active (information comes from Probe Responses) or beacon table (information comes from Measurement Pilots).
Frame	Request-Report	Frame count information	For every transmitter address, the measuring station reports the number of frames received, the power level and the BSSID
Channel Load	Request-Report	Channel utilization	The fraction of measurement time that the channel is busy for the measuring station.
Noise histogram	Request-Report	Power histogram of non-IEEE 802.11 noise power	Provides the average noise plus interference power measured while the channel is idle.
STA Statistics	Request-Report	Frame counters, BSS average delays and channel utilization	The requests are made for groups of statistics, each group containing a different amount of information
Location Configuration Information	Request-Report	Location information	Location in terms of latitude, longitude, altitude and optionally, azimuth. A station can make a request for its own position or the position of the measuring station
Neighbor Report	Request-Report	Known neighboring APs	Action frame. This information can be used before a transition to another BSS, identifying potential candidates before the actual transition.
Link Measurement	Request-Report	Instantaneous quality of a link	Sent in Action frames
Transmit Stream/ Category Measurements	Request-Report	Condition of an ongoing stream link between two QoS stations	The report can be triggered by conditions included in the request
Measurement Pause	Request only	Request the measurements to be paused or suspended	Used to delay execution of consecutive Measurement requests
Measurement Pilot	Report only	Transmitted periodically by an AP, more frequently than Beacons, to help STAs with scanning	Measurement Pilot frames are Public Action frames, which allow communications inter-BSS and between an AP and unassociated stations.

the 2.4GHz band or 802.11i aimed at improving security mechanisms. Whereas IEEE 802.11-2012 [11] incorporates optimizations for fast BSS transitions, as in amendment 802.11r, operations for interworking with external networks, as in 802.11u, or address the necessities of wireless networks in a different environment, as in 802.11p for vehicular scenarios.

7.3 IEEE 802.11k/r/v functionality

In this Section we introduce the main functionality of each of the amendments that we consider, including their motivation and the modifications that they apply to the standard. Understanding the mechanisms proposed by each of them is relevant to evaluate the impact of their implementation and their applicability to WiFi-enabled devices.

7.3.1 IEEE 802.11k: Radio resource measurement

This amendment was approved in 2008 [15] with the aim of paving the way towards next generation WLANs. In the last years there has been a shift in the kind of traffic being exchanged through wireless networks, and some applications would benefit from having direct information about the state of the wireless access. In addition, if every wireless station[1] had knowledge of the current link conditions and its environment, they could adapt their operation to these conditions, or use this information in case of a given event, e.g., a handover [109]. 802.11k addresses this issue, providing measurements to improve

[1]Note that "station" refers to any kind of station in the WLAN, an AP or a STA, according to the standard's notation.

performance and reliability of the wireless network, as well as interfaces to expose this information to upper layers. It is worth pointing out that these measurements are not vendor-dependent, as it happens for instance with the measurement of the Received Signal Strength Indicator (RSSI) provided by current wireless cards, whose measurement ranges depend on the hardware manufacturer.

A station can perform its own local measurements or request them to other stations. To that purpose, the amendment defines modifications to the body of Management frames, to make every station aware that the Radio Resource Measurement capabilities are available and to enable the exchange of the measurements among the stations. The result of most of the measurements can be exchanged in a request-report fashion, while some of them are report-only or request-only. Table 7.1 summarizes the measurements and the information provided.

A station can inform about the measurements it can perform, if any. The requests and reports for radio measurements, link measurements and neighbor reports are sent in the body of Action frames. An Action frame is a kind of Management frame that triggers an action, in this case, it makes a station perform measurements upon other station's request. The standard considers the case in which a station that is requested to perform a measurement may reject to do it, but does not restrict the reasons to do that. In addition, it is not specified how the station should carry out the measurements, and it is worth pointing out that some of these measurements may prevent the station from its normal operation, having to scan other channels or perform an operation not compatible with receiving or transmitting data.

This amendment enables stations to increase their knowledge about the network conditions and therefore, it may lead to a more efficient management of the resources and performance optimization. For instance, including information such as the current load in the BSS in the body of Beacons and Probe Response frames lets stations have an estimation on which resources would be available at potential candidate APs for handover, before taking the actual decision. However, there are other hitches that should also be considered. For instance, the nature of the wireless link is very dynamic, and the result from some measurements may differ considerably even if they are spaced a few milliseconds. The standard defines the duration of the measurements, but it does not specify when they have to be performed. Should a station defer its current tasks to perform a requested measurement or can it be delayed? Would a delayed measurement be still useful for the station that requested it? Defining the timeliness of the measurement is a key issue that has been left out of scope.

In addition, a station might need to use several resources or applications in order to perform some measurements, and not all of them are easy to obtain. The location information in the way it is defined in the standard, with a very fine-grained precision on latitude, longitude, altitude and even azimuth coordinates might not be easy to obtain

by every station.

7.3.2 IEEE 802.11r: Fast Basic Service Set (BSS) transition

This amendment, also approved in 2008 [16], addresses the movement of a station from one BSS to another within the same ESS, trying to minimize the interruption in data connectivity. When a wireless station roams from the AP where it is currently associated to another one, it needs to complete several steps in order to establish the new connection. The scanning phase that lets the station discover neighboring APs, and the authentication process to secure the communication are time consuming tasks that need to be performed in every transition. To this matter, 802.11r aims at reducing the time needed for a transition by providing mechanisms for authentication, with equivalent security as the one provided by 802.11i, and QoS negotiation for QoS support as in 802.11e, but shortening the delay in the transition. In a fast transition, the negotiation of QoS and the authentication exchange are done prior to reassociation, which speeds up the process [110].

Regarding authentication, IEEE 802.11 uses two mechanisms: *Open System* authentication, where any station can join the network, and *Shared Key* authentication, which uses 802.1X. In addition to those, this amendment includes a Fast Transition (FT) authentication to enable the authentication to a different AP before reassociation. Fast transition mechanism only applies to transitions between APs belonging to the same ESS, where the set of BSSs that allow fast transitions between them is referred to as a *mobility domain*. During the first association of a station to this domain, 802.11r establishes a key management mechanism for the whole mobility domain and a FT 4-way handshake that will allow the station to authenticate to the new AP before the actual transition.

After this initial association to the mobility domain, for the next BSS transitions the station can authenticate and negotiate QoS capabilities with the new AP before the reassociation by using two FT protocols: *i)* FT protocol and *ii)* FT Resource Request Protocol, depending on whether the station requires a resource request for ensuring QoS before the transition. The messages exchanged for these two protocols can take place in two different ways: *i)* "Over the air", where the station communicates directly with the target AP or *ii)* "Over the DS"(Distribution System), where the station sends data intended to the target AP to the currently associated AP (format of Action frames defined for this purpose), and this one connects to the target AP through the DS infrastructure.

7.3.3 IEEE 802.11v: Wireless network management

This amendment was approved in 2011 [17] and it addresses improvements to be applied to the standard and the previous amendments for them to support wireless network management in a variety of fields, from multicast transmission to beaconing or sleep

Table 7.2: Summary of the services provided by 802.11v

Service	Information	Utility
BSS Max Idle Period Management	An AP can inform the amount of time that it does not disassociate stations due to absence of frames received	Power saving and AP resource management
BSS transition management	An AP indicates a set of preferred APs to a station for a transition or request it to reassociate to a given AP	Load balance and handover enhancement
Channel usage	The AP recommends channels to a station for non-infrastructure networks	Interference avoidance
Collocated interference reporting	A station can get information about interference level at another stations, so that its own transmissions minimize the effect of interference from other radios at the measuring station	Interference avoidance
Diagnostic report	A station can request other stations on hardware, configuration and capabilities to diagnose and solve problems in the network	Resource management and troubleshooting
Directed Multicast Service (DMS)	A station can ask the AP to send group addressed frames addressed to it as unicast frames	Multicast transmission
Event reporting	A station can request other stations to send a message upon certain events (e.g. transitions, security, log reports or link status)	Handover, troubleshooting, resource management
Flexible Multicast Service (FMS)	A station can request to receive group addressed frames at a different interval. Its implementation is optional	Multicast transmission, power management
Location services	Location information can be requested by the stations (Radio ResourceMeasurements) or provided by the AP	Resource management
Multicast diagnostic reporting	A station can provide statistics of the multicast traffic received successfully	Multicast transmission, resource management
Multiple BSSID capability	Several BSSIDs can use a single Beacon or Probe Response frame to announce its capabilities. Its implementation is optional	Resource management
Proxy ARP	An AP can indicate that a station will not receive ARP frames	Power saving
QoS Traffic capability	A station can announce its own capability to support QoS traffic of a given priority	Resource management
SSID list	A station can request information from a list of SSIDs, instead of sending several separate Probe Request frames	Resource management
Triggered STA statistics	According to a predefined threshold, stations can generate a statistics report	Resource management
TIM broadcast	A station can reduce the time that it is awake by receiving an indication of buffered traffic independent of the Beacon frame. Its implementation is optional	Power saving
Timing measurement	This service allows a station to have an accurate estimate of its own offset with respect to another station's clock	Synchronization
Traffic filtering service	An AP, upon request by a station, can filter the traffic it sends to the station discarding the traffic that does not match the imposed criteria	Power saving, resource management
U-APSD coexistence	APs and stations can agree the most likely interval to transmit data avoiding interference	Interference avoidance, resource management, power saving
WNM-Notification	Stations can notify to each other a management event. The only event defined is the firmware update notification	Resource management
WNM-Sleep mode	A station can notify the AP the amount of time that it will be in sleep mode. Its implementation is optional	Power saving, resource management

modes.

Wireless network management has the objective of increasing the knowledge of every station about the status of the network. This amendment provides a mechanism for the stations to exchange information and use it to manage their own configuration, contributing to a general performance enhancement. The new services that this amendment introduces are summarized in Table 7.2. The exchange of data involved in the new services takes place through modifications to the format of Management frames and defining the format of Action frames for Wireless Network Management. It is straightforward to relate some of these services with the measurements introduced by 802.11k, such as the Location report or the station statistics collection. The common line between these two amendments is to increase the knowledge of the stations about the network conditions, enabling them with the ability to manage their own resources to enhance the overall performance, or let the AP manage the resources in the most convenient way. Furthermore, they are not only related, but 802.11v also amends 802.11k, for instance, modifying the

content of the Neighbor report. In addition, it complements the information provided for the Location service, including more complete and accurate information about how a station can determine this data. Moreover, this amendment is also aligned with the fast transition between BSS defined by 802.11r. Therefore, the three amendments can be considered together, for instance, to improve handover performance [111].

Note that, as in the case of 802.11k, the changes introduced by Wireless Network Measurements can be implemented in software. For instance, a station may use an application to track its location and present it in the correct format for the Location service, or apply some processing to the received frames to check whether they match the filtering criteria (e.g. source IP address or destination port). By implementing 802.11v, a station can save energy and obtain information about network events, whereas in addition to that, an AP can gain more control over the network resources in the BSS. It is worth to mention, however, that the way in which the station is able to obtain the information to be exchanged in the different services, as well as some information being used (for instance, the multicast groups of which a station is a member) are left out of the scope of the standard, being the manufacturer the only responsible for that. Therefore, this may have an impact on the commercial adoption of the amendments into final products.

7.4 Impact evaluation

The amendments on which we focus introduce interesting features to help optimizing performance in the BSS. If they are applied and most of the stations and APs support them, the management of the resources and the increased knowledge about the network status can lead to better adapt WiFi technology to next generation networks. Note that 802.11k, 802.11r and 802.11v have significant synergies and can be used jointly to enhance handover and minimize traffic interruption, while coordinating and optimizing the transmission of the rest of stations and APs within the ESS. In this section we are going to evaluate the penetration of these amendments both in the research community and in the implementation on wireless devices being commercialized nowadays.

7.4.1 Impact on the research community

The research community has directed its attention to the features introduced by these amendments even before they were officially approved. Just to give an estimation of their impact on the research carried out in wireless networking, Figure 7.1 shows the chronological evolution of the publications related to the three amendments evaluated over the years. The figure does not aim for completeness, but it shows the results for a search in *Google scholar* for the keywords 802.11k, 802.11r and 802.11v (last access: June 2014).

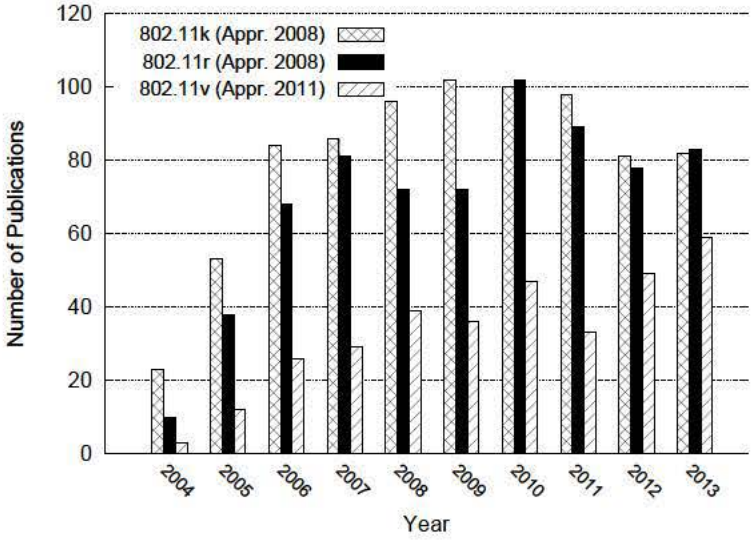


Figure 7.1: Presence of 802.11k/r/v amendments in research literature

Figure 7.1 shows that the amendment with less influence in the research works published is 802.11v, having fairly the same number of publications every year, with a small increase from 2012, the year after it was approved. On the contrary, 802.11k and 802.11r are very much aligned with each other and often used together [112], mainly because the fast transitions can make use of the radio resource measurements and they were approved with only 1-month difference. Despite the fact that 802.11v extends the radio resource measurements to offer a more complete solution, it calls our attention its little relation with 802.11r in the literature, while most of the attention has been addressed to improving multicast performance, for instance: studying the Flexible Multicast Service (FMS) [113], the Directed Multicast Service (DMS) [114] or extending the standard proposal to design new leader-based multicast solutions [115].

Although the impact of these amendments in the research literature has not been remarkable, the interest in improving performance of WLANs is a hot topic, specially in terms of handover latency, minimization of connectivity interruption, network discovery (e.g. scanning phase) and dense wireless deployments. Current activity in the IEEE 802.11 group targets the discovery of new services prior to association (Task Group aq), the reduction of association time (Task Group ai) and the increase in network efficiency (High Efficiency WLAN, HEW, Study Group), in terms of spectrum usage, interference avoidance or dense heterogeneous scenarios.

7.4.2 Impact on commercial implementations

The arrival of these three amendments to commercial wireless devices has been gradual. Similarly to the case of the research works, 802.11k and 802.11r are frequently implemented jointly, while 802.11v is considered in most of the cases for the management

of fast transitions. Therefore, applicability of 802.11v is related to the one of its two predecessors, because it also extends 802.11k functionality.

The most renown implementation of amendments 802.11k and 802.11r is the one in Apple's devices, as they are included from their iOS6.[2] They remark that not all the hardware vendors support these features, and we believe it is not a coincidence that Cisco implemented 802.11k for assisted roaming in some of its devices. At the same time, Cisco enables fast transitions in its devices and includes customized solutions for easy configuration of Apple devices [116], as they are the most notable among the wireless clients implementing these amendments. Therefore, the joint efforts of these two powerful brands could position them in advantage of their competitors. In addition, 802.11r is already deployed in *eduroam*, the wireless worldwide access infrastructure for educational and research institutions. However, the need for an AAA infrastructure and its unsuitability for home deployments, which is a numerous application for 802.11 wireless access, play an important role against the wide adoption of 802.11r.

It is worth pointing out that the modifications introduced by these amendments can be adopted by software implementations. This is the case of Realtek drivers, which implement among others, 802.11k radio resource measurements. In addition, Cisco and Aruba implement solutions to help stations not supporting 802.11k improving their performance for BSS transitions.

Although most of the attention has been addressed to 802.11k and 802.11r, the joint usage of the three amendments can optimize performance. This fact has been recognized by Aerohive networks, which include support for the three of them in their software release 6.0 of HiveOS and HiveManager.[3] To account for the relevance of these three amendments, the WiFi-Alliance released a new certification, namely Voice-Enterprise. Voice-Enterprise addresses the requirements of networks of different size (enterprise environment, not just one AP and several stations) for supporting enhanced voice traffic, enabling fast transitions and managing security. This certification should allow for a widespread adoption of these three amendments in commercial products in the near future and their presence in a variety of scenarios. The three amendments are part of the current IEEE 802.11 standard, so the devices that claim to be compliant with this specification must comply with the protocol implementation conformance statement (PICS) proforma [11]. For the moment, the support of radio measurements, wireless network management and fast transitions is optional, which clearly influences their adoption in commercial products.

7.4.3 Factors to contribute to this impact

The impact of the three amendments considered hereby is uneven. On the one hand, 802.11k and 802.11r are present in a considerable amount of research works published in

[2] `http://support.apple.com/kb/HT5535`

[3] `http://www.aerohive.com/products/access-points/products/software-management/hiveos`

the recent years and they are being implemented in commercial wireless devices. On the other hand, the adoption of 802.11v is much less notorious. In the following we categorize the possible factors that contribute to their sparsely spread influence:

- **Complexity**: We argue that the first factor against the commercial implementation of the new amendments is time. The modifications that need to be made to the implementations already developed need to go through development, testing and debugging processes. In addition, not all the devices can support more complex operations than the ones they were designed for, accounting for an additional evaluation of the cost in complexity that any further modification would introduce taking into account that, for these amendments to be effective, both clients and infrastructure need to support them.

- **Hardware feasibility**: This should not be a reason against the adoption of these amendments, as most changes can be adopted in software. Therefore, most devices could be upgraded to include these functionalities. There are already some products implementing 802.11k and 802.11r, and we argue that 802.11v enhances and complements 802.11k, for we assume it is only a matter of time that its implementation is more widely spread among commercial wireless devices.

- **Timeliness**: The presence of amendments 802.11k and 802.11r in commercial devices is very recent. Taking into account that both amendments were approved very close in time, and that it has been already 6 years since then, we expect 802.11v to be adopted gradually, assuming it is just a matter of time to enhance its implementation before introducing it into the market.

- **Modification of previous existing products**: The adoption of these amendments implies support for new frame formats, and enabling existing products with enhanced capabilities to perform the measurements and managing their configuration. There is a considerable amount of wireless products being commercialized, so the penetration of new devices cannot happen in the short term.

- **Compatibility with non-supporting APs and stations**: Related to the previous point, every new wireless device, either station or AP, needs to ensure compatibility with already existing devices. Therefore, manufacturers and developers must support the new features introduced by an amendment, also making sure that devices not supporting these features can interact in the same network.

- **WiFi Alliance certification**: Certification of wireless devices plays an important role on their commercial impact, even more when it certifies a new feature maybe not supported by many vendors. The inclusion of these three amendments in the WiFi Alliance certification program (Voice-Enterprise) definitely contributes to their adoption.

7.5 Summary

We have summarized the functionality introduced by three amendments recently added to the IEEE 802.11 standard: 802.11k, 802.11r and 802.11v. These amendments address radio resource measurements, fast BSS transitions and wireless network management, respectively. Their purpose is very much aligned, and their support can influence the overall performance of the wireless network. We have introduced their functionalities, how they relate and how they can take advantage of one another, specially for enabling fast transitions and minimize the interruption of data traffic being exchanged by the roaming station. Moreover, we have presented an evaluation of the interest raised in the research community and the commercialization efforts around them. We have also evaluated the potential factors that contribute in favor and against their implementation. Our conclusion is that although they contribute to enhance network performance and toward next-generation networking concepts, their widespread adoption in still rarely deployed, as current wireless devices are not ready to support them and there are other amendments that bring benefits directly noticeable by the final user (e.g. 802.11n or 802.11ac) whose support is more urgently added to WiFi-compliant products. The standardization of these amendments has been ahead, foreseeing potential solutions to enable fast transitions and improve management of resources in the network. However, the full adoption of these amendments to commercial devices, has been delayed, as overall compatibility of every wireless device in the network needs to be ensured.

Chapter 8

Multi-interface energy savings

8.1 Introduction

The design of an efficient mechanism to have the cellular and the WLAN connections sharing the traffic load benefits both the network operators and the final users. By offloading the traffic from the cellular network to a WLAN, a network operator can reduce the load on its network and reuse the freed resources for users that cannot handoff their traffic. On the other side, offloading and flow mobility in general can also improve the user experience, although usually this part is overlooked since the main driver of it is to alleviate the operators problems. For example, the mobile users can experience a better quality due to the higher bandwidth that a WLAN can offer compared to 3G, or even use both interfaces at the same time in order to achieve a higher available bandwidth. Although 3G offload can be used to just refer to the simple handover of all IP flows from one interface, e.g., 3G to a secondary one, for instance, WiFi, the opportunities that this technology enables are maximized when a fine-grained flow selection is allowed. For example, an operator might prefer not to offload VoIP flows, due to the inherited difficulties in providing QoS guarantees on an unmanaged WiFi access, while video traffic might be always offloaded to a technology providing higher bandwidth. Throughout this thesis our research is centered on networks where the mobile terminal has different network interfaces, namely cellular (3G) and WiFi. In this chapter, we focus on the energy consumption due to having both interfaces operative at the same time. In addition, we provide insights of an extra-benefit to the end user that has not been previously studied, the increased energy efficiency at the terminal that can be achieved by access technologies such as WiFi, which consumes less energy than 3G. Through several measurements comparing the energy consumed by each interface on its different transmission states, we prove that flow mobility allows a longer battery lifetime.

8.2 Energy consumption assessment

Enabling flow mobility implies benefits both for the operator, that can save resources in the radio access, and also for the user, that is able to take advantage of an increased available bandwidth. However, in order to fully assess the suitability of this mechanism, it is essential to evaluate it in terms of complexity and of another component usually forgotten, its energy consumption. Energy consumption is specially critical for mobile devices and smartphones, which already suffer from battery-drain issues due to continuous and exhaustive use along the day. Despite the fact that 3G connection is heavily consuming the battery of the device, it is generally configured to be the default access connection and is almost always on. Therefore, in order to implement a flow mobility solution, it is reasonable to assume that in addition to this intensive usage of the 3G interface, we will need to add the energy consumption corresponding to additional network interfaces. Nevertheless, our experimental results show that the energy consumed by the 3G interface is higher than the one by the WLAN interface, so offloading the 3G connection helps reducing this consumption. Through this section we provide experimental results supporting the claim that flow mobility can also be beneficial for the user in terms of achievable energy consumption savings.

Modern terminals such as Android or iPhone smartphones do not allow by default the simultaneous use of 3G and WiFi interfaces. To overcome this issue and perform an experimental assessment of the energy cost derived from enabling IP flow mobility (i.e., use of multiple network interfaces at the same time) we perform real power consumption measurements on a multi-mode device, equipped with a WLAN IEEE 802.11a/b/g and a 3G UMTS (HSDPA capable) interface. In order to be able to control as much as possible the used devices, capture traffic sent and received at the network interfaces, as well as closely monitor the device, we decided to use a small residential router, Asus WL-500GP v1.0, based on a Linux firmware. The measurements provided through this procedure are later validated by the analysis of the battery lifetime of an Android smartphone while using 3G and WLAN interfaces separately. Finally we derive our main conclusions through a synthetic use case that allows us to provide quantitative gains on the percentage of battery spent through the use of the proposed flow mobility mechanism.

8.2.1 Power consumption of the join operation of 3G and WiFi

The following section is devoted to perform the experimental assessment of the energy consumption associated to our flow mobility solution. To measure the power consumption of each technology we have chosen a small residential router: the Asus WL-500g Premium. This router is equipped with a 266 MHz processor, an IEEE 802.11b/g WLAN interface and an IEEE 802.3 Ethernet interface connected to a VLAN capable 5-port switch. This version of the router has a mini-PCI slot that allows changing the original wireless card.

We replaced the original Broadcom card an Atheros based 802.11a/b/g (Alfa Networks AWPCI085S) one, which is supported by the Madwifi[1] driver. In order to mitigate as much as possible the impact of collisions and interference in the power consumption measurements, we avoided the 2.4GHz band (IEEE 802.11b/g) – which is very crowded in our lab, as reported in [117] – and configured the WLAN interface in 802.11a mode.

We replaced the original firmware of the router by installing a lightweight Linux-based version, which gives us more flexibility in the configuration. We choose the distribution Kamikaze 8.09.2 of OpenWRT[2] with a Linux-2.6 kernel and this allows the support of a 3G USB modem. In our tests, we used a Huawei E160 HSDPA USB stick.[3]

Power consumption was measured using a PCE-PA 6000 power analyzer.[4] Power measurements were carried out using a PCE-PA-ADP current adapter where the power supply of the router was plugged in. Measurement data was transferred from the power analyzer to a computer via an RS-232 interface for its processing.

Using this setup we performed the measurements described next. We first calibrated the power analyzer by measuring the consumption when both the WLAN and 3G interfaces are switched off. All reported results are relative to this level. For the actual measurements, we are interested in the power consumption when the network interfaces are in the following states:

- OFF: the interface is switched off.
- IDLE: the interface is on but it does not send or receive any data traffic. For the case of WLAN, this means that the card is associated to an access point (so the card is receiving beacon frames) without sending or receiving any user data traffic. For the case of 3G, this means that the interface is up, a PDP context has been activated and a PPP interface has been set up, but no data is exchanged.
- ON: the interface is on and engaged in a data traffic exchange. In our tests, this means that a file is downloaded from a server using HTTP. By using TCP, the card is receiving at the maximum available rate, and traffic is sent in both directions (downlink: mostly data segments, uplink: mostly TCP acknowledgments).

We measured the power consumption for the different states of the WLAN and 3G interfaces. Table 8.1 shows the mean and 95% confidence interval of the results extracted from five 300-second experiments. We focus on the scenarios where at least one of the interfaces is actively sending or receiving traffic, as those are the cases in which it is important to evaluate the energy cost associated with having a second active interface. This second interface may be either receiving or sending traffic or just idle, ready to

[1] http://www.madwifi.org/
[2] http://www.openwrt.org/
[3] http://www.huawei.com/mobileweb/en/products/view.do?id=1960
[4] http://www.industrial-needs.com/technical-data/power-analyser-PCE-PA-6000.htm

Table 8.1: Power consumption results

3G ON		WLAN ON	
WLAN OFF	1.80 ± 0.10 W	3G OFF	1.03 ± 0.08 W
WLAN IDLE	1.86 ± 0.08 W	3G IDLE	1.21 ± 0.16 W
WLAN ON	2.16 ± 0.13 W	3G ON	2.16 ± 0.13 W

operate. Results show that the 3G interface consumes more energy than the WLAN interface, but the difference between using only the 3G interface and using simultaneously the 3G and the WLAN interfaces is only of 20%. Note that this additional cost is only incurred when both interfaces are actively engaged in a data transfer, and that by using them simultaneously, the time required to send a given amount of data via WLAN would be shorter – since the throughput obtained via a WLAN network is typically higher than the one that can be obtained via a 3G network – and this would also contribute to a lower power consumption. The extra power consumption caused by activating the WLAN interface (IDLE state) is just around 3%, which besides would only be needed when the mobile is sending or receiving traffic, as it is then when the network operator and the user may benefit from offloading traffic from the 3G infrastructure to a WLAN hotspot, if available. It is important to highlight that the actual values of energy consumption of the device are not directly comparable with the results we would obtain with a smartphone, since the level of integration provided in such a platform is much higher, allowing further improvements in the energy consumed by the device. Due to this, and to be able to compare, we only focus on the relative difference between the 3G and WiFi consumption profiles which, as we will see in the next section, follow the same trend in both device families.

8.2.2 Energy consumption profiles of 3G and WiFi in an Android device

In order to confirm the results obtained in section 9.3, we measured the battery duration of an HTC Legend device. The operating system of the device under test is Android 2.1 (Eclair). Apart from the device, we used a desktop to monitor and configure the parameters of interest in the mobile terminal. We also configured the WLAN interface of this desktop as Access Point to which the mobile terminal would associate. To measure the energy consumption we developed an application running in the background to monitor the battery continuously, keeping track of the voltage level in order to compute the power consumed. As this application is a service running in the background, it consumes negligible CPU resources, minimizing the impact on the energy consumption measurements. In addition, no interaction with the user (or the tester in this case) is required, as all the information is saved to a text file. All the measurements have been performed keeping one element active and the rest inactive, in order to isolate the contribution of each individual element to the total power consumption of the device.

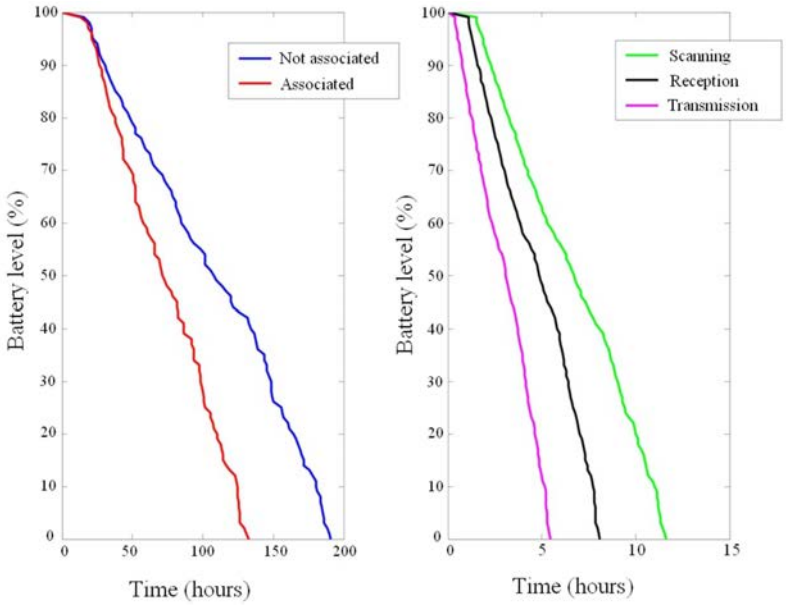


Figure 8.1: Battery drainage for the different WiFi states.

Figs. 8.1 and 8.2 show the battery drainage curves for WiFi and 3G respectively. Figure 8.1 presents results for each of the possible states of the WiFi interface, as explained in section 9.3, while Figure 8.2 considers only the states we can control on the 3G interface, namely: transmission, reception and disconnection states. The results from these measurements show that the battery life of the device is much shorter when the 3G interface is on than when the WLAN interface is at its maximum battery consumption task, which is transmitting packets. The battery of the mobile terminal can last 200 hours when the 3G interface is inactive, against a duration of less than 5 hours when there is incoming or outgoing traffic. However, in the case of the WLAN interface, the difference between the transmission and reception states are much more evident than those of the 3G interface, shrinking from 8 hours to 5 hours, respectively.

In addition, through the analysis of the slopes of each curve, we can obtain the relative difference between the cases of a single active interface (3G) and the case when both interfaces (3G+WiFi) are simultaneously used. Supposing a transmission and reception cycle of a 50%, the overall difference between both cases is approximately 15%. As expected, the relative difference between both cases is lower for the smartphone device, we argue that this difference is due to a higher integration of the components in the smartphone compared to the router.

Finally, and in order to conclude this analysis, let us make a synthetic example of the energy saving that such an approach would provide to the end user. First let us consider some assumptions, for the sake of the simplicity of this analysis, which aims at assessing if a typical mobile user could afford the additional power consumption introduced by the use of flow mobility extensions. Several studies, such as [118], point out that users of

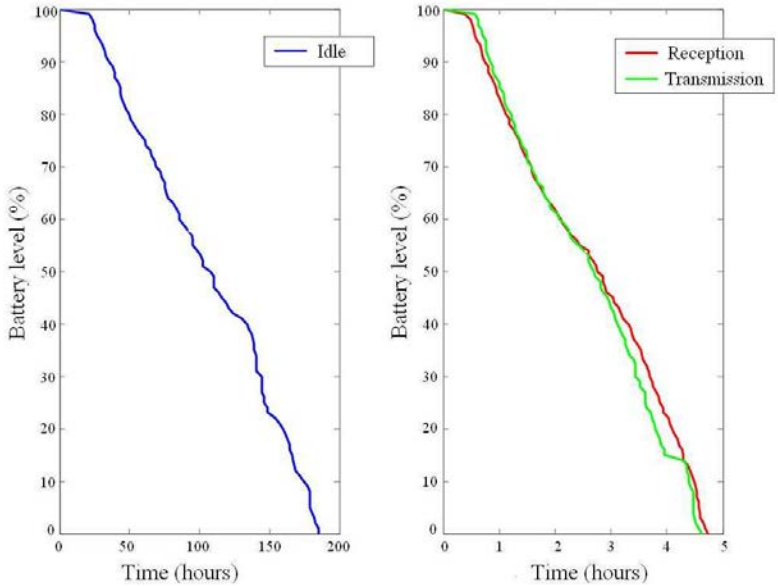


Figure 8.2: Battery drainage for the different 3G states.

smart hand-held devices download an average of 20 MBytes per day via 3G. Considering Figure 8.2, and an average 3G speed of 1 Mbps, the download would take 160 seconds and consume around an 0.8% of the battery[5]. In case a flow mobility solution was deployed and the terminal was able to use WiFi to download the same amount of information, it would use the WiFi interface for approximately 6.4 seconds (assuming IEEE 802.11a extended rates and a real throughput of approximately 25 Mbps). During this time, the terminal will use a 15-20% more energy compared to the case of using only 3G, but the overall time would be highly reduced. This implies that the terminal would have spent less than a 0.1% of the battery downloading the file. This simple analysis does not aim at providing rigorous and precise figures, but just at roughly assessing if a flow mobility solution is affordable from the perspective of power consumption. Based on the obtained results, we can conclude that selectively using more than one network interface results in an affordable additional cost.

From these experimental results we can derive that the use of the WLAN interface is considerably more efficient in terms of energy consumption than the use of the cellular 3G connection. In addition, the throughput and the achievable bandwidth by using a WLAN access are also higher than the ones that the 3G connection can offer. Therefore, we can take advantage from the higher bandwidth offered by the IEEE 802.11 access network, offloading the cellular connection and freeing resources for other users while reducing the energy consumption of our devices.

[5]This value matches perfectly with a real measurement of the power consumed by an iPhone 3GS downloading a 20 MBytes file.

8.3 Summary

In this Chapter we have focused on the IP flow mobility and we analyze it from the perspective of energy efficiency. We have identified the advantages that flow mobility brings both for the network operator and for the end user, and in order to argue the energy consumption we present some experimental results on commercial devices. Networking research is evolving towards a greener framework, analyzing the causes of battery draining and searching for optimizations or new solutions that allow to diminish the power consumption of networking protocols and communication devices. These results enable us to claim that the flow mobility solution provided is also affordable in terms of battery consumption, which is a key element under study in the research community.

Chapter 9

Connectivity management in smartphones

9.1 Introduction

The use of smartphones has also modified the traditional patterns of mobile data consumption. The typical smartphone user is craving for data-based services, imposing a high burden on the operators, which see their investments on network deployments pushed to the limits due to greater bandwidth requirements. Due to the shift in user profile and data service demand experienced in the recent years, smartphones have become a powerful tool in most people's daily life. In addition, the enhanced capabilities and fast upgrades of hardware in handheld devices have considerably increased their usage. These facts pave the way to advanced research and development that relies on the use of smartphones for carrying out innovative tasks, mostly related to health care or behavioral studies and using the smartphone as a measurement instrument [119]. Moreover, there is a trend towards specialized, almost personalized services, which could benefit from an accurate knowledge on the capabilities supported by smartphones and how they manage their resources.

The smartphone operating system provides a set of mobility-related functions from which an application can benefit in case it decides to handle mobility. These functionalities depend on the operating system (e.g., Android, iOS, Windows Phone 8) and include connectivity events such as network up or down events, and commands exposed to the application layer to extract information on connection availability. Usually the terminal connectivity is handled by a service widely known as the *Connection Manager*, in charge of deciding which is the best connection for the terminal in a specific moment, and the application has to deal with those decisions.

This Chapter presents the analysis of the current state of the art on the mobility support at the Connection Manager in different terminals, providing a functional view of the differences between the major operating systems and the different improvements that

can be done to optimize the mobile user experience. Our main contributions are:

- We analyze the default network connectivity management of the three currently most popular families of mobile OSes: Android, iOS and Windows Phone 8,[1] including iOS8 and Android Lollipop, in their latest versions supported to date. We test the same OS versions in different terminals, to avoid biased conclusions, derived from the performance of the terminal rather than from the OS behavior.

- We study how the smartphones running these different OSes perform inter- and intra-technology handover, considering the most widely used access networks: cellular and IEEE 802.11. For this study, we measure the handover latency in different scenarios and we evaluate the differences and similarities in the management and configuration of the networking parameters in each device.

- We evaluate how the handover performance affects the user experience by considering different applications, and whether they can survive to a change in connectivity: changes in IP address and changes in access technology.

- As a result of our experimentation, we identify the challenges and open issues that are present in current smartphones and discuss on potential improvements for the connectivity management that are feasible but not yet implemented and on the integration of connectivity management with current mobility protocols.

9.2 Mobile terminal networking stack

Even though the developer community for the three families of OSes is considerably large, there is no official documentation on the networking stack and the network management of the system. The effort of the community is focused on the application layer, thereby the main interest of a developer is centered at checking whether Internet connection is available, rather than making an efficient usage of the networking resources and optimizing performance. Still, we identify the five most representative elements that define the network management in Android, iOS and Windows Phone 8, and we introduce them comparatively in this section.

9.2.1 Android

Android is an open source software stack released by Google, and publicly available under Apache or GNU General Public Licenses. It is based on Linux kernel 3.X (kernel

[1]Product names, logos, brands and other trademarks are registered and remain property of their respective holders (Google Inc., Apple Inc. and Microsoft Corporation). Their use throughout this thesis aims only at describing the results and the work performed and in no way it indicates any relationship with the holders of such trademarks.

2.6 in versions up to Android 4.0). On top of the Linux kernel, you can find libraries, [2] and the Android runtime. The Android runtime consists on a Dalvik virtual machine –where the applications run– and on core libraries, specific for Android devices and used by the applications. In Android Lollipop, the new Android Runtime (ART) replaces Dalvik by default. An application framework interacts between the lower layers and the applications, which are on top of the system architecture.

1. **Default interface**: the Android API provides tools for an application to configure its preference on a default interface. It is important to isolate this behavior from the terminal's own networking preferences. The default interface in Android is the cellular one, and simultaneous active connections over the cellular and the WLAN interfaces are not supported in versions previous to Lollipop. However, it is possible to modify the default Android behavior to use both interfaces at the same time [61]. In Android Lollipop, each interface has its own routing table and the cellular connection is kept for 30 seconds after switching to WiFi. The ongoing communications started over the cellular interface will remain there. In addition, the terminal will not connect to a WiFi AP that has no Internet connection.

2. **WLAN interface**: the hardware abstraction layer (HAL) contains the software modules that talk directly to the kernel wireless stack and drivers. The HAL is a user-space layer developed in C/C++ used by the application framework to interact with the *wpa_supplicant* module,[3] which runs in the background and controls the wireless configuration.

3. **Network related events**: Android offers, by means of the Android Debug Bridge (ADB), the possibility of tracking system logs and monitor different system events. ADB is a command line tool to communicate from a computer to emulated or physical Android devices. For example, we can monitor the active network connection (WIFI or cellular) and its coarse-grained status (*disconnected*, *connecting*, *connected*, *disconnecting*, *suspended*, *unknown*). In the case of the WiFi connection, it is also possible to monitor the state of the *wpa_supplicant* module, which also reports changes on its status (*associated*, *associating*, *authenticating*, *completed*, *disconnected*, *dormant*, *four_way_handshake*, *group_handshake*, *inactive*, *interface_disabled*, *invalid*, *scanning*, *uninitialized*). This information is also made available to application developers by the API framework.

4. **Application Programming Interface (API)**: the Android API framework is

[2] Commonly found also in Linux systems. Note that not all the Linux libraries are fully supported by Android and some changes have also been introduced to the architecture, such as *wake locks* and power management.

[3] `http://w1.fi/hostap.git`

published in API levels.[4] Each API level is an integer that identifies the API revision (current API level 21). Applications must support the API level for the specific Android version, and they usually are backward compatible. Network access is handled by the Android core libraries. As we have previously stated, the API provides applications with information of the network connection status and the HAL and kernel modules access to the network interfaces and are in charge of the (re)configuration. However, the network management still keeps a simplistic or conservative approach, as it is very limited in terms of optimization or using the two access network interfaces at the same time. Interestingly enough, the API provides constants and methods to check whether the signal strength has changed, to know when a scanning has been performed and information about surrounding APs is available, compute the difference in signal strength or change the state or configuration of the WiFi connection, so that applications could make a much wiser usage of the network connectivity.[5]

5. **Flexibility and restrictions to the user**: Android, being Linux-based, offers a high degree of flexibility in configuration and networking support. In addition, the lax licensing has favored the distribution of customized versions of the firmware.[6] By default, Android users are not given root access, but root access can be configured, empowering the accessibility to all the terminal features. The root access and the Linux characteristics make the Android OS the most flexible and accessible of the systems under study. In this way, we have full access to the system logs record, we can monitor the status of the network interfaces and we can capture traffic with a network analyzer such as *tcpdump*.[7]

9.2.2 iOS

iOSn is the operating system running on Apple mobile devices, being n the version number, currently the latest version is iOS8, released in September 2014. iOS is based on the open source OS *Darwin*, however, iOS remains as closed-source. It is built upon 4 abstraction layers, ordered from top to bottom: *i)* Cocoa Touch layer, *ii)* Media layer, *iii)* Core Services layer and *iv)* Core OS layer. The Core OS layer has access to the kernel, drivers and networking features as the interface to BSD sockets. However, developers are recommended to implement applications by using the framework in the highest level as possible, as the complexity of handling networking events or configuration is hidden by

[4]Not necessarily a new Android version has to support a new API level, but commonly a new version upgrades the API too.

[5]The interested reader is referred to `http://developer.android.com/reference/packages.html` for a complete guide of the Android API framework.

[6]One of the most popular ones is CyanogenMod, which reports more than 12 million installs (`http://www.cyanogenmod.org/`).

[7]`http://www.tcpdump.org/`

the OS.

1. **Default interface**: the cellular interface is assumed to provide always-on connectivity, but as soon as a wireless AP appears in range, the WiFi connection takes over the cellular connection.

2. **WLAN interface**: the WLAN interface is selected as the primary interface, over the cellular connection.

3. **Network related events**: the development platform does not offer access to the lower layers that talk directly to the hardware. However, the application can react to network connectivity changes, for instance, by means of the *SCNetworkReachability* API (available in the System Configuration framework in Core Services API) to diagnose the cause of the failure of a connection and determine availability of different connections.

4. **Application Programming Interface (API)**: iOS provides three different networking API layers: Foundation layer, Core Foundation layer (these two are specific to iOS) and POSIX layer (as in other UNIX systems). The three of them support common networking tasks, being recommended to use the highest level API that fulfills the developer's requirements. The networking API provided by iOS7 can be categorized into three main groups: BSD sockets, web access and Bonjour. The interested reader is referred to [120] for further details.

5. **Flexibility and restrictions to the user**: the iOS system is closed source thereby the user is not given much flexibility of configuration with respect to network preferences. Even though the operating system has been hacked and it can be *jailbroken* it does not change the networking behavior or configuration, but increases flexibility application-wise.

9.2.3 Windows Phone 8

Windows Phone 8 (WP8) is developed by Microsoft, and, as the OS version for PC, they changed completely what was established in previous releases. Therefore, due to a change in the architecture, applications designed for WP8 cannot run in previous OS versions or devices running an older version cannot upgrade to WP8. As part of the features that Microsoft tried to improve in this new version, they include the network stack, focusing on speeding up the connection and reducing power consumption.

1. **Default interface**: one of the main changes in the networking stack in Windows 8 is the prioritization of network connections. Despite having a priority, multiple interfaces can be connected simultaneously. By default, the cellular interface is the

one that is actively connected, but in the moment that an already visited WiFi network becomes available, the phone will try to connect to it. However, the connection through one interface does not kill a previous connection in another interface. It is only after a short period of time that the cellular connection will be terminated, unless it is being used by an application. In addition to that, existing connections at the moment of the new attachment, are kept alive and only the new connections will use the new interface. WP8 even offers the applications the possibility to select the network interface to use.

2. **WLAN interface**: the wireless network stack builds on top of the hardware device (and its firmware) with the driver and the Wi-Fi service of the OS. With this new generation of their OS, Microsoft targets to optimize power consumption and connection delay, which worsens user experience. In order to do so, they have tried to integrate as much operations as possible into the hardware layer.

3. **Network related events**: as mentioned, the access to the configuration of the WP8 device is more restricted to the user, providing no information on the current connections or the networking events, other than enabling or disabling a network interface and changing the priority of a WiFi network in the list of preferred networks.

4. **Application Programming Interface (API)**: WP8 exposes a set of APIs that comprises the ones for previous releases (WP7) and the ones that the manufacturer recommends for applications built from scratch for WP8. We are only going to mention here the ones recommended for this system, which are *.NET HttpClient* and *WinRT Sockets*, and the so-called "Native code" (*IXML HttpRequest2* and *WinSock*). Although there are several development frameworks, we can differentiate two parts: the one devoted to the web browsing (HTTP-related) and the one directly related to the network connections (sockets). Despite the restrained flexibility, WP8 offers its applications information about the network connectivity and when a change in connectivity occurs. An application can even set preferences on the use of one interface over the other. Moreover, the emulator provided by the development framework can simulate changes in the network connectivity to test the application under different scenarios.

5. **Flexibility and restrictions to the user**: WP8 tries to optimize mobile user experience and is designed considering the performance of current devices (touch-screen, portable devices, wireless connectivity). However, this optimization, achieved by implementing a more integrated system, speeds up some processes but no involves more freedom to the user in configuration nor accessibility. The manufacturer's claim to favor this design is that the user just wants to be connected, but

does not care about how they get connected.

9.3 Experimental setup

Table 9.1: Main characteristics of the analyzed smartphones

	LG Nexus 4 E960	LG Nexus 5	iPhone 3GS	iPhone 4	iPhone 5	HTC 8S
OS Version	Android 4.2.2, 5.0	Android 5.0	iOS 6.0	iOS 7.0.4	iOS 8.1.2	Windows Phone 8.0
Chipset	Qualcomm Snapdragon S4 Pro APQ8064	Qualcomm MSM8974 Snapdragon 800	Samsung APL0298C05	Apple A4	Apple A6	Qualcomm Snapdragon S4 Plus MSM8227
CPU	Quad-core 1.5 GHz Krait	Quad-core 2.3GHz Krait 400	600 MHz Cortex-A8	1 GHz Cortex-A8	Dual-core 1.3GHz Swift	Dual-core 1GHz Krait
RAM	2GB	2GB	256MB	512 MB	1GB	512MB
WLAN	Atheros WCN3660 Murata SS2908001	Broadcom BCM4339 (5G WiFi combo)	Broadcom BCM4325	Broadcom BCM4329	Murata 339S0171 Broadcom BCM4334	Atheros WCN3660

This section describes the characteristics of the mobile terminals under test and provides an overview of the different experiments. We aim at studying the connection manager in several mobile devices running the most representative OSes – iOS, Windows Phone and Android. The characteristics of the smartphones in our experiments are collected in Table 9.1. We have included in our analysis the latest versions available of the OSes, including iOS8 and Android Lollipop. We have tested the same operating system version running in different terminals, so we can avoid dependency on the terminal rather than on the OS. For the sake of fairness we have avoided performing any modification to the terminals.

We deploy two IEEE 802.11 Access Points that provide Internet access. These Access Points are under our complete control to keep track of the behavior of the terminals attached to them and to be able to modify network parameters, such as the ESSID (Extended Service Set Identifier), the wireless channel in which the Access Point (AP) operates and the IP subnet managed by the access router.

We intend to characterize the Connection Manager on the systems under study, identifying the main strengths and weaknesses of network connectivity management and comparing their behavior. Our analysis focuses on understanding the following mechanisms:

1. **Initial attachment procedure to an 802.11 network**. Regarding this mechanism, we aim at understanding: *i)* how the attachment to a WLAN is carried out by every device, analyzing the differences among them, if any, and, *ii)* how the network selection algorithm works and what criteria are used to choose among the different candidate networks. This is explained in Section 9.4.1.

2. **Initial configuration of the protocol stack**. Once the mobile terminal has attached to a point of access, we aim at understanding the main steps and the protocols used to complete its networking stack configuration. Note that, if this procedure takes place entirely whenever there is a change in the point of attachment

to the network, and not only as an initial configuration, it may enable potential optimizations for the handover process. This operation is explained in Section 9.4.2.

3. **Horizontal handover**. We examine the handover procedure between two IEEE 802.11 APs. We play with different network parameters to have a wide view of the performance for the different mobile terminals. Specifically, the current AP and the target one may have the same or different ESSID, operate in the same or different channels and manage the same or different IP subnets, in which case the handover would imply also a layer three reconfiguration. Through this analysis, we aim at knowing whether there are any dominant factors when the mobile device changes its point of attachment to the network and to what extent the different changes impact the configuration and connectivity management. The horizontal handover is explained in detail in Section 9.5.1.

4. **Vertical handover**. We evaluate the handover procedure when it involves a change in the access technology. We aim at understanding how the mobile devices handle the inter-technology handover, whether they can keep both technologies operative simultaneously and whether they handle the survival of ongoing connections. Characterizing the inter-technology handover is essential to design potential optimizations and flow mobility solutions. However, due to restrictions by the terminal and the network operator, we were not able to obtain direct measurements from the cellular interface. The vertical handover results are presented in Section 9.5.2.

5. **Application behavior**. We study application survival to handover in the cases already explained. The objective is to know the perception of the user when an application is running and there is a change in connectivity. In addition, we get to know whether applications can handle an interruption due to horizontal or vertical handover seamlessly. These experiments are presented in Section 9.5.3.

We start our analysis in Section 9.4, with the evaluation of the initial attachment procedure to an IEEE 802.11 network.

9.4 IEEE 802.11 Initial attachment procedure

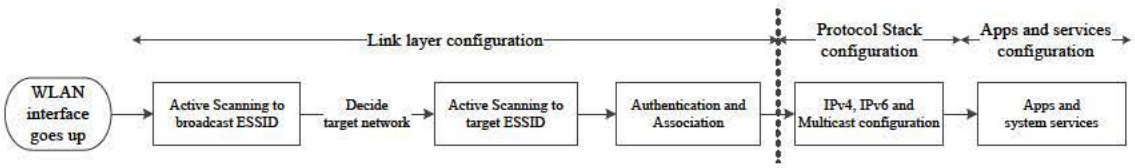


Figure 9.1: Initial attachment to the WLAN.

In order to test the default attachment to an IEEE 802.11 [11] WLAN, we deploy a wireless access point and run thirty experiments for each smartphone. In each experiment, which lasts for 60 seconds, we monitor the traffic by means of a network analyzer.[8] The monitor interface is located close to the access point, so we can likely capture all the frames involved in every exchange. Initially, the WLAN interface of the mobile terminal is down, so we do not miss any packet or reach misleading results because of having the device already connected to another network. We start our experiment by bringing up the interface and checking that the device actually connects to the AP under our control. This experiment describes the case in which the terminal finds an already known network and connects successfully. Note that to connect to a network for the very first time the user must identify and select manually the network to connect to.

Figure 9.1 presents in a diagram the different steps performed during the initial attachment to the IEEE 802.11 network. Note that this is a general diagram and it does not try to highlight the differences among the terminals, but to illustrate the common procedures followed by all of them. In the following, each of the steps in the initial attachment is explained in detail, highlighting the differences among terminals.

9.4.1 IEEE 802.11 attachment

Active scanning to broadcast address: when the WLAN interface of a mobile terminal goes up, it detects all the surrounding wireless networks available by initiating an active scanning procedure. The terminal sends Probe Request frames to a wildcard ESSID (and to every previously visited ESSID) in every channel sequentially. Neighboring access points will receive these Probe Requests and answer with Probe Response messages, indicating their capabilities and providing synchronization information. This active scanning phase differs slightly in the systems under study.

By monitoring the traffic in different channels we are able to measure the delay induced by the scanning procedure. The results show high variability in the amount of Probe Requests being broadcast as well as in the time interval between them, but we can identify different patterns:[9]

- The Android terminal scans approximately every 10 seconds in every channel, sending a number of consecutive Probe Requests inter-spaced approximately 15 ms.

- The Windows terminal spends approximately 6 seconds between consecutive scans in the same channel. The time lapse between consecutive Probe Requests is very variable, but shows two different dominant values: the terminal sends several requests every 6 ms and one after 60 ms to continue with 6 ms interval again.

[8] http://www.wireshark.org/

[9] Due to the high variability we just provide rough numbers.

- The iPhone terminal presents an interval of approximately 9 seconds between the scan in every channel and the delay between consecutive Probe Requests is around 20 ms. As the WP8 terminal, there are several requests spaced 20 ms (much higher interval than WP8) another one 700 ms after that, to continue sending every 20 ms again.

It is also interesting to evaluate the behavior of the terminals in channel 14, which is not allowed in Europe, where we are based. The Android and Windows phones do not list the networks operating in that frequency as available, but the iPhone does, regardless of the regulatory domain. However, the three terminals send Probe Request frames in every channel including that out of the allowed frequency band (at a different interval than in the other channels, though). Noteworthy, Android and Windows devices do not give the user the opportunity to attach to these networks although they actively scan that channel.

The scanning policy followed by a terminal has several potential effects. First, the number of Probe messages sent during the scanning phase impacts directly the energy consumption. Second, a terminal can only obtain information regarding the signal level from the AP by the frames received, hence receiving more responses before deciding the target point of attachment can be beneficial. Last, the delay in the attachment to a WLAN AP is directly influenced by the time spent in the active scanning, and so is the handover delay if the terminal has to scan again before connecting to a new AP.

Target Network Decision: this step corresponds to the actual decision on the AP to connect to. It seems reasonable to expect the terminals to perform some kind of complex algorithm considering, for instance, the signal level received from the different APs. After the analysis, we have discovered through our extensive tests that all the terminals use a simple rule to decide where to connect to. If the AP used in the immediately previous connection is available, the terminals will connect to it, no matter its signal level. In case the last visited AP is not available, the terminal connects to the one previous to the last one, and so on. In the case secured and open networks are available, only the iPhone terminal shows a preference for secured networks if the immediately previous connection is not possible. Note that we have confirmed this behaviour also with several wireless networks that are available at our laboratory, not only the two APs we deploy in our set of experiments. It is worth to note that the Android phone includes an option in the WiFi settings to connect to a different WLAN or to the cellular network if the signal is weak. However, even if this option is enabled, the terminal follows the same approach and disregards signal strength information to connect to an AP. This way of choosing the target network leads to weak connections and poor performance.

Active scanning to selected ESSID: in this step, the terminal addresses a Probe Request message directly to the AP selected previously and indicates the target ESSID in the correspondent field in the frame.

Table 9.2: Initial attachment delay

OS Version	Link layer delay (s)	Network layer delay (s)		Total (s)	
		IPv4	IPv6	IPv4	IPv6
Android 4.2 Nexus 4	0.42 ±0.12	1.09 ±0.31	4.75 ±0.43	1.51 ±0.33	5.17 ±0.45
Android 5.0 Nexus 4	0.47 ±0.05	0.68 ±0.36	9.58 ±3.53	1.15 ±0.33	10.05 ±3.54
Android 5.0 Nexus 5	2.58 ±0.05	0.50 ±0.04	3.33 ±1.15	3.09 ±0.06	5.90 ±1.14
iOS6	1.91 ±0.08	0.13 ±0.01	1.16 ±0.06	2.05 ±0.07	3.08 ±0.14
iOS7	1.98 ±0.26	0.13 ±0.02	1.12 ±0.58	2.12 ±0.26	3.1 ±0.67
iOS8	2.28 ±0.47	0.25 ±0.11	0.87 ±0.25	2.54 ±0.45	3.25 ±0.51
WP8	1.08 ±0.28	1.1 ±0.12	1.28 ±0.15	2.18 ±0.30	2.36 ±0.32

Authentication and Association: these are the last two steps before being attached to a WLAN AP. Both of them are standard procedures and are equally executed in the different terminals. Security procedures are out of scope as the main focus in this work is on the Connection Manager.

Table 9.2 gathers the average and standard deviation of the delay during the initial attachment to a WLAN measured for the different OSes versions. We distinguish between link layer and network layer configuration, which in turn is measured for IPv4 and IPv6 configuration, in order to provide more information on the influence of both processes. The link layer delay is measured as the time from the interface going up until the terminal is associated to the AP. The scanning policies presented previously influence considerably these delays. The Android terminal clearly outperforms the rest of the systems. Although it sends a higher number of frames before attaching to the selected AP, these frames are more frequent, whereas the other systems have a longer interval and therefore, the association is delayed. However, Android Lollipop has clearly impaired performance in the connection for Nexus 5 devices (well-known issues are being reported by Nexus 5 users since updated to Lollipop). The WP8 delay doubles that of Android 4.2 and Nexus 4 Lollipop, although the scanning time in every channel is lower for Windows. There is no clear difference in performance between iOS6 and iOS7, so this process seems not to have suffered major modifications from one version to the other. Still, the latest version, iOS8 increases the delay, as it happened with the Android update. We have had access to a new iPhone 6 and been able to perform the same tests, finding no difference, on average, in our measurements with respect to the ones in Table 9.2, which correspond to an iPhone 5. It calls our attention the usage of *CTS-to-self* frames sent by Android Lollipop in Nexus 5 and iOS8 in iPhone 6, but not in previous terminals running the same operating system. Moreover, Nexus 5 just sends one frame right before the authentication frame. The process and the delay for IP configuration is explained in detail in the next subsection.

9.4.2 Protocol stack initial configuration

IPv4, IPv6 and Multicast configuration: we group these configuration steps because they are very similar for the evaluated systems. Table 9.2 shows the delay during the initial attachment due to the configuration of an IP address in the wireless interface of the mobile terminal, once it is associated to the AP (under the column "Network layer delay"). In the case of IPv4, the different terminals follow the same mechanism and use DHCP (Dynamic Host Control Protocol) [121] to configure the IP address when they attach to the network. In our experiments, the DHCP server is located in a node belonging to the same network but different from the AP. The iPhone terminals are the fastest ones in this case because they start the process much earlier, while Android 4.2 (ICS) and WP8 present a very similar delay, close to one second over the one from iOS. Contrarily to the attachment to the WiFi AP, the IPv4 address configuration has been made faster in the new Android version. Another interesting difference is that the new iOS version, iOS8, uses gratuitous ARP in the IPv4 configuration when the wireless interface goes up, for both iPhone 5 and iPhone 6. Neither Android nor WP8 do that.

Regarding IPv6 configuration, the WP8 device tries to configure an IPv6 address by means of DHCPv6 [122], but as we have no DHCPv6 server in our network, all the terminals configure local and global IPv6 addresses following the SLAAC (Stateless Address Auto Configuration) procedure as specified by [123] and [124] for configuration and DAD (Duplicate Address Detection). First, the mobile node acquires a link local IPv6 address and joins the all-nodes and the solicited-node multicast addresses when the connection is established in the interface. Then, in order to perform the DAD the terminal sends a Neighbor Solicitation message to the solicited-node multicast address. As the source address of this message is the unspecified address, any other node will not respond to that message and will identify the tentative target address and know that address cannot be used. Whether the node itself receives its own Neighbor Solicitation depends on the particular implementation of the multicast loopback. In the case of iOS devices, the delivery to upper layers of their own multicast message is disabled, so, if no other node in the network has the same IPv6 address as the target one, no Neighbor Advertisement from any other node will be received and the mobile terminal will silently configure the interface with the target IPv6 address. However, the Android and Windows terminals send out a solicited Neighbor Advertisement with their own source address upon receiving their own Neighbor Solicitation messages to announce this configuration. The Android device unicasts the advertisement to the router, while the Windows device broadcasts it to all the nodes.

In light of the results in Table 9.2, the delay for the IPv6 configuration is comparable in the iOS and Windows Phone systems, but the Android device takes significantly more time, which delays any IPv6 connection. It is worth to mention that the three families of mobile OSes follow [50] considerations to protect privacy. According to SLAAC rules, the

IPv6 address is configured from an interface identifier (EUI-64 identifier), and the second half of the global IPv6 address (without considering the 8-byte prefix announced by the router for configuration) is the same regardless of the location, so the device could be tracked. Therefore, IPv6 privacy extensions are defined so the network interfaces are configured with randomized strings, which change over time, instead of the interface identifier in order to complicate the activity correlation. RFC 7217 [51] provides the specification for the generation of these random interface identifiers while keeping IPv6 addresses stable in each visited subnet. Typically, the address derived from the EUI-64 identifier is kept, in addition to a temporary address built upon randomized identifiers. In our tests, we have observed that only the Android device configures an IPv6 address matching its EUI-64 identifier and a randomized one. The WP8 and the iOS devices configure the two IPv6 addresses from random strings. According to the RFC, "devices implementing this specification MUST provide a way for the end user to explicitly enable or disable the use of temporary addresses"; however, none of the systems are compliant with this statement. Table 9.2 shows a considerably higher delay for network layer configuration for the Android (Nexus 4) terminal. It starts the IP address configuration process approximately 1 second after the association, starting with the Router Solicitation message, and approximately 4 seconds after the association, it starts with the SLAAC, which leads to the highest delay among the terminals studied. Unfortunately, this process has been impaired significantly in the updated version.

Finally, the use of multicast for the interface configuration is slightly different in the three families of systems. For instance, Windows Phone makes use of LLMNR (Link Local Multicast Name Resolution) protocol [125] in addition to IGMP (Internet Group Management Protocol) [126] and MLDv2 (Multicast Listener Discovery) [127, 128] which are used in the iPhone signaling. The Android phones just make use of MLDv2 messages, as IGMP is not supported. This behavior is specific of some Android devices and constitutes a known issue in the community.[10]

Higher layer configuration: As part of the process to gain Internet connectivity, all

Table 9.3: DNS queried for initial configuration of services on WLAN interface start-up

Service	Android	iOS	WP8
Initial connection to central servers	clients3.google.com (IPv4 and IPv6)	www.apple.com; (IPv4 and IPv6)	login.live.com clientconfig.passport.net
Network status indication	clients3.google.com/generate_204 www.google.com/blank.html	http://www.apple.com/library/test/success.html (iOS6) randomized (iOS7)	www.msftncsi.com
push notification service	mtalk.google.com	18-courier.push.apple.com (IPv4 and IPv6)	push.live.net
Software updates	play.googleapis.com android.clients.google.com (IPv4 and IPv6)	www.apple.com; (IPv4 and IPv6)	ctldl.windowsupdate.com crl.microsoft.com (IPv4 and IPv6)

mobile OSes perform a technique called Network Connectivity Indicator, to detect captive

[10]Issue 51195: many devices have multicast disabled in the kernel. `http://code.google.com/p/android/issues/detail?id=51195`

portals. This protocol issues a DNS query to establish a TCP connection intended to send an HTTP GET method and retrieve a light-weighted web page, which is mainly void. This procedure serves to check the Internet connection at the device and prompt the users with a login form to introduce their credentials, if required by the WLAN administrator. The API in Android includes a way to access the information on whether the terminal is connected or connecting to the Internet, through which interface and allows registering to event notifications in case of network status changes. The connection manager can detect the status of the interface and if it is connected, and in addition, application developers have the option to try and reach a website from their application when it starts running to check that there is actual connectivity. With regard to Apple's devices, it is worth highlighting a recent change performed in iOS7 compared to iOS6. The captive portal detection in iOS6 was performed issuing a query to the web page `http://www.apple.com/library/test/success.html`, while in iOS7 this web page check has been swapped by a randomized query to a URL selected in a list identified by Apple. Table 9.3 collects the service connections that are preceded by a DNS query when the different terminals attach to a WLAN. As soon as the mobile terminals gain Internet connectivity, all the terminals try to reach the central servers of their correspondent manufacturers to update their location, configure some services - time synchronization or push notification service – and re-initiate some connections – as GTalk, in the case of Android. IP addresses on the server side are expected to change, so the terminal connects by hostname, issuing DNS queries, rather than by IP address. Commonly, servers implement a load balancing scheme, so it is possible that the same query returns a different IP address for the same host name. For that reason, to identify the connections we track an IP address block, instead of a specific name or address. In addition we observe that many providers and applications use CDN (Content Distribution Network) nodes to offer their services, which complicates the identification of the service as the connections are hidden by the CDN. For instance, iPhone uses the Akamai network [129] for all Apple related services.

To sum up, the results show that the Android terminal associates to the AP much faster in the previous versions of the system, and the network layer configuration time is comparable to that of WP8 for IPv4, but it takes significantly longer than in the other terminals for IPv6 configuration and Duplicate Address Detection (DAD). However, it calls our attention that, as Linux does, the first DNS query sent by the Android phone is always a request for an IPv6 address (type AAAA), so IPv6 takes precedence over IPv4. On the contrary, the WP8 device issues the IPv4 query before the IPv6 one. Note that the differences in hardware, as reported in Table 9.1, should not be responsible for the differences in performance, specially in the case of the Android phone, which is the one with the slowest initial connection. In the case of the iOS terminals, they require much less time than the Windows and Android devices to configure an IPv4 address, but the

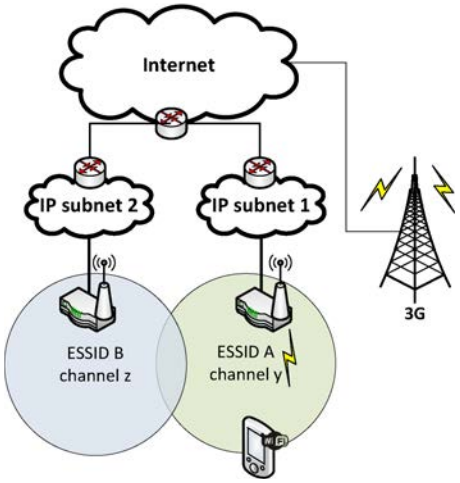


Figure 9.2: General scenario for handover tests.

association to the wireless AP takes longer (except for Nexus 5). The fast configuration in IPv4 makes possible to have comparable total times for iOS and Windows devices, while for IPv6 the Windows terminal is almost 1 second faster than iOS and almost 3 seconds faster than Android.

9.5 Handover dissection

This section presents a thorough study of different handover scenarios and their effect on applications and user experience. For these experiments, we vary the configuration of two 802.11 APs in order to cover as many different scenarios as possible. The different variations in the setup are presented in Figure 10.4. The access points provide Internet access to the terminals attached to them and we play with different parameters of the network – ESSID, channel and IP subnet – to evaluate their influence when the handover process takes place from one AP to the other.

We do not only analyze the handover from an 802.11 AP to another, but also the inter-technology, or vertical handover, moving the connection from the cellular interface to the WLAN one and vice versa. Lastly, we evaluate the behavior of several applications when a handover takes place to check handover impact from the point of view of the user.

9.5.1 WLAN Horizontal Handover

9.5.1.1 Initial considerations

In this section we focus on the main core of the handover analysis, which is the intra-technology or horizontal handover, where the mobile node changes the point of attachment to the WLAN and connects to a different WLAN AP. Table 9.4 gathers the link layer delay measured for the different experiments. We have indicated the differences in ESSID

Table 9.4: Layer 2 handover delay [s] for the different terminals

			Handover mechanism			
			Manual	Forget	disconnect AP	
					AP deauthenticates	Connection lost
Same ESSID	Same channel	A 4.2	N/A	N/A	0.92 ±0.13	5.54 ±0.09
		A 5.0 Nexus 4	N/A	N/A	7.20 ±2.54	21.09 ±6.39
		A 5.0 Nexus 5	N/A	N/A	9.12 ±1.72	2.96 ±1.06 (reassoc) 13.14 ±2.66
		iOS6	N/A	N/A	1.94 ±0.36	4.63 ±1.12 (reassoc)
		iOS7	N/A	N/A	2.48 ±0.29	4.59 ±0.59(reassoc)
		iOS8	N/A	N/A	2.44 ±0.81	11.23 ±3.33 (reassoc)
		WP8	N/A	N/A	0.24 ±0.1	
	Different channel	A 4.2	N/A	N/A	0.99 ±0.19	5.40 ±0.11
		Lollipop Nexus 4	N/A	N/A	3.25 ±0.26 (reassoc)	2.71 ±0.37 (reassoc)
		Lollipop Nexus 5	N/A	N/A	9.05 ±4.68	13.40 ±4.60
		iOS6	N/A	N/A	1.57 ±0.34	5.66 ±1.38 (reassoc)
		iOS7	N/A	N/A	2.38 ±0.32	4.88 ±0.97 (reassoc)
		iOS8	N/A	N/A	14.80 ±5.94	18.87 ±1.87
		WP8	N/A	N/A	0.4 ±0.097	
Different ESSID	Same channel	A 4.2	0.95 ±0.08	0.9 ±0.03	0.78 ±0.29	5.42 ±0.1
		A 5.0 Nexus 4	0.10 ±0.08	1.04 ±0.45	5.31 ±3.02	10.73 ±3.18
		A 5.0 Nexus 5	0.09 ±0.04	4.70 ±1.78	8.53 ±3.81	9.75 ±4.83
		iOS6	0.14 ±0.054	1.53 ±0.10	1.36 ±0.53	11.87 ±0.83
		iOS7	0.22 ±0.03	1.64 ±0.33	2.41 ±0.67	11.88 ±0.36
		iOS8	0.12 ±0.06	0.27 ±0.06	3.90 ±1.90	11.44 ±2.23
		WP8	0.87 ±0.34	1.52 ±0.62	10.14 ±1.83	
	Different channel	A 4.2	0.93 ±0.07	0.91 ±0.025	0.91 ±0.06	5.41 ±0.15
		Lollipop Nexus 4	0.27 ±0.28	3.29 ±0.31	9.07 ±1.65	15.09 ±3.39
		Lollipop Nexus 5	0.10 ±0.08	4.24 ±1.30	6.87 ±3.26	13.12 ±4.48
		iOS6	0.22 ±0.06	1.43 ±0.45	1.09 ±0.82	12.00 ±0.51
		iOS7	0.28 ±0.04	1.69 ±0.61	1.20 ±0.26	12.16 ±0.887
		iOS8	4.92 ±0.03	0.17 ±0.08	8.61 ±4.12	12.12 ±2.38
		WP8	0.73 ±0.05	0.75 ±0.05	10.41 ±0.065	

and channel of operation between the two APs. Our analysis is centered on the link layer handover, so we do not change the IP subnet. As none of the three mobile OS families bases a handover decision on the received signal strength from the current AP or link quality degradation, we force a handover in our experiments by initiating it in the network or in the terminal side following three different approaches: *i)* the AP deauthenticates the station (since the mobile terminals do not react to signal strength or link quality degradation, we turn off the AP). This is presented in Table 9.4 as "disconnect AP" in light of the obtained results we differentiate two subcases; *ii)* the mobile user terminates the connection by manually omitting – "forgetting" – that network (referred to in Table 9.4 as "forget") or *iii)* the mobile user switches the connection directly to other network ("manual" in Table 9.4). The difference between the second and the third option is that, in the latter, the user explicitly indicates the target network to switch to.

Another preliminary consideration is that the cellular interface is the default interface in smartphones, since it provides always-on connectivity. Therefore, we also evaluate the influence of having this interface enabled or disabled in the case of a horizontal WLAN handover. We have noticed that the analyzed OS families always fall back to the cellular connection as soon as the current WLAN connection fails, even if there are other WiFi networks available. This change involves another variation of the IP address that provides

global connectivity to the device, however, it does not necessarily worsen the handover latency and the interruption experienced by the user. On the contrary, the change to the cellular connection actually does not increase the handover delay and, as we will see in Section 9.5.3, improves the performance in case the handover implies a change of IP subnet, contributing to the survival of a running application, depending on the implementation. For the experiments presented in this section, we have confirmed that having the cellular data connection enabled or disabled makes no significant difference in the horizontal handover delay, so we do not distinguish these two cases in Table 9.4 for the sake of clarity.

9.5.1.2 Handover latency

The first issue that calls our attention is the considerable handover latency for the systems under study in every scenario. However, it is remarkable that this latency does not always translate into a complete loss of connectivity or killing a running application – see Section 9.5.3.

From the figures in Table 9.4 we can clearly see that the fastest way of handing over from one WiFi network to the other is to manually change the connection. All the terminals can change the connection in less than a second, but iOS devices are particularly fast, with times around 200 ms. In the case the handover is initiated by the user but just deciding to disconnect from the current AP, without directly connecting to the new one ("forgetting" the current connection) the handover latency increases to values around 1.5 seconds as the mobile terminal scans again in every channel. However, the Android terminal presents a stable delay as the handover delay remains around 0.9 s when the mobile terminal hands off between two APs from different ESSs (Extended Service Sets), regardless whether the handover is initiated by the terminal or the network. For the rest of the systems, instead, it varies significantly. It is remarkable the sticky client implementation in iOS8 wireless client, which tries to remain connected to the same AP regardless of network conditions or even user choices, especially when having to change to a different channel, which explains the 4.92 s delay in the "manual" handover and the 0.17 s for the "forget" handover, as the terminal cannot try to remain connected to the previous AP.

In the case of roaming between two APs within the same ESS, the only possibility is to hand off by disconnecting the AP, because the user cannot choose manually to which AP in the ESS it connects. When the handover is triggered from the network side, the AP sends a deauthentication frame when it disconnects. In this case, we have identified two differentiated behaviors that repeat in our experiments, except for the WP8 terminal. In the first case, the mobile terminal receives the deauthentication from the AP and starts active scanning for a new AP to connect to. This case shows lower delays and the signalling is similar to the one described in Section 9.4 for the initial attachment. On the

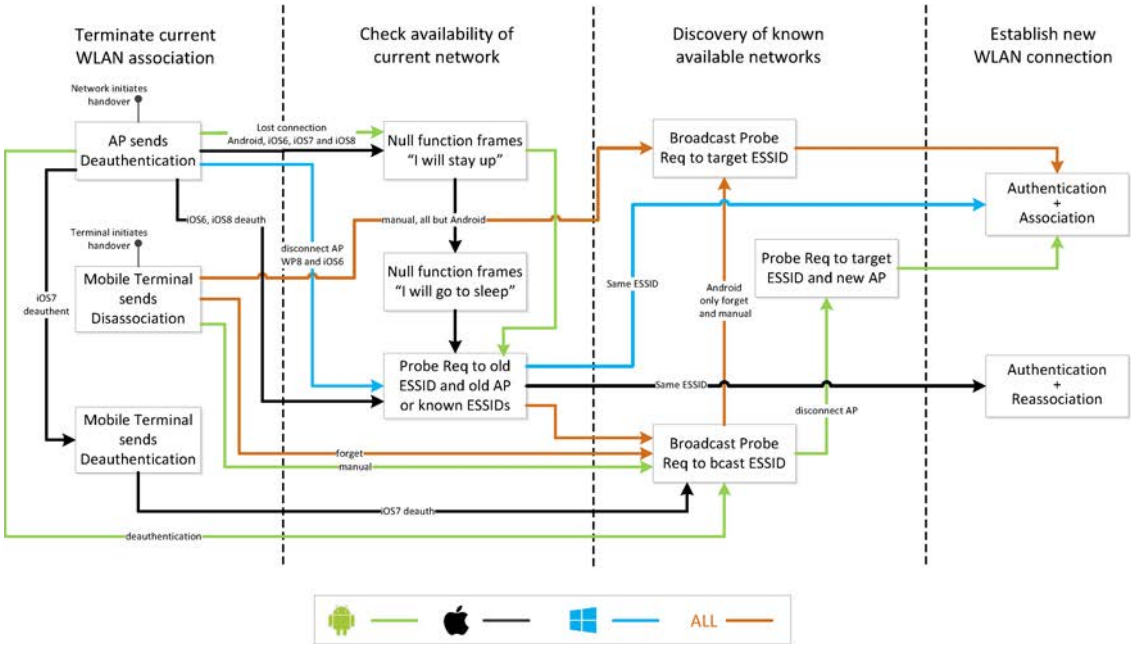


Figure 9.3: Flow diagram of a handover procedure for the different OS families.

contrary, in the other case we have measured much higher delays until the mobile terminal associates to the new AP. The reason for this difference is that the mobile terminal does not get disconnected because it recognizes the deauthentication from the AP, but because the connection is lost (e.g. missed Beacons). The mobile terminal tries to reconnect to the lost AP by sending Null Function frames and Probe Requests. As the AP does not respond anymore, the mobile terminal needs to scan to look for other available APs. Once again, the Android terminal presents a similar delay irrespective of whether the former and the new AP are part of the same ESS (around 5 seconds). However, the iOS devices perform a re-association in approximately 5 seconds when roaming within the same ESS, whereas the delay raises to 12 seconds when the two APs are in different ESSs. We have traced the events that take place in the case with the higher delays in the wireless networking stack until the wireless driver. However, we cannot confirm if this is a buggy behavior of the implementation, but it clearly gives us some room for improvement in the connection management. The Windows Phone terminal does not present these differences in its behavior. Its handover process is more stable although moving within the same ESS clearly decreases the delay and reduces the scanning phase. It is worth to mention that, when roaming within the same ESS the iOS devices send a Reassociation Request frame instead of an Association Request to the new AP. The main difference between these two frames is that the Reassociation includes the BSSID (Basic Service Set Identifier) of the previous AP that the mobile terminal was connected to. This is because from iOS6 Apple has implemented support for 802.11r (amendment for Fast BSS transition). Since our APs do not implement 802.11r mechanisms, this feature reduces to a regular handover,

although having it enabled, should influence significantly aspects like security or QoS.

If the handover is performed between two APs within the same ESS, this change should have an effect only at the link layer, being transparent to the IP layer. Therefore, the mobile terminal should not renew its DHCP lease until it is expired even though it changes from one AP to another inside the same ESS. This behavior is confirmed by the iPhone terminal, but both Android and Windows phones initiate a DHCP discover process when they connect to the new AP, regardless of being part of the same ESS.

In these experiments we do not consider changes in the IP layer, as we are focusing on the link layer delay. Nevertheless, it is worth pointing out that for the Android and Windows phones any kind of handover involves a reconfiguration in the IP layer (new DHCP and IPv6 configuration), even though the target network is managing the same IP subnet.

9.5.1.3 Link layer behavior

Figure 9.3 illustrates the signaling differences among the systems under study for the four handover cases we have analyzed. We indicate the steps common to all the systems with a brown line, and we differentiate the steps for each system with green for Android, blue for WP8 and black for iOS. In the following, we highlight the main differences among the three. The handover process starts either when the AP sends the deauthentication to the mobile terminal ("disconnect AP" case in Table 9.4) or the mobile terminal disassociates ("manual" and "forget" cases in Table 9.4). One of the main differences we can appreciate is that the Android terminal does not differentiate between the "manual" and "forget" handovers. Both trigger the active scanning with broadcast Probe Request frames to the wildcard ESSID; then, it follows a broadcast Probe Request to the target ESSID (when the AP sends the deauthentication to the Android terminal, this Probe Request frame is not broadcast) and finally authenticate and associate. For WP8, we notice it follows a shorter path than the other systems when the AP sends the deauthentication (and the ESSID does not change), which matches the low delay of the WP8 handover in this case. We notice that only the iOS7 device sends a Deauthentication frame when it receives the deauthentication from the AP. Finally, the two different cases for the "disconnect AP" handover are illustrated by the two different paths that part from the "AP sends Deauthentication" box. The case that incurs the highest delay is the path through the Null Function frames, while the direct path to the active scanning box also involves considerably lower delays (as abovementioned in Table 9.4). In this last case, the terminals may follow two different paths, sending Null Function frames or not, depending on whether they accept the deauthentication from the AP or think the connection has been lost.

The use of Null Function frames is a common practice and, as their usage is not defined by the standard, different implementations make different use of them. On the

one hand, they are used for power saving, notifying the AP when the station is going to sleep mode and waking up. On the other hand, they are used during active scanning to get the AP buffering the frames addressed to the station while it is sensing other channels to avoid retransmissions. A third alternative is to use the Null Function frames as a keep alive for the connection. However, the flexibility provided by these frames is also applied for attacking wireless networks [130].

9.5.1.4 Upper layers behavior

It is worth mentioning that the handover delay increases significantly in the case of not having an application running. Moreover, although the behavior on the link layer is quite similar in the different devices for any of the scenarios, the behavior in the upper layers is very diverse.

We aim to characterize also the interruption at upper layers due to the handover. However, we have not been able to extract similar results to the link layer measurements, due to the variability in the behavior of the application running and the high delay to re-establish the flow with the main servers. The application behavior is not under our control, but several cases can be identified. In the case of Android, every handover case involves a complete reconfiguration also at the network layer, reissuing DHCP discovery and performing DAD. Therefore, the interruption of any traffic flow is considerably long. When the device gains connectivity again, the TCP connection is reset (RST flag set) and any connection of an application running at the moment of the handover has to be renewed, sometimes even connecting to different servers – e.g. due to load balancing at the server. On the other hand, in the case both APs handle different IP subnets, this re-issue of DHCP discoveries enables a faster re-configuration of the IP layer. Otherwise, even though the connection at link layer takes place successfully, no data will be delivered to or from the mobile terminal as it does not reconfigure its IP address according to the new network. However, passing through the cellular interface during the transition between two WLAN APs enables the IP re-configuration, also within the same ESS. This is the case for iOS devices, which do not re-issue a DHCP discovery when there is a handover within the same ESS if the cellular interface is not enabled.

9.5.2 3G-WLAN Vertical Handover

One of the main differences among the operating systems under study is in the simultaneous usage of the cellular and 802.11 interfaces. iOS does not allow one interface to remain active when the other is getting started too, e.g. finishes the open connections and turns down the cellular interface when a known WLAN appears in range and tries to connect to it. On the other side, Android keeps the cellular connection on until the IP address is configured in the WLAN interface. Finally, Windows Phone allows the simulta-

neous connectivity through both interfaces, even keeps an active communication through the cellular interface although it gets attached to a WLAN AP, while applications started from that moment use the WLAN. This is an important advancement over its competitors, as for instance, it allows to keep a VoIP (Voice over IP) conversation over the cellular network while connecting to an 802.11 AP, ensuring that the call will not be interrupted. Android 5.0 includes a significant change in the network management. First of all, the simultaneous usage of cellular and WiFi interfaces is allowed. Therefore, the connections established through the cellular connection remain active through that interface even in the case the mobile terminal connects to a WLAN. If there are not active connections, the WLAN interface is still the preferred one, deactivating the IP connectivity in the cellular interface 30 seconds after the WiFi interface gets a connection. In order to introduce these changes, the development team has defined a routing table per interface, instead of a system-wide routing table, as in the previous Android versions. This has allowed to improve, as we show in 9.5.3 the performance of applications running when a handover occurs, for instance, in the case of an ongoing Skype call.

9.5.3 Application survival to handover

Following the different approaches for handover described in Section 9.5, we have studied the influence of these handover procedures in the behavior of several applications. The test procedure consists on starting the application under stable network conditions and perform a handover to analyze the effect of this change. The results of these experiments are presented in Tables 9.5 and 9.6 for intra-technology WLAN handover and inter-technology handover respectively. We have observed that the mobile devices under study fall back to the cellular interface as soon as the WLAN connection is lost, although other 802.11 networks are available and even attach to one of them immediately. Because of that, we have included another variant in our experiments apart from the different configurations of the WLAN, which is to have the cellular data connection in the mobile terminals enabled or disabled. However, regardless whether the cellular interface is on or off, the delay (presented in Section 9.5.1) is not affected but only has an influence on the survival of the application, which is more likely to overcome the handover when the cellular interface is on. We have identified different application behaviors:

a) The application interrupts and does not recover even when there is global connectivity unless you restart the application. This is indicated in the table by a black cell.

b) The application interrupts for a small lapse, continues working intermittently and finally stops. This is indicated by a dark grey cell.

c) The application interrupts for a small interval, but it continues working properly

and go back to its normal operation with a noticeable but acceptable glitch for the user. This is indicated by a light grey cell.

d) The application tolerates the handover, which happens smoothly and the interruption is seamless for the user. This is indicated by a white cell.

We have chosen some of the most widely used applications serving for different purposes, also to check if there are differences in the way they access to the network. The applications we have evaluated are Skype, as a VoIP application (Android v3.2.0.6673, v5.1.0.57240 (Nexus4 Lollipop) and v5.1.0.58677 (Nexus 5 Lollipop); iOS6 v4.2.2601; iOS7 v4.17.0.123; iOS8 v5.11; WP8 v2.1.0.241); and three applications for radio streaming, which will be referred to in the following as Radio 1[11] (Android v1.08.16 (ICS), v4.1.2 (Nexus4 Lollipop) and v1.08.33 (Nexus 5 Lollipop); iOS6 v2.1.7216; iOS7 v2.1.9327; iOS8 v3.0.0; WP8 v2.1.0.0), Radio 2[12] (Android v1.0 and v2.1.3 (Nexus 4 and Nexus 5 Lollipop); iOS6 v2.4; iOS7 v3.0; iOS8 v3.5; WP8 v1.3) and Radio 3[13] (Android v2.2.2 and v2.2.6 (Nexus 4 and Nexus 5 Lollipop); iOS6 v2.1.1; iOS7 v2.2.1; iOS8 v2.2.2; WP8 v1.2). We choose three different radio stations to avoid biasing the conclusions, because some features may be implementation-dependent. We have also considered in our analysis Youtube, as a video streaming application (Android 4.2 v4.2.16 and v6.0.13 (Nexus 4 and Nexus 5 Lollipop); iOS6 v1.1.0; iOS7 v2.3.1.11214; iOS8 v10.09.11358); and Facebook, as the most relevant social network at the moment[14] (Android v3.3, v23.0.0.22.14 (Nexus 4 Lollipop) and v24.0.0.30.15 (Nexus 5 Lollipop); iOS6 v5.6; iOS7 v6.8; iOS8 v26.0; WP8 v4.1.0.0). However, Youtube and Facebook are not included in the comparison of the results. Youtube has been excluded from the table because there is no difference in the behavior of this application in any of the handover combinations. As long as the buffer does not empty, the user will be able to watch the video without noticing that there is a handover in progress. However, the application may stop if the handover does not allow to keep filling the buffer to continue playing the video. Facebook does not appear in the table neither, as this application can always recover from the loss of connectivity, retrying to load the user profile (or the target page) in several attempts, either automatically triggered by the application or reloaded by the user.

As recommended by the developer documentation of the three OS families, HTTP or HTTPS are the way to send or receive small pieces of information, and this is the way that all the Radio applications use for streaming their content. The only application in our experiment set that uses a different protocol is Skype.

In order to mitigate the interruption in network connectivity, the different mobile terminals analyzed fall back to the cellular connection as soon as the current WLAN

[11]The application tested is the one by *Los 40* radio station.
[12]The application tested is the one by *Cadena 100* radio station.
[13]The application tested is the one by *RNE* radio station.
[14]http://www.dreamgrow.com/top-10-social-networking-sites-by-market-share-of-visits-may-2013/

Table 9.5: Survival to handover for applications of different nature in the three OS families. Intra-technology handover

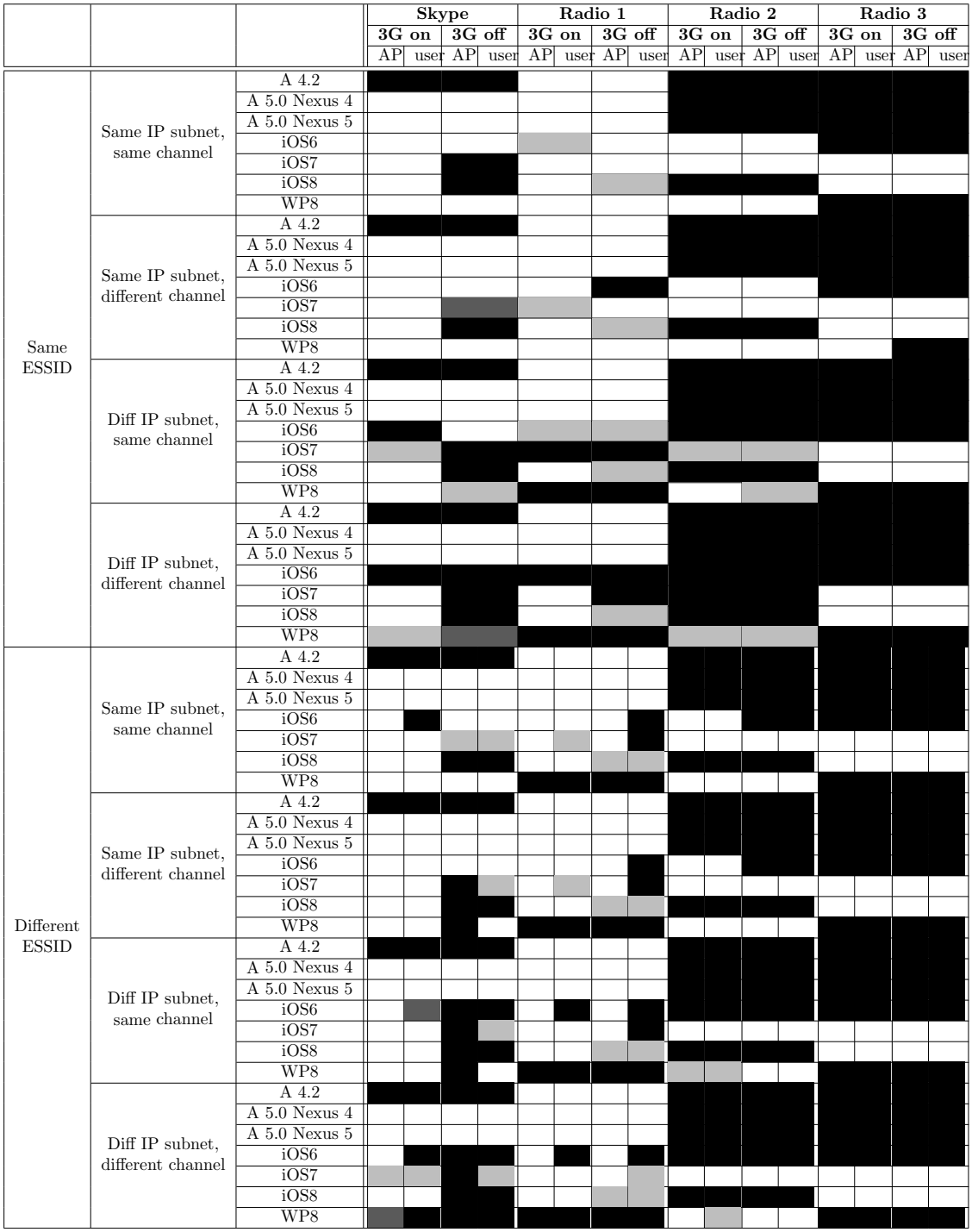


Table 9.6: Survival to handover for applications of different nature in the three OS families. Inter-technology handover

		Skype	Radio 1	Radio 2	Radio 3
3G → WiFi	A 4.2	■		■	■
	A 5.0 Nexus 4			■	■
	A 5.0 Nexus 5			■	■
	iOS6				
	iOS7				
	iOS8				
	WP8				■
WiFi → 3G	A 4.2	■		■	■
	A 5.0 Nexus 4	▒		■	■
	A 5.0 Nexus 5	▒		■	■
	iOS6	■			■
	iOS7	▒	▒		
	iOS8				
	WP8	▓	■		■

Color guide for application behavior during handover in Table 9.5 and Table 9.6:
■ Application interrupts ▓ Annoying, finally interrupts ▒ Acceptable glitches □ Seamless handover

connection fails, even if there are other known WiFi networks available. Regaining IP connectivity by the 3G, gives more time to the mobile terminal to scan in the WiFi channels and connect to another AP if there are any other networks available. However, the change in the technology involves a change in the IP address in use for currently ongoing connections, and this may impact open connections more than the link layer handover, depending on the implementation. There are applications that can survive these two changes in the current connection, while others are interrupted as soon as the current network interface loses connectivity. Note that the applications that easily survive are those that can benefit from buffering the content (Youtube) but note that buffering does not necessarily imply a seamless handover, as not all the applications that buffer the content can survive a handover (Radio 3) concluding that the survival of an application is implementation dependent. It does not depend neither on the OS nor the API offered to application developers, as in the same system different applications can overcome the handover interruption whereas others do not regain connectivity anymore – Radio 1 and Radio 3 in Windows Phone, respectively. In addition, in the case of the applications that hand-off successfully, even though the change of access technology back to 3G involves an additional change in the IP configuration, it helps improving the user experience making the interruption smoother.

In the light of the results in Tables 9.5 and 9.6 the first issue that calls our attention is the poor performance of the application Radio 3, which cannot survive the change of

the point of attachment to the network. Secondly, we identify that Radio 2 performs significantly better in the Windows Phone 8 than in Android or iOS. A user of this application in the Windows terminal can continue listening to the radio station with a seamless handover or with a minimal interruption in every case, while an Android or iPhone user would stop being provided the service. However, while the Android Radio 2 application fails in every handover, under any circumstances, the iPhone Radio 2 application shows different behavior under different scenarios and for iOS6, iOS7 and iOS8: *i)* for iOS6, the application cannot handle a handover that involves a change in the IP subnet; a change in the point of attachment within the same ESS is supported but, changing to a different ESS this application only survives if the 3G interface is enabled *ii)* for iOS7, Radio 2 interrupts only when IP subnet and channel change within the same ESS; *iii)* for iOS6, iOS7 and iOS8, the application tolerates inter-technology handovers, although the application for iOS8 is the one that interrupts the most. It is also remarkable the improvement from iOS6 to iOS7 versions of the same application, especially for Radio 2 and Radio 3, as well as in the case of iOS8, where Radio 1 does not interrupt when 3G connection is on and Radio 3 can always maintain the connection. However, Radio 2 does not show an improvement, as it interrupts playing.

The issue of changing the IP address is not trivial. A similar behavior is observed for Radio 1: the Windows terminal can handle the change of AP unless it involves a change in the IP address of the target subnet. Similarly, the iPhone terminal starts experiencing trouble even when the target wireless network has the same ESSID, but a different IP. When the target AP operates in the same channel, as the scanning process takes less time, the application can recover, but that is the only case. When the ESSID is different between both networks, the application running on the iPhone terminal can only handle the handover when the action that triggers the change comes from the network, but not when the terminal decides to terminate the connection. This has been overcome in iOS7, only if the connection can fall back to the cellular interface, as waiting for having configured the new WLAN network connection adds too much delay. However, the Android terminal can manage the change in connectivity for Radio 1 without interruption in every case.

By observing the results in Table 9.5 we can see that the applications running in the Android 4.2 device present a more homogeneous behavior and mostly do not tolerate the handover, but for the application Radio 1. However, the Android update 5.0, has allowed to maintain an ongoing Skype call in several scenarios, while applications Radio 2 and Radio 3 still show the same poor performance and interrupt when handover occurs. It is important to note that although the application Radio 1 seems to keep the transmission without interruption, the user actually hears the last packets before the handover twice. That is to say, the sound keeps being heard, but if, for instance, listening a song, a part of it will be heard twice before recovering the connection, with the subsequent impairment

of user experience.

Last, in the case of Skype, the application running on the Windows phone outperforms the other two versions. This result was expected, being a proprietary solution by the same manufacturer. Among the rest, the Android implementation offers the poorest performance, not being able to survive the handover in any case. It is worth pointing out that in the case of inter-technology handover from 3G to WiFi, Windows Phone 8 and iPhone terminals are able to keep the call active for two different reasons: the iOS device turns down the 3G interface when the WiFi network becomes available, then, the call just survives the change of connection and takes advantage of the higher bandwidth of the WiFi network to overcome packet losses during the interruption. On the other hand, the Windows Phone 8 device keeps the ongoing call on the 3G connection, even though the WiFi connection becomes ready and active to be used by other applications. We have confirmed that this only happens when the WiFi connection becomes available while there is an ongoing call, but if the call starts when the device is already connected to a WLAN, every call will use that connection. In any case, Skype is the most sensitive application among the ones we tested. We cannot claim that it does not handle a change in network connectivity, but it will interrupt an ongoing call in case the communication between the two endpoints is lost for more than 15 seconds.

9.6 Overview

In this section we summarize the main findings for the different families of OSes and highlight the differences and features that called our attention, categorizing them into five groups:

Simultaneous usage of network interfaces: Out of the three OS families under study, Windows Phone 8 and Android Lollipop allow simultaneous usage of cellular and WLAN interfaces. The Windows Phone 8 device keeps active connections over the cellular and the WLAN interfaces simultaneously. Android, only in its latest version (Lollipop) modified its policy to allow simultaneous connection through both interfaces. The cellular connection remains active for 30 seconds after ongoing sessions finish. iOS devices finish the open connections on the cellular interface when a connection to a WLAN is established.

Network selection: None of the systems under study perform a network selection algorithm to decide on the best network available to connect to. They rely on the network used in the last connection, if it is available. In addition, none of the systems uses information on link quality or performance to change point of attachment if needed. iOS8 WiFi client is particularly sticky, even not responding to user choices and remaining attached to the current AP if the signal is not lost.

Connection establishment: Android and Windows Phone 8 renew their IP address by reissuing a DHCP discovery every time they connect to a different AP, but iOS does

not if the change of AP takes place within the same ESS. In the initial attachment to a WLAN, Android outperforms the other systems in the link layer attachment, but it is considerably slower in the IP configuration. The WP8 terminal has proven to be the fastest one, on average, for initial attachment to an already visited WLAN. It calls our attention the remarkable impairment of performance in the connection establishment to WiFi networks in Android Lollipop running on Nexus 5. The delay in the connection establishment for iOS8 is also slightly higher than in previous versions, but the difference is not as notorious as for Lollipop (on Nexus 5). Although the IP configuration has been made faster, the connection to the AP has been considerably damaged. It is also new with respect to previous versions, the use of Gratuitous ARP in iOS8 and the CTS-to-self frames sent before authentication frames by iOS8 (iPhone 6) and Lollipop (Nexus 5).

IPv6 configuration: The three OS families implement privacy extensions for SLAAC [50], configuring an IPv6 address that does not match their respective EUI-64 identifiers. Actually, this is not the only IPv6 address configured in the terminal interface, so applications should handle this and take into account that this kind of address will change over time, which may affect ongoing sessions. The first DNS query sent by the Android phone always requests an IPv6 address, so IPv6 takes precedence over IPv4. However, the IPv6 configuration takes a significantly longer period to be completed. On the contrary, WP8 issues first the IPv4 query but the delays for IPv4 and IPv6 configurations are comparable. The cellular networks available for our experiments do not offer IPv6 support for the moment. Although standardization bodies have provided guidelines for the migration to IPv6, to our knowledge, at the time of writing only some LTE networks in North America and Europe support IPv6 access.

Handover: Not having any application running increases delay in case of a handover. Android (except for Lollipop) presents a more regular behavior, having a handover latency of 0.9 s for most of the cases evaluated. WP8 and iOS devices present a more variable performance. Particularly, when a handover is initiated by a deauthentication from the AP (between different ESSIDs), the interruption in connectivity through the WLAN can take approximately 5 s, 10 s or 12 s on average for Android, WP8 and iOS systems respectively. However, when the handover happens within the same ESS it is completed in 0.24 s or 0.4 s by the WP8 phone, a result which outperforms the other two systems. When the handover is initiated by the terminal, the fastest handover (90 ms) is performed by the Nexus 5 Android Lollipop device if the user is manually indicating the target network, closely followed by iOS devices. Note that changing to a different channel, even if the user does it manually, increases delay considerably in iOS8, due to its sticky client implementation. However, Android and WP8 offer lower delays than iOS if the terminal needs to scan for available networks to decide on the new AP to connect to ("forget" case in our experimental results). Nevertheless, the new Android version offers a significantly higher delay (around 4 s) and the new iOS version (iOS8) overcomes the issues with

the manual connection and lowers the delay to just 170 ms. Only the iOS devices and Android Lollipop perform a re-association when the handover takes place within the same ESS. This diminishes the handover delay for the Lollipop devices, but not for the iOS terminals. Although all the systems fall back to the cellular connection when they lose WLAN connectivity, this change does not increase the delay in the WiFi-WiFi handover. The change to the cellular network shades the interruption in the WiFi interface, allowing for time to perform the scanning and the association to the new AP. The remarkably bad performance of Android Lollipop and iOS8 deserve a special mention, especially in the case of changing to a different channel manually for iOS8 and as a general consideration for Android Lollipop.

Multicast and network traffic: Unsurprisingly, HTTP is the dominant traffic as it is also the recommended way to access remote content in the development documentation. It calls our attention the intensive use of IEEE 802.11 Null function frames. The use of these frames is not specified by the standard, but WiFi clients, especially in smartphones, send a significant amount of these frames regularly. The most extended use of Null Function frames is power management, so the station informs the AP when it goes to sleep or awakes. However, these frames are sent very regularly, and, specifically, when connection to the current AP is lost. In our handover experiments, we have detected that Android and iOS devices try to reach the AP, whose signal is lost, by sending numerous Null Function frames and Probe Requests, delaying the connection to a new point of attachment. It is also remarkable the number of DNS requests required every time that any connection is open. Regarding multicast support, IGMP is not supported in some Android devices, including Nexus 4 and Nexus 5, which we used in our experiments.

9.7 Open issues and future directions

The thorough assessment of the connection management in the three mobile operating system families under study highlights some flaws in current implementations. Our study reveals that the design of current mobile terminal OS and applications only takes into account the availability of Internet connection, but does not consider the presence of several access networks as a resource. In addition, current implementations do not optimize network access selection or handover and pay minimal attention to connection management, beyond identifying the interface being used and detecting Internet connection. To fill these gaps we identify some potential implementation changes that would be feasible, even easy to implement, and that would enhance user experience, reducing latency in the connection and the handover and improving efficiency in the handling of several interfaces.

9.7.1 Enhanced network selection

If several known WLANs are in range, all the systems analyzed connect to the one they were connected the last time. None of them takes into account the signal strength or link quality towards the different APs as a criterion to choose what network to connect to. If this were considered, handover after a short time could be avoided. Current drivers and firmware as the ones in smartphones have full access and capabilities to monitor certain key indicators of the performance or link quality. Keeping track of changes in these parameters and other decision policies are easily implementable in software tools like *wpa_supplicant*. Even further improvements, as supporting some of the recent IEEE 802.11 standard amendments, are already included in recent versions. The potential changes that we suggest as an example for access network selection are mainly related to the WLAN connection:

- **Connection to the best WLAN**: when the WLAN detects that already visited networks are available, the smartphone connects to the last visited one. Our suggestion is to connect to the network that provides the best link quality at that moment. Another variant of this selection is to keep track of the performance offered by the network in the previous connection – or keep an average measurement of historic connections – at that given location (similar approaches exist in the context of vehicular networks [44]). This choice would be customizable and applicable to different criteria, like security or delay instead of just throughput or signal quality when several of the networks offer similar characteristics.

- **Reduced scanning during handover**: since the mobile terminal keeps sending Probe Request frames after attaching to the WLAN AP, this information could be used to speed up the process of handover, shrinking the interval that the terminal spends scanning again after disassociating from the previous AP and connecting to the new one. Moreover, the mobile terminals keep sending Probe Requests frames to all the visited networks. As the terminal already has all the information about the user location and movement, the scanning could adapt to the user location, only scanning for nearby networks, reducing considerably the number of frames being sent regularly.

- **Early detection of low link quality**: similarly to prioritizing the connection to the AP that offers the highest signal quality, the link quality should be monitored, given the dynamic nature of wireless networks. This monitoring would allow a quick reaction when the link quality gets low, making the current connection likely to fail. Moreover, we could take advantage of the considerable amount of frames that are exchanged with the AP constantly, as we have checked that, apart from the active scanning, the three OS families that we have evaluated send 802.11 Null Function

frames at all times for power management and signaling purposes. Depending on the terminal capabilities, the current open connections could be handed-off to the cellular interface and start trying to connect to a new WLAN in order to avoid interruptions before the current one fails.

- **Get information from network repositories**: standardization bodies have made an effort in the specification of different information repositories and distribution such as ANDSF [131], ANQP [132], ALTO [133] and MIIS [13] [134]. Their generalized deployment and the access to this information can help the mobile device to choose the best access network to connect to, increase performance in network-assisted handovers and contribute to more efficient network management.

9.7.2 Multi-interface management and integration of mobility protocols

The IP layer constitutes a reasonable level to offer inter-technology mobility support, being the most widespread network layer protocol and in use with different access technologies underneath. IP mobility has been a research topic for a long time and the different solutions designed to allow a user to freely roam across different points of attachment are a clear example of the evolution of the research on this topic, continuously adapting to the new requirements imposed by operators. Although there are hundreds of different solutions, none of them has been a clear market success and none is massively deployed. The current panorama on mobility management is somehow mixed, since mobility solutions have only been deployed within the cellular operator boundaries, e.g., a user can freely roam across the different access networks defined by 3GPP, but there is no solution for inter technology handover or IP mobility within non-3GPP technologies in the wide sense,[15] due to the lack of support in the network and in the terminal. The lack of a common IP mobility solution implemented in the majority of smartphones and networks results in the inter-dependence between the mobile user experience and the smartphone mobile services exposed to the applications.

The network connectivity management in multi-interfaced devices can mainly follow two models, widely known as *weak host* and *strong host* models. The weak host model will accept any packet destined to one of its IP addresses, regardless of the interface where the packet is received. On the contrary, the strong host model will only accept the packet if the destination address matches that one of the interface which received it. Different operating systems decide to implement one or the other. For instance, Linux implements the weak host model, whereas Windows Vista and Windows 7 default to the strong host, although weak host model behavior is configurable. Such implementation decisions affect the performance of the devices, especially when different access technologies are available.

[15]Some technologies have their own mobility support at link layer but the connections at the terminal will not be able to survive an IP address change.

We argue that a flow mobility solution [135,136] may enhance the user experience when a handover takes place with an ongoing communication. Current smartphones, which have multiple interfaces and can connect to different access technologies, need a connection manager that enables this feature. For Internet access, the two main technologies currently used are cellular (UMTS, LTE) and IEEE 802.11. Nevertheless, the most common approach is to use only one of the interfaces at a time, missing the benefits that the usage of both interfaces simultaneously could provide. One of these benefits, currently being discussed by standardization bodies is 3G offloading. By offloading some of the flows at the mobile terminal over a congested cellular network to a WiFi network whenever it is possible not only is convenient for the user, who enjoys greater bandwidth with less delay and at a lower cost, but also for the network operator that frees resources to serve other users. The offloading can be selective depending on the application running or respond to user requirements.

From our experiments, the only OSes that make possible the simultaneous usage of 3G and WLAN interfaces are Windows Phone 8 and the latest version of Android (Lollipop). In this way, e.g., a Skype call can continue without interruption even though the mobile terminal attaches to a WLAN that just became available. The attachment to the new network is totally seamless for the ongoing call as it keeps going through the cellular interface, but the applications that can tolerate a hand off, or that start after it, will be bound to the WLAN interface. This, as we have reported in Section 9.5.3 improves considerably the performance in case of an inter-technology handover, which is seamless for the user. However, once the ongoing session has finished, the new connections will use the WiFi interface mandatorily, without a choice, for instance, if the user needs to establish a longer session and they will be on the move or if they need to ensure session continuity. According to the IETF [49] there are different approaches for connectivity management in multi-interface devices: *i)* per-application, *ii)* centralized, system-wide or based on user input *iii)* stack-level solutions to specific problems. If flow mobility were enabled, applications could choose their default interface, mobility requirements if more than one access technology is available or specify minimum resources or QoS needs according to the network interface. Both hardware and software tools in current smartphones allow the implementation of this kind of policies, although it increases complexity both in the development of the applications and the connectivity management. In addition, the mobile OS has to deal with multiple applications running in parallel, which may or may not specify network connectivity requirements in the same way. Therefore, the system needs to provide a combined approach, with a default policy, system-wide, based on information available and at the same time offer the possibility of a more advanced connectivity management per-application, if it is specified. Moreover, the management of simultaneous connectivity opens issues as routing, default address selection [137] and the selection of parameters to be configured on a per-interface basis [49].

9.8 Summary

In this Chapter we have studied a feature that commonly goes unnoticed, not existing much documentation about its operation: the network manager in smartphones. To that purpose, we have analyzed the connection procedure of the three most popular mobile OS families – Android, iOS and Windows Phone 8 – and we have studied the performance when a handover, both inter- and intra-technology takes place. We have also examined how the handover impacts the performance of some applications and affects user experience. Finally, we introduced some optimizations that are directly extracted from the conclusions gathered as a result of our experiments. The potential optimizations that we propose are the base for our future lines of research. The mobile terminal that presented the least network attachment delay and the most advanced features in terms of connectivity management is the Windows Phone 8 terminal. WP8 and Android Lollipop allow both the cellular and WiFi interfaces to be active at the same time. Unfortunately, WP8 is also the one that offers the most restricted access to the device's features and the least flexibility for potential modifications resulting from our research. Indeed, in terms of flexibility and room for modifications, the mobile terminal chosen for performing further improvements is the Android device. The open source licensing and the root access provide a convenient development environment to continue improving the connectivity management in current mobile devices. As we have confirmed in this study, the three analyzed OS families access the network in a very similar way and perform also similarly when the point of attachment changes. Due to these similarities, the conclusions of the potential enhancements to the connection manager performed in Android, could be ported to the other two platforms, by adapting it to their software stack.

Chapter 10

Mobile data traffic characterization

10.1 Introduction

The discussion about mobility is often centered in the network characteristics and the mobility management protocols, but it is essential not to forget that mobile data traffic characteristics and application design influence the requirements for mobility and impact the performance perceived by the end user.

Thanks to a collaboration in the FP7 ICT project CROWD with a network operator, we have been able to analyze the characteristics of the mobile data traffic being exchanged in their core network. All this information provides insights on the needs and the requirements to be fulfilled by mobility protocols.

In addition, the choice for the most suitable mobility solution can depend on the characteristics of the traffic being exchanged which will set the standard for which performance metrics can be considered acceptable (e.g., long flows with small inter-arrival packet time different from short flows with low data rate, voice traffic impose heavier requirements and require shorter handovers than web browsing.)

We analyze the mobility experienced by the traffic in a real operator environment, and we aim at confirming whether the application of mobility protocols based on the distributed mobility management concept were worth the effort considering the structure of the network, deployment characteristics and the real user traffic. To answer this question, we directly analyze the data traffic in one of the operator's core interfaces looking for mobility and traffic characteristics, required to understand the performance of the mobility protocol. Through this direct approach we found two main problems: *i)* The overwhelming number of flows going through the interface and their lack of information matching location of the user and *ii)* the need for a mechanism to correlate the information in the data and the control paths, to match flows to users and cells. To overcome

these challenges, we resorted to the use of big data analytic techniques. Therefore, we focus *i)* on the development of a platform to perform Big Data analysis over packet traces in a real operator environment in Turkey under privacy and regulation concerns, and *ii)* to showcase the utility of this new framework by the analysis of the suitability of applying a new mobility management concept, DMM, to the current network deployment. To gain an understanding of the whole network operation, we also report on the challenges of working with packet captures taken on the data path of the operator's core, which can carry thousands of flows, and the need of correlating these flows with the control data obtained from a different interface in the operator's core.

10.2 Dataset and collection of information

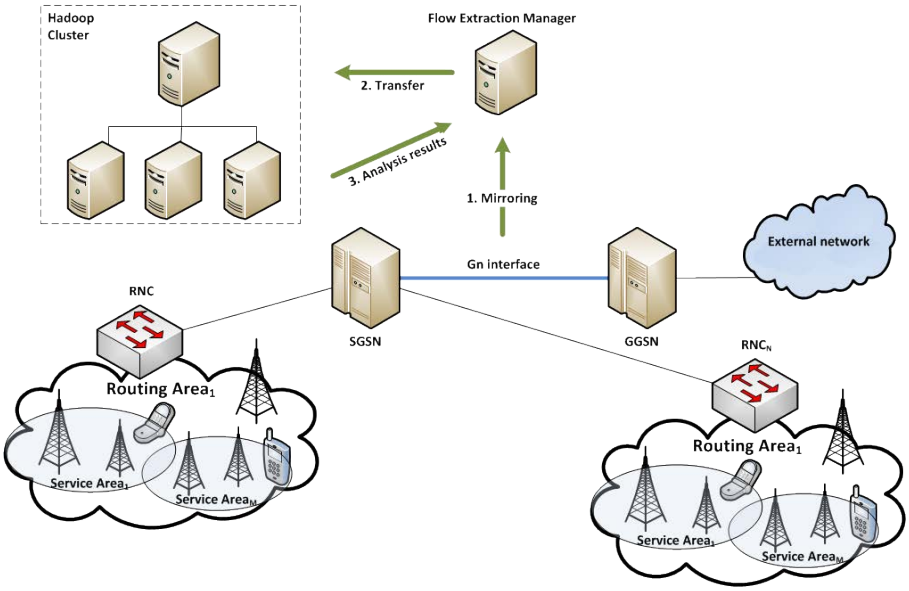


Figure 10.1: Flow Information Extraction Architecture

Identifying the structural patterns in the data traffic is of high importance for mobile operators to apply optimal mobility management techniques within their network. Mobile network traffic has a highly complex and massive structure, making it tough to analyze and reveal structural patterns. It is not unusual to have terabytes of data per second flowing in a typical mobile operator consisting of 10 to 20 million subscribers, which translates into roughly exabytes monthly. The scale of this problem rules out direct sniffing approaches [138], with the additional problem of data packets not carrying information of the user location. The analysis of mobility requirements necessitates the extraction of handover-related information, which can be tracked both in radio access and core network nodes. Accessing handover related information from the radio access nodes is difficult since the amount of probes that needs to be placed into the infrastruc-

ture can be large. Moreover, log data from different entities, such as the Radio Network Controller (RNC), can be too hard to extract for further mobility analysis due to the unavailability of appropriate tools, which are mostly vendor specific. In Universal Mobile Telecommunication System (UMTS), the core network is only notified of Location Area (LA) and Routing Area (RA) updates, when the mobile terminal is in the idle state, since in connected state the access network can still locate it and report to the core network when necessary. Moreover, our approach aims to link user roaming across the network with the characteristics of the mobile traffic being exchanged.

We propose a system based on the extraction of handover related information observed in the core network nodes, correlating the control message headers with the information of the flows in the user data plane, obtained by capturing the data and control packets in the Gn interface. One of the constraints imposed in the system design was the lack of existing measurement tools over the interfaces for mapping flow information with the location information inside the operator domain. Specifically, we track the *Create Context* and *Update Context* messages of the Packet Data Protocol (PDP) [139]. In this way, we can simplify the requirements for monitoring user mobility, avoiding the need for multiple probes. For our analysis, we do not need to track the exact location of the user at the precise moment that it happens, but detect the changes in their point of attachment and characterize the traffic being exchanged by roaming users. We focus our analysis on handovers involving RA changes, which is inline with the current DMM architectures discussed in the IETF.[1]

The final outcome of this process provides a combined listing of the control and user plane packets indicating location of mobile terminals via the information included in PDP *Create Context* and *Update Context* messages. This trace is further processed to characterize mobile data traffic (see Section 10.3) and to obtain insights on the applicability of the DMM concept in an operator's network, as explained in Section 10.4.

10.2.1 System Description

A general view of the architecture for the extraction of flow information is provided in Figure 10.1. The system is composed of two elements, the Flow Extraction Manager (FEM) and the Processing Cluster that has been implemented using Hadoop [76]. The Gn interface in the core network between Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN) is mirrored and collected in the FEM, which applies initial processing and transfers the data to be analysed and filtered into the Hadoop Cluster. The extracted flow information is sent back to FEM from the Hadoop Cluster for collecting analysis results. The details of the extraction process are given in Section 10.2.2. In the following we detail some of the operations of the system:

[1]Note that routing areas may include one to several groups of cells connected to same RNC.

10.2.1.1 GTP-U and GTP-C correlation on Gn Interface

Network packets sent from a User Equipment (UE) to the packet data network (PDN), e.g. Internet, pass through the SGSN which tunnels them towards the GGSN. The GPRS Tunnelling Protocol (GTP) is used for tunnelling the packets in the Gn interface [139].

The operation of GTP protocol differentiates user and control planes. The user plane packets on the Gn interface flow on the GTP User (GTP-U) [139], which is a relatively simple IP based tunneling protocol allowing several tunnels between each set of end points. When used in a UMTS network, each subscriber will have one or more GTP tunnels, corresponding to each active PDP context, as well as possibly having separate tunnels for specific connections with different quality of service (QoS) requirements. Each tunnel is identified by a TEID (Tunnel Endpoint Identifier) in the GTP-U messages, which should be a dynamically allocated random number. The control plane packets on the Gn interface are encapsulated on the control section of the GTP, namely GTP-C [139]. When a subscriber requests a PDP context, the SGSN will send a *Create PDP Context Request* GTP-C message to the GGSN giving details of the subscriber's request. The GGSN will then respond with a *Create PDP Context Response* GTP-C message which will either give details of the activated PDP context or indicate a failure and give a reason for that failure. We make use of a Hadoop based platform (explained below) to aggregate packets in flows and correlated GTP-U and GTP-C information (described in Section 10.2.2). In this way we infer the mobility characteristics of the flows and associated GTP tunnels.

10.2.1.2 Mirroring of Gn interface

The portion of the network considered consists of an area covered by 10 SGSNs. The average total traffic over all regional areas consists of approximately 15 billion packets in uplink direction and 20 billion packets in the downlink direction daily. This corresponds to approximately 80 TB of total data flowing in uplink and downlink daily in the mobile operator's core network. The importance of this work can be seen by the exponential increase in data traffic that has to be handled by a mobile operator. For example, in 2012, the approximate total data traffic was over 7TB in both uplink and downlink daily. The packets are captured by a mirroring device which was already in place on the operator premises as part of their already deployed measurement system. Hence, this work takes benefit of already deployed probes without requiring any additional deployment. We tested our method on real-world Gn interface Internet traffic data. The flow traces obtained from the mobile operator of interest are collected by a server on a high speed link of 200 Mbit/sec at peak hours between 8-9 pm.

10.2.1.3 Hadoop platform

Among the available Big Data platforms, Hadoop [76] stands out as the most notable one as it is an open source solution. It is made up of a storage module, namely HDFS (Hadoop Distributed File System) and a computation module, namely MapReduce. Whereas HDFS can have centralized or distributed implementations, MapReduce inherently has a distributed structure that enables it to execute jobs in parallel on multiple nodes.

10.2.2 Flow Information and Mobility Characteristics extraction

The proposed process extracts and matches the user data TEID (TEID_DATA) field in the GTP-C messages with the TEID in GTP-U packets, to add the corresponding location information to the traffic in the user plane.[2].GTP-C messages include an Information Element containing the location of the mobile terminal, expressed by the Cell Global Identification (CGI), which is formed of the country code (MCC), the network code (MNC) the location area code (LAC, which corresponds to the Routing Area identifier) and the cell identifier (Cell-ID) or service area code (SAC) in UMTS. The Service Area (SA) identifies an area of one or more cells of the same location area and it is identified with a Service Area Code (SAC), unique within that location area. As the MCC and MNC will remain unchanged for every packet in our trace, we monitor the LAC and SAC fields. A Location Area (LA) is a set of cells, which are grouped to decrease signaling overhead. Larger LAs reduce signaling for location updates because users hardly move out of the LA. However, the overhead introduced by paging is very high because there are many cells. Therefore, the size of location areas for circuit switched services can be larger than for packet switched services, which is why the term RA, which is a sub-division of a LA, is introduced in packet switched services. The differentiation between the RAs and LAs depends on the decision of the network operator. Typically, tens or even hundreds of base stations are present in a given LA. As we work with data transmissions, we will use the term RA from now on to refer to the greater area that groups several SAs.

A TEID uniquely identifies a tunnel endpoint on the receiving end of the GTP tunnel [139]. A local TEID value is assigned at the receiving end of a GTP tunnel. The GTP-C packets contain the information that identifies the location of the user (LAC and SAC fields) as well as a TEID_DATA field, pointing to the identifier of the corresponding tunnel in the data plane. We extract from the GTP-U packets, information regarding the characteristics of the mobile traffic, such as packet size or duration of the flows. The flows are identified by the five tuples namely, layer 4 protocol (e.g. TCP, UDP, etc), source and destination IP addresses and port numbers. Matching the unique TEID and TEID_DATA values on data and control planes respectively, the GTP-C and GTP-U in-

[2]Note that this location information is not carried in the data plane packets.

formation is joined by the Hadoop cluster through successive map and reduce operations, to obtain a merged table matching flow characteristics and user location. The complexity of the operation comes from the fact that, in a given PDP session with a specific TEID, there can be multiple flows (and each TEID belongs to a specific user).

The process gives a collection of anonymized rows, representing data packets, providing the time it is collected, its size, a flow identifier, TEID and unique (SAC-LAC) information. This last tuple uniquely identifies the location of the user generating the flow. The mechanism is applicable to all mobile operators using the 3GPP standard Gn interface and can be put into practice immediately, since it does not require changing the monitoring platform of the operator.

10.3 Mobile data traffic analysis

In this section, we analyze the main characteristics of our dataset, built as described in Section 10.2. We have collected two 1-hour traces at times with different traffic load in the network: one close to the "peak" and to the "low" hours.

The Hadoop cluster in our platform is based on Cloudera's Distribution Including Apache Hadoop (CDH4) [140] version on four nodes with Intel Xeon E5-2670 CPUs, 32 cores, 20 TB hard drive and 132 GB RAM.

Table 10.1: Summary results from data traces.

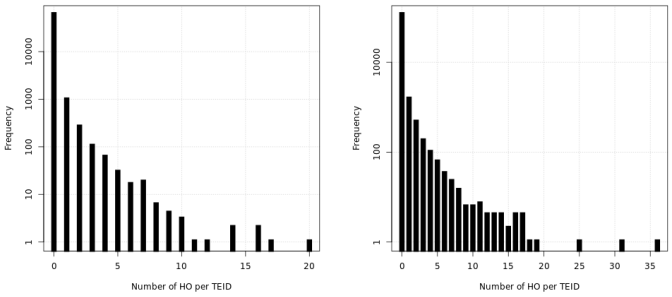
	Low hour	Peak hour
Total duration of the traffic trace	1 hour	1 hour
Total number of control packets	2722234	4294077
Total number of data packets	18978264	28582691
Flows analyzed	1054566	1926062
Total number of data TEIDs in C-trace	671078	1020178
Total number of data TEIDs in U-trace	83510	154650
Total number of data TEIDs analyzed	60932	116529
Number of data TEID experience handover	1464	2413
Total number of flow handovers	7402	18854
Total number of TEID handovers	2624	4750
Data TEIDs without handover	59468 (97.6%)	114116 (97.9%)
Data flows without handover	1047380 (99.3%)	1907759 (99%)

Table 10.1 gathers the main characteristics of our dataset, after filtering and properly grouping the information available in the trace file. We present the total amount of data and control information, expressing it in terms of flows and tunnels. It calls our attention the number of control messages exchanged linked to data tunnels (identified by the TEID_DATA field) that are not exchanging any data packet. That translates into

a considerable overhead in the already challenged network to maintain context for data tunnels not transmitting data.

Regarding mobility, with more than a million flows spread over more than 100 thousand data tunnels, we highlight the high percentage of data traffic that does not experience a handover (99% of flows and 98% of tunnels in the peak hour trace). Still, this traffic is provided with mobility support (default in current networks) leading to an inefficient usage of resources in current network deployments, where mobility is granted to all the traffic being transmitted. Therefore, future network deployments could take into account these characteristics to improve scalability and free resources.

Figure 10.2 shows the number of handovers per tunnel for the low (Figure 10.2a) and peak hour (Figure 10.2b). We have included the number of tunnels not moving in the network (tallest bars in x=0; note the histogram in logarithmic scale). In the peak hour, the number of handovers increase, even though the higher number of RA changes are not frequent.



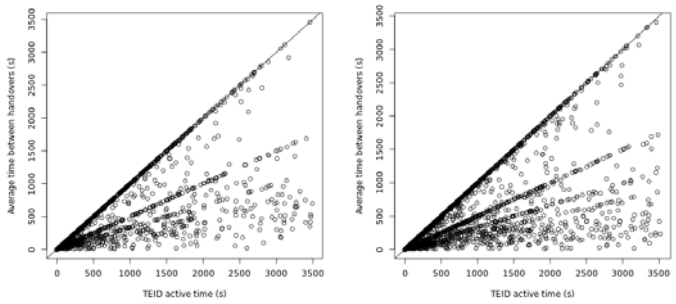
(a) Low hour trace (b) Peak hour trace

Figure 10.2: Histogram for the number of handovers performed per tunnel for the low and peak hours.

We do not evaluate changes of SA, for similarity to an initial deployment of DMM, which would be done by replacing P or S-GW [141] (terms for GGSN and SGSN in LTE nomenclature). In addition, SAs are groups of cells, and they can overlap, which could lead us to misleading conclusions about user mobility. In any case, and just to confirm that the low mobility between routing areas is not due to the size of the area considered, we have measured the SA changes within a given RA. The number of tunnels that changes SA accounts to 5.6% of the tunnels in our dataset, confirming our previous conclusion on the low mobility.

Figure 10.3 represents the duration of tunnel identifiers and the median of the time between routing area handovers for every tunnel. The points are concentrated in the lower X and Y range, meaning that from the flows experiencing a handover, there are a majority which are short lived and highly mobile. It can also be seen how the points form several lines across the plot (e.g., $X = Y$). These lines correspond to the points which

ratio between the average time between handovers and TEID active time is constant. In fact these lines show the number of handovers that are more common between the users. The first of the lines (starting from the top of the graph) corresponds to tunnels surviving one handover. It is also worth explaining the reason behind the lack of data for $X < Y$. We measure the time between handovers through the changes in LAC and SAC of the active tunnels, hence for $X < Y$ the tunnel is already disconnected and data is not available.



(a) Low hour (b) Peak hour

Figure 10.3: Duration of tunnel identifiers vs average of the time between handovers for the low and peak hours.

10.4 Mobility management evaluation

In this section, we evaluate the usage of DMM as an alternative to current deployments and we use the mobile data traffic characteristics to back our comparison with actual data.

Based on the analysis by Giust et al. [142], we compare the packet delivery gains attained by the use of DMM principles, comparing to a PMIPv6 (or GTP) architecture. In a DMM architecture, the number of active prefixes (for relating to our scenario, active tunnels) is directly related to the number of handovers performed by the mobile node. In [142], it is proven that the gains of DMM over PMIPv6 are given by Equation 10.1, where $\overline{N}_{PR}$ is the average number of active prefixes at a handover and $C_{MAAR-MAAR}$ and $C_{LMA-MAG}$ are expressed in terms of delay between MAARs and between LMA-MAG respectively.

$$\frac{C_{DMM}}{C_{PMIP}} = (\overline{N}_{PR} - 1)\frac{C_{MAAR-MAAR}}{C_{LMA-MAG}} \tag{10.1}$$

We propose the scenario in Figure 10.4 to compare PMIPv6 and DMM approaches. We see that PMIPv6 is conceptually similar to GTP, and the packet delivery cost comparison between DMM and PMIPv6 can be similarly extended to GTP. We propose as reasonable scenario to have the LMA co-located with the GGSN and the MAG with the SGSN. For DMM, we propose to place a MAAR in every RA (i.e. RNC). Therefore, to follow

the analysis given by Equation 10.1 the number of active prefixes applying DMM to our scenario is given by the number of tunnels performing a RA change. As the analysis in [142] involves the average number of prefixes active per user and we have the aggregated information of active tunnels in the network, for comparison with PMIPv6, we need to take into account all the active tunnels in the network (which is equal to 1 active prefix per user in [142]). We have tracked the RA changes per tunnel, as reported in Section 10.3, to find out that roughly 2% of the tunnels in both low and peak hour experience a handover. With this information and the number of packets delivered per tunnel, we can say that the packet delivery cost ratio is given by Equation 10.2, where λ_{mov} is the number of data packets sent in the tunnels that experienced handover, λ_{tot} is the total number of data packets, N_{mov} is the number of tunnels that experience handover and N_{tot} is the total number of tunnels. This is equivalent to the ratio between the traffic that experiences a handover (and should be provided mobility) and the total amount of traffic being served in the network. Note that, for λ_{mov} we are considering a worst case scenario, as the amount of data packets that would be tunneled by DMM would be the packets that are sent after the handover, but not in the routing area in which the flow is originated.

$$\frac{C_{DMM}}{C_{PMIP}} = \frac{\lambda_{mov}/N_{mov}}{\lambda_{tot}/N_{tot}} \frac{D_{MAAR-MAAR}}{D_{LMA-MAG}} \tag{10.2}$$

Table 10.2: Value of the parameters in Eq. 10.2

	Low hour	Peak hour
λ_{mov}	41467	49927
λ_{tot}	18978264	28582691
N_{mov}	1464	2413
N_{tot}	60932	116529

Substituting the values in Table 10.2 in Equation 10.2 for the peak hour trace we obtain:

$$\frac{C_{DMM}}{C_{PMIP}} = 0.08 \frac{D_{MAAR-MAAR}}{D_{LMA-MAG}} \tag{10.3}$$

Computing the delay between MAARs that makes equal the packet delivery cost of applying PMIPv6 and DMM, we have for the peak and low hours respectively:

$$D^{peak\_hour}_{MAAR-MAAR} = 11.85 D_{LMA-MAG} \tag{10.4}$$

$$D^{low\_hour}_{MAAR-MAAR} = 10.99 D_{LMA-MAG} \tag{10.5}$$

This result is explained given the low mobility in the network. For instance, taking the peak hour case, the number of tunnels that are provided mobility is so low compared to the total amount of traffic in the network that the distance between MAARs (in terms of delay) can be up to 11.85 times the distance between LMA and MAG for the packet

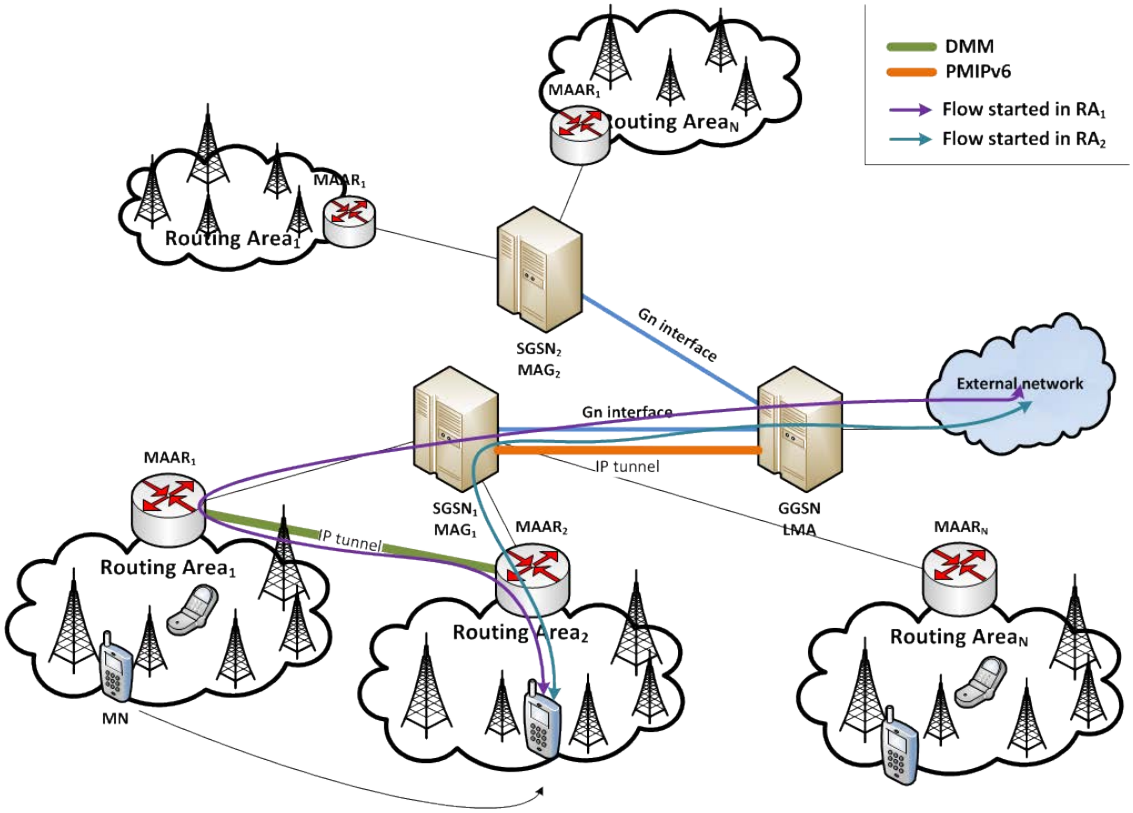


Figure 10.4: Cellular network architecture with PMIPv6 and DMM mobility entities co-located.

delivery cost of DMM and PMIPv6 to be equal. This analysis, based on real operator's network traces, indicates that this network can be flattened by a factor of 11, with no extra overhead in the network, freeing resources in the core and providing on-demand mobility to the user. We claim the use of this technology is key for the development of the 5G network architectures, characterized by extreme consumption of resources in the network.

10.5 Summary

Network operators are challenged by a significant increase of mobile traffic demand. In addition, all the traffic being served in the network is provided with mobility support. However, most of the traffic flowing through the network does not experience a handover. In this Chapter, we have analyzed the data flow characteristics of real traffic from the Gn interface of a UMTS network operator. We have matched the location information in the control plane, which is missing in the user plane packets, with the characteristics of the data flow (e.g. duration, size) and our analysis shows that flows are not long enough to experience several handovers, and in fact, most of the traffic does not experience a change

in the point of attachment to the network. In this scenario, a DMM-based mobility solution would flatten the current deployment to handle mobility more efficiently. The new mobility management solution that we propose will provide increased scalability to the operator network, without additional overhead. The limits on the profitability of this solution will depend on the specific characteristics of the operator's network and the mobile data traffic mobility. As future work we plan to extend the period of time analyzed, searching for time patterns and providing guidelines for deployment of mobility solutions.

Part IV

Conclusion and Further Research Directions

Chapter 11

Conclusion and Future Work

Provisioning networks to ensure good user experience dealing with the outstanding increase in traffic demand currently challenges network operators. Therefore, they need new deployment scenarios that help meet the high traffic requirements. One of the implementation decisions being adopted is the deployment of different technologies, transforming traditional networks into heterogeneous networks. However, there is also an increase in complexity to manage all the available resources without losing performance or incurring higher costs.

Driven by this heterogeneity, in this thesis we analyze and experimentally evaluate the connectivity management in different scenarios. First, we have experimentally evaluated and proposed the redesign of a route optimization protocol for VANETs, which establishes a route through an ad hoc network using 802.11 technology instead of cellular access when the two endpoints of the communication are in the same area. This route optimization mechanism [12] allows lower delays and more efficient routing. Due to the characteristics of the vehicular scenario this protocol is challenged by the dynamism of the links, and the lifetime of the established paths will depend on the speed and trajectory of the vehicles as well. Thanks to our experimental evaluation we re-define some aspects of the original specification to provide more reliability to the optimization process and increase the duration of a valid connection in the VANET. We also designed a mechanism based on Received Signal Strength Indicator (RSSI) to detect link quality degradation in the VANET and anticipate data losses by falling back to the default route through the cellular interface.

Next, we have proposed an architecture that integrates optical broadband networks with mobility management, including a mechanism to enhance handover. We have leveraged the similarities in the architecture of an EPON and PMIPv6 protocol to co-locate the OLT with the LMA and the ONUs with the MAGs. In this way, we avoid the need for tunneling in the mobility management, thanks to the topology of the EPON. In addition, being co-located with the OLT, the LMA has complete information about the network

(e.g. location of every AP, current traffic load) so it is able to optimize network selection on behalf of a mobile node or reduce the delay due to scanning for target networks. To improve the handover process we integrate 802.21 capabilities so the network prepares and executes the handover minimizing data loss. Thanks to the EPON topology, when a handover occurs we can bicast data for the roaming node to both the current and the target ONU-MAGs. Note that this bicasting does not imply any overhead in the network, as in the downlink the operation of the EPON is broadcast-and-select.

Third, we have introduced and deployed an architecture to apply software-defined networking to mobility management. The architecture is proposed in the scope of the FP7 ICT CROWD project, and consists in a two-tier hierarchical set of controllers that handle connectivity management in single technology domains called *districts*. The mobility between two districts is based on the concepts of Distributed Mobility Management. In our experimental setup the access technology is 802.11, although the solution is envisioned to work as well with other technologies and there is also an alternative to cover mobility to a non-CROWD compliant domain, based on host mobility.

In these heterogeneous scenarios several access networks are available to the user, typically equipped with a mobile terminal with multiple network interfaces. To that purpose, we have carefully analyzed and experimentally evaluated the connectivity management in current smartphones for different operating system families and different devices. Indeed, smartphones are becoming more powerful every day and our study shows that the networking connectivity management is often disregarded in favor of application performance. However, connectivity management and handover performance have a strong impact on user experience. It is feasible, both in terms of hardware and software, to have a more intelligent connection manager, even more integrated with application behavior (traffic characteristics) and more flexible in terms of the network interface to be used, multi-homing and routing table management. On the other hand, developed applications are so heterogeneous that it is necessary to implement control mechanisms to allow that flexibility while keeping consistency and managing connections efficiently. Although some operators have favored the network-based approach for mobility management, involving the mobile node in its own connectivity management will be unavoidable, given the network access heterogeneity. The design of a connection manager that is able to decide and prioritize network selection and mobility management based on terminal capabilities, user traffic and mobility patterns should be the trend to follow for boosting the mobile user experience.

Finally, we have also focused our attention on the characteristics of mobile data traffic to study the suitability of IP mobility solutions. Although all the traffic flowing through a cellular network is provided with mobility support, our analysis to the traffic from a real operator's network shows that more than 90% of the traffic does not experience a handover. In this situation, a mobility management solution like DMM would be more

efficient and scalable than traditional approaches. There is an overwhelming amount of traffic being served, which toghether with the control information makes essential to look for more efficient ways to handle it. Our study reveals that, depending on the specific network operator infrastructure and the traffic characteristics, a DDM-like approach would consume less resources to provide mobility support.

The work in this thesis opens new research directions for future work: The more delicate aspects of the route optimization proposed by VARON are the establishment of the connection in the VANET and how to anticipate the degradation of the link. Therefore, we would like to experiment with IEEE 802.11p as the wireless technology in the VANET and explore a different mechanism to know when it is necessary to withdraw the ad hoc route and fall back to the route through the cellular interface.

We also would like to increase the complexity of the performance evaluation of our software-defined networking solution for mobility management. In order to provide a more stable architecture, we must check the scalability and the resilience to failures. In addition, we could implement further policies to increase the flexibility of the solution.

A key takeaway message from this thesis is the necessity for the development of a smarter connection manager in mobile devices. The connection manager should be agnostic for the application, but enhance performance of the application and let developers not to worry about connectivity issues. Given the highly dynamic environment, the connection manager needs to be flexible to adapt to the different scenarios and implement a smarter algorithm for network selection. Additionally, there is a trend towards customization and personalized services. By studying traffic and mobility patterns and including a training period, a smartphone can get easily used to the routines of the user.

Our most recent work includes the analysis of a dataset from real network traces. We would like to extend our analysis and apply a machine learning solution to implement a predictive mechanism for mobility management. We plan to increase the dataset, comparing different times of day, different areas (e.g. rural vs. urban areas, working hours vs. weekend) and extract information on the characteristics of the traffic that experience handover.

References

[1] Open Networking Foundation, "OpenFlow Switch Specification,Version 1.5.1," March 2015.

[2] Open Networking Foundation, "OpenFlow-enabled mobile and wireless networks," September 2013.

[3] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019," White paper, Cisco, Feb. 2015. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf

[4] E. C. Publications office, "Why EU is betting big on 5G," *Research*eu Focus Magazine*, vol. 15, 2015.

[5] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. Sukhavasi, C. Patel, and S. Geirhofer, "Network densification: the dominant theme for wireless evolution into 5G," *Communications Magazine, IEEE*, vol. 52, no. 2, pp. 82–89, February 2014.

[6] "Ericsson mobility report: On the pulse of the networked society," White paper, Ericsson, Jun. 2015. [Online]. Available: http://www.ericsson.com/res/docs/2015/ericsson-mobility-report-june-2015.pdf

[7] C. Perkins, "IP Mobility Support for IPv4, Revised," RFC 5944 (Proposed Standard), Internet Engineering Task Force, Nov. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5944.txt

[8] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6," RFC 5213 (Proposed Standard), Internet Engineering Task Force, Aug. 2008, updated by RFC 6543. [Online]. Available: http://www.ietf.org/rfc/rfc5213.txt

[9] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963 (Proposed Standard), Internet Engineering Task Force, Jan. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3963.txt

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

[11] "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, 2012.

[12] C. J. Bernardos, I. Soto, M. Calderón, F. Boavida, and A. Azcorra, "VARON: Vehicular Ad hoc Route Optimisation for NEMO," *Comput. Commun.*, vol. 30, no. 8, pp. 1765–1784, Jun. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2007.02.011

[13] *IEEE Standard for Local and Metropolitan Area Networks- Part 21: Media Independent Handover*, Std., 21 2009.

[14] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[15] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs," *IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007)*, pp. 1–244, 2008.

[16] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Fast Basic Service Set (BSS) Transition," *IEEE Std 802.11r-2008 (Amendment to IEEE Std 802.11-2007)*, pp. 1–126, 2008.

[17] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: IEEE 802.11 Wireless Network Management," *IEEE Std 802.11v-2011 (Amendment to IEEE Std 802.11-2007)*, pp. 1–433, 2011.

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

[19] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management," RFC 5380 (Proposed Standard), Internet Engineering Task Force, Oct. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5380.txt

[20] R. Koodli, "Mobile IPv6 Fast Handovers," RFC 5568 (Proposed Standard), Internet Engineering Task Force, Jul. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5568.txt

[21] X. Pérez-Costa, M. Torrent-Moreno, and H. Hartenstein, "A performance comparison of mobile ipv6, hierarchical mobile ipv6, fast handovers for mobile ipv6 and their combination," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 4, pp. 5–19, Oct. 2003. [Online]. Available: http://doi.acm.org/10.1145/965732.965736

[22] H. Jung, H. Soliman, S. Koh, and J. Y. Lee, "Fast Handover for Hierarchical MIPv6 (F-HMIPv6)," Internet Draft, Internet Engineering Task Force, Apr. 2005. [Online]. Available: https://tools.ietf.org/html/draft-jung-mobopts-fhmipv6-00

[23] X. P. Costa and H. Hartenstein, "A simulation study on the performance of Mobile IPv6 in a WLAN-based cellular network," *Computer Networks*, vol. 40, no. 1, pp. 191 – 204, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128602002748

[24] N. Montavont and T. Noel, "Handover management for mobile nodes in IPv6 networks," *Communications Magazine, IEEE*, vol. 40, no. 8, pp. 38–43, Aug 2002.

[25] K.-S. Kong, W. Lee, Y.-H. Han, M.-K. Shin, and H. You, "Mobility management for all-IP mobile networks: mobile IPv6 vs. proxy mobile IPv6," *Wireless Communications, IEEE*, vol. 15, no. 2, pp. 36–45, April 2008.

[26] J.-H. Lee, J.-M. Bonnin, I. You, and T.-M. Chung, "Comparative Handover Performance Analysis of IPv6 Mobility Management Protocols," *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 3, pp. 1077–1088, March 2013.

[27] I. Soto, C. J. Bernardos, M. Calderon, A. Banchs, and A. Azcorra, "NEMO-enabled localized mobility support for internet access in automotive scenarios," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 152–159, 2009.

[28] Z. Yan, S. Zhang, H. Zhou, H. Zhang, and I. You, "Network mobility support in pmipv6 network," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, ser. IWCMC '10. New York, NY, USA: ACM, 2010, pp. 890–894. [Online]. Available: http://doi.acm.org/10.1145/1815396.1815600

[29] J.-H. Lee, T. Ernst, and N. Chilamkurti, "Performance analysis of pmipv6-based network mobility for intelligent transportation systems," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 1, pp. 74–85, Jan 2012.

[30] H. Chan, D. Liu, P.Seite, H. Yokota, and J. Korhonen, "Requirements for Distributed Mobility Management," RFC 7333, Internet Engineering Task Force, Aug. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7333.txt

[31] 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*, vol. 52, no. 4, pp. 1989–2002, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11235-011-9479-3

[32] "IETF DMM WG: http://datatracker.ietf.org/wg/dmm/charter/ ."

[33] J. Zuniga, C. Bernardos, A. De La Oliva, T. Melia, R. Costa, and A. Reznik, "Distributed mobility management: A standards landscape," *Communications Magazine, IEEE*, vol. 51, no. 3, pp. 80–87, March 2013.

[34] D.-H. Shin, D. Moses, M. Venkatachalam, and S. Bagchi, "Distributed mobility management for efficient video delivery over all-IP mobile networks: Competing approaches," *Network, IEEE*, vol. 27, no. 2, pp. 28–33, March 2013.

[35] F. Giust, L. Cominardi, and C. Bernardos, "Distributed mobility management for future 5G networks: overview and analysis of existing approaches," *Communications Magazine, IEEE*, vol. 53, no. 1, pp. 142–149, January 2015.

[36] K. Sanzgiri, D. LaFlamme, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, "Authenticated Routing for Ad hoc Networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 3, pp. 598–610, 2005.

[37] P. Lutterotti, G. Pau, D. Jiang, M. Gerla, and L. Delgrossi, "C-VeT, the UCLA Vehicular Testbed: An open platform for vehicular networking and urban sensing," in *International Conference on Wireless Access for Vehicular Environments (WAVE 2008)*, 2008.

[38] R. Mahajan, J. Zahorjan, and B. Zill, "Understanding wifi-based connectivity from moving vehicles," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 321–326. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298351

[39] H. Soroush, N. Banerjee, A. Balasubramanian, M. D. Corner, B. N. Levine, and B. Lynn, "DOME: a diverse outdoor mobile testbed," in *Proceedings of the 1st ACM International Workshop on Hot Topics of Planet-Scale Mobility Measurements*, ser. HotPlanet '09. ACM, 2009, pp. 2:1–2:6. [Online]. Available: http://doi.acm.org/10.1145/1651428.1651431

[40] P. Deshpande, X. Hou, and S. R. Das, "Performance comparison of 3G and metro-scale WiFi for vehicular network access," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, ser. IMC '10. New York, NY, USA: ACM, 2010, pp. 301–307. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879180

[41] A. Giannoulis, M. Fiore, and E. W. Knightly, "Supporting vehicular mobility in urban multi-hop wireless networks," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*, ser. MobiSys '08. New York, NY, USA: ACM, 2008, pp. 54–66. [Online]. Available: http://doi.acm.org/10.1145/1378600.1378608

[42] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: vehicular content delivery using WiFi," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, ser. MobiCom '08. New York, NY, USA: ACM, 2008, pp. 199–210. [Online]. Available: http://doi.acm.org/10.1145/1409944.1409968

[43] F. Martelli, M. Elena Renda, G. Resta, and P. Santi, "A measurement-based study of beaconing performance in IEEE 802.11p vehicular networks," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 1503–1511.

[44] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. N. Levine, and J. Zahorjan, "Interactive WiFi connectivity for moving vehicles," in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, ser. SIGCOMM '08. New York, NY, USA: ACM, 2008, pp. 427–438. [Online]. Available: http://doi.acm.org/10.1145/1402958.1403006

[45] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using WiFi," in *MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 209–222.

[46] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: a distributed mobile sensor computing system," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys '06. New York, NY, USA: ACM, 2006, pp. 125–138. [Online]. Available: http://doi.acm.org/10.1145/1182807.1182821

[47] P. Deshpande, A. Kashyap, C. Sung, and S. R. Das, "Predictive methods for improved vehicular WiFi access," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys '09. ACM, 2009, pp. 263–276.

[48] M. Blanchet and P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement," RFC 6418 (Informational), Internet Engineering Task Force, Nov. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6418.txt

[49] M. Wasserman and P. Seite, "Current Practices for Multiple-Interface Hosts," RFC 6419 (Informational), Internet Engineering Task Force, Nov. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6419.txt

[50] T. Narten, R. Draves, and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6," RFC 4941 (Draft Standard), Internet Engineering Task Force, Sep. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4941.txt

[51] F. Gont, "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)," RFC 7217, Internet Engineering Task Force, Apr. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7217.txt

[52] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, "Diversity in Smartphone Usage," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 179–194. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814453

[53] W. Song, Y. Kim, H. Kim, J. Lim, and J. Kim, "Personalized Optimization for Android Smartphones," *ACM Transanctions on Embedded Computing Systems*, vol. 13, no. 2s, pp. 60:1–60:25, Jan. 2014. [Online]. Available: http://doi.acm.org/10.1145/2544375.2544380

[54] J. Sommers and P. Barford, "Cell vs. WiFi: On the Performance of Metro Area Mobile Connections," in *Proceedings of the 2012 ACM conference on Internet measurement conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 301–314. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398808

[55] S. Liu and A. Striegel, "Casting Doubts on the Viability of WiFi Offloading," in *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, ser. CellNet '12. New York, NY, USA: ACM, 2012, pp. 25–30. [Online]. Available: http://doi.acm.org/10.1145/2342468.2342475

[56] K. Fukuda and K. Nagami, "A Measurement of Mobile Traffic Offloading," in *Proceedings of the 14th international conference on Passive and Active Measurement*, ser. PAM'13. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 73–82, http://dx.doi.org/10.1007/978-3-642-36516-4_8.

[57] X. Chen, R. Jin, K. Suh, B. Wang, and W. Wei, "Network Performance of Smart Mobile Handhelds in a University Campus WiFi Network," in *Proceedings of the 2012 ACM conference on Internet measurement conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 315–328. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398809

[58] J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl, "Anatomizing Application Performance Differences on Smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 165–178. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814452

[59] Q. Zhang, C. Guo, Z. Guo, and W. Zhu, "Efficient mobility management for vertical handoff between WWAN and WLAN," *IEEE Communications Magazine*, vol. 41, no. 11, pp. 102–108, Nov 2003. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2003.1244929

[60] C. Guo, Z. Guo, Q. Zhang, and W. Zhu, "A seamless and proactive end-to-end mobility solution for roaming across heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 5, pp. 834–848, June 2004.

[61] G. Bollano, D. Panno, F. Ricciato, M. Turolla, N. Vaccaro, and D. Ettorre, "Enhanced Android Connection Manager: An Application-Based Solution to Manage Mobile Data Traffic," in *World Telecommunications Congress (WTC), 2012*, 2012, pp. 1–6.

[62] Y. Liu, F. Li, L. Guo, B. Shen, and S. Chen, "A Comparative Study of Android and iOS for Accessing Internet Streaming Services," in *Proceedings of the 14th international conference on Passive and Active Measurement*, ser. PAM'13. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 104–114. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36516-4_11

[63] A. de la Oliva, A. Banchs, I. Soto, T. Melia, and A. Vidal, "An overview of IEEE 802.21: Media-Independent Handover Services," *IEEE Wireless Communications*, vol. 15, no. 4, pp. 96–103, 2008.

[64] B. Pfaff and B. Davie, "The Open vSwitch Database Management Protocol," RFC 7047 (Informational), Internet Engineering Task Force, Dec. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc7047.txt

[65] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a Globally-deployed Software Defined WAN," in *SIGCOMM '13*. New York, NY, USA: ACM, 2013, pp. 3–14.

[66] K.-K. Yap, M. Kobayashi, D. Underhill, S. Seetharaman, P. Kazemian, and N. McKeown, "The Stanford OpenRoads Deployment," in *WINTECH '09*. New York, NY, USA: ACM, 2009, pp. 59–66.

[67] K.-K. Yap, R. Sherwood, M. Kobayashi, T.-Y. Huang, M. Chan, N. Handigol, N. McKeown, and G. Parulkar, "Blueprint for Introducing Innovation into Wireless Mobile Networks," in *ACM SIGCOMM VISA Workshop 2010*. New York, NY, USA: ACM, 2010, pp. 25–32.

[68] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, "OpenRoads: Empowering Research in Mobile Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 125–126, Jan. 2010.

[69] K. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: Toward software-defined mobile networks," *Communications Magazine, IEEE*, vol. 51, no. 7, pp. 44–53, July 2013.

[70] J. Kempf, B. Johansson, S. Pettersson, H. Luning, and T. Nilsson, "Moving the mobile Evolved Packet Core to the cloud," in *WiMob 2012, IEEE*, Oct 2012, pp. 784–791.

[71] P. Dely, A. Kassler, and N. Bayer, "OpenFlow for Wireless Mesh Networks," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, July 2011, pp. 1–6.

[72] A. Detti, C. Pisa, S. Salsano, and N. Blefari-Melazzi, "Wireless Mesh Software Defined Networks (wmSDN)," in *WiMob 2013, IEEE*, Oct 2013, pp. 89–95.

[73] A. Imran and A. Zoha, "Challenges in 5G: how to empower SON with big data for enabling 5G," *Network, IEEE*, vol. 28, no. 6, pp. 27–33, Nov 2014.

[74] F. Ricciato, E. Hasenleithner, and P. Romirer-Maierhofer, "Traffic analysis at short timescales: an empirical case study from a 3G cellular network," *Network and Service Management, IEEE Transactions on*, vol. 5, no. 1, pp. 11–21, March 2008.

[75] J. Liu, F. Liu, and N. Ansari, "Monitoring and analyzing big traffic data of a large-scale cellular network with Hadoop," *Network, IEEE*, vol. 28, no. 4, pp. 32–39, July 2014.

[76] "Apache Hadoop project," http://hadoop.apache.org/, 2015, [Online; accessed 02-April-2015].

[77] A. Karatepe and E. Zeydan, "Anomaly detection in cellular network data using big data analytics," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, May 2014, pp. 1–5.

[78] O. Celebi, E. Zeydan, O. Kurt, O. Dedeoglu, O. Iieri, B. AykutSungur, A. Akan, and S. Ergut, "On use of big data for enhancing network coverage analysis," in *Telecommunications (ICT), 2013 20th International Conference on*, May 2013, pp. 1–5.

[79] E. Bastug, M. Bennis, E. Zeydan, M. Kader, A. Karatepe, A. Salih Er, and M. Debbah, "Big data meets telcos: A proactive caching perspective," *Journal of Communications and Networks, Special Issue on Big Data Networking-Challenges and Applications (to appear)*, December 2015.

[80] M. Abdel Kader, E. Bastug, M. Bennis, E. Zeydan, A. Karatepe, A. Salih Er, and M. Debbah, "Leveraging big data analytics for cache-enabled wireless networks," in *IEEE Global Communications Conference (GLOBECOM) Workshop; Proceedings of*, December 2015.

[81] J. Magnusson and T. Kvernvik, "Subscriber classification within telecom networks utilizing big data technologies and machine learning," in *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, ser. BigMine '12. New York, NY, USA: ACM, 2012, pp. 77–84. [Online]. Available: http://doi.acm.org/10.1145/2351316.2351327

[82] G. Kramer, B. Mukherjee, and G. Pesavento, "IPACT: A dynamic protocol for an Ethernet PON (EPON)," *IEEE Commumincations Magazine*, vol. 40, no. 2, pp. 74–80, 2002.

[83] S. Sarkar, S. Dixit, and B. Mukherjee, "Hybrid Wireless-Optical Broadband Access Network (WOBAN): A review of relevant challenges," *IEEE/OSA J. Lightwave Technology*, vol. 25, no. 11, pp. 3329–3340, Nov. 2007.

[84] P. Chowdhury, B. Mukherjee, S. Sarkar, G. Kramer, and S. Dixit, "Hybrid Wireless-Optical Broadband Access Network (WOBAN): Prototype development and research challenges," *IEEE Network*, vol. 23, no. 3, pp. 41–48, 2009.

[85] K. Yang, S. Ou, K. Guild, and H.-H. Chen, "Convergence of Ethernet PON and IEEE 802.16 Broadband Access Networks and its QoS-aware dynamic bandwidth allocation scheme," *IEEE J. Selected Topics in Communications*, vol. 27, no. 2, pp. 101–116, Feb. 2009.

[86] G. Shen, R. Tucker, and C.-J. Chae, "Fixed Mobile Convergence Architectures for Broadband Access: Integration of EPON and WiMAX," *IEEE Communications Magazine*, vol. 45, no. 8, pp. 44–50, Aug. 2007.

[87] A. Reaz, V. Ramamurthi, S. Sarkar, D. Ghosal, S. Dixit, and B. Mukeherjee, "CaDAR: An efficient routing algorithm for a Wireless-Optical Broadband Access Network," *IEE/OSA J. Optical Communications and Networking*, vol. 1, no. 5, pp. 392–403, Oct. 2009.

[88] P. Chowdhury, M. Tornatore, S. Sarkar, and B. Mukherjee, "Building a Green Wireless-Optical Broadband Access Network (WOBAN)," *IEEE J. Lightwave Technology*, vol. 28, no. 16, pp. 2219–2229, Aug. 2010.

[89] S.-H. Lee, J. Kim, and M.-H. Kang, "Performance enhancement in future PON and mobile convergence networks," in *11th International Conference on Advanced Communication Technology (ICACT 2009)*, vol. 1, Feb. 2009, pp. 233–236.

[90] S. Newaz, Y. Bae, M. Ahsan, and J. Choi, "A study on PMIP deployment over EPON," in *9th International Conference on Optical Internet (COIN 2010)*, Jul. 2010, pp. 1–3.

[91] G. Giaretta, "Interactions between Proxy Mobile IPv6 (PMIPv6) and Mobile IPv6 (MIPv6): Scenarios and Related Issues," RFC 6612 (Informational), Internet Engineering Task Force, May 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6612.txt

[92] P. Serrano, A. de la Oliva, C. Bernardos, I. Soto, A. Banchs, and A. Azcorra, "A CARMEN mesh experience: deployment and results," in *IEEE Workshop on Hot Topics in Mesh Networking, HotMESH'09*, 2009.

[93] T. Pagtzis, "Advanced IPv6 mobility management for next generation wireless access networks," Ph.D. dissertation, University College London, 2005.

[94] R. Aguiar, A. Banchs, C. J. Bernardos, M. Calderon, M. Liebsch, T. Melia, P. Pacyna, S. Sargento, and I. Soto, "Scalable QoS-aware Mobility for Future Mobile Operators," *IEEE Communications Magazine*, vol. 44, no. 6, pp. 95–102, Jun. 2006.

[95] Y. Han, J. Choi, and S. H. Hwang, "Reactive handover optimization in IPv6-based mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 9, pp. 1758–1772, 2006.

[96] R. G. Clegg, "A practical guide to measuring the Hurst parameter," *Arxiv preprint math/0610756*, 2006.

[97] J. A. Hernandez and I. W. Phillips, "Weibull mixture model to characterise end-to-end Internet delay at coarse time-scales," *IEE Proceedings-Communications*, vol. 153, no. 2, pp. 295–304, 2006.

[98] C. Vogt and M. Zitterbart, "Efficient and scalable, end-to-end mobility support for reactive and proactive handoffs in IPv6," *Communications Magazine*, vol. 44, no. 6, pp. 74 – 82, June 2006.

[99] 3GPP, "Architecture enhancements for non-3GPP accesses," 3rd Generation Partnership Project (3GPP), TS 23.402, v11.2.0, 2012. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/23402.htm

[100] T. Melia, A. de la Oliva, I. Soto, C. Bernardos, and A. Vidal, "Analysis of the effect of mobile terminal speed on WLAN/3G vertical handovers," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, 27 2006-dec. 1 2006, pp. 1 –6.

[101] H. Yokota, K. Chowdhury, R. Koodli, B. Patil, and F. Xia, "Fast Handovers for Proxy Mobile IPv6," RFC 5949 (Proposed Standard), Internet Engineering Task Force, Sep. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5949.txt

[102] A. de la Oliva, I. Soto, M. Calderon, C. J. Bernardos, and M. I. Sanchez, "The costs and benefits of combining different IP mobility standards," *Computer Standards & Interfaces*, vol. 35, no. 2, pp. 205 – 217, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092054891200092X

[103] B. Lannoo, L. Verslegers, D. Colle, M. Pickavet, M. Gagnaire, and P. Demeester, "Analytical model for the IPACT dynamic bandwidth allocation algorithm for EPONs," *OSA J. Optical Networking*, vol. 6, no. 6, pp. 66–688, 2007.

[104] T. Melia, A. de la Oliva, I. Soto, P. Serrano, and R. Aguiar, "Network controlled handovers: challenges and possibilities," *Wireless Personal Communications*, vol. 43, no. 3, pp. 959–974, 2007.

[105] G. Kramer, B. Mukherjee, and G. Pesavento, "Ethernet PON (ePON): Design and analysis of an optical access network," *Photonic Network Communications*, vol. 3, no. 3, pp. 307–319, 2001.

[106] A. Myers, E. Ng, and H. Zhang, "Rethinking the service model: Scaling Ethernet to a million nodes," in *Proc. ACM SIGCOMM Workshop on Hot Topics in Networking*, 2004.

[107] G. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. Costa, and B. Walke, "The IEEE 802.11 universe," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 62–70, 2010.

[108] "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007*, 2007.

[109] G. Athanasiou, T. Korakis, and L. Tassiulas, "An 802.11k compliant framework for cooperative handoff in wireless networks," *EURASIP J. Wirel. Commun. Netw.*, pp. 23:1–23:14, Jan. 2009.

[110] S. Bangolae, C. Bell, and E. Qi, "Performance study of fast BSS transition using IEEE 802.11r," ser. IWCMC '06. New York, NY, USA: ACM, 2006, pp. 737–742.

[111] R. Meschke, M. Krohn, R. Daher, A. Gladisch, and D. Tavangarian, "Novel handoff concepts for roadside networks using mechanisms of IEEE 802.11k & IEEE 802.11v," in *ICUMT 2010*, 2010, pp. 1232–1238.

[112] T. Oliveira, M. Silva, K. Cardoso, and J. Rezende, "Virtualization for Load Balancing on IEEE 802.11 Networks," ser. 7th International ICST Conference, MobiQuitous 2010, Sydney, Australia, December 6-9, 2010.

[113] Y. Morisawa, Y. Kawahara, and T. Asami, "Reducing power consumption of IEEE802.11 stations in flexible multicast services," ser. MobiCom '13. New York, NY, USA: ACM, 2013, pp. 215–218.

[114] Y. Shin, M. Choi, J. Koo, and S. Choi, "Video multicast over WLANs: Power saving and reliability perspectives," *Network, IEEE*, vol. 27, no. 2, pp. 40–46, 2013.

[115] N. Choi, Y. Seok, T. Kwon, and Y. Choi, "Leader-Based Multicast Service in IEEE 802.11v Networks," in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, 2010, pp. 1–5.

[116] "Enterprise Best Practices for Apple Mobile Devices on Cisco Wireless LANs," 2013.

[117] P. Serrano, C. J. Bernardos, A. de la Oliva, A. Banchs, I. Soto, and M. Zink, "FloorNet: Deployment and Evaluation of a Multihop Wireless 802.11 Testbed," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, 2010.

[118] P. Rysavy, "Mobile Broadband Capacity Constraints and the Need for Optimization," February 2010.

[119] P. Daponte, L. D. Vito, F. Picariello, and M. Riccio, "State of the art and future developments of measurement applications on smartphones," *Measurement*, vol. 46, no. 9, pp. 3291 – 3307, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S026322411300198X

[120] Apple Inc., "iOS7 Developer Library," [Online]. Available: https://developer.apple.com/library/ios/navigation.

[121] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131 (Draft Standard), Internet Engineering Task Force, Mar. 1997, updated by RFCs 3396, 4361, 5494, 6842. [Online]. Available: http://www.ietf.org/rfc/rfc2131.txt

[122] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315 (Proposed Standard), Internet Engineering Task Force, Jul. 2003, updated by RFCs 4361, 5494, 6221, 6422, 6644, 7083. [Online]. Available: http://www.ietf.org/rfc/rfc3315.txt

[123] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861 (Draft Standard), Internet Engineering Task Force, Sep. 2007, updated by RFCs 5942, 6980, 7048. [Online]. Available: http://www.ietf.org/rfc/rfc4861.txt

[124] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," RFC 4862 (Draft Standard), Internet Engineering Task Force, Sep. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4862.txt

[125] B. Aboba, D. Thaler, and L. Esibov, "Link-local Multicast Name Resolution (LLMNR)," RFC 4795 (Informational), Internet Engineering Task Force, Jan. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4795.txt

[126] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet Group Management Protocol, Version 3," RFC 3376 (Proposed Standard), Internet Engineering Task Force, Oct. 2002, updated by RFC 4604. [Online]. Available: http://www.ietf.org/rfc/rfc3376.txt

[127] R. Vida and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6," RFC 3810 (Proposed Standard), Internet Engineering Task Force, Jun. 2004, updated by RFC 4604. [Online]. Available: http://www.ietf.org/rfc/rfc3810.txt

[128] H. Holbrook, B. Cain, and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast," RFC 4604 (Proposed Standard), Internet Engineering Task Force, Aug. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4604.txt

[129] E. Nygren, R. K. Sitaraman, and J. Sun, "The akamai network: A platform for high-performance internet applications," *SIGOPS Oper. Syst. Rev.*, vol. 44, no. 3, pp. 2–19, Aug. 2010. [Online]. Available: http://doi.acm.org/10.1145/1842733.1842736

[130] W. Gu, Z. Yang, D. Xuan, W. Jia, and C. Que, "Null Data Frame: A Double-Edged Sword in IEEE 802.11 WLANs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 7, pp. 897–910, July 2010.

[131] G. T. 23.402, "Architecture Enhancement for Non-3GPP Accesses (Release 12) v12.5.0," June 2014.

[132] "IEEE Standard for Information Technology-Telecommunications and information exchange between systems. Local and Metropolitan networks. Specific requirements. Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 9: Interworking with External Networks," *Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11w-2009, IEEE Std 802.11n-2009, IEEE Std 802.11p-2010, IEEE Std 802.11z-2010, and IEEE Std 802.11v-2011*, pp. 1–208, Feb 2011.

[133] J. Seedorf and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement," RFC 5693 (Informational), Internet Engineering Task Force, Oct. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5693.txt

[134] L. Sarakis, G. Kormentzas, and F. Guirao, "Seamless service provision for multi heterogeneous access," *Wireless Communications, IEEE*, vol. 16, no. 5, pp. 32–40, October 2009.

[135] 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*, vol. 61, no. 4, pp. 603–627, 2011. [Online]. Available: http://dx.doi.org/10.1007/s11277-011-0423-3

[136] A. De La Oliva, C. Bernardos, M. Calderon, T. Melia, and J. Zuniga, "IP flow mobility: smart traffic offload for future wireless networks," *IEEE Communications Magazine*, vol. 49, no. 10, pp. 124–132, Oct 2011.

[137] D. Thaler, R. Draves, A. Matsumoto, and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)," RFC 6724 (Proposed Standard), Internet Engineering Task Force, Sep. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6724.txt

[138] F. Ricciato, R. Pilz, and E. Hasenleithner, "Measurement-Based Optimization of a 3G Core Network: A Case Study," in *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, ser. Lecture Notes in Computer Science, Y. Koucheryavy, J. Harju, and V. Iversen, Eds. Springer Berlin Heidelberg, 2006, vol. 4003, pp. 70–82. [Online]. Available: http://dx.doi.org/10.1007/11759355_9

[139] "3GPP TS 29.060: General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface," http://www.3gpp.org/DynaReport/29060.htm, [Available online].

[140] "Cloudera," http://www.cloudera.com/content/cloudera/en/documentation.html, 2015, [Online; accessed 02-April-2015].

[141] W. Hahn, "3GPP Evolved Packet Core support for distributed mobility anchors: Control enhancements for GW relocation," in *ITS Telecommunications (ITST), 2011 11th International Conference on*, Aug 2011, pp. 264–267.

[142] F. Giust, C. Bernardos, and A. De La Oliva, "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, Nov 2014.