Technical Report

UCAM-CL-TR-656
ISSN 1476-2986

Number 656

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Seamless mobility in 4G systems

Pablo Vidales

November 2005

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted May 2005 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Girton College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/TechReports/

ISSN 1476-2986

Abstract

The proliferation of radio access technologies, wireless networking devices, and mobile services has encouraged intensive nomadic computing activity. When travelling, mobile users experience connectivity disturbances, particularly when they handoff between two access points that belong to the same wireless network and when they change from one access technology to another. Nowadays, an average mobile user might connect to many different wireless networks in the course of a day to obtain diverse services, whilst demanding transparent operation. Current protocols offer portability and transparent mobility, however, they fail to cope with huge delays caused by different link-layer characteristics when roaming between independent disparate networks. In this dissertation, I address this deficiency by introducing and evaluating practical methods and solutions that minimise connection disruptions and support transparent mobility in future communication systems.

Firstly, I show that repercussions on the link can be minimised by using the appropriate architecture. Recently, it has been considered necessary to integrate commercially-available wireless networks into a universal access platform, also known as the fourth generation (4G) communication system. I present the design and deployment of an experimental testbed that implements this 4G model, enabling roaming between the most popular wireless and wired technologies. In addition, the testbed is used to evaluate Mobile IPv6 performance in heterogeneous environments, and identify the weaknesses of this protocol to manage seamless roaming between different technologies.

Secondly, I demonstrate that Mobile IPv6 can be optimised by modifying its current specification without altering the core functionality or adding significant overhead. My experimental results show the improvements in vertical handover latency, which varied between small time reductions and "zero" latency, for the case of soft handovers.

Finally, I claim that minimising the latency is not enough to enable seamless roaming in highly-integrated and diverse networks. Therefore, I show the design and implementation of PROTON, a policy-based solution to assist nomadic users. To validate the strength of my proposal, I evaluate PROTON using the testbed, and demonstrate that it is possible to offer suitable mobility support in next generation systems, whilst handling the complexities and dynamic behaviour posed by the environment.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The explosion of radio access technologies and wireless networking devices over recent years has triggered the intensive use of nomadic computing. Mobile devices receive intermittent network access, and alternate between connected and disconnected states. However, today's personal gadgets have more networking capabilities, wireless network coverage is becoming ubiquitous, and always-on IP-based services are now closer to reality. An average mobile user might connect to a variety of wireless networks in the course of a day to obtain services, for which they demand operational transparency. Seamless networking is no longer a fuzzy concept; many efforts are being made towards the deployment of novel architectures, protocols, and support systems.

This dissertation is concerned with the design and implementation of a practical solution to support seamless mobility in future integrated heterogeneous networks. Through this work, I show that the complexities imposed by these new environments can be properly handled. My results demonstrate that current mobility protocols and solutions cannot cope with drastic link-layer variations—when roaming between independent networks—nor handle the complexities and intrinsic dynamics of upcoming networks.

In this chapter, I briefly introduce the most important terminology used throughout this work, followed by a description of the background issues that motivated my research. I then state the main contributions that are described in this dissertation, and summarise the contents of each chapter.

1.1 Terminology

This dissertation makes frequent use of a number of technical terms which are defined and discussed here for clarity and simplicity (Figure 1.1). Many of these terms are also included in Appendix A, where a more complete glossary is included.

The term **handover** is used to describe when a mobile terminal changes its attachment point to the Internet. However, the process of performing a handover is referred to as **handoff**. Rather confusingly, there are many types of handover and it is fundamental to understand what makes them different.

Handovers can occur between two access routers (AR) that belong to the same technology or different technologies; I reserve the term **homogeneous handover** for the former case and **heterogeneous handover** for the latter. If the previous two definitions are applied to a network architecture such as the one described in Chapter 3—in which the

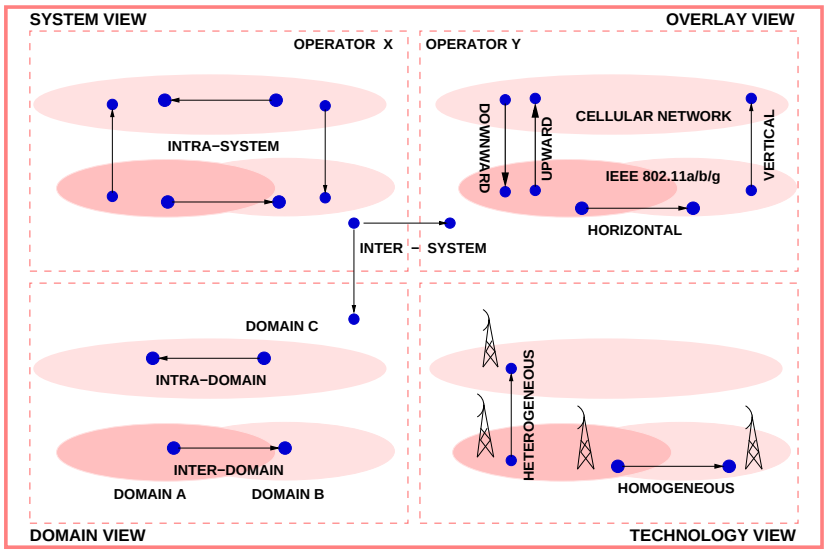


Figure 1.1: Handover taxonomy and views.

different access networks are organised in an overlay model and all the access points of a particular technology belong to the same overlay—I consider a **horizontal handover** to be when a mobile terminal performs a handover between two access points belonging to the same network overlay, and **vertical handover** when these belong to different ones. Furthermore, when the mobile terminal "moves up" in the overlay model, it is performing an **upward handover**—the old access router belongs to a technology with smaller coverage but more bandwidth than the new access router—and the opposite case is termed **downward handover**.

When performing a handover, the mobile node can roam within the same network domain—this is an **intra-domain handover**—but it can also cross more than one, executing an **inter-domain handover**. Finally, when the mobile node hands off between two independent systems—controlled by different network operators—I term it **inter-system handover**, and when the current and future access router are part of the same system it is a **intra-system handover**.

Another commonly-used term is **seamless mobility**, and in this dissertation it is defined as the capability to change the mobile node's point of attachment to an IP-based network, without losing ongoing connections and without disruptions in the communication. There are two scenarios where this concept is applied: **homogeneous systems** and **heterogeneous systems**. The latter implies the use of different integrated networks, and the former having many access points of the same technology available.

There are several different interpretations of what **4G technologies** actually are. In this dissertation, the transition to 4G does not imply a change in interface technology—as in previous transitions. Instead, 4G technology proposes to integrate various wireless technologies into a unified "ubiquitous" access platform—from indoor networks such as WLAN and Bluetooth, to cellular networks, TV broadcasting, and satellite. The ideal is a seamless merger, thus users of mobile devices can roam freely. This poses many challenges however, as will be described later.

1.2 Motivation

The seamless migration of ongoing data-flows from one access point to another is one of the fundamental issues in the deployment of the so-called next generation (i.e. fourth generation wireless networks). Migration is a challenge mainly due to the intrinsic limitations of mobile devices, the complexities in the upcoming highly integrated and largely disparate wireless networks and the highly dynamic behaviour of mobile environments. These constraints lead to many challenges in seamless networking:

- Delays due to inter-system roaming should be eliminated if we want to support real-time services in integrated environments. To do so, current optimisations to the Mobile IPv6 protocol proposed to anticipate handovers using link-layer hints, and to reduce signalling overhead by organising the network in a hierarchical topology. However, these methods cannot be directly applied to 4G communication systems. For example, before organising two independent networks hierarchically, we need to deploy an appropriate integration model, and this can result in incompatible topologies that do not follow a hierarchical model when combined.

- During inter-system roaming, disruptions are exacerbated by the adjustment of the connection to the new link. The adaptation to a different network can have an enormous impact on performance, affecting user experience. The analysis and characterisation of this effect need to be taken into account in the design of the next generation protocol stack and network architecture.

Furthermore, system designers increasingly believe that integration is necessary to satisfy the demands of broadband wireless services. For example, the trade-off, physically imposed, between coverage and bandwidth could be partially overcome by coupling low-bandwidth and ubiquitous coverage networks with high-bandwidth and local coverage technologies. This means the merger of cellular networks with wireless local area networks.

Unfortunately, the integration of disparate wireless overlays increases the complexity of the handover process, and can aggravate the effects of mobility on the protocol stack and user experience, in a number of ways:

- The concept of inter-system handover arises from this integration process, and it widens the spectrum of decisions and possibilities during roaming, which was limited in the horizontal scenario.

- In cellular systems, mobility affects only the dependent layers (physical and link layers), whereas in highly-coupled IP-based networks, the handoff occurs in layer three, causing disturbances in the upper layers, particularly the transport layer, and decreasing overall performance.

- The addition of new technologies, devices and services to networking environments dramatically increases resource management complexity. Furthermore, the demand for seamless operation and permanent connectivity increases the need for solutions that can handle these complexities, and hide them from the user.

There is a growing interest in practical mechanisms and solutions to facilitate seamless mobility in upcoming 4G communication systems, formed by highly-coupled and heterogeneous networks. By merging disparate wireless technologies it is hoped that users will enjoy ubiquitous access to plenty of services on the move. Unfortunately, the current reality is that most proposals are complex, insufficient, and impractical. This dissertation addresses this situation by presenting, deploying, and evaluating an integration architecture that facilitates seamless inter-system roaming. In addition, practical optimisations and solutions are proposed to support mobility in these scenarios.

1.3 Contribution

It is my thesis that transparent migration of ongoing data flows between two access points belonging to independent heterogeneous technologies is achievable, and that tools and mechanisms for supporting this type of mobility should be placed within the next generation networking architectures. Existing protocols do not cope with the impact of heterogeneity on mobile networking—in this dissertation I describe the necessary mechanisms and tools, which together can offer a feasible solution and whose performance and practicality can surpass current approaches.

- My first contribution is the design and deployment of an integration architecture that enables vertical handovers between the most popular wireless and wired technologies. Although many architectures have been discussed in previous work, the model described in this thesis has been fully deployed into an experimental testbed. The main points that distinguish our testbed from other setups are that it evaluates MIPv6 in heterogeneous environments—other projects are constrained by the architecture itself—and we use Vodafone's GSM/GPRS live network. As far as I am aware, our testbed is the only one that enables a GSM/GPRS overlay using the actual provider's network—previous projects emulate a GPRS link or install isolated GPRS base stations (BSs), suffering from the absence of real operating conditions during the experiments.

- Another contribution is a suite of Mobile IPv6 optimisation mechanisms for different conditions in future networking environments. These are interesting not only because of the reduction in handover latency, but also because they demonstrate that not every solution applied in horizontal scenarios works in the vertical case as well. Performance results show that these modifications to the current Mobile IPv6 specification can improve performance without affecting the core functionality or adding significant overheads. Furthermore, the analysis of Mobile IPv6 performance in heterogeneous environments, coupled with my findings, represent a further contribution to the state of the art in mobility management.

- My last contribution is the design and implementation of a software solution to enable complete mobility support for nomadic users in 4G communication systems. By fully identifying the characteristics of next generation environments (e.g., heterogeneity, high complexity, dynamic behaviour, and integration), it is possible to describe the real challenges in supporting transparent mobility. Contrary to current belief, today's solutions are not sufficient to assist mobile users in future networking environments.

1.4 Outline of the rest of the dissertation

In Chapter 2 I briefly describe the evolution of computing systems, concentrating on the most relevant aspects from the perspective of this dissertation such as terminal mobility and networking protocols. Then, I compare the main solutions to optimise roaming in homogeneous and heterogeneous systems. Finally, I discuss the related work to policy-based systems in network management, and introduce the concepts of context awareness and autonomic communications. A recurring issue is that enhanced terminal mobility is now essential to enable the continued expansion of wireless services and networks.

In Chapter 3, I examine the integration of wireless networks into a ubiquitous access platform, and tackle some of the main challenges that this poses. A discussion of different merging strategies is included, deepened by the detailed comparison of the testbed with similar projects. I conclude the chapter with the design and deployment issues of the experimental setup.

In Chapter 4, I present the performance analysis for MIPv6 during inter-system roaming, for which I used the testbed described in Chapter 3. In addition, I introduce the design and evaluation of four mechanisms to reduce latency: RA frequency, RA caching, BU simulcasting, and soft handover.

In Chapter 5, I discuss the idea of applying autonomic computing principles linked with policy-based systems, to offer assistance for users in future communication environments. I explain the architecture, design, and implementation of PROTON [1], as an example of such approach.

In Chapter 6, I explain how I tested my mobility assistance solution for correctness. I then present experimental results that demonstrate the practicality of my novel solution, its scalability in mobile environments, and the run-time performance. I close this chapter with a qualitative analysis of my solution compared to previous work. PROTON, together with the mechanisms covered in Chapter 4, can offer suitable roaming support in 4G mobile systems.

Finally in Chapter 7, I conclude this dissertation, list the open issues stemming from this work, and suggest areas for further research.

[1] **P**olicy-based system to **RO**am **T**ransparently among **O**verlay **N**etworks

Chapter 2

Mobility Management Overview

The advent of mobile computing implies the creation of novel communication architectures and the modification of computer networks, operating systems, and applications [39]. This chapter summarises previous work relating to mobility management and discusses the limitations which place existing solutions beyond practical use in 4G communication systems.

In 1994, Randy H. Katz [49] stated that the next logical step in the natural evolution of computing systems were *Wireless Information Systems* (Figure 2.1 shows an updated time-line based on Katz's original). We can see that Katz's vision was not far from the state of today's technology. However, we are still far away from what the research community considers as ubiquitous networking.

The enhancements made in networking and computing capabilities in mobile devices, together with the deployment of wireless networks, has improved the nomadic computing experience considerably. This revolution in the mobile world has extended users' presence by augmenting resource availability, increasing their productivity while travelling. These kind of facilities were not previously possible and now the pieces are in place to enable seamless terminal mobility, moving us one step closer to truly ubiquitous networking.

The remainder of this chapter is organised as follows. A brief introduction to mobility management protocols is included in Section 2.1. The challenge of terminal mobility is described in Section 2.2, focusing on the problem of transparent roaming in 4G systems. Then, Section 2.2.1 outlines the basic Mobile IPv6 (MIPv6) functionality. Section 2.3 presents protocols relating to terminal mobility and discusses why none of the proposed approaches are viable for future heterogeneous roaming. Finally, Section 2.4 motivates the use of policy models for mobility management.

2.1 IPv6: Next generation Internet protocol

The Internet Protocol (IP) was developed in the early 1980s with the aim of supporting connectivity within research networks, as part of *Catenet* [12]. However, in the last decade IP has become the leading network-layer protocol. It is the basic tool for a broad variety of client/server and peer-to-peer networks; it predominates in both wired and wireless worlds, and now the current scale of deployment is straining many aspects of its twenty five-year old design. To overcome the limitations inherent in the IP version 4 design, IPv6 [27] has been proposed as the new protocol that will provide a firmer

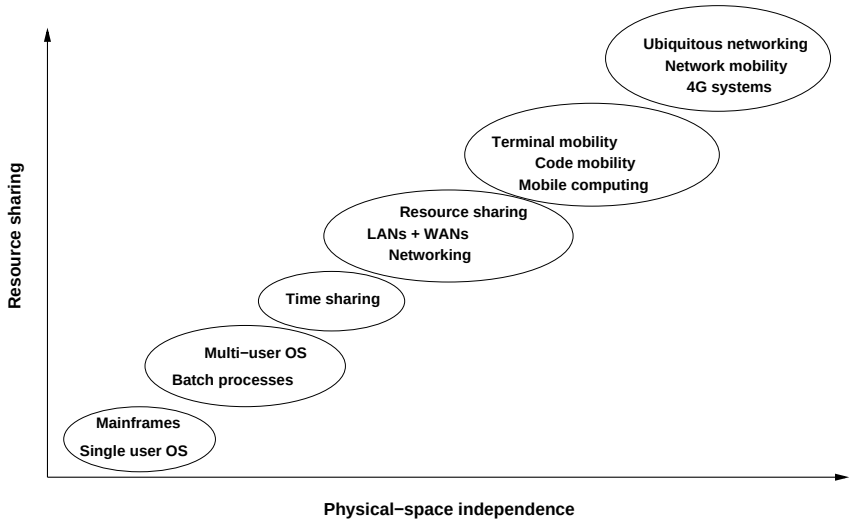


Figure 2.1: The evolution of mobile computing.

base for the continued growth of today's inter-networks. IPv6 was designed to improve IPv4's scalability, security, ease-of-configuration, and network management; these issues are central to the competitiveness and performance of network-related processes [52]. IPv6 was devised to enable high-performance scalable inter-networks in the fixed world. Moreover, when IPv4 was conceived, the wireless world was not even close to what it is today, thus IPv6 proposes solutions for some major IPv4 gaps in the support of mobility.

In forthcoming years, more people will access the Internet via wireless than via wired connections, and each user will have a set of wireless devices inter-connected that will be accessing a great variety of IP-based services. Currently, there are approximately 1 billion mobile phones in the world, and this number is expected to continue its exponential growth in the next few years. In light of this, IPv4 faces many problems related to its addressing and mobility capabilities when considered in the context of the mobile world. Current extensions to the protocol tackle these problems. However, IPv6 contemplates them within its design, enabling a more robust solution.

The Internet has experienced enormous change in recent years, and the number of users accessing services on the move has also grown exponentially. Every mobile device is potentially capable of accessing IP services, Wi-Fi networks are becoming wide-spread; the spurt in the hotspot market is being accompanied by similar growth in other wireless technologies such as Bluetooth, UltraWideband, and satellite, posing the urgent need for a larger address space and an adequate support for mobility.

The life of IPv4 has been extended via a technology called Network Address Translation (NAT)—a mechanism that conserves scarce IPv4 addresses. Essentially, NAT allows enterprises to deploy potentially large networks using shared IP-addressing space, and translating their Internet-bound traffic at their network edge to unique addresses assigned to their enterprise. In this way, an enterprise can deploy a thousand computer systems and only consume a handful of unique IP-addresses.

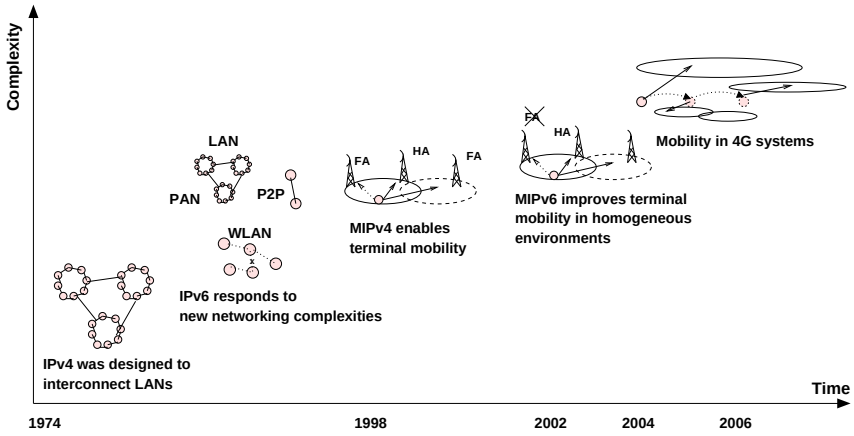


Figure 2.2: Networking evolution and driver protocols.

Thus, NAT is effective in the way it allows more nodes to join the network than would be possible if all nodes required routable addresses. This capability comes at a cost—that cost is the loss of key functionality. As the Internet develops, more advanced applications will require end-to-end connectivity throughout the network–just the capability that is not provided to NAT-enabled nodes. Additionally, many of these applications will require extensive changes to facilitate operations through NAT, creating technically complicated architectures. One last additional drawback for NATs is related to scalability constraints. The shared IP-address needs to be converted to a valid IP-address in every outgoing and incoming packet, which represents a huge constraint in large-scale networks.

Whereas the main thrust of IPv6 is to increase the address space and overcome NAT's drawbacks, it also provides important functions to enable mobility (e.g., scaling and ease-of-configuration) mainly derived from the larger address space. IPv4 has difficulties managing mobile terminals for several reasons such as address configuration and location management.

To drive the evolution in the mobile world, Mobile IPv6 [48] was proposed as a protocol that exploits the added features in IPv6 to extend it and enable micro-mobility (roaming between access points within the same network domain) and macro-mobility (roaming to access points outside the current domain). With the introduction of IPv6, many of the disadvantages of the previous version of mobility support (i.e. Mobile IPv4 [75]) were eliminated. However, for Mobile IPv4 and Mobile IPv6 supporting seamless roaming in heterogeneous environments is outside the requirements.

The rapidly growing demand for "anywhere, anytime" high-speed access to IP-based services is becoming one of the major challenges for mobile networks. As the demand for mobility increases, mobile terminals need to roam freely across heterogeneous networks, posing the challenge of network integration into an All-IP ubiquitous access platform [113]. Mobile IPv6 stands as *de facto* solution for mobility management in next-generation systems. Highlights of networking evolution are shown in Figure 2.2; IP and Mobile IP protocols are considered key players in the unfolding of networking in past, present, and future mobile communication systems.

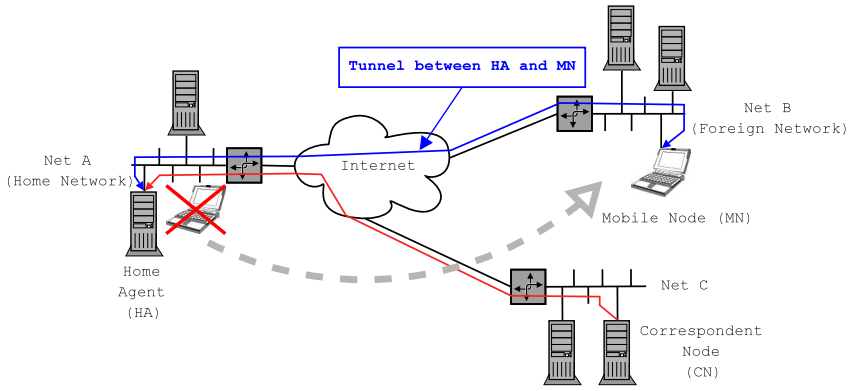


Figure 2.3: Basic Mobile IPv6 case example.

2.2 Terminal mobility

IPv4/v6 addressing and routing schemes entail that a host address relates to the point at which the host is connected to the network. This is exactly the opposite of what is needed for mobility, because a mobile node frequently changes its attachment point and therefore its address configuration. In fact, the TCP/IP network architecture has an imperfect layered structure, in which the transport layer not only uses source and destination TCP port numbers (ports are the network abstraction at the TCP layer) but also the source and destination IP addresses as the connection identifier. This leads to a bad use of the IP address where it has dual functionality: (1) to identify connections above the network layer and (2) to determine the packet's routing. Thus, the host needs to have a stable address to identify ongoing connections and it should acquire a routable address from the network to which it is currently connected.

Although during the last decade protocols for dynamic IP address assignment—for example, DHCP [30]—have been designed, these solutions only provide portability and not transparent mobility to roaming nodes. Portability is the terminal mobility capability that allows a host to change its location. However the terminal is required to stop and restart its upper layer connections (e.g., TCP flows). In contrast, transparent mobility allows a terminal to move between different networks without dropping any ongoing connection. IPv6 provides portability because of its mechanisms for automatic address configuration, but not transparent mobility. MIPv4 and MIPv6 are the protocols defined to provide support for reachability and transparent mobility in future All-IP networks.

The problem of mobility management in IP-based networks is twofold. The problem of managing relationships between home addresses and care-of addresses, and the problem of using the appropriate type of address in relation to the context.

2.2.1 Mobile IPv6: How does it work?

Figure 2.3 shows the basic MIPv6 functionality as described in the IETF RFC at the time of writing. Mobile IPv6 defines mechanisms that allow a terminal to change its point of attachment to the Internet, and remain reachable through a permanent address whilst preserving its active connections when travelling to its new location.

While a Mobile Node (MN) is connected to its Home Network (HN), i.e. the network where its Home Address (HoA) is located, no special mode of operation is needed and packets are forwarded (using normal IP routing) between the mobile node and any other node it is communicating with (the correspondent node).

When a MN is connected to a network other than its home network, the MN acquires an IPv6 address belonging to the address space of the Foreign Network (FN)—called the Care-of Address (CoA). The MN announces its CoA by sending a Binding Update message (BU) to an special entity knows as the Home Agent (HA) that is located in the MN's home network. This special router, the home agent, acts on behalf of the MN in its absence and usually serves all mobile nodes in a home network. The HA traces all the MN's movements, maintaining the mapping between HoA and CoAs using BUs sent by the nodes. This is known as the binding cache.

The HA intercepts the packets sent to the MN's home address while the MN is away from its home network and establishes a bidirectional tunnel terminating at the MN's CoA. This tunnel is used to redirect intercepted packets to any MN's current location. The MN also uses this tunnel to send its traffic to the Correspondent Nodes (CN), thus avoiding ingress filtering.

Furthermore, MIPv6 also defines a Route Optimisation (RO) procedure to avoid the suboptimal routing problem caused by the use of bidirectional tunnelling through the HA. This procedure enables the MN to also send binding updates to the CNs. Packets sent by the MN then have the MN's CoA as source address, but also carry a special IPv6 *home address destination option*, containing the MN's home address, allowing a CN to use this address as the source address when delivering the received packets to its upper layers (i.e. mobility is transparent to the layers above IP). In the reverse direction, the CN sends the packets addressed to the MN's CoA, but also inserts a routing header Type II (see Section 15.9 of [48]) with the MN's HoA as a unique next hop. In this way, the MN can also manage mobility in a transparent way with respect to those layers above IP.

Hence, we can see that Mobile IPv6 is suitable for providing support for roaming between networks and that it can be used from an Ethernet network to a wireless network, between homogeneous networks and, of more relevance to this work, between diverse access technologies.

Nevertheless, note that Mobile IPv6 has been conceived to support macro-mobility and it is less suitable for micro-mobility. This dissertation includes the analysis of MIPv6 to support macro-mobility in 4G systems, in which MNs will normally roam between disparate wireless technologies to obtain ubiquitous connectivity. Issues related to MIPv6 performance and potential optimisations are discussed in chapters 3 and 4.

2.3 Terminal mobility protocols

Many projects have proposed schemes to support host mobility. These solutions can be classified into *micro-mobility* and *macro-mobility* according to the type of roaming that they support. When a MN executes a vertical handover it usually changes its domain, and appropriate macro-mobility support is required. In this Section, popular schemes for micro- and macro-mobility are contrasted using the features listed in Table 2.1.

Unlike traditional link layer handovers (e.g., those in cellular networks) vertical handovers take place in different layers according to the level of integration between the

different access technologies (see Section 3.1.3). Thus, vertical handovers can occur at the network layer, like Mobile IP, or even at higher layers, e.g., using TCP Migrate [86] or SIP [106]. Moreover, there are significant differences between the solutions depending on where the handover occurs.

The aim of any mobility support scheme is to provide optimised and efficient roaming in every possible scenario. Unfortunately, this may not always be possible. Even some link-layer handover latencies can still be very high, as much as 1000 ms in GSM, if handover decision and execution timeshare are included [96], and latencies increase when dealing with inter-domain or network-layer handovers. Most communication systems can exploit domain coverage overlap to reduce handover latencies by employing a "make-before-break" soft handover approach. To further minimise handover latencies, access networks are also organised hierarchically to decrease update latency and localise signalling overhead. This results in handover latencies generally suitable for voice traffic.

Transparent host mobility across heterogeneous networks demands a universal, physical-layer independent host mobility solution at the network layer or higher. Most common approaches are aimed at reducing handover latency, signalling load, and improving scalability and robustness.

Table 2.1 compares several schemes, each of them with particular advantages for host mobility. These solutions do not always preclude the simultaneous usage of others listed in the table; instead, they may complement each other to offer improved performance. For example, the Intra-Domain Mobility Management Protocol (IDMP) leverages existing macro-mobility management protocols, such as Mobile IP and SIP [106], for locating roaming nodes.

Table 2.1 includes a set of characteristics that are relevant for both micro- and macro-mobility. These aspects are defined below:

- **Handover latency**: To improve user experience, it is important to maintain handover latency as low as possible.

- **Mobility**: It is difficult to define what is high and low mobility. However, in the context of this work high mobility means the capability to cope with more than 10 handovers per minute. In contrast, a host performing less than 10 handovers is considered to have low mobility.

- **Low signalling**: In order to reduce latency, it is important to maintain low signalling and decrease the use of bandwidth for sending control or management data.

- **Real time**: This characteristic evaluates the capability of the solution to reduce delays caused by host mobility and achieve acceptable values for real time applications.

- **QoS support and AAA support**: Another requirement for a complete solution is to offer QoS and AAA support during roaming. Most of the mobility management protocols keep functionalities separated to avoid complexities.

- **Scalable**: This property evaluates if there is some part of the solution that affects its scalability e.g., signalling, delays, etc.

	Micro-mobility			Macro-mobility			
Solutions	IDMP	CellularIP	HAWAII	SIP	TeleMIP	S-MIP	TCP Migrate
Low handover latency	√	√	√	×	√	√	×
High mobility	√	√	√	×	×	√	×
Low signalling	√	√	√	√	√	√	√
Real-time	×	√	√	√	×	×	×
QoS support	×	×	√	√	×	×	×
Scalable	√	√	√	√	√	√	√
AAA support	×	×	×	×	×	×	×
MIPv6 compatible	√	√	√	√	√	√	√
Optimisations	√	√	√	√	√	√	√
Global mobility	×	×	×	√	√	√	√

Table 2.1: Host mobility support.

- **MIPv6 compatible**: Compatibility with MIPv6 is considered because it is de facto standard in industry and academia. Table 2.1 indicates if the solution is compatible or not with MIPv6.
- **Optimisations**: There are some well defined methods to optimise mobility management. This property indicates if any of them are used in the solution e.g., HMIPv6, L2-hints, etc.
- **Global mobility**: It is also shown if the protocol supports global roaming (micro- and macro- mobility), or if it needs other solutions to handle complete mobility.

2.3.1 Micro-mobility solutions

IDMP [25, 67] is a standalone approach that provides a multiple CoA intra-domain mobility solution, and is one of the several IP-based hierarchical mobility management solutions that attempts to minimise handover latency. But unlike micro-mobility solutions like HAWAII or Cellular-IP, it can be made to work, completely independently, with Mobile IP for enabling global host mobility.

IDMP can enhance Mobile IP in micro-mobility environments with high handover frequency (more than 1 handover per second) and also could decrease signalling load in a global solution. Furthermore, it can offer paging services to locate mobile nodes within a particular domain and save power, an important consideration for next generation's resource-limited or constrained devices.

However, the protocol is not enough to manage mobility in 4G communication systems because it cannot offer macro-mobility support by itself, despite the fact that it optimises intra-domain roaming. Heterogeneous roaming demands a global solution such as Mobile IP, but one that is optimised to perform well in inter- and intra- domain roaming. IDMP originally was deployed using the Linux Mobile IP code of the Standford University Mosquito Project that did not study vertical roaming [55].

Similar approaches have used methods originally applied to cellular systems to optimise network-level handovers. Micro-mobility protocols such as Cellular IP [11] and HAWAII [80] optimise handover latency and reduce signalling load by distinguishing between the movement of the MN within the domain and outside the domain (i.e. hierarchical handovers). Also, other optimisations are based on the availability of cross-layer information, particularly from the link layer, to anticipate handovers, so called "fast handovers".

The CellularIP protocol [10] from Columbia University and Ericsson Research supports paging and a number of handover techniques and optimisations. To minimise signalling, regular packets are sent to update host location information. CellularIP avoids paging to minimise signalling and reduce power consumption. However, CellularIP is also limited to supporting heterogeneous roaming between different domains, and relying on some inter-domain mobility management protocol to support global mobility, such as Mobile IP.

HAWAII, from Lucent Technologies, proposes a separate routing protocol for micro-mobility. HAWAII [80] relies on Mobile IP for inter-domain roaming. One important aspect is that HAWAII is not a standalone solution, it extends Mobile IP to provide intra-domain mobility with QoS support. When Mobile IP is used for micro-mobility, it results in high control overhead due to the frequent notifications sent to the home agent, and high latency causing disruptions during handover. Also, in the case of a QoS-enabled host, acquiring a new care-of address on every handover would trigger the establishment of new QoS reservations along the complete path to the CN, even when the majority of it remains unchanged. HAWAII leverages Mobile IP to enable QoS-aware micro-mobility.

2.3.2 Macro-mobility solutions

To overcome the limitation of global mobility, TeleMIP [26] combines IDMP and Mobile IP for intra-domain and inter-domain mobility support respectively, to provide an attractive and scalable mobility management solution for All-IP networks. Although hierarchical extensions to Mobile IP clearly optimise high-frequency updates, TeleMIP's authors argue that introducing multiple levels of hierarchy in a commercial multi-level provider environment can often lead to network management and security issues. Instead of having a multi-level hierarchy, TeleMIP attempts to achieve a balance between the problems of high update latency and complex management architectures by introducing an structure that makes use of only a two-level hierarchy.

Despite the fact that TeleMIP improves Mobile IP performance in the intra-domain scenario, it does not modify MIP for inter-domain roaming. This aspect makes TeleMIP unsuitable for heterogeneous environments where roaming between different domains is a very common situation.

S-MIP (Seamless Handoff architecture for Mobile IP) differs from previous schemes because it combines the advantages of both fast handovers [53] and hierarchical mobility schemes [88] to enable what it calls "smarter" handovers. It introduces the concept of Synchronised Packet-based Simulcast (SPS) by simulcasting packets to the current and new networks, thus minimising packet loss during handovers. S-MIP [43] builds on the structure of Hierarchical Mobile IPv6 (HMIPv6) with fast handovers and operates similarly to the Mobile Node Initiated Fast handover scheme [53]. Unlike the HMIPv6 with fast-handover approach that uses layer-two triggers, in S-MIP the network uses the MN's location and movement patterns to instruct the MN when to handoff. S-MIP uses physical context data to enable context-aware handovers in a similar manner as the mobility support solution presented in Chapter 5. In [42] the authors present a simulation of S-MIP that compares it with plain HMIPv6 and fast handovers for homogeneous micro- and macro-mobility. Similarly, this work evaluates MIPv6 during vertical handovers using an experimental testbed that emulates a 4G system.

At the time of writing, apart from MIPv4/MIPv6, the IETF proposes two main solutions to manage mobility: Hierarchical MIPv6 and a Fast Handover Protocol. Even though MIPv6 offers several benefits, the signalling overhead's effect on the network load can at times be significant and the handover process can be long. Hierarchical MIPv6 focuses on local movements to reduce the signalling load on the network. The idea behind HMIPv6 is to divide the global Internet into logical regions defining domains that are independent from subnets. This mitigates the signalling load on the network. On the other hand, the Fast Handover Protocol, also an extension to MIPv6, allows access routers to offer services to the MN in order to anticipate handovers. The movement anticipation of the MN is typically based on layer-two (L2-level) triggers. These two optimisations will be discussed further in Section 4.1. An IETF working group, called *Seamoby* [50], is considering the complex interaction of parameters and protocols needed for seamless handover. The two main issues dealt with at the time of writing in Seamoby are the dormant mode host alerting problem (i.e. paging) and context transfers between nodes in an IP access network.

TCP Migrate [86, 87] provides a way of achieving session-layer host mobility. Here, TCP is modified on both the mobile and correspondent nodes such that it can withstand changes in IP address during a connection. Using DNS, the correspondent node learns the current address of the MN, with the DNS being updated every time the host moves. However, TCP Migrate lacks support for location privacy, and cannot have two mobile nodes communicating simultaneously, making it suitable only for client-server type of applications (e.g., email, web downloads, etc.) and not appropriate for peer-to-peer topologies.

Finally, SIP [106] supports higher-layer (application-level) host mobility. SIP exploits knowledge about the traffic at a higher layer to benefit real-time flows. This scheme is quite similar to MIPv6 (or MIP with route optimisations) and is especially advantageous for real-time traffic, both voice and video, as it reduces end-to-end latency by allowing a CN to directly communicate with the mobile node's CoA, without needing direct traffic tunnelling through the home agent.

In conclusion, many solutions have been proposed to solve transparent mobility. The perfect protocol does not exist; trade offs between overhead and performance are always present (see Table 2.1). Mobile IPv6 has become the standard for industry and academia, but its deployment in 4G systems poses performance issues that need to be addressed. A wide body of research in mobility has resulted in a number of approaches to optimise Mobile IPv6. Chapters 3 and 4 discuss practical mechanisms proposed to tackle this challenging issue.

2.4 Policy models to enable seamless mobility

Networking complexities in future multi-homed devices equipped with several interfaces, which may belong to different access technologies, and upcoming 4G systems pose the need for assistance during roaming. Currently, few solutions propose any means for the user or application to be able to dynamically influence the roaming process (network selection, handover initiation, execution, etc.) However, in forthcoming communication systems, tools to control network roaming will be essential.

There has been considerable effort dedicated to developing policy models to control data storage [85], quality of service [108, 36], and system security [89, 29]. There are models to influence the behaviour of mobile components [17, 18] for adapting them to evolving system conditions. Of relevance to this work, policies are being used in mobile environments [40] to provide means of specifying the adaptive behaviour in networks and devices [84], and to enable seamless mobility in future ubiquitous environments [101].

Chapter 5 describes a policy-based system, PROTON, that provides mobility management support in 4G systems. To the best of the author's knowledge, the proposed model is the first that addresses mobility assistance in such a fashion [100]. Other projects exploit advantages offered by policies to tackle mobility management, yet there are meaningful differences that are shown in the qualitative evaluation included in Chapter 6—in which PROTON is extensively contrasted with other approaches.

Many important issues related to mobility management have already been addressed using policy models, however, a general solution has not yet been implemented. Most of the work done focuses on particular problems, mainly due to the complexity implicit in offering complete mobility support—it can be beyond mobile devices' capabilities.

One of the problems that was encountered when the concept of overlay networks was first proposed at UC Berkeley, was controlling handovers between independent networks. In [104] this problem was first addressed, the proposed handover mechanism being based on simple decision policies according to cost functions, and it was limited to the selection of the best access network. The authors mentioned that offering full assistance would result in an excessive increase in complexity; for this reason, it is argued in this work that it would be better to use simple policies, instead of cost functions, as they are more flexible and have lower computation overhead. They alluded to the possibility of the inclusion of more parameters in the policy model to enrich the decision-making process.

Further projects followed this work and developed other approaches such as fuzzy logic algorithms [14] and more comprehensive policy models. Some of these solutions utilised complex decision-making tools—neural networks or fuzzifiers—that are disadvantageous for the dynamics of mobile environments. In the handover process the basic problem is not to find the "best" solution, but a fast and convenient decision. Multi-homed devices were enhanced with interface selection mechanisms supported by policies [112].

These mechanisms allowed dynamic decision-making during the operation of the mobile device. However, initial approaches were unable to use more than one network interface simultaneously. More recent solutions for policy-based routing enabled mobile devices to utilise all active network interfaces and provided optimum interface selection. Most of them were based on Mobile IPv4/v6.

Wakikawa et al. [103] proposed modifications to Mobile IPv6 to enable the use of multiple network interfaces. They managed to enhance the protocol, but not to enable a seamless handover process due to the latency of switching interfaces (this fundamental obstacle is tackled in Section 4.1) and in the TCP congestion control algorithm owing to packet losses (discussed in Section 4.5.4).

Mobile devices began to proliferate and wireless networks became insufficient for the number of users per access point. Smart network selection began to be useless without a supporting access architecture. Therefore, the use of policy models to enable intelligent access [71], admission control, and mobility management were explored.

After solving the basic problem of handover management, research moved towards seamless mobility. At the same time, emerging protocols such as Mobile IPv4 and Mobile IPv6 triggered the design of new solutions. The use of signal strength information—usually related to link-layer handoffs—to enhance network-layer Mobile IP handovers was proposed and successful mechanisms were developed. In [4] the authors proposed the use of link-layer hints to improve networking, posing the idea of cross-layer data exchange. The concept of designing data-aware handover mechanisms is taken further in this dissertation with PROTON that uses a rich dataset (Networking Context), which includes data from every layer, from the user, and from the physical environment.

Policy-based handover initiation methods [13] reduce the effect of mobility on performance, permitting the support of a greater number of mobile services. A new wave of horizontal handover mechanisms facilitated the deployment of mobile real-time applications, by providing transparent mobility between two hotspots [9].

Patanapongpibul and Mapp [74] suggested caching incoming RAs from every potential access point, and controlling the cache with some sort of replacement criterion to ensure RA validity. This method reduces horizontal handover latency by avoiding network discovery delays. They claimed a latency reduction from approximately 2200 ms to 250 ms (without packet loss) for TCP flows.

At this point, a problem emerged as a consequence of the broad variety of mobile devices, wireless technologies, and services. Heterogeneity became the problem tackled by another generation of policy-based solutions. Decision-making happens to be more complex in these environments, and as a response novel context-aware policy models arose.

Context knowledge was essential to improve networking performance. Connectivity showed high dependencies on surrounding conditions and many networking tasks became context-dependent. Limiting decision-making to the received signal strength was no longer possible. Context-aware solutions were developed and supported by policy-based architectures, strongly motivating the creation of policy models to handle the complexities posed by pervasive interaction in heterogeneous systems.

In [47] a network-oriented methodology to offer context-aware services is presented. This European project (CONTEXT) proposes a scenario where context information is collected by the network, and used to provide context-aware services to mobile clients. Thus, when MNs connect to the CONTEXT-based network, they receive the "best possible" QoS based on current network context.

Making use of this momentum, PROTON proposes a formal policy representation that allows MN's to react to context data. It can be seen as a collection of context-triggered actions adapted to hostile mobile environments. PROTON provides assistance for wireless worlds, in which devices are exposed to unpredictably changing contexts, such as unreliable connectivity conditions, by actively or passively migrating through one environment or between multiple environments. This together with the heterogeneity in access networks increases the number of parameters to be considered when reasoning about networking resources.

Context awareness increases autonomy in systems and represents a good way to hide unnecessary complexities from users. A context-aware system is one in which applications have knowledge of their surrounding physical and computing environments. This knowledge can be applied to minimise user participation by the definition of the governing rules

between networking resources and user activities in 4G systems. The research community recognises the general issue of complexity as the crucial inhibiting factor of technological advancement and proposes context-aware policy models as a holistic solution. However, complexities posed by forthcoming 4G systems can go beyond the capabilities of policy-based solutions forcing a new evolution.

In its visionary manifesto IBM, anticipating the need for more autonomous solutions, expands the frontiers of policy models. Foreseeing the insufficiency of current models, it outlines the concept of *autonomic computing* [44]. By deploying novel autonomic solutions, the following issues could be addressed: reducing networking complexity, coping with proliferation of wireless access technologies and networking paradigms, supporting mobility, providing personalised services, and adaptive computing, and improving performance and efficiency. Constraints and challenges in mobile environments need to be studied and novel solutions proposed before taking further steps towards seamless mobility.

Chapter 3

The LCE-CL Experimental Setup

We are witnessing the development and deployment of a large number of wireless networking technologies including 3G, WLANs, Bluetooth, and UltraWideband. At the same time we are seeing a convergence of core networking infrastructure based on the Internet Protocol Suite (IP) [94]. IPv4 is widely deployed throughout the world and there is now a serious effort to deploy IPv6, which simplifies mobility support.

There is a significant need for a unified approach that integrates all disparate wireless technologies (see Table 3), and enables mobile users to seamlessly roam between networks while accessing applications with different service requirements. This convergence poses many challenges, which need to be solved before the deployment of a real 4G system.

Vertical handovers are challenging for current transport protocols, because packets may get lost, disordered, or delayed during the handover and therefore affect performance. Moreover, methods to minimise latency during vertical handovers are needed to support real time applications in these future systems.

Network	Coverage	Data Rates	Cost
Satellite	World	Max. 144 kb/s	High
GSM/GPRS	Aprox. 35 km	9.6 kb/s up to 144 kb/s	High
IEEE 802.16a	Aprox. 30 km	Max. 70 Mb/s	Medium
IEEE 802.20	Aprox. 20 m	1-9 Mb/s	High
UMTS	20 km	up to 2 Mb/s	High
HIPERLAN 2	70 up to 300 m	25 Mb/s	Low
IEEE 802.11a	50 up to 300 m	54 Mb/s	Low
IEEE 802.11b	50 up to 300 m	11 Mb/s	Low
Bluetooth	10 m	Max. 700 kb/s	Low

Table 3.1: Diversity in existing and emerging wireless technologies.

In 2002, the Laboratory for Communication Engineering (LCE) and the Computer Laboratory (CL) at the University of Cambridge came together to develop the LCE-CL testbed, which has been used to study the challenges related to the deployment of forthcoming mobile systems. This chapter presents the design, integration architecture, and deployment of the testbed. The goal is to build a platform that fully integrates heterogeneous wireless technologies, anticipating that in the near future mobile devices will have several wireless interfaces and users will expect connections to be seamlessly

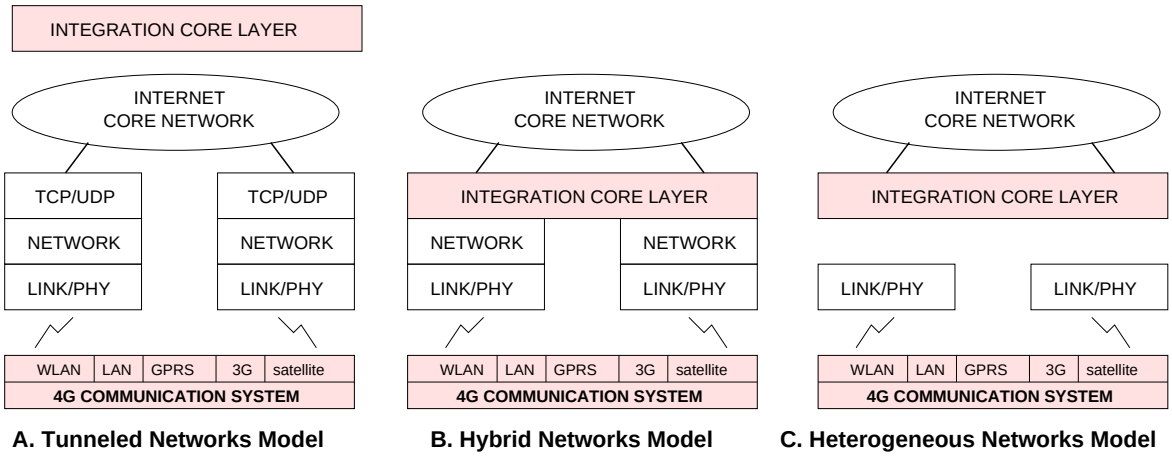


Figure 3.1: Taxonomy according to the integration layer.

managed. In that sense, the testbed can be regarded as a prototype of a 4G system with particular focus on mobility. Experimental activities play a vital role in the development and deployment of novel radio access networks. In particular the move from 3G to 4G poses new challenges, which need to be solved using practical approaches such as testbeds.

The remainder of this chapter is structured as follows: Section 3.1 describes several possible integration techniques from the perspective of the OSI network model. The techniques used to build the testbed are described in a three-dimensional taxonomy. Section 3.2 introduces the testbed, mentioning important aspects that are used to contrast it with similar work. Section 3.3 compares the LCE-CL testbed with previous projects, focusing on the aspects that make it unique and different from the others. Section 3.4 includes important hardware-related information for the deployment of the testbed. The relevant software components are mentioned in Section 3.5. Finally, Section 3.6 closes the chapter summarising the main objectives of the LCE-CL testbed.

3.1 Integration techniques

One of the main features of the 4G communication systems is the inter-operation of multiple radio access technologies (RATs). Contrary to homogeneous environments, many approaches can be taken depending on the level of integration desired between different RATs; the level of integration achieved is strongly correlated with the degree of modification required for each individual technology. This section places the LCE-CL testbed into a three-dimensional taxonomy: (1) the OSI-model layer where the integration takes place, (2) the component that connects the disassociated technologies, and (3) the common and independent functionality in the architecture.

3.1.1 OSI-layer integration

There are several architectures using multiple RATs. The basic models—considering the integration layer—are shown in Figure 3.1 [109]. The LCE-CL testbed integrates disparate access networks using a core IP-layer to manage networking (i.e. a heterogeneous networks model).

- **Tunnelled Networks:** Upper layers access the different technologies independently. According to policies, the best network is selected and the *integration layer* tunnels the traffic across the Internet and the chosen RAT. Thus, no modifications are required to the existing network stacks. However, service latency increases, mainly because of duplication and lack of integration in the lower layers.

- **Hybrid Networks:** In this model, the individual RATs implement the three lowest layers (Physical, Link, and Network layers). There is a hybrid core that interfaces between the Internet and the different wireless access networks. The main drawback of this model is that networking activities are duplicated, however, the stack does not need to be modified. Nevertheless, the service latency reduces because there is not as much redundancy in functionality as in tunnelled networks.

- **Heterogeneous Networks:** In this model there is a core layer that deals with all the network functionality and operates as a single network with respect to the upper layers. Thus, different RATs implement only the physical and link layers, which are specifically related to each technology. A major obstacle of this model is that the different access networks must converge, which requires a huge standardisation effort and operator commitment.

Nevertheless, heterogeneous networks are a promising solution for 4G systems. Using a module-based design to minimise impact on the networking stack, current protocols can be changed to include Mobile IP and thereby provide the required level of integration.

3.1.2 Networking-component integration

Inter-networking between wireless technologies was considered by the *3GPP TSG* [1] working group. This group drafted a feasibility study in which they presented four levels of integration between RATs, according to the component where coupling takes place [2]. The main integration scenarios are shown in Figure 3.2 and listed below.

- **Open Coupling:** There is no real integration effort between two or more access technologies. Thus, separated sub-processes take place, however the billing system is shared between networks. These models do not enable seamless inter-technology handovers. When the terminal changes its current access router to another, the ongoing session is terminated.

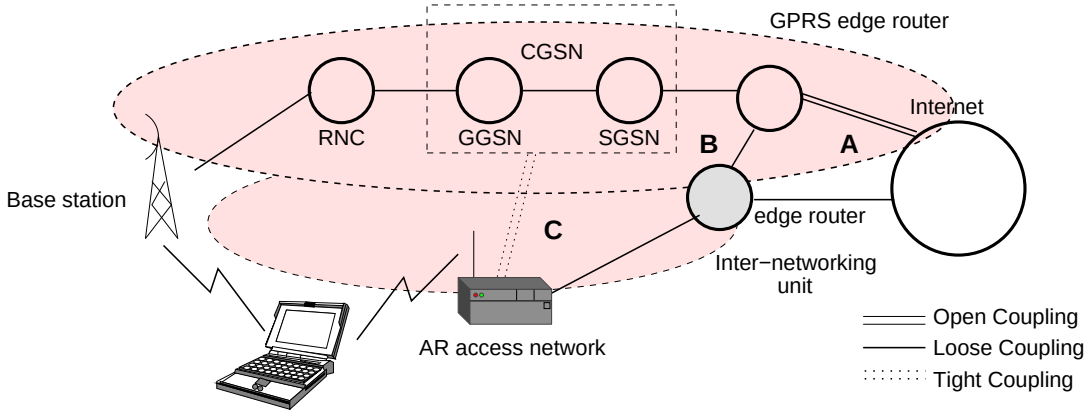


Figure 3.2: Integration Models: Depending on the integration components.

- **Loose Coupling:** Defined as the utilisation of a generic RAT (e.g., WLAN) as an access network complementary to current 3G access networks. It uses a common subscriber database without any user plane Iu interface[1], i.e. avoiding the SGSN and GGSN nodes. Thus, the RATs are integrated in the network layer by adding special purpose inter-networking components.
- **Tight Coupling:** The key characteristic of this model is that the generic access networks (e.g., WLAN) are connected to the core network (e.g., GSM/GPRS) sharing the Iu interfaces. Thus, the level of integration impacts the core components—the GGSN and SGSN in the case of Vodafone's GPRS architecture. This enables the integration of most of the operational capabilities into a single platform. There is one more type of integration considering the components that need to be modified. Networks can also be fully integrated, meaning that the integration affects core components in both, core and access networks.

Broadly speaking, integration architectures have been classified into loose coupling and tight coupling, as the main difference among them is the ability to offer service continuity [2].

3.1.3 OSI-functionality integration

There are other reasons to have the integration built into the network layer or above. The OSI model separates functionality into layers, lower ones are responsible for connection control and upper layers are in charge of presenting the data. The lower two layers (i.e. physical and link layers) are strongly attached to the specific access technology in use. Signalling and control for a specific technology happen between these two layers, whereas IP facilitates the integration of heterogeneous networks because it only includes control and signalling (e.g., addressing, routing, encapsulation of packets or datagrams) common to every technology.

[1]The Iu interface provides connection between the Radio Network Controllers (RNCs) and certain core nodes in the GPRS network

[2]Service Continuity is the capability to maintain services during the process of changing access network technology [2]

Thus, the OSI model can be split into two levels: (1) network-dependent functions and (2) application-oriented functions. This means that an application can utilise a network without knowing how it operates or what protocols are used for transmission. The transport layer forms the interface between application-oriented services and network-dependent functions.

This means that network integration can be done in any layer apart from layers one and two without dealing with any technology particularities. However, the added overhead due to this kind of integration above the network layer is enormous, as well as the impact on overall performance and user experience. Almost all the functionality is duplicated, including key networking services that are the bottleneck in data transmission (i.e. IP-layer functions).

In the OSI model, the next level of integration is the network layer. Although it is network-dependent, due to the explosive growth of TCP/IP-based applications and services most of the technologies have converged on IP. Most of the current heterogeneous radio networks use different mappings of IP onto the data plane, and hide the signalling (e.g., connection control, handover, networking signalling, etc) from the network layer.

Thus, the IP layer does not include any signalling or network control functions, however, there is nothing to prevent the inclusion of these kinds of functions in any layer. Nevertheless, in forthcoming 4G communication systems, composed of highly heterogeneous integrated networks, the inclusion of additional functionality into the IP layer can have a negative impact due to existing dependencies between the network layer and the radio technologies underneath, which can drastically increase complexities during the integration process.

The same situation arises when discussing application-related services, e.g., Quality of Service mechanisms are specified for every technology (Integrated Services, Differentiated Services, Multiple Packet Label Switching, or best-effort mechanisms) and these are visible to the layers above IP. This heterogeneity in QoS makes IP-layer integration difficult. Thus, mobility support approaches for 4G systems need to separate application- and data-related functions in order to offer a simple solution. In contrast, if these functionalities are mixed, either cross-layer interaction needs to be considered or QoS mechanisms should be added to IP or MIP [15].

The LCE-CL testbed proposes a *loose-coupled* architecture that enables seamless mobility between heterogeneous environments by integrating the RATs using a common IP layer (network layer). The testbed's architecture clearly separates data and control functionality, but is presently limited in terms of offering QoS guarantees to mobile users. It enables service continuity between access and core networks without affecting core network components or adding functional complexities.

3.2 The testbed

The testbed supports connectivity to the most relevant access networks: IEEE 802.3, IEEE 802.11a, and GSM/GPRS networks. The GPRS infrastructure comprises base stations that are linked to the SGSN (Serving GPRS Support Node) which is then connected to a GGSN (Gateway GPRS Support node). In the current Vodafone configuration, both the SGSN and the GGSN are co-located in a single CGSN (Combined GPRS Support Node). A well provisioned virtual private network (VPN) connects the Lab network

to that of the Vodafone's backbone via an IPSec tunnel over the public Internet. A separate "operator-type" RADIUS server is provisioned to authenticate GPRS mobile users/terminals and also assign IP addresses.

For access to the 4G integrated network, mobile nodes (e.g., laptops) connect to the local WLAN network and simultaneously to GPRS via a Phone/Card modem. The mobile node's MIPv6 implementation is based on that developed by the MediaPoli project [66], chosen for its completeness and open source nature. A semi-permanent IPv6 subnet from BTExact's IPv6 Network, connects the testbed to the 6BONE. Using this address space, it is possible to allocate static IPv6 addresses to all the IPv6 enabled mobile nodes. A router in the lab acts an IPv6/IPv4 tunnel end-point to BTExact's IPv6 network. This router is also an IPv6 access router for the lab's fixed-internal IPv6-enabled network and for internal WLANs. Routing has been configured such that all GPRS/WLAN user traffic going to and from mobile clients passes through the internal router, enabling traffic monitoring.

Since the GPRS cellular network currently operates only on IPv4, A SIT (Simple Internet Translation) tunnel forwards all IPv6 packets as IPv4 packets between the mobile node and a machine providing IPv6-enabled access router functionality on behalf of the GPRS network. Ideally, the GGSN in the GPRS network would provide this functionality directly, but using the tunnel causes only minor overhead, and it represents the current state in IPv4-to-IPv6 migration.

The testbed integrates several independent IP networks, including three IEEE 802.11b sub-networks, Vodafone's GSM/GPRS network, and the Lab's local area network. This setup was designed to do experimental analysis of intra-network and inter-network handovers—also known as horizontal and vertical handovers respectively.

The mobile node's home agent provides network connectivity through an IEEE 802.11b access point. Additionally, foreign networks (e.g., IEEE 802.11b sub-networks, GSM/GPRS Vodafone's network, M-DVB link, and Ethernet LAN) allow the mobile node to stay connected by the HA emitting router advertisements and providing secondary radio coverage. This experimental testbed enables the mobile node to perform seamless roaming between heterogeneous technologies, and maintain connectivity with its correspondent nodes.

3.3 Related work

Several testbeds have been proposed (and some of them implemented) that emulate 4G systems. Most of these environments provide limited mobility between heterogeneous networks based on existing mobility management protocols (e.g., MIPv4, MIPv6, and SIP). This section contrasts the LCE-CL testbed with previous projects. Firstly, the most relevant testbeds are described. Then, the functionality needed to evaluate a 4G system is mentioned. Finally, fundamental differences with previous work are highlighted.

The following projects represent a short-list of the most relevant testbeds deployed over the years. The main work done on the correspondent testbeds is summarised below.

- **BARWAN testbed:** The concepts of wireless overlay networks and vertical handover were introduced in 1996, as part of the BARWAN project at Berkeley [49, 90]. The first overlay networks testbed, the BARWAN testbed, included WaveLAN, Infrared, and Ricochet wireless networks. This testbed was based on MIPv4 and was a pioneering work in the area of mobile networking.

- **MosquitoNet testbed:** Other researchers proposed many testbeds and simulations [8, 111], concentrating on the evaluation of MIPv4 during intra-technology handovers. MosquitoNet was one of these testbeds, deployed at Stanford University [55]. Later on, with the growth of IPv6 and the deployment of MIPv6, a new generation of testbeds appeared. MosquitoNet continued working on minimising delays during horizontal handovers and later on started evaluating MIPv6 performance in heterogeneous environments.

- **EURESCOM testbed:** In 2002, EURESCOM [33] funded an European testbed [35, 34] to evaluate the use of Mobile IP in an All-IP core network. At the end of this project [83], they implemented a MIPv4-based testbed that integrated GPRS, LAN, and WLAN as sample technologies, and they evaluated MIPv4 and Cellular IP as mobility management protocols. However, this project focused on recommendations and did not consider practical results related to handover issues in detail.

- **Moby Dick testbed:** MIPv6-based testbeds have been used to study the integration of different radio access technologies into one IP-based core infrastructure. A recent example, *Moby Dick* [22, 59], proposed and implemented a global end-to-end MIPv6-based architecture to offer QoS in heterogeneous environments. The testbed included UMTS-like TD-CDMA wireless access technology, IEEE 802.11b WLANs, and wired connectivity. Further work is being done as part of a new initiative: **Daidalos** project [23].

- **Nomad testbed:** The Nomad project [73] terminated in June 2004 [76], after successfully setting up and measuring a MIPv4-based testbed [54, 37]. They evaluated seamless roaming between heterogeneous networks based on MIPv4.

- **MIND testbed:** MIND [65] is the follow up of the IST project **BRAIN** (Broadband Radio Access for IP based Networks). These two projects implemented an experimental setup, which integrated IEEE 802.11b, UMTS TDD, and GPRS. They evaluated MIPv6 during inter- and intra-technology handovers [64, 82].

These testbeds have produced very interesting results for the improvement of mobility management and wireless networking. However, they present limitations compare to the LCE-CL testbed. The general features of a good testbed to study 4G systems are:

- Flexibility for the adherence of access technologies and modification of core networking components and software.

- A good level of integration to support seamless mobility without incurring overhead in process duplication.

- Inclusion of real networking conditions to increase accuracy in the experimental results.

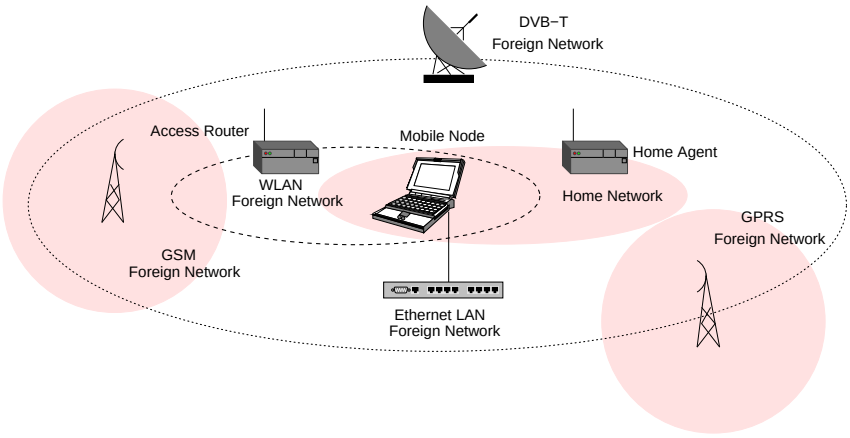


Figure 3.3: LCE-CL testbed and MIPv6 entities.

To emulate a next generation integrated networking environment, the LCE-CL setup consists of a loosely-coupled, Mobile IPv6-based GPRS-WLAN-LAN testbed and has been operational since March 2003. The superiority of the LCE-CL testbed can be resumed in the following characteristics:

- **Access to operator's network:** The lack of access to a real operator's 3G network is the main difference between the listed projects and the LCE-CL setup. There are two important deficiencies in comparison with the LCE-CL testbed. First, other testbeds do not have an appropriate integration between networks (i.e. open coupling), which implies duplicity in all the processes, adding overhead during mobile node's roaming. Second, some testbeds cannot use direct IPv6 over IPv4 tunnelling because MNs are behind a NAT (Network Address Translator) and this situation cannot be avoid without access to core network components. The advantage of accessing the operator's core network allows tight integration of independent networks using core components as common networking devices (loose coupling).

- **Common network layer:** The use of MIPv6 enables the integration of networks using the heterogeneous networks model (core integration at the IP layer). MIPv6 is a better candidate for mobility in 4G systems as it has been optimised according to the demands imposed by integrated networks.

- **Real operation conditions:** The MIPv6/IPv4 traffic generated by the mobile nodes is sent over the Vodafone's network. Ideally, the GGSN will cover the access router functionality directly, with no encapsulation or forwarding. By using a SIT tunnel from source to destination and implementing the appropriate rules, the problems with firewalls placed in the path between the mobile nodes and the CGSN are overcome—otherwise IP traffic will be filtered—allowing the experiments to show the effects of real networking conditions on vertical handovers.

- **Flexibility:** The LCE-CL is flexible in both, hardware and software aspects. It is ready to cope with the addition of recently proposed technologies such as IEEE 802.20 or Wi-Max, and more software can be added when needed.

	Mobile node #1	Mobile node #2	Mobile node #3
Device	Workstation	Laptop	PDA
Networking role	Multi-mode MN	Multi-mode MN	Multi-mode MN
Software	Red Hat 7.1 Linux 2.4.16 MIPL 0.9.1	Red Hat 7.1 Linux 2.4.16 MIPL 0.9.1	Linux 2.4.16 MIPL 0.9.1
Network devices	GSM/GPRS, WLAN, LAN	GSM/GPRS, WLAN, LAN	GSM/GPRS, WLAN
Hardware	P-II 300 MHz RAM 364 MB	P-III 600 MHz RAM 356 MB	StrongARM II RAM 64 MB

Table 3.2: The LCE-CL testbed enables three multi-mode devices.

3.4 Hardware

The LCE-CL testbed contains 6 workstations, two PDAs, and one laptop, which fulfil different network functionality. Figure 3.3 shows a MN-centric view of the network architecture. Mobile nodes can connect to the testbed using 100 Mb/s Ethernet LAN, WLAN, and the live Vodafone GSM/GPRS network. Furthermore, the experiments were performed with three types of mobile devices: a workstation, a laptop, and the PDAs [3] (see Table 2).

As mentioned, the testbed integrates heterogeneous wireless and wired technologies with a common IP-layer (i.e. network layer). However, below this layer each radio access technology has an independent protocol stack and disparate network characteristics.

For Ethernet connectivity, the mobile node is equipped with a IEEE 802.3 network card and supports IPv4 and IPv6 stacks. The Ethernet connection is established through the LCE local network using a cable link. This overlay enables static high-speed networking (10–100 Mb/s) with small RTTs (1 ms).

For WLAN connectivity, there are three IEEE 802.11b sub-networks exclusively supporting IPv6. These sub-networks are inter-connected forming a private LCE-IPv6 local network. The LCE-IPv6 network includes the gateway that connects the testbed to 6BONE [3]. The mobile node is equipped with WLAN cards and the corresponding network profiles to connect to any of these sub-networks. This overlay enables the mobile node to connect using up to 11 Mb/s, with RTTs as low as 10 ms, and medium mobility.

For GSM connectivity, the mobile node has three different GSM phones to connect to the Vodafone's network: a Nokia D211 GSM/GPRS card phone, a Sierra wireless AirCard750, and a Motorola T260. The main purpose for this variety was to evaluate the properties of various hardware providers and products. The mobile node connects to the Vodafone network via a serial port (for the Motorola T260) or via a PCI slot to establish a permanent connection using PPP (Point-to-Point Protocol). This overlay enable voice connection with high mobility.

For GPRS connectivity and to enable low-speed data connection (up to 49.5 kb/s), the mobile node has three different GPRS radio access devices that connect to the Vodafone network using a SIT tunnel (to encapsulate IPv6 traffic into IPv4 packets). The Nokia and Motorola phones can establish a "3+1" connection, which means three channels for

[3]The PDA has only one PCMCIA slot, it is not possible to have the three interfaces working simultaneously.

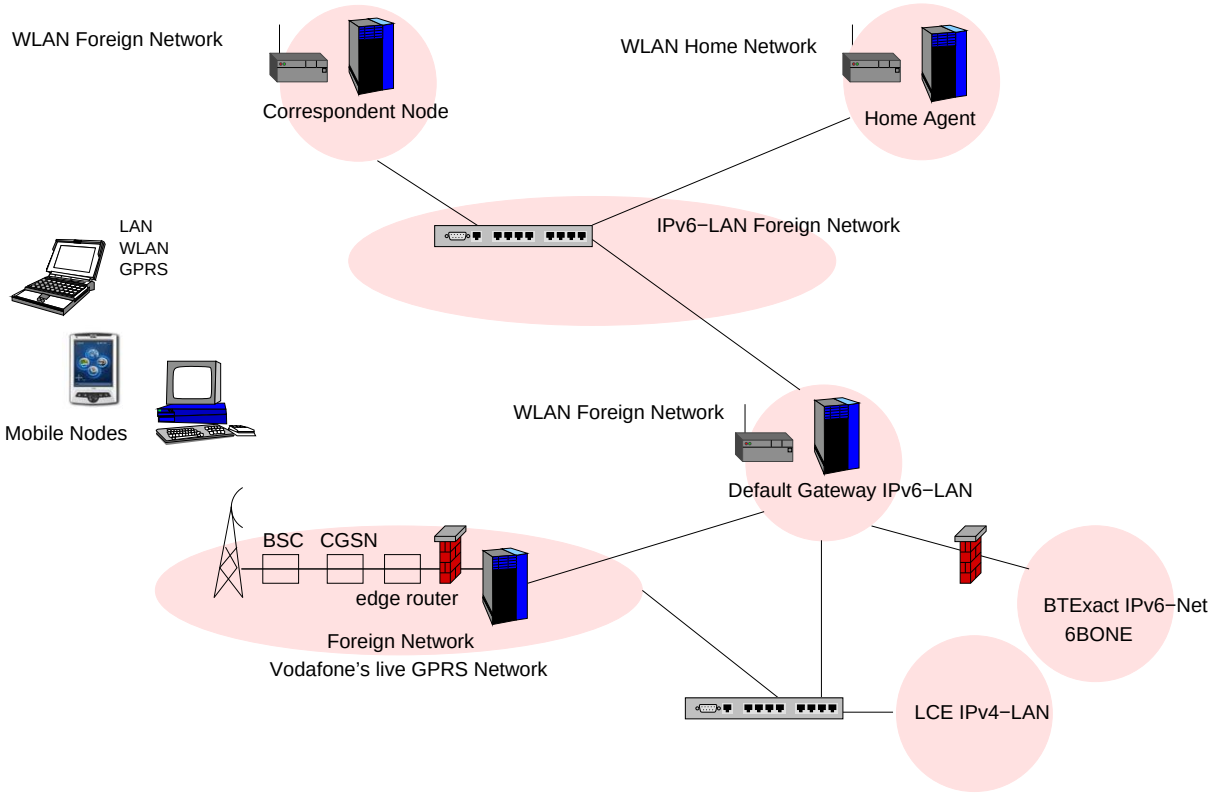


Figure 3.4: Network-centric view of the LCE-CL testbed.

the downlink (39.6 kb/s) and one for the uplink. The AirCard750 connects to the network with a maximum downlink speed of 52.8 kb/s and 13.2 kb/s for the uplink, using a "4+1" connection. This overlay enables low-speed, high mobility data connection with RTTs around 800 ms.

Figure 3.4 shows a network-centric view of the testbed. The LCE-CL enables horizontal and vertical handovers through the integration of seven networks based on different technologies. There are three WLANs, one functioning as home network and the other two as foreign networks (i.e. visited networks), which enable homogeneous handovers. An IPv6-LAN allows the interconnection of the WLANs and mobile nodes to the IPv6 backbone (i.e. 6BONE), and it enables wired-to-wireless handovers (functioning as a foreign network). Finally, the IPv4-based LCE local network connected to the Vodafone's network allows access to the live GPRS RAT. The testbed allows roaming from both, wired to wireless and cellular to wireless.

3.4.1 The Sentient Car

A vehicle-based mobile node was implement for experimental work related to context-aware networking, and as an extension of the LCE-CL testbed for additional testing.

In mobile computing, operating conditions change as clients move. This is particularly true for highly mobile nodes that not only change their location, but also their direction and velocity. The Sentient Car [102], shown in Figure 3.5, gathers experimental results from a highly mobile node and was the outcome of the joint collaboration between the

Figure 3.5: The Sentient Car.

LCE and the Cambridge-MIT Institute (CMI). The Sentient Car is context-aware; it has a priori information of its location using GPS tracking system, movement direction, and its velocity using speed sensors.

This information is then available on a computer terminal, which is also fully equipped to function as a mobile node. The collected data can be used to test sophisticated handover mechanisms, meant for 4G communication environments, that are aware of the immediate context. For example, based on the exact position (e.g., available from a GPS system) and velocity information available at the mobile node, a co-located proxy in the infrastructure can assist host mobility by tracking and accurately predicting when the handover can occur; in other words context-aware, anticipated handovers. Once it is equipped with this information, it can communicate this to the correspondent nodes, which in turn can assist flow adaptation proactively.

The fully-integrated LCE-CL testbed, together with the Sentient Car, allows experimental emulation of a next generation communication system in a wide variety of scenarios.

3.5 Software

This section describes the main software components used in the testbed and mentions relevant modifications done to them. These changes were made to fulfil the conditions needed to perform the proposed experiments (detailed in the next chapter). This section is divided in two parts: one describes the software that support the operation of the testbed, and the other contains the analysis tools.

3.5.1 Operational software

MIPL (Mobile IPv6 for Linux) is used to enable basic mobility in the LCE-CL testbed. This implementation of the MIPv6 specification was developed by Helsinki Technical University [66] in Finland. Lancaster University has the oldest implementation, however, they stopped supporting this distribution in 1998. The last kernel supported by this group was 2.1.9 and the implementation has stopped being compatible since then.

Another reason, MIPL is open source and it is possible to modify the code and implement the mechanisms described in the next chapter. Significant changes were made to the code aiming to optimise the protocol and improve its current performance in 4G systems. Chapter 4 includes the detailed description of the optimisation mechanisms, but the main modifications made to MIPL are listed below.

- The Linux IPv6 Router ADVertisement Daemon (RADVD) is responsible for sending RAs according to configuration parameters (unsolicited RAs) or when responding to a router solicitation [72]. This signalling is required to support IPv6 stateless autoconfiguration. Support was added to allow different *RA frequency* (Section 4.5.1) below the ones recommended in [48]. Additionally, the modified RADVD is able to send alarms from kernel to user space when an incoming RA arrives to any interface—this modification was made to support PROTON (Chapter 5).

- Support was added for *BU simulcasting* (Section 4.5.3) by modifying the MN's source, in particular the *send-binding update* and *send options* routines. Also, the modified MIPL version support *soft handovers*—this optimisation is described in Section 4.10. The MN was modified to accept on-the-air packets from the old interface, even after the registration process had completed.

- In order to force handovers, the IPv6 routing table needs to be modified in correspondence to the expected scenario and conditions. A suite of scripts were developed to control the data in the routing table and dynamically allocate the appropriate routes.

3.5.2 Analysis tools

- **MGEN** was used to generate UDP/IPv6 traffic and perform trace analysis to calculate raw network layer MIPv6 latency [63].

- **TRPR** was used to review the collected traces and produce suitable plots to show the results [97].

- **TCPDUMP** was used to collect traces. The traffic passes through an intermediate router, and using tcpdump it is collected for further analysis. The collected protocols are: UDP/IPv6, TCP/IPv6, and ICMPv6 [93].

- **TCPTRACE** was used to obtain performance graphs. Transport level plots were generated using tcptrace. Since the latest version of tcptrace for analysing TCP traces does not offer any support for MIPv6, this analysis tool was also modified to handle Mobile IPv6 connections [95].

3.6 Remarks

Experimental environments represent an essential tool in the deployment of future mobile networks. In this chapter, a loosely-coupled testbed was described. The main points that distinguish the LCE-CL testbed from other setups are that (1) it fully evaluates MIPv6 in heterogeneous environments, while other projects focus on the evaluation of intra-technology handovers using MIPv6 or inter-technology handovers using MIPv4 due to constraints in the architecture, and (2) Vodafone's GSM/GPRS network. The LCE-CL testbed enables a GSM/GPRS overlay using the actual provider's network. Other projects emulate a GPRS link or install an isolated GPRS network for experimental purposes, both cases lack real-world conditions.

The testbed is used as the platform for analysing, solving, developing, and testing the following technical issues:

- Seamless horizontal handovers using MIPv6
- MIPv6 performance during vertical handovers
- A client-based solution to improve performance in horizontal handovers for WLANs [74]
- Methods to minimise vertical handover latency
- Policy-based solution to manage multiple interfaces and support mobility
- QoS-based handover algorithms for overlay networks
- Context-aware algorithms for overlay networks

To conclude, it is fair to comment that one of the main advantages in the experimental setup presented in Chapter 3 is also its major limitation. On one hand, it was fundamental for this work to have access to core components in the Vodafone GSM/GPRS network, and to perform experiments under real network conditions. On the other hand, the required agreement with Vodafone posed some limitations on the publication of results, although this did not affect the outcome of this dissertation.

Chapter 4

Evaluation and Networking Improvements for 4G Systems

This chapter summarises the practical experiments performed to evaluate the impact of inter-networking (i.e. vertical handovers) on the network and transport layers. Based on experiments and observations, a number of inter-technology handover optimisation techniques are proposed and evaluated in this chapter—RA frequency, BU simulcasting, RA caching, and soft handovers.

This work targets networking issues in future 4G communication systems, in particular, those related to the impact of vertical handovers and terminal mobility on the performance of the TCP/IP stack.

The main objectives are:

1. **Measure** the latency when the terminal is roaming between heterogeneous networks,

2. **Characterise** the vertical handover latency, identifying its main components,

3. **Identify** the main performance problems during vertical handovers and their impact on the TCP/IP stack,

4. **Explore** different optimisation methods that decrease the overall latency, and evaluate each of them using the LCE-CL testbed,

5. **Demonstrate** the efficiency of different optimisations, and the need for a high-level mobility support middleware to enable informed decisions and drive the handover process.

In this chapter a detailed evaluation of Mobile IPv6 (using the testbed introduce in Chapter 3) is presented. The most important improvements to this protocol are mentioned in Section 4.1. The testing environment and conditions are mentioned in Section 4.2. The vertical handover latency study is summarised in Section 4.3 and latency partition detailed in Section 4.4. Different handover optimisations are explored, and results are presented in Section 4.5. Related work is presented in Section 4.6, while Section 4.7 concludes the chapter.

4.1 Optimisations to Mobile IPv6

The handover latency using basic MIPv6 is proportional to the round-trip time necessary for a binding update message to reach either the MN's home agent or a correspondent node. This is further analysed in Section 4.4.

Network-layer latency on wireless links can be very high, especially for interactive applications that have real-time requirements. Therefore, the research community is working on mechanisms that decrease this latency as much as possible, at least to levels that support real-time applications in a seamless manner.

Two of the most significant proposals are Fast Handovers for Mobile IPv6 (FMIPv6) [53] and Hierarchical Mobile IPv6 (HMIPv6) [88]. FMIPv6 [53] aims to decrease the total latency to almost only the layer-two handover time. This approach has been shown to perform well in intra-technology (i.e. horizontal) handovers [22, 7].

The HMIPv6 approach is designed to reduce the degree of signalling required and to improve handover speed for mobile connections by managing local mobility in a more efficient way. Previous work [19, 20] has analysed which of these approaches (i.e. FMIPv6 and HMIPv6) performs better, the conclusion being that a combined approach would be optimal. However, given the implementation complexity that this would require, the FMIPv6 optimisation by itself is sufficient (this has been experimentally evaluated in [7]).

Future 4G systems, in which heterogeneity will be the rule instead of the exception, present some challenging characteristics when performing vertical (also known as inter-technology) handovers:

- Bandwidth, delay, and packet losses can be radically different among the different candidate access network technologies (e.g., IEEE 802.3, IEEE 802.11, GPRS, UMTS, Bluetooth, etc.)

- There are certain technologies that are usually globally available (e.g., GPRS, UMTS), whereas there are others that are only present in certain locations (e.g., WLAN, Bluetooth). This fact dictates an overlay-based model, where "moving up" (handing off from a locally available access technology to a globally available network) is not equivalent to "moving down" (roaming from a globally available network to a locally available technology). This needs to be taken into account, while in horizontal handovers the situation does not arise.

Therefore, the vertical handover process is affected by many different parameters, and the solutions needed to improve it must handle this complexity. Moreover, optimisation methods that have been tested for homogeneous environments do not necessary work well in vertical handovers, and it should not be assumed that everything in the horizontal scenario brings benefits to the heterogeneous case.

This chapter aims to clarify this situation; a thorough evaluation of MIPv6 is performed and the main effects of mobility on the TCP/IP stack are mentioned. During this process, the possible handover scenarios and optimisations are studied with the intention of defining the best combination and its scope. Suggestions for particular optimisation methods, which can be applied under a certain set of conditions are formulated. Finally, the need for middleware to support mobility and handle these intrinsic complexities in 4G networks is motivated.

4.2 Experimental environment

The testbed was operated under the following conditions:

1. The movement detection was performed based on Neighbour Discovery (i.e. router advertisements). In this case, the mobile node waits to receive a router advertisement after it arrives at the visited network.

2. In the case of the vertical handover, when the new interface has a lower priority, compared to the previous one, the mobile node waits until the old access router is unreachable and then generates a router solicitation message.

3. Unless stated otherwise, all access routers are set to multicast RAs, according to the recommended parameters and values specified in [92] (the effects of these values are analysed in Section 4.5.1)

4. For all the cases considered in these tests, the multi-mode mobile device had all of its interfaces (e.g., LAN, WLAN and GPRS) powered on and listening to their specific networks.

5. The soft handover and RA caching optimisations follow the "make-before-break" philosophy, meaning that at least one handover-related process starts before breaking the connection with the previous access router.

6. The RA frequency and BU simulcasting methods follow the "break-before-make" philosophy. The MN waits until the current access router is unreachable, and then it begins the handover process.

7. The results presented in Section 4.5.4 (i.e. soft handovers) consider the MN listening on both interfaces simultaneously, while performing the vertical handover. In contrast, a *hard handover* occurs when the MN listens exclusively on one interface—expecting packet losses.

8. All hosts run Red Hat 7.1, with Linux 2.4.16 and the MNs run MIPL 0.9.1. For these experiments the workstation and the laptop are used for the experiments, equipped with an Orinoco wireless silver card and a Sierra Wireless Aircard 750 for IEEE 802.11b and GPRS connections, respectively.

For the tests conducted, handovers were forced by filtering incoming RAs at the network layer using IP6tables-based filters [46]. For testing handovers, data transfers were initiated by the multi-mode device over the Visited Network access router. During this data transfer (e.g., ICMP packets or HTTP), vertical handovers to different visited networks are conducted (i.e. a different RAT). In the testbed, all traffic passes through an intermediate router, and traffic traces are taken using tcpdump [93], to/from the correspondent node, allowing us to collect traces at this point for further analysis.

The following systematic errors need to be considered in every experimental measurements:

- Clock synchronisation: The network machines are synchronised using the Network Time Protocol (NTP), which has a clock accuracy of 10 ms.
- Clock accuracy: The packet traces were collected using tcpdump [93]. This tool sets a time-stamp for each packet that arrives. It has an accuracy of 1 ms.

Therefore, all the experimental values presented in the current chapter and Chapter 6 should include a maximum error value of 11 ms.

4.3 Experiments to evaluate MIPv6

Firstly, experiments to evaluate MIPv6 during vertical handovers are performed. These tests aim to measure the impact of mobility on the TCP/IP stack. Therefore effects on layer 3 and layer 4 are analysed for the following scenarios: GPRS-to-WLAN, WLAN-to-GPRS, GPRS-to-LAN, LAN-to-GPRS, WLAN-to-LAN, and LAN-to-WLAN (the experiments were carried out using the LCE-CL testbed). The most relevant results from the study are explained in the next section.

- Measurements of MIPv6 performance during vertical handovers at the network layer for every possible scenario, using ICMP as a test case protocol. Due to the simplicity of this protocol, it is the most appropriate way to measure network layer latency. ICMP does not add too much overhead to the networking process.
- The most popular test scenario in heterogeneous environments is the handover between WLAN and GPRS networks, for which results are included showing the UDP over MIPv6 handover latency.
- MIPv6 transport layer measurements to determine the effects of mobility on TCP using HTTP as a test case application, due to the fact that almost 85% of the Internet traffic is generated by this type of connection [62].
- Packet overhead added by different protocols, considering routing between the MN and the CN and between mobile nodes, in the different test cases.

4.3.1 Mobile IPv6 network layer performance (IP)

To calculate the latency at the network layer based on packet loss, ICMPv6 traffic is generated between the correspondent node and the mobile node. The correspondent node sends small ICMPv6 packets (104 bytes) every 200 ms. Thus, the handover latency is the product of the number of packets lost multiplied by the time interval between the packets (see Figure 4.1).

Results shown in Figure 4.1 suggest that Mobile IPv6 protocol was designed for mobility management without caring about the underlying technologies. When dealing with heterogeneous handovers, MIPv6 does not perform as expected and in most scenarios

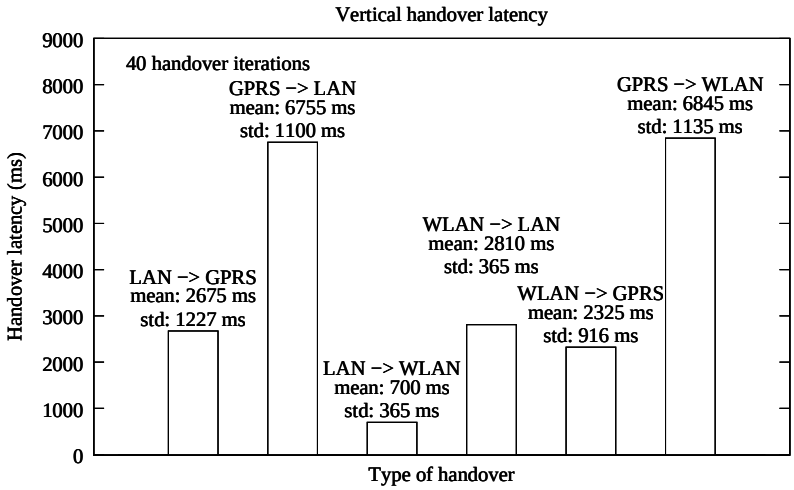


Figure 4.1: MIPv6 network layer vertical handover latency.

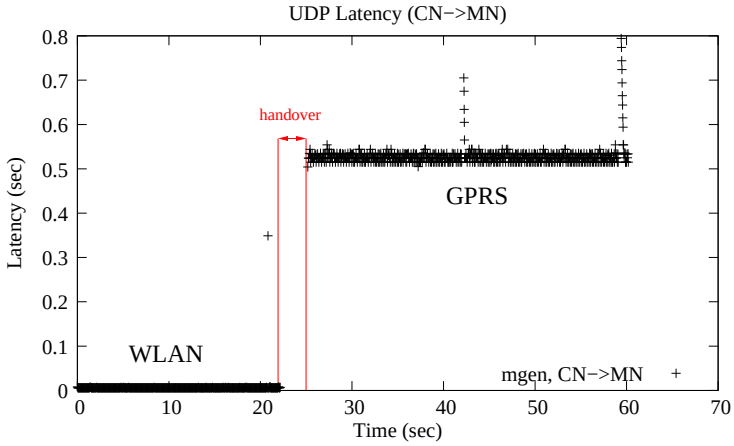


Figure 4.2: UDP over MIPv6 vertical handover latency.

latency exceeds acceptable limits (not even close to support for real-time applications). The values presented can have a precision error of 200 ms due to the experimental setup, thus handover latency from IEEE 802.11b to GPRS is between 2200 ms and 2400 ms. In fact, the mean handover latency for this scenario is 2325 ms. The mean and standard deviation values shown were calculated using the data collected over 40 iterations for each handover scenario.

4.3.2 Mobile IPv6 transport layer performance (UDP)

In addition to the MIPv6 performance at the network layer, it is also interesting to investigate how current transport protocols are affected by mobility in heterogeneous environments. This section discusses related results collected during some experiments that were performed using the LCE-CL testbed.

Several tests were done using MGEN [63] in order to analyse UDP effects on MIPv6. During the experiments, the receiver (i.e. mobile node) collects transmitted packets and creates logs, which are then analysed to extract statistical data (using TRPR [97]) such

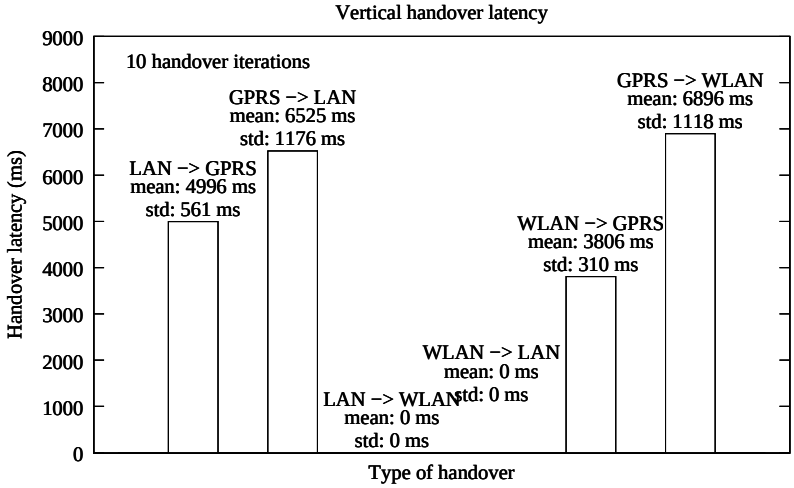


Figure 4.3: MIPv6 transport layer (TCP) vertical handover latency.

as perceived latency at the mobile node.

Figure 4.2 shows the latency at the mobile node during an average iteration. For this trial, the correspondent node starts sending UDP packets (packet size 80 bytes, every 50 ms). After 22 s a handover occurred and 57 packets are lost. This implies a handover latency of 2850 ms, which is a value very close to the network layer latency for the same technologies, around 2325 ms as shown in Figure 4.1.

4.3.3 Mobile IPv6 impact on the transport layer (TCP)

The impact of vertical handovers on the transport layer is evaluated by collecting traces whilst downloading a file from the CN to the mobile terminal. Results are included for every possible scenario using the LCE-CL testbed. The values in Figure 4.3 show the total delay when the MN moves between two access points that belong to different technologies. Using the complete picture offered by Figures 4.1 and 4.3, Some early patterns in the behaviour of MIPv6 when dealing with heterogeneous technologies can be observed, as well as the mobility impact on the TCP/IP protocol stack.

When the MN moves "up" in the model (from a faster network to a slower one with higher RTTs) the latency is smaller, e.g., to handoff from WLAN to GPRS takes 5.5 s, whereas when the terminal moves to a lower layer (i.e. to a faster access technology), the disruptions last for longer (the handover from GPRS to WLAN takes about 7.4 s). This relation is also true for the latency values at the network layer shown in Figure 4.1.

The values are larger for the TCP latency in every scenario, compared to the network-layer latency. This is due to the residual TCP back-off time or the interval for which the TCP flow remains (exponentially) backed-off even after the IP-level handover. This delay is considered part of what I termed adaptation component, which affects the handover latency at layer 4 (Section 4.4 explains this concept). The mean and standard deviation values shown were calculated using the data collected over 40 iterations for each handover scenario.

4.3.4 Packet overhead

This section introduces an analytical study of the protocol overhead introduced by Mobile IPv6, by comparing it with the overhead present in IPv4 and plain IPv6.

Mobile IPv6 adds an extra IPv6 header (40 bytes) to every packet sent/received by the MN in the MN-CN portion of the path followed by the data packets. If the Route Optimisation support is enabled in an ongoing communication between a MN and a CN, a 24-byte overhead (due to the addition of a home address destination option in the packets sent by the MN or a Type II Routing Header in the packets sent by the CN) is added instead of the IPv6 header. For the case of two MNs that are communicating, a 48-byte overhead is present, because both the home address destination option and the routing header are present in every packet of the communication.

Figure 4.4 shows graphically the packet overhead for different kinds of IP payloads:

- **TCP 40 bytes.** Mostly TCP packets which carry TCP acknowledgements, but not payload. This is the minimum TCP packet size.
- **TCP 552 bytes.** This packet-size (or 576 bytes) is used by those TCP implementations that do not use path MTU discovery.
- **TCP 1500 bytes.** This is the maximum standard Ethernet payload size.

  Because a large proportion of the TCP traffic is generated by bulk transfer applications, such as HTTP and FTP, the majority of the packets seen in the Internet are one of the previous sizes [62].
- **UDP VoIP GSM**. UDP-RTP[1] packets carrying VoIP payload using the GSM codec (33 bytes per packet).
- **UDP VoIP G723.1**. UDP-RTP packets carrying VoIP payload using the G723.1 codec (20 bytes per packet).
- **UDP VoIP G711**. UDP-RTP packets carrying VoIP payload using the G711 codec (240 bytes per packet).
- **UDP VoIP LPC10**. UDP-RTP packets carrying VoIP payload using the LPC10 codec (7 bytes per packet).

UDP-RTP VoIP packets are analysed because VoIP has been one of the emerging services in the last several years and also because the packet overhead is a critical parameter in this application (even when using IPv4).

Figure 4.4 shows that even for 552-byte TCP packets, the overhead is not negligible when Mobile IPv6 is used (about 14–16%). The use of Mobile IPv6 has a great impact on the packet overhead in VoIP packets—sometimes even tripling it compared with the IPv4 case. Therefore, special care should be taken in the design of the network (i.e. resources, queue management, QoS mechanisms, etc.), especially if applications request QoS guarantees. Thus, the added overhead to support mobility in 4G networks needs to be considered; current deployments (designed for IPv4 traffic) could be insufficient for carrying the new IPv6-mobility-enabled traffic.

[1]The RTP header is 12 bytes

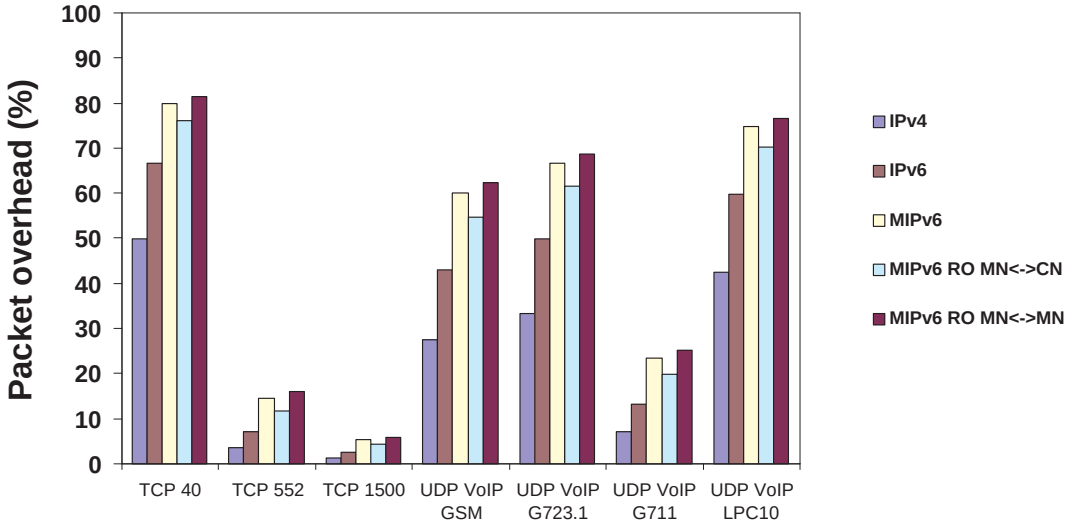


Figure 4.4: Packet overhead for different types of payloads.

A handover process between two disparate wireless technologies is usually divided into three main stages: (1) network discovery, (2) network selection, and (3) handover execution. However, for the case of the transport-layer handover using TCP (i.e. connection oriented), a fourth phase to the handover process is added (adaptation). Derived from the results collected from experimental work done in the LCE-CL testbed, it is concluded that after the handover execution there is a time period during which the mobile host needs to adapt to the new link characteristics and this delay affects the handover latency depending on the old and the new networks. Figure 4.5 shows the relative values for the total handover latency and the adaptation component measured from a representative iteration of a handover between a WLAN and the Vodafone cellular network.

4.4 Vertical handover latency characterisation

The stages are therefore:

- *Detection Period (t_d).* It is the time taken by the mobile terminal to discover the available network(s), using link-layer signalling or the network layer detection mechanism. The Mobile IPv6 generic movement detection mechanism uses the facilities of IPv6 Neighbour Discovery. When the MN is not sending traffic, it listens to IPv6 router advertisements to determine the network prefix.

- *Configuration Interval (t_c).* This is the interval from the moment a mobile device receives a router advertisement, to the time it takes to update the routing table and assign its new care-of address based on the received network prefix, including the DAD delay. This interval depends on the terminal characteristics (e.g., memory, processing power, etc.)

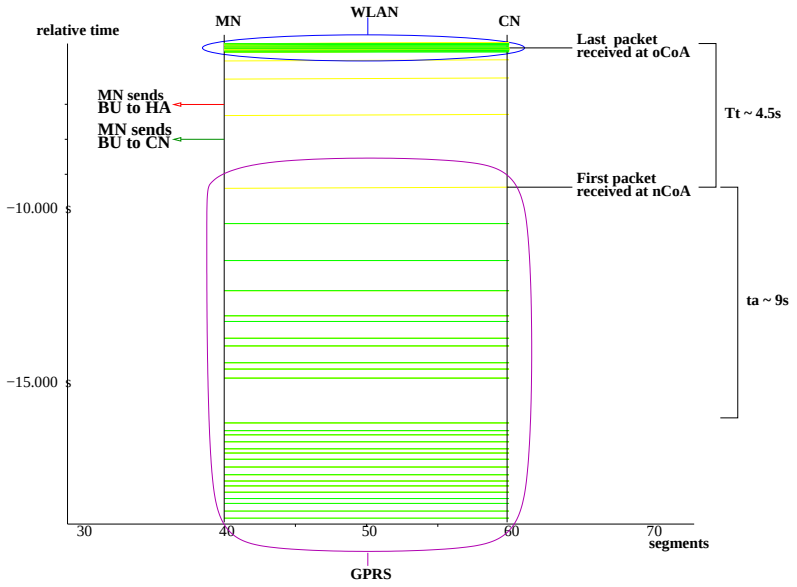


Figure 4.5: Adaptation component for the test scenario WLAN-to-GPRS.

The DAD delay was not significant in the described scenario due to the lack of concurrent mobile hosts roaming simultaneously. In an optimistic case, the DAD delay is very small compared to the other latency components. However, when dealing with more mobile nodes the probability of address collision increases and the DAD delay should be considered. For experimental results on this topic, an environment where the number of mobile nodes moving can be controlled needs to be available. Further discussion on this issue is included in [5, 70].

- *Registration Time (t_r).* This is the delay between the delivery of the binding update to the home agent and correspondent nodes, and the reception of the first packet at the new interface—with the new care–of address as the destination address. This time component increases if the mobile terminal is configured to wait for the binding acknowledgement sent by the correspondent node, as mentioned in Section 11.7.2 of [48].

- *Adaptation Time (t_a).* When dealing with vertical handovers at the transport level, t_a needs to be considered in the total handover latency. This delay only happens when the mobile host adapts the connection to the new technology at the transport layer, adjusting the TCP state machine parameters (e.g., congestion window size, timeout timers, etc.), due to the heterogeneous nature of the technologies—this is caused by the normal TCP settling period, but augmented by the differences in the link characteristics. Thus for the case of TCP transmissions, the transport-layer latency (T_t) is equivalent to the network-layer latency (T_n) plus the adaptation component, as shown in Figure 4.6.

The definition of handover latency in Mobile IPv6 is limited to the period of time when the mobile node is unable to receive IPv6 packets both due to the link (i.e. layer 2) switching delay and IP protocol operations (i.e. network-layer handover), as defined in [48]. However, this chapter analyses the latencies in both UDP/MIPv6 and TCP/MIPv6

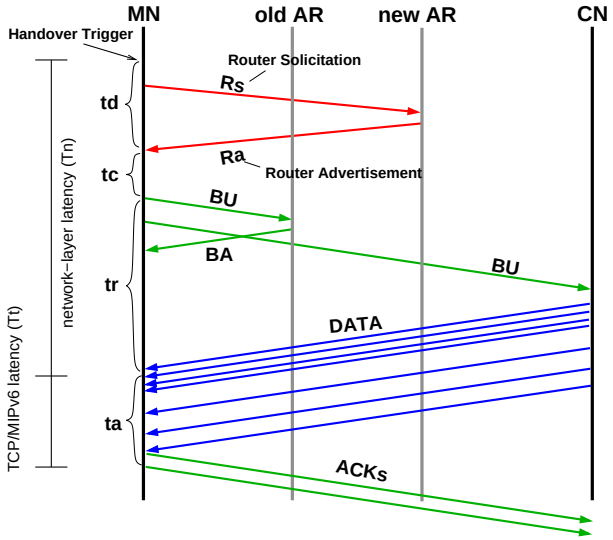


Figure 4.6: Network and transport layers' latency partition.

for transport-layer handovers. For the UDP case, the latency is equivalent to the network-layer latency. However, for TCP transmissions the term of handover latency is extended to consider the t_a component.

In this study handover latency is divided into: network-layer handover latency and TCP handover latency. The former matches the definition included in [48], and the latter is defined as the period of time when the mobile node is unable to receive IPv6 packets with a stable pattern (meaning that the throughput is close to the link's normal behaviour) due to the link switching delay, the IP protocol operations, and the adjustments performed at the transport layer due to huge disparities in link characteristics.

It has been observed that 85% of the traffic in the Internet is generated by TCP connections [62]. Therefore, proposals to minimise TCP handover latency during vertical handovers are essential for future seamless roaming—a possible approach is to reduce t_a in order to improve the overall latency.

4.4.1 Analytical representation of latency partition

As mentioned above, the impact of vertical handovers on the network layer is characterised using ICMPv6 protocol, and the effects at the transport layer when using UDP and TCP as transport protocols. The cases of the GPRS-WLAN-LAN testbed are included based on current standard wireless technologies to outline an experimental study that determines the impact of each individual component in the overall handover latency.

The network-layer latency is given by:

$$T_n = t_d + t_c + t_r \tag{4.1}$$

where

t_d is a random variable with probability $p(t_d)$:

$$p(t_d) = \frac{1 - \int_0^{t_d} p_{t_{RA}}(t)}{\overline{t_{RA}}} \tag{4.2}$$

$$p_{t_{RA}}(t) = \begin{cases} \frac{1}{RAint_{MAX} - RAint_{MIN}} & \text{, if } RAint_{MIN} \\ & \leq t \leq RAint_{MAX} \\ 0 & \text{, otherwise} \end{cases}$$

$$t_c = t_{DAD_{LL}} + t_{DR} + t_{CoA} + t_{DAD_{NL}} \tag{4.3}$$

$$t_r = t_{RR} + t_{BU} \tag{4.4}$$

Tn = Total latency perceived in the network layer,
t_{RA} = Time period between consecutive RAs,
$RAint_{MAX}$ = Maximum RA interval (i.e. time between two consecutive router advertisements),
$RAint_{MIN}$ = Minimum RA interval (i.e. time between two consecutive router advertisements),
$t_{DAD_{LL}}$ = Time taken for Duplicate Address Detection for link local address,
t_{DR} = Time taken for default router configuration,
t_{CoA} = Time taken for configuring the new CoA,
$t_{DAD_{NL}}$ = Time taken for Duplicate Address Detection for CoA,
t_{RR} = Time taken for entire Return Routability procedure (= 1.5 RTTs in the best case scenario—i.e. no packet losses)
t_{BU} = Time taken for update of the binding in the HA (= 1 RTT from the MN to the HA using the NAR) and this can be done simultaneously while updating the CNs (an optimisation to minimise this delay is explained in Section 4.5.3)

The overall latency is found by summing the delays to discover the new network (t_d), to build the binding update message using the prefix from the new access router (t_c), and to register the recently formed CoA with the home agent and correspondent nodes (t_r), as shown in Equation 4.1. The network discovery delay depends on the movement detection mechanism, but if the generic L3 Neighbour Discovery based mechanism is used by the MN, this time depends mostly on the router advertisement frequency and also on the policy that the MN uses to consider a router unreachable. Generally, the MN may use the Advertisement Interval (if included in the router advertisements received by the MN) field as an indication of the frequency with which the current default router is sending these messages. Therefore, if during this time interval (which indicates the maximum time between two consecutive RAs) the MN does not receive any new RA from the current default router (i.e. at least one RA has been lost), the MN can use the event of losing a certain number of RAs as a possible L3 handover indication (in MIPL, for example, a figure of 2 lost RAs is used, triggering the MN to send router solicitation messages on all the available interfaces to discover new reachable routers).

The configuration time depends on the terminal characteristics. However, some methods have been proposed to reduce this delay such as avoiding the DAD process to minimise disruptions when roaming between access routers [5]. The last component (t_r) is dictated by the RTT of the network used for the registration process, as shown in Equation 4.

WLAN ⇒ GPRS	Mean	Std. Dev.	Min.	Max.
Detection time (t_d)	808	320	200	1148
Configuration time (t_c)	1	0	1	1
Registration time (t_r)	2997	416	2339	3649
Total handover latency (t_n)	3806	327	3323	4438
GPRS ⇒ WLAN	Mean	Std. Dev.	Min.	Max.
Detection time (t_d)	2241	968	739	3803
Configuration time (t_c)	1	0	0	1
Registration time (t_r)	4654	1698	2585	7639
Total handover latency (t_n)	6897	1178	5322	8833
LAN ⇒ GPRS	Mean	Std. Dev.	Min.	Max.
Detection time (t_d)	1168	460	347	2070
Configuration time (t_c)	1	0	1	1
Registration time (t_r)	3307	585	2299	4759
Total handover latency (t_n)	4476	520	2806	5107
GPRS ⇒ LAN	Mean	Std. Dev.	Min.	Max.
Detection time (t_d)	2058	1030	1	3257
Configuration time (t_c)	1	0	1	1
Registration time (t_r)	4466	1449	2357	7183
Total handover latency (t_n)	6525	1229	4011	8197

Table 4.1: Latency partition for vertical handovers using MIPL(milliseconds).

4.4.2 Experimental latency partition

The impact that hard handovers (i.e. only one active interface at the same time) have on the network and transport layers have been studied. To demonstrate latency partitioning, the cases of GPRS-WLAN-GPRS and GPRS-LAN-GPRS are included because these are the most common scenarios. Table 4.1 shows the average values over 40 iterations of the latency components for different handover scenarios.

The raw latency values included in the tables are too high to even consider using MIPv6 (or its implementation, MIPL) with real-time applications for which the tolerated delays are less than 100 ms. For example, the handover latency from a hotspot to the cellular system takes around 6.8 seconds to complete, and in the scenario where the terminal moves from a cellular system to an IEEE 802.11b access point there is has an overall delay of 3.8 seconds.

As t_d is a random variable, that depends on the RA frequency, the values for the standard deviation in every case are very large—around 40% and 50% of the mean value. Ideally, when the mobile host moves from a high-RTT and low-bandwidth network (e.g. GPRS) to a low-RTT and high-bandwidth access technology (e.g. WLAN), the registration time should be smaller than the opposite case—moving from a lower layer to an upper layer—as RA frequencies are higher in the lower networks and also because the time required for the signalling to be completed is lower. Table 4.1 shows that this is not necessarily true for all the cases.

The registration time is basically related to the RTT of the network; since WLAN offers links that have low RTTs, then t_r for GPRS to WLAN handover should be smaller

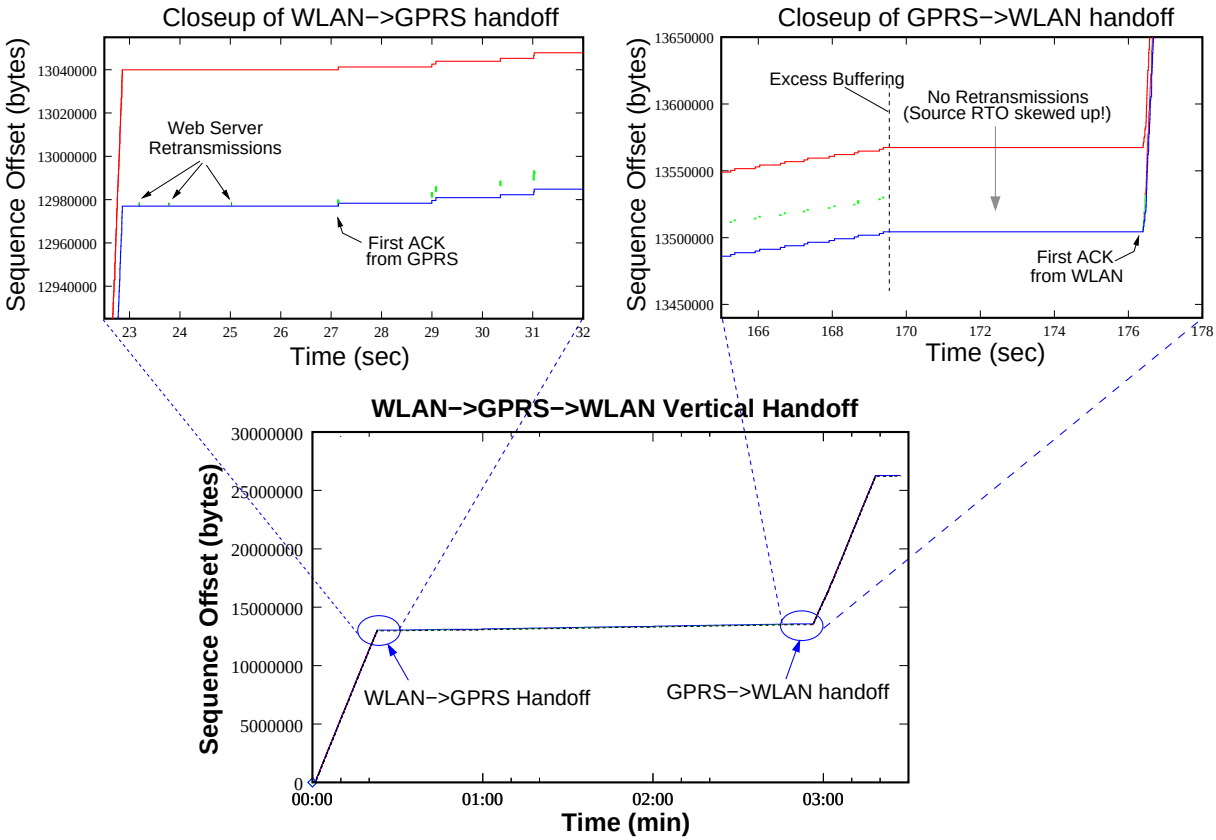


Figure 4.7: Close-up of a handover showing the effects of excess buffering.

than for its counterpart (i.e. WLAN to GPRS). The measured values show the opposite. This is because the Mobile IPv6 implementation used in these tests (MIPL) tries to send the BU to the CNs piggybacked in a data packet. Therefore, the BU is sent in the first data packet after the movement detection (or after a certain maximum amount of time, to avoid delaying the binding update so much). Because of the high buffering in the GPRS GGSN, the MN perceives an inflated RTT, and hence, an increased Retransmission Timeout (RTO). Consequently, the registration process can complete only after the source eventually retransmits with a high value of the RTO. This leads to a very high latency when moving from the cellular network to a hotspot (see Figure 4.7 [2]), affecting the t_r delay.

There is also an additional feature that affects the overall handover latency, which is related to the Movement Detection mechanism specified in Section 11.5 of the MIPv6 RFC document [48], and directly affects the t_d delay. The specification does not include policies to determine when the mobile node needs to perform Router Discovery, due to the loss of connectivity through the current network. However, it does enumerate certain indications that should be considered (e.g. Advertisement Interval field) to detect unreachability of the current access router.

For the case of MIPL [66], which is the MIPv6 implementation used in the LCE-CL testbed, there are different policies according to certain conditions when deciding the

[2]Source: R. Chacravorty

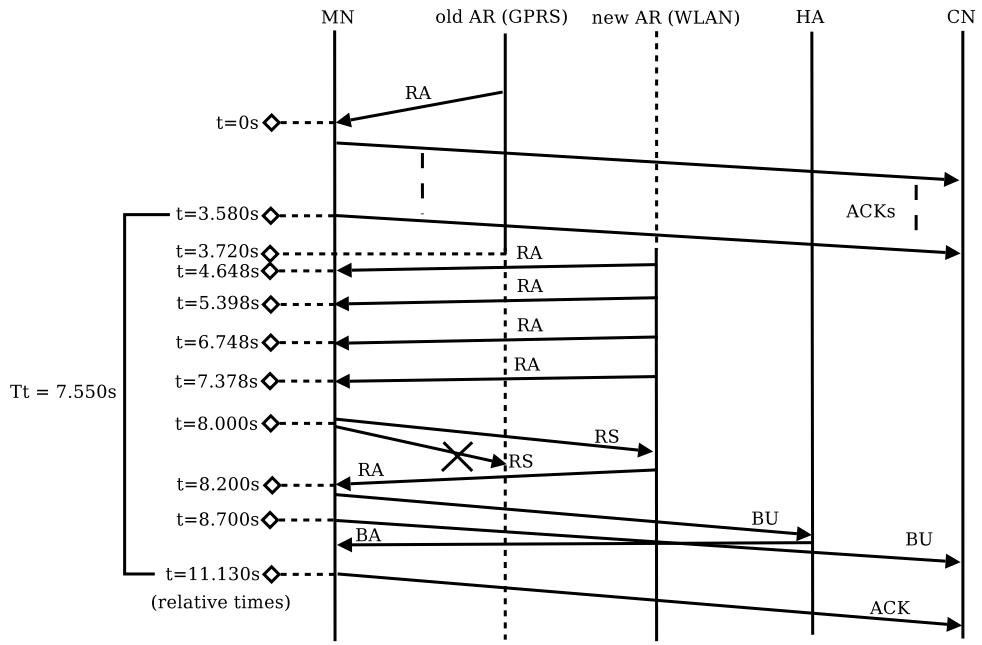


Figure 4.8: Lazy cell switching.

handover. The Movement Detection mechanism included in MIPL covers the following situations:

- *No movement detection.* If the detected router advertisement comes from the current router, then the MN does not perform any handover action. Interpreting this, the MN is still within the coverage of the current access router and there is no other network available.

- *Horizontal Handover.* This occurs when the incoming RA is received through the same interface—and it is not sent by the current access router, which implies that both the new and old access routers are within the same layer (i.e. wireless technology). Under these conditions, MIPL executes the handover using the *Eager Cell Switching* (ECS) algorithm. The RA received is processed immediately, followed by the configuration and registration procedures.

- *Vertical Handover.* The incoming RA comes from a different access router, and it is received on a different interface from the one being used. This means that the new and old routers of attachment belong to different access technologies. In this case MIPL checks if the new network interface (NIC) is preferred over the old interface. If the new one is preferred, then the RA is processed, and if not, MIPL triggers a the *Lazy Cell Switching* (LCS) algorithm for vertical handovers.

The LCS algorithm implemented in MIPL needs to be described in order to understand its effect on the handover latency. If the interface that receives the RA from the new access router (i.e. recently available network) is different from the one being used, then the terminal waits until the current network is unreachable. As shown in Figure 4.8, the MN detects that the old access router is not reachable when it does not receive RAs for

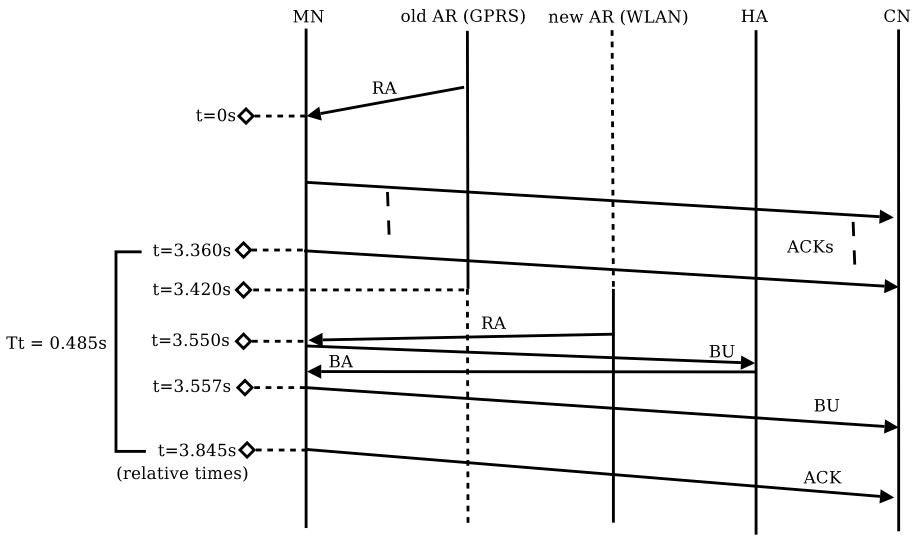


Figure 4.9: Eager cell switching.

a period of time equal to twice the RA interval ($2 * RA_{interval}$). The MN sends a router solicitation after 8 s because the RA interval is configured to 4000 ms for the example shown in Figure 4.8.[3]

For this implementation of the LCS in MIPL, the current network has higher priority than any other technology available, without considering the link characteristics. Thus, if no other policy is taken before the handover execution, a slower network can have a higher priority than a faster technology. In this case, the mobile node applies the LCS algorithm instead of ECS when moving from the cellular system to an available hotspot, resulting in a bigger value of t_r as shown in Table 4.1 on page 58.

The importance of applying the appropriate handover execution method can be observed (in MIPL either LCS or ECS) in figures 4.8 and 4.9. When ECS is applied, whilst performing a downward (to handoff from a global coverage to a local coverage access technology) handover, the total latency was approximately 500 ms as shown in Figure 4.9, whereas for the case where LCS was used (i.e. when the slower interface had a higher preference value), the handover process took approximately 7.5 s in total (see Figure 4.8).

This situation motivates the need for a policy-based system to take well-informed handover-related decisions, and deal with the complexities posed by heterogeneous networks. The handover performance can be improved just by having an *a priori* policy to set the interfaces' preferences according to the link characteristics. For example, the GPRS preference is set to a lower value than the IEEE 802.11a interface due to its higher RTT values. Thus, when a downward handover occurs, the ECS algorithm is applied and not the LCS execution method, improving the vertical handover performance in this scenario.

[3]These values are the ones recommended in the Mobile IPv6 RFC.

RA Interval MinRA–MaxRA	WLAN− >GPRS t_d (GPRS overhead, %)	GPRS− >WLAN t_d (WLAN overhead, %)
300 ms–400 ms	551 ms (4.7)	234 ms (0.0210)
200 ms–300 ms	360 ms (9.5)	242 ms (0.0317)
100 ms–200 ms	324 ms (19.0)	174 ms (0.0633)
40 ms–70 ms	217 ms (47.5)	86 ms (0.1583)

Table 4.2: RA frequency variation effects on WLAN and GPRS networks.

4.5 Impacting MIPv6 latency

As shown in the previous section, the total MIPv6 handover latency is the sum of the IP layer latency (T_n) and the TCP adaptation time. This section describes mechanisms to improve the MIPv6 latency and minimise disruptions in the connectivity while roaming between heterogeneous networks. The following methods are discussed: RA frequency, RA caching, BU simulcasting, and soft handover to improve networking performance.

4.5.1 RA frequency

Higher RA frequency improves performance due to the reduction in t_d, which results in smaller latencies during vertical handovers. There are two methods related to the Neighbour Discovery protocol that can reduce the network discovery time: (1) incrementing the RA frequency and (2) generating an on-demand router solicitation.

Related to incrementing the RA frequency, the current Mobile IPv6 specification [48] considers this topic and proposes shorter intervals—30 ms (MinRtrAdvInterval) to 70 ms (MinRtrAdvInterval). However, decreasing the RA interval is not without cost, as this has a direct impact on the link performance, caused by the added overhead. Thus, there is an obvious trade-off involved, especially over reduced-capacity links such as GPRS networks.

In order to evaluate the RA frequency's impact on vertical handovers, a modified version of the Linux IPv6 RA daemon (radvd [79]) is used to perform experiments using different RA interval values, including the ones specified in [48]. Table 4.2 shows the effect of varying RA interval on mean t_d values obtained from over 20 runs. Based on these results, it can be observed that although increasing the RA frequency reduces the total latency, it is not the best option for low-bandwidth networks such as GPRS.

For example, in the case of WLAN, increasing the RA frequency from 300–400 ms (min–max) to 40–70 ms represents a marginal increase in the overhead—considering a bandwidth of 8 Mb/s in the link—and the benefits in t_d (as a consequence, in the overall latency) are very significant (obtaining a reduction of 65%), whereas this same situation on the GPRS link has completely different effects. Reducing the RA interval to 40–70 ms means using almost 50% of the link to transmit RAs (considering a 36.9 kb/s link of a '3+1' GPRS phone), however, there is a 61% reduction in the t_d delay. After this analysis, for the case of the GPRS overlay the following RA interval values are suggested to maintain a convenient trade-off between added overhead and benefits: MinRtrAdvInterval = **500** ms and MaxRtrAdvInterval = **1000** ms (see [13] for details).

Upward handover	No BU simulcasting	BU simulcasting	Reduction
LAN to WLAN	7.5 ms	1.9 ms	75.0%
WLAN to 3G	750 ms	156 ms	79.2%
3G to GPRS	2500 ms	1000 ms	60.0%
WLAN to GPRS	2500 ms	506 ms	79.8%

Table 4.3: BU-simulcasting reduction (best case analysis).

4.5.2 RA caching

This method aims to eliminate the discovery time (t_d) from the total latency. The main principle behind this optimisation is that the mobile node caches every incoming RA, and when it needs to perform a vertical handover (specifically upward handovers) because of the loss of coverage, the mobile terminal does not wait for the next incoming RA from the currently available access network, but instead it uses a previously cached RA when possible, reducing (t_d) to zero.

It is important to mention that this optimisation is only useful when there is an overlap between the coverage of the old and new access routers. The algorithm was originally designed for horizontal handovers between overlapping cells. A detailed description of this version is included in [74]. This algorithm was extended to support vertical handovers. However it only reduces t_d when a cached RA exists (i.e. for the upward handover scenario).

The optimisation was implemented as a module, part of a middleware system to support complete mobility (see Chapter 5). The RAs are extracted from the network stack before they reach the MIPv6 module. These RAs are maintained in the RA-cache (according to certain policies such as time-to-live and signal strength). Then, when the MN performs a handover, high-level policies decide if an upward handover will occur and check for the cached RA to process it.

4.5.3 BU simulcasting

During IP-level handovers, the time required to communicate the new care-of address to the home agent and correspondent nodes (t_r) is typically limited by the RTT to the CN and the HA. A technique that can be used to minimise the impact of t_r on the overall latency (T_n) is to simulcast the BUs. Thus, the registration time is limited by the RTT of the fastest network and not by the latency of the new network as specified in the Mobile IPv6 specification, (Section 11.7 [48]). The t_r delay is given by: $R_t = T_{RR} + t_{BU} \geq 2.5\text{RTT}$. Table 4.3 shows the reduction values for different scenarios, using estimated RTT values. The reduction in (t_d) is calculated using the following RTT values: LAN=200 ms, WLAN=3 ms, 3G=300 ms, and GPRS=1000 ms.

4.5.4 Soft handover

The optimisations discussed thus far have been related to hard handover; the MN disconnects (stops listening) from the old interface, and just then starts listening to the new interface. As a result, packets that were already on-the-air, sent by the CN before it realises that the MN has moved to another network or those destined to the old network interface, are discarded.

These packets need to be retransmitted by the source, leading to a reduction in performance because of vertical handovers. However, vertical handovers can be made "soft" to improve inter-networking.

Traditionally, *soft handovers* (the MN attaches to the new network before breaking the connection with the old access point) have been exploited for link-layer handovers in cellular networks. This chapter analyses soft vertical handovers between highly heterogeneous networks. To enable this study using the testbed, the source code of MIPL [66] was modified so that during the handover process every on-the-air packet destined to the previous interface (i.e. old interface) is read and passed to the application, despite the incongruity between the packet's destination address (old interface's address) and the current terminal's CoA (new interface's address).

Thus, the MN keeps receiving packets from the previous network and meanwhile, it completes the registration process with its new CoA and starts receiving packets through the new interface, which has the CoA assigned. It can be thought that using soft handovers would enable the seamless roaming. However, the benefits are not always evident due to drastic differences in link characteristics (heterogeneous networks).

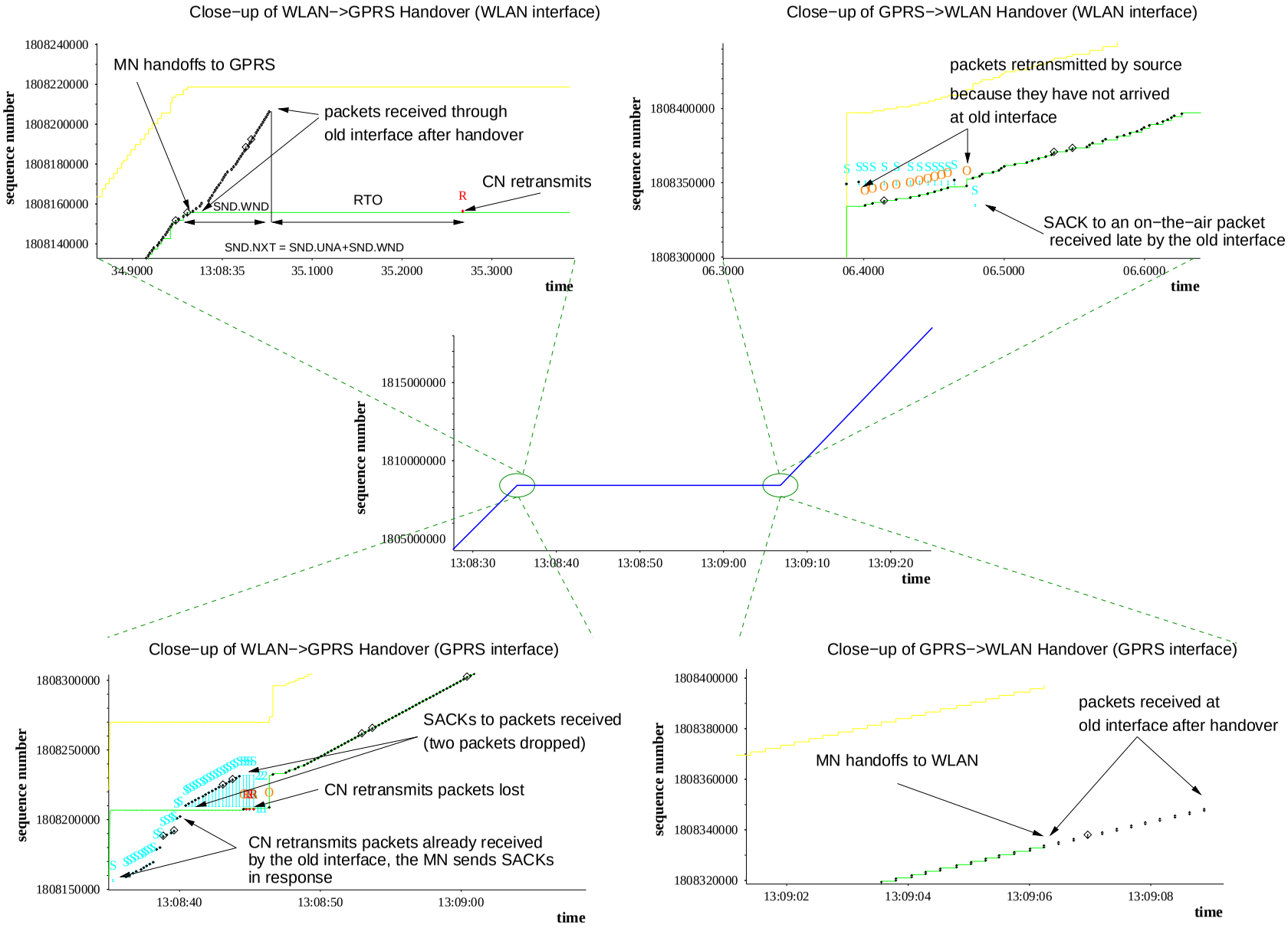


Figure 4.10: Soft GPRS-WLAN handover.

Figure 4.10 shows an example trace collected using a modified version of MIPL, showing the benefits and drawbacks of soft handovers in vertical scenarios; this figure shows the WLAN-GPRS-WLAN case, sending TCP traffic between the CN and the MN. As evident in the plot (centre), performing soft handovers leads to dramatic reductions in handover latency (there is no packet loss during the handover, although packet reordering can occur).

The MN is connected to the WLAN (upper left corner) and initiates a handover to the cellular network. However, the source (CN) continues sending packets destined to the old interface for approximately 1 ms (until SND.UNA+SND.WND-SND.NXT reduces to zero). The mobile terminal receives the on-the-air packets through the old interface and responds sending ACKs using the new interface, which is, in this scenario, much slower than the previous one. The CN times out and starts retransmitting these packets, whereas the MN sends SACKs to the source because it is receiving duplicated packets (see lower left corner plot). It can be observe that when the MN performs an anticipated upward handover (WLAN-GPRS), the disparity in the link characteristics affects the TCP connection, retransmissions occur, and the benefits of soft handovers are less dramatic.

For the other case (right side), the MN is connected to the GPRS network and starts a handover to the WLAN. Some packets are on-the-air, until the CN realises that the MN has moved—in this case, the registration process delay is very small, compared to the time that it takes for the packets to arrive at the old interface (lower right plot). The source starts retransmitting the on-the-air packets because these have not arrived to the MN due to the RTTs in the GPRS network. Finally, the MN sends some SACKs as it receives delayed on-the-air packets on the old interface, which have been already retransmitted by the source. It can be seen that because of the huge differences between networks, the source retransmits packets that are already at the MN, wasting bandwidth and reducing the benefits of soft handovers. TCP modifications to reduce the impact of link disparities in soft handovers are an avenue for future work.

The deployment of this kind of handover mechanism is crucial to offer real-time services such as video streaming. Thus, handover mechanisms that retain on-the-air packets are critical for seamless roaming.

4.6 Related work

The concepts of wireless overlay networks and vertical handover were introduced in 1996, as part of the BARWAN project at Berkeley [49, 90]. The first overlay networks testbed, the BARWAN testbed, included WaveLAN, Infrared, and Ricochet wireless networks. Obviously, this project, which was the pioneer one in the area of mobile networking, was based on MIPv4.

Other researchers [8, 111] have worked on the evaluation of MIPv4 during intra-technology handovers (i.e. horizontal handovers), for example, the project at Stanford, MosquitoNet [55].

Now, with novel protocols, researchers concentrate on minimising delays in order to enable seamless handovers, needed for supporting real-time applications. It should be noted that the type of handover (i.e. horizontal or vertical) is a very important factor that should be taken into account when dealing with the optimisation of handover performance. There are solutions defined within the IETF MIPSHOP WG, like Fast Handovers for

Mobile IPv6 [53] and Hierarchical Mobile IPv6 [88], that perform quite well in horizontal scenarios [7], as these protocols were designed with wlan-wlan handover scenarios in mind. On the other hand, the FMIPv6 solution [53] does not apply to GPRS ↔ Wi-Fi handover scenarios, because the handover signalling is something to be exchanged at the very edge of the network (i.e. access routers) and exchanging FMIPv6 signalling between an AR in the Wi-Fi network and the GGSN in the GPRS core network requires the flows to traverse too many hops, not making the process reactive as is desired in a Fast Handover approach. The vertical handover scenario poses additional challenges, because of the presence of heterogeneous networks that may present disparate characteristics.

The Moby Dick project [22, 59] proposed and implemented a global end-to-end MIPv6-based architecture to offer QoS in heterogeneous environments. The testbed included UMTS-like TD-CDMA wireless access technology, IEEE 802.11b WLANs, and wired connectivity. The mobility management approach followed by this project is based on Fast Handovers for Mobile IPV6 [53] and results, focused on intra-technology (horizontal) handovers, can be found in [7]. Further work is being done as part of a new initiative: the *Daidalos* project [23]. Nevertheless, the lack of access to a real operator's 3G network is the main difference between these projects and the LCE-CL testbed.

Projects like MIND[65], which is the follow up of the BRAIN (Broadband Radio Access for IP based Networks) IST Project, proposed their own local mobility management optimisation: BCMP (Brain Candidate Mobile Protocol), which combines properties of HMIPv6 and FMIPv6. Performance results are good for the WLAN horizontal scenario [64], although the network complexity in terms of required infrastructure is high. Moreover, the obtained performance results obtained were only for the horizontal handover scenario.

The Nomad [73] project terminated in June 2004 [76], and having successfully set up a MIPv4-based testbed [54, 37]. While this evaluated seamless roaming between heterogeneous networks based on MIPv4—assuming the presence of foreign agents in each visited network, they did not analyse the performance of MIPv6 in 4G networks.

4.7 Remarks

One of the main challenges in future communication systems is heterogeneity, when mobile devices roam between networks. The diversity in these environments augments the complexity in every stage of the handover process: network discovery, network selection, execution, and adaptation (see Section 4.4). In particular, one of the main networking-related problems is the latency of vertical handovers that results from the sum of the partial delays related to the aforementioned stages. Using MIPv6, sets of experiments were conducted to measure, characterise, and improve latency in 4G environments.

In this chapter the practical analysis of Mobile IPv6 performance is included, focusing on the vertical handover latency at the network and transport layers, highlighting the main effects of it on the protocol stack. The means to enable transparent mobility in heterogeneous environments are discussed, through the reduction of MIPv6 latencies using different optimisation methods. The optimisations described and evaluated in this chapter impact different latency components and have an effect only under certain conditions—networking in 4G environments will strongly depend on context, as a result of heterogeneity among access technologies.

Vertical and horizontal handovers are affected in different ways by terminal mobility, thus, not every effect or optimisation is equivalent for both cases. For example, it is not true that when a mobile terminal performs a downward handover, the registration time is smaller. This has two explanations: (1) the high buffering in upper layers (i.e. Vodafone's GSM/GPRS network) and (2) the Movement Detection mechanism included in [48], as was mentioned in Section 4.4.2.

One of the most interesting and unexpected results is included in Section 4.5.4. It might be thought that the so-called soft handover would always be advantageous for vertical handovers as this is the case for homogeneous environments. However, it is shown that the current TCP/IP stack soft vertical handovers exhibit a different behaviour due to the drastic changes in link characteristics. Thus, adjustments to TCP protocols would be necessary to improve soft handovers in heterogeneous environments [61].

The correct optimisation method should be applied in every scenario to obtain improved performance. These types of handover-related decisions add complexity to the roaming process that needs to be hidden from the users. Therefore, a middleware system to enable informed decisions during the handover process is one of the main drivers towards truly ubiquitous computing.

Performance differences, network heterogeneity, and context dependency are some of the reasons that lead to the need for a policy-based solution where every important condition is considered. PROTON (Chapter 5) has been built to demonstrate the usefulness of a cross-layer design approach, combined with concepts from autonomic communication and policy-based systems. This client-based middleware enables mobility support for 4G networks, and it can be considered as an early attempt to apply autonomic communications in future networks.

Future networking environments (4G) will be composed of multiple heterogeneous access networks, highly integrated to offer ubiquitous connectivity to a plethora of IP-based mobile services from different devices. This vision poses the clear need for a policy-based solution (e.g., PROTON, Windows XP Wireless Zero Configuration (WZC), etc.) to control the complex relation between demanding services and networking resources, based on context.

In conclusion, this research contributes to the evaluation of MIPv6 as *de facto* solution for mobility management in future networks. An integrated network was used to emulate a 4G system and collect real experiences (the most relevant of which are included in this chapter) that helped us understand the current challenges.

Chapter 5

Autonomic System for Future Networks

The ubiquitous explosion of Internet services and the rapid proliferation of mobile networked devices, as well as radio access technologies, creates a unique challenge for networking researchers. The next generation of communication systems will involve mobile users interacting with a pervasive computing environment that adapts accordingly. New solutions are required for managing interactions among the plethora of inter-connected networks, wireless devices and IP-based services.

There is a wide range of wireless access networks becoming available such as infrared, Bluetooth, 802.11-based wireless LANs, cellular wireless, and satellite networks, which will combine to provide a highly integrated wireless access platform. Katz, et al., termed this model as Wireless Overlay Networks [49]. The wireless networks that form the overlay have different characteristics, and there is a trade-off between bandwidth and coverage (typically, smaller/local coverage has higher bandwidth).

The evolution in wireless access technologies shows that the trade-offs between coverage and bandwidth will exist. Ideally, a wireless access technology with unlimited coverage and infinite bandwidth would be desirable. Since this is not easy to achieve (due to spectrum and mobility constraints), researchers are focusing on creating an integrated platform architecture able to provide better access for mobile users. Thus, the vision for the next generation of wireless architecture (4G) builds on the key notion of heterogeneous wireless integration and inter-networking.

Also, the growth in the popularity of Internet services among mobile users, together with the higher QoS required by novel applications, requires improving resource management capabilities in mobile devices to offer a better user experience. High mobility, seamless roaming, high data access rates and transparent connectivity to services from "any" device are dominant trends in the 4G vision and the basic reasons to think that *autonomic computing* provides a plausible solution for emerging challenges.

Autonomic computing is an approach to self-managed systems with a minimum of human interference. This new computing paradigm means that the design and implementation of an autonomic system must exhibit these fundamentals from the user perspective: flexibility, accessibility, and transparency [44].

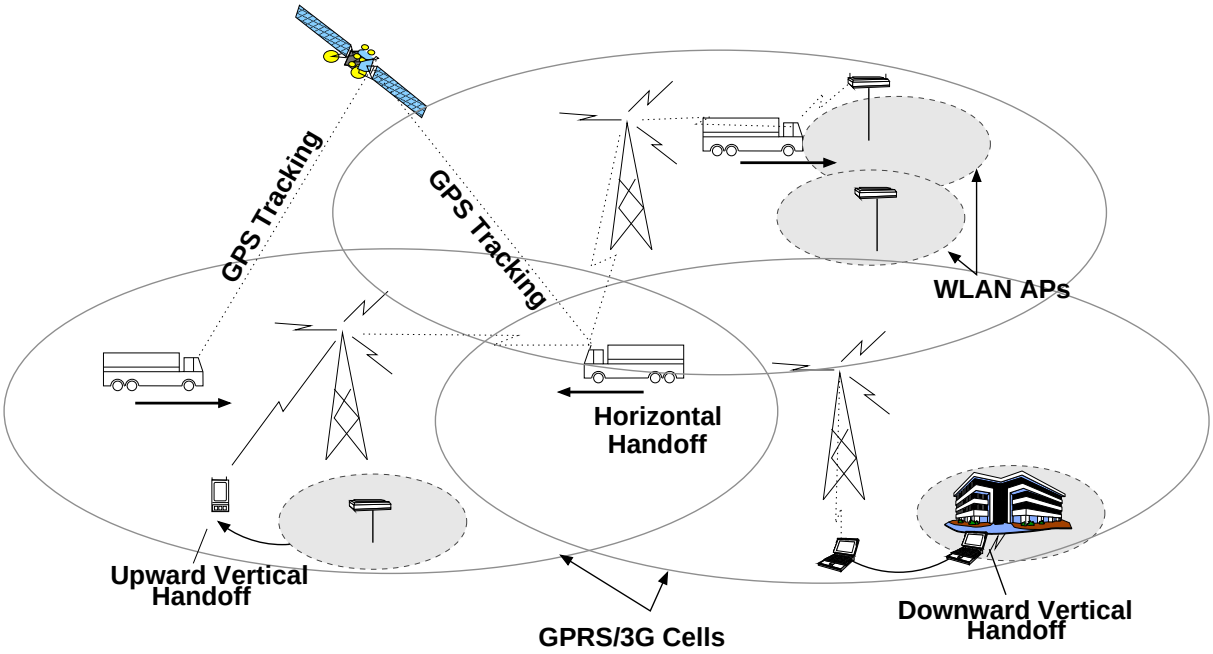


Figure 5.1: Future 4G communication system.

I hold that some principles of autonomic computing should be applied to the design of solutions to support mobility and deal with complexity in the next generation of wireless networks. This Chapter describes PROTON [101], a solution that blends concepts of autonomic computing, policy-based systems, and a novel model based on *Finite State Automata* (FSA), to solve mobility management issues in 4G networks.

This FSA-based model uses a new metric called *Tautness Function* (TF), and a new kind of automata called *Finite State Transducer with Tautness Functions and Identities* (TFFST). The TF and the TFFST were defined to model policies and resolve potential conflicts. Conflict resolution has been one of the main obstacles for policy-based systems and the model handles conflicts with good run-time performance while greatly reducing human intervention.

5.1 The Problem: Seamless complexity

Future wireless environments will not consist simply of one radio access technology such as current cellular systems (e.g., GSM, WCDMA, or EDGE), but will integrate multiple access networks, adding complexity to mobility management systems. Moreover, seamless inter-networking (as shown in Figure 5.1 [1]) will be a basic feature in mobile terminals to allow connectivity in this pervasive computing environment.

Giving such capability to users across heterogeneous networks is much more complicated than in homogeneous scenarios. In this case, where multiple disparate networks are accessible from a mobile terminal, detecting the possible options and choosing the optimal combination of network resources and active applications at the correct moment, becomes a complex procedure.

[1]Source: R. Chakravorty

Homogeneous Networks	Heterogeneous Networks
Detection of access points belonging to same system.	Detection of access points belonging to multiple systems.
Mobile host decides where to handoff among access points belonging to the same technology.	Mobile host decides where to handoff among access points belonging to multiple technologies.
Handover initiation triggered mainly by signal strength fading.	Handover initiation triggered by multiple events.
The execution methods can be applied in every situation.	The execution methods depend on context and not all methods can be applied in every scenario.
Adaptation process is not as important because the mobile host roams between similar conditions (same technology).	Adaptation is essential, the mobile host roams between disparate technologies and conditions change drastically.

Table 5.1: Complexities in 4G systems compared to homogeneous environments.

In contrast to traditional algorithms, mobility management systems will need many parameters to support vertical handover-related processes. Table 5.1 shows the main challenges in 4G systems, mobile devices need more intelligent solutions to handle these complexities, while maintaining transparency to avoid affecting usability.

5.1.1 Autonomic solution for 4G systems

IBM research outlined eight defining characteristics of an autonomic system. Considering heterogeneity, dynamics, and complexity added in 4G environments; an appropriate support should endeavour to possess these key elements with the intention of offering a complete seamless solution [44]. From these concepts, the following characteristics were integrated in PROTON's design:

To be autonomic, a system needs to "know itself". An autonomic system will need detailed knowledge of its components, current status, and ultimate capacity, as well as possible connections with other systems. PROTON's architecture (described in Section 5.2) allows the system to access a detailed *Networking Context*, which includes important data about mobile host's network resources, activity, physical environment, as well as users' preferences at all times. This gives the device capability to know the extent of its own resources and decide how to use them.

An autonomic system must configure and reconfigure itself under varying and unpredictable conditions. PROTON uses the knowledge about its context (i.e. Networking Context) to feed a policy-based model that controls terminals' initial configuration as well as its ongoing behaviour according to the generated events (e.g., connection/disconnection, activity variations, and users' preferences changes).

An autonomic system never settles for the status quo—it always looks for ways to optimise its workings. In this sense, considering dynamics in the conditions when dealing with mobility, PROTON always senses the environment and evaluates policies to look for the best possible *quality of service* considering terminal activity and connectivity resources.

An autonomic system knows its environment and context surrounding its activity, and acts accordingly. It is essential for PROTON to sense its context and produce events to trigger policies that drive mobile node's behaviour.

An autonomic system cannot exist in a hermetic environment. In this sense, PROTON is compatible with the TCP/IP stack and it helps in the integration process of heterogeneous networks, creating an open IP-based platform to access mobile services.

An autonomic system will anticipate the optimised resources needed while keeping its complexity hidden. PROTON offers seamless mobility support, coping with the complexity posed by 4G systems, hiding it from the users.

Currently, a system incorporating the eight elements [44] will be very difficult to build. However, the solution presented in this Chapter can be considered as an early attempt to critically examine such concepts. An autonomic system seems appropriate to tackle the complexity posed by future integrated heterogeneous environments formed by diverse access networks and services, and a huge variety of mobile terminals interacting.

5.1.2 A novel approach

Using well-known entities such as finite state machines, this work proposes interpretations and adaptations to the basic theory to suit the domain of policy-based systems. Policies are modelled as finite state transducers that consume events, and a function called *tautness function* is defined for the transitions.

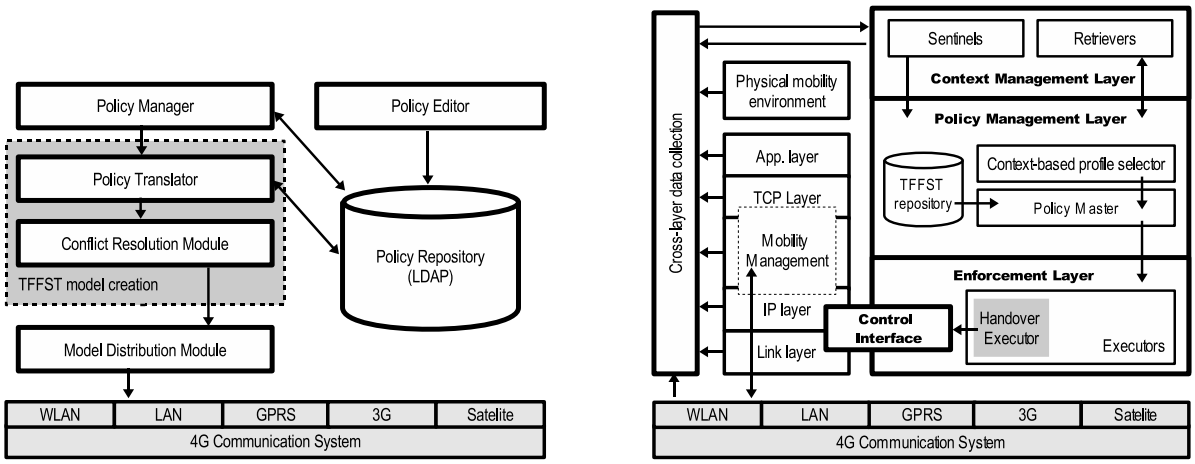
The operations of finite state transducers are revised accordingly. In particular, determinisation and intersection operations have been adapted to mimic the modality conflict resolution process between policies. Also, it is shown how the composition of transducers could be used to express constraints, e.g., the restriction of a particular subset of policies according to the actual context.

To adapt this approach to resource-constraint mobile devices, all the tasks associated with the conflict resolution process can be done beforehand, and the computing of a policy-evaluation is linear in the number of events, and independent of the number of policies. This makes it possible to deal with the complexity and dynamics of 4G systems while keeping a light-weight solution.

The next section describes PROTON's architecture that is divided into network- and host-side components. Section 5.3 introduces the concept of Networking Context, defining the three datasets that form it. Then, Section 5.4 explains the policy model based on *Finite State Transducers* (FST). Section 5.5, describes the processes related to the generation, distribution, and evaluation of TFFSTs. Finally, Section 5.6 includes some remarks.

5.2 Architecture

PROTON components are divided into network-side and host-side components. The reason for this is that because of the number of decisions required to fully support the handover process, the raw policy set can get too complex to maintain within a limited mobile device. However, the functionality still being completely based on the mobile host, only the highly demanding pre-processing tasks related to the policy evaluation model are placed in the network—where computing constraints are much more relaxed (see Figure



(a) Network-side components

(b) Host-side components

Figure 5.2: PROTON's architecture.

5.2(a)). The host-side components are organised into a three-layered system: *Context Management Layer*, *Policy Management Layer*, and *Enforcement Layer*, which sit on top of layer 3 in the protocol stack. The network-side contains the components related to the specification and distribution of the policies.

5.2.1 Network-side components

Those components that involve operator's management or high computational cost are located in the network to minimise complexities at the mobile terminal. This is the case of policy definition, storage, and conflict resolution. Network-side components are shown in Figure 5.2(a).

Policy Editor—To create the system policies, the operator must write them in a high-level policy specification language. Ponder was used as the high-level language because of its expressiveness and deployed tools. In particular, PROTON uses the *Ponder Policy Editor* and its compiler [78] to create the first internal Java representation of the policies.

Policy Repository—The policy repository is implemented using a light-weight Directory Access Protocol (LDAP) server, which intends to store system policies in their high-level representation as well as in the internal Java representation.

Policy Translator—This component translates the policies specified in the Ponder language [24] into the evaluation model described in Subsection 5.4.2.

Conflict Resolution Module (CRM)—The conflict resolution module builds the deterministic Finite State Machine (FSM) modelling every active policy. The CRM performs two main tasks: (1) it combines the policies considering the system constraints and (2) it resolves conflicts among those rules. During this task, all possible static and dynamic conflicts are foreseen. Therefore, the algorithms that are executed have a high computational cost. The main benefit of adding such overhead on the network-side is to avoid heavy tasks in the mobile device (usually a terminal with limited computational power and memory capacity). Furthermore, after resolving conflicts and constructing the deterministic FSM, the mobile device can react quickly to incoming events.

Model Distribution Module (MDM)—Once all policies and constraints are combined in a TFFST, it is delivered to the mobile device and installed into its *policy master* to drive its decisions. One TFFST is created and deployed for each mobility profile and this module takes care of coding and transmitting the transducers to each mobile device.

5.2.2 Host-side components

Context Management Layer (CML)—This layer has two types of components responsible for collecting Networking Context: *Sentinels* and *Retrievers*. The former are responsible for collecting dynamic elements, and the latter manage static elements. There is a responsible object for each context element, and it has individual settings (e.g., polling frequency and local rules) depending on the complexity and dynamics of a particular fragment. For example, *VelocitySentinel* only polls the velocity every second due to the constraints in the GPS receiver. The local rule (shown in Section 5.3) filters the collected data according to the current velocity and acceleration. Thus, not every reported measurement generates an event.

Policy Management Layer (PML)—Responsible for the control and evaluation of the policies to drive the behaviour of the mobile device and it has the following elements:

Policy Master: This component acts as the Policy Decision Point (PDP) in the policy system [69]. It receives events (e.g., Transition-Pedestrian produced by the VelocitySentinel) from the CML, and according to these inputs, it decides the possible actions to execute, which are immediately sent to the Enforcement Layer.

Context-based profile selector: The fact that only a small portion of sensory input is relevant under certain conditions is used to improve the performance of the system. Some inputs can generate special events (i.e. *macro-events*) which are then used by the selector to load a profile that defines a valid subset of policies to evaluate, i.e. the appropriate TFFST . An example of a macro-event is velocity—if host speed is more than 90 km/h the only active policies are those that produce an upward handover as an action. This means that mobile users should never attempt to connect to a lower layer when moving at very high speeds.

TFFST Repository: The TFFSTs are produced in the network side, as mentioned in Subsection 5.2.1, and then deployed into the mobile device where they are kept in the TFFST repository. Thereafter, the selected TFFST and its evaluation are decided according to the events received from the CML.

Enforcement Layer (EL)—Formed by different *Executors* that are the Policy Enforcement Points (PEPs) of the system [69]. They are responsible for performing the actions that result from evaluating the TFFST. The EL connects with the lower layers through a *Control Interface* (CI) that captures incoming router advertisements just before they reach the Mobile IPv6 module—prior to the handover procedure. The CI executes different scripts, which receive the selected interface as a parameter and outline the execution handover method.

Communication protocols—For the connection CML-PML and the communication within the PML, a generic asynchronous notification service called *Elvin* is used [38]. This service was designed as a middleware for distributed systems. However, many research projects have used Elvin due to its simplicity. Ponder uses this messaging service in its framework, and it is use in the present system as well.

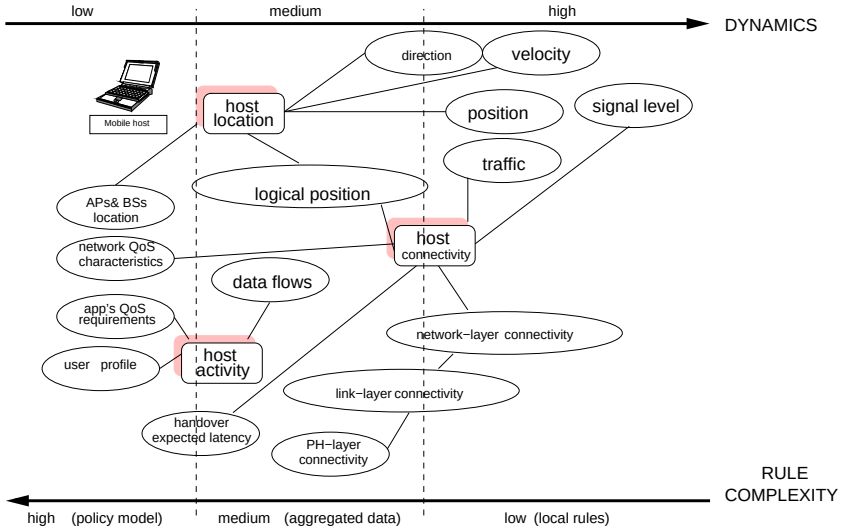


Figure 5.3: Networking Context components.

5.3 Networking Context

Context is defined as any information sensed from the environment which may be used to define the behaviour of a system. The effectiveness of PROTON's assistance depends on three main tasks: accurate extraction, combination, and expression of unsteady measurements collected from the environment. These tasks are constrained by three factors: frequency in sensory capture, complexity in context fusion, and limited inference capability, respectively.

Since in a highly dynamic environment, the instability of the sensed data has a negative impact on the amount of information that can be extracted from a particular context fragment, PROTON organises sensed data (i.e. Networking Context) into a three-level hierarchy according to: *dynamics of sensed data* and *complexity of the rules applied*. This taxonomy results in the definition of three datasets, each of which has a particular combination of rule complexity and component dynamics.

Collected dataset—Every dynamic fragment gathered by a sentinel is part of this dataset (high and medium dynamics components). Sentinels poll data from many different sources, and then filter it according to simple local rules that only affect the specific context element. The output of the collected dataset is smaller than the input, which reduces the processing overhead in the mobile device.

For example, the local rule shown below corresponds to the VelocitySentinel, it filters the collected data (every second) according to current speed and the increment in velocity. Therefore, this minimises processing and assures that generated events respond to meaningful context changes.

```
void localRule(double currentVelocity) {
  double velDiff;
  velDiff = Math.abs(currentVelocity - Host.getVelocity());
  if (currentVelocity < PEDESTRIAN & velDiff > 2.5)
     /*Generate event Transition-pedestrian*/
     event = new NamedEvent();
     String[] params = new String[10];
     params[0] = "Transition-Pedestrian";
     params[1] = "double";
     params[2] = Double.toString(velDiff);
     event.HandleEvent(params);
  if (currentVelocity < LOW_AUTOMOBILE & velDiff > 5)
     /*Generate event Transition-low-automobile*/
  if (currentVelocity < HIGH_AUTOMOBILE & velDiff > 10)
     /*Generate event Transition-high-automobile*/
  if (currentVelocity > HIGH_AUTOMOBILE & velDiff > 20)
     /*Generate event Transition-high-speed*/
}
```

Aggregated dataset—It groups the filtered and retrieved information coming from the CML. The former is the output of the collected dataset after applying the corresponding local rules. The latter derives from the low-dynamic components, which are managed by Retrievers, e.g., user preferences retriever or application profile retriever.
Networking Context dataset—It is a snapshot of the Aggregated and Collected datasets used by the Policy Master to select the path and evaluate the conditions in the TFFST. The Networking Context allows the mobile host to have complete knowledge of its resources, context, and activity at all times.

5.4 Policy model

Multi-mode mobile devices must be flexible and proactive to cope with dynamics and changes in 4G systems. PROTON has to cover several aspects that derive from this premise:

- The solution must include physical context (e.g., velocity and position).
- Adaptation must be supported in the system.
- PROTON must lead to unambiguous decisions in the shortest possible time.

After pondering these requirements, an effective approach to address the problem is a policy-based system to assist users in future mobile scenarios. Moreover, considering the constraints of dynamics and complexity, context was broken into simpler and more intuitive fragments (as shown in Section 5.3) and policies were written using these elements as conditions.

Thus, complexity is transferred to the combination of policies and decision-making, instead of having it in the individual rules. Therefore, using a policy-based system enables easier tuning of the system's behaviour. Employing cost functions to drive decisions can often lead to static and over-complicated solutions, as the complexity is related to the number of parameters.

Furthermore, breaking down context into fragments allows us to use independent normalisation functions for each element. This leads to a more accurate transformation of the parameters, while cost functions are more static. In conclusion, a policy-based system is more flexible and can express more than a cost function.

The policy model uses Ponder [24] as a high-level language for policy specification. This framework is used to obtain an initial Java representation from the high-level policy. The Ponder language provides a common means of specifying policies that map onto various actors within a network. However, adaptations are required in order to use Ponder in a particular application as the implementation of an autonomic solution for 4G systems.

5.4.1 Policy specification

PROTON follows the *Event-Condition-Action* (ECA) paradigm where policies are rules that specify actions to be performed in response to predefined conditions, triggered by events (see sample policy below).

```
    Rule 1:

inst oblig /ProtonPolicies/Obligs/CheckupPolicy {
  on PhysicalConnection(nic);
  subject /ProtonPMAs/HandoverPMA;
  target t = /ProtonTargets/HandoverExecutor;
  do t.networkSelectionEvent(nic);
  when t.isLinked(nic);
}
```

The policy shown above, *CheckupPolicy*, is triggered when a new radio access interface is connected to the mobile host—the event *PhysicalConnection* is sent by the *Attached-Sentinel*. The policy target, *HandoverExecutor*, checks the connectivity in the network interface executing the method *isLinked(nic)*. Then, when the new NIC (Network Interface Card) is linked the policy target sends an event to initiate the process of network selection by executing the method *networkSelectionEvent(nic)*. This high-level policy is compiled into an initial Java representation and translated into TFFSTs.

5.4.2 An evaluation model based on FSTs

Finite state automata are classical computational devices used in a variety of large-scale applications. FSTs, in particular, are automata whose transitions are labelled with both an input and an output label. They have been useful in a wide range of fields, but particularly in Natural Language Processing. This discipline makes intensive use of grammatical rules, which are ambiguous by nature, and requires quick decisions based on those rules, in particular in fields such as speech recognition with major performance requirements.

Additionally, FSM-based solutions are typically light-weight. They can be implemented as arrays of states, transitions, and pointers among them without falling into computationally expensive management structures.

Policies are represented as *deterministic transducers* that are a category of transducers without ambiguities. This means that at any state of such transducers, only one outgoing arc has a label with a given symbol or class of symbols.

Deterministic transducers are computationally interesting because their computation order does not depend on the size of the transducer, but rather only on the length of the input since the computation consists of following the only possible path corresponding to the input and writing consecutive output labels along the path [81].

For representing policies with FSTs the model presented in [6] was used (see Appendix B). It is based on a modification of predicate augmented FSTs [98], in which predicates were replaced by a metric representing the relation between a policy and a given event.

A policy has a condition delimiting a region where a given event can or cannot lie. When such an event is inside two or more overlapping regions a modality conflict may arise. The concern is to know how *tautly* a condition fits to an event instead of how far from the border it is. Thus, the preferred condition will be that which is the most *taut* around the event under consideration.

In order to quantitatively represent the aforementioned *tautness*, a metric called Tautness Function is used, a real number in the interval $[-1, 1]$ so that the more taut a condition is, the closer its TF is to zero.

Definition 1 *A Tautness Function associated with a condition c, denoted τ_c, establishes a mapping from $E \times C$ to the real interval $[-1, 1]$ where:*

- *E is the set of possible network events or attempted actions,*
- *C is the set of policy conditions,*
- *$\tau_c(e) \in [-1, 0) \Leftrightarrow e$ does not satisfy c,*
- *$\tau_c(e) \in (0, 1] \Leftrightarrow e$ satisfies c,*
- *$\tau_c(e) = 1 \Leftrightarrow \forall f \in E, f$ satisfies c,*

When the TF is modelling the condition part of the rule, in condition c the subject or any other property of the condition such as temporal constraints, are included. In the same manner, when the TF is modelling the action part of the rule, condition c includes the target or any property of the action.

To provide an intuitive example of TF, let us assume that one policy specifies *wireless interfaces* in general and another policy specifies *IEEE 802.11b interface* (a subset of *wireless interfaces*). For an action attempted by a IEEE 802.11b interface, the second policy should define a TF that is closer to 0 than the first policy. However, as with the distance-to-a-policy concept, much more complicated expressions could be computed, for example using the associated traffic types to the interface or the QoS characteristics.

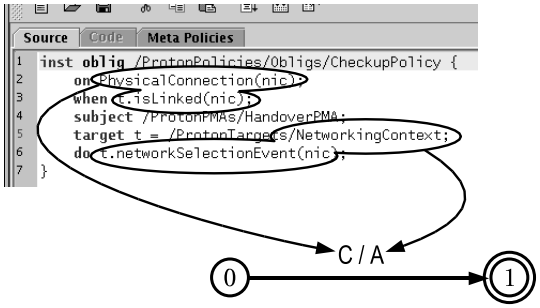


Figure 5.4: Translation process.

Notice that in the TF definition only the general rules are stated, with which a TF should comply. This non-specificity is deliberate, because how it should be implemented or how it maps events and conditions to real numbers should be decided in the context of a specific policy-based system and technology. Thus, a TF is an abstraction layer of technology-dependent issues that allow us to work in a more general fashion.

In Subsection 5.5.6 some examples of how to compute TFs are shown. The most outstanding advantage of using TFs in PROTON is the capacity to define a different way to evaluate each networking context fragment and combine them using the algebra for TFs, which defines the basic logic operators *disjunction*, *conjunction* and *negation*, plus two new operators called *tauter-than* ($\rightarrow_\tau$) and *as-taut-as* ($\rightleftarrows_\tau$) specially formulated to express the concept of distance in the TFs (see detailed algebra definition in a summary of [6] included in Appendix B). The definition of the transducers that use TFs to model policies internally on the host side follows.

Definition 2 *A Finite state transducer with tautness functions and identities M is a tuple (Q, E, T, Π, S, F) where:*

- *Q is a finite set of states,*
- *E is a set of symbols,*
- *T is a set of tautness functions over E.*
- *Π is a finite set of transitions $Q \times (T \cup \{\epsilon\}) \times (T \cup \{\epsilon\}) \times Q \times \{-1, 0, 1\}^2$.*
- *$S \subseteq Q$ is a set of start states.*
- *$F \subseteq Q$ is a set of final states.*
- *For all transitions $(p, d, r, q, 1)$ it must be the case that $d = r \neq \epsilon$.*

In the implementation, an extension of the above definition is used to let the transducer deal with strings of events and actions in each transition. Policy rules are modelled using TFFSTs, in which the incoming label represents the condition and the outgoing label the action.

[2]The final component of a transition is an "identity flag" used to indicate when an incoming event must be replicated in the output.

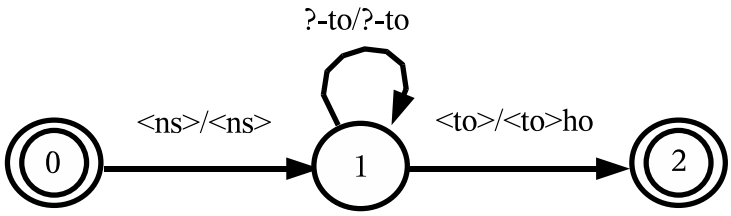


Figure 5.5: TFFST model for the obligation in Rule 2.

5.4.3 Modelling policies with TFFSTs

To understand how the entities introduced before are used for modelling policies, it is fundamental to present how *obligations* and *constraints* are expressed. TFFSTs can also model authorisations, prohibitions and dispensations [6], but the following two policy types are expressive enough to deal with current PROTON requirements.

Obligations—An *obligation* is a rule expressing that when an event fulfils a particular condition, a given action must be executed. It is represented as a transducer with a main link that has an event as the input and the action as the output. Typically, the incoming event will report the occurrence of a fact and the outgoing event will order the execution of a given action. However, other combinations are possible as well, for instance, to be *unobtrusive* (as defined by [16]), the incoming event could be replicated in the output.

A clear view of the links between objects generated by Ponder tools and the TFFST structure is shown in Figure 5.4. The Ponder distribution [78] was modified to handle the new TFFST structures and support the translation process.

Some actions can be conditioned on the occurrence of more than one event. This is the case of *lazy switching handover method*, in which after initiating the handover (receiving the *NetworkSelected(nic)* event) the action needs to be delayed—wait for the *TimerOver(delay)* event.

To express an action as a consequence of a set of events, e.g., Rule 2, a transducer such as the one in Figure 5.5 is deployed.

```
    Rule 2:

inst oblig /ProtonPolicies/Obligs/LazyHandover {
  on NetworkSelected(nic) → TimerOver(delay);
  subject /ProtonPMAs/HandoverPMA;
  target t = /ProtonTargets/HandoverExecutor;
  do t.handoff(nic);
  when t.isRAreceived(nic);
}
```

In the figures, the symbol "?" represents the TF associated with the *all-event* condition while the "-" symbol represents set substraction and "ϵ" means a null event. The *NetworkSelected event* is indicated using "ns", the *TimeOver event* with "to", the *FadingSignal event* is "fs", and the *Handoff action* uses the "ho" abbreviation. By convention, state 0 is *initial* and a double-circled state is *final*. Symbols "<" and ">" enclosing TFs in the labels, express identity between inputs and outputs (see [81]).

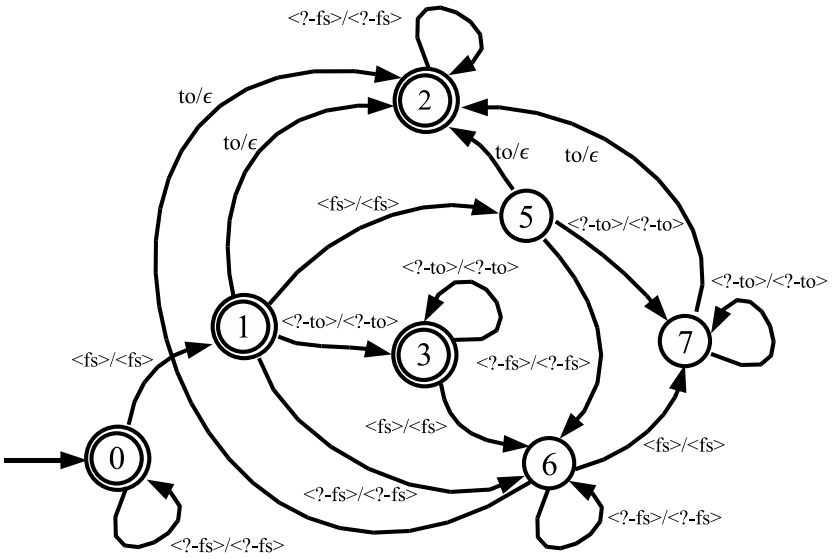


Figure 5.6: TFFST model for the constraint in Rule 3.

For the sake of simplicity, the fact that events can arrive without order is disregarded, and other possible events before and after the sequence of interest in the model are not included.

Constraints—Constraints are expressed using the *composition* TFFST operation included in Appendix B, an analogue operation to composition between functions. After all the obligations are represented in a single transducer, the transducer representing constraints should be subsequently composed.

To see how constraints work, let us assume the *InsertHysteresis* example of Rule 3. If the model relies only on the plain policy, if a *FadingSignal(nic)* event occurs, the host can fall into the ping-pong effect. One possibility for avoiding this situation is to create the following constraint:

```
    Rule 3:


inst oblig /ProtonPolicies/Obligs/InsertHysteresis {
  on FadingSignal(nic);
  subject /ProtonPMAs/HandoverPMA;
  target t = /ProtonTargets/HandoverExecutor;
  do t.ignoreFadingEvent(nic);
  when t.hysteresisPeriod(time);
}
```

The transducer shown in Figure 5.6 represents this constraint. Computing the *composition* of both transducers produces the solution shown in Figure 5.7, in which all possible system responses are analysed a priori in the network side.

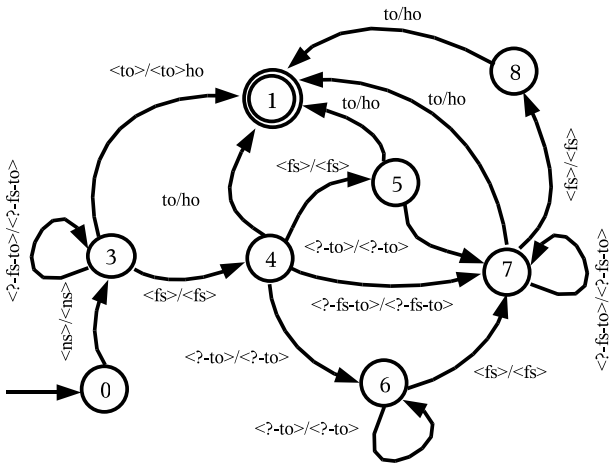


Figure 5.7: TFFST model for composition of rules 2 and 3.

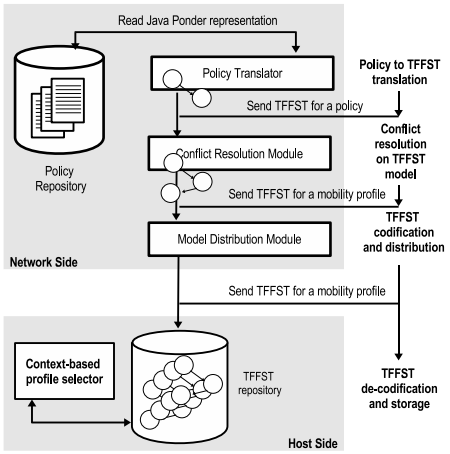


Figure 5.8: Model distribution process.

5.5 Processes

This section describes the processes related to the policy model. Several tasks have to be performed to generate, deploy, and evaluate the TFFST corresponding to a policy set (see Figure 5.8). These tasks are: policy translation, conflict resolution, model distribution, context gathering, policy evaluation and tautness function computation.

5.5.1 Policy translation

Policy translation from high-level languages into internal policy evaluation models can be a complex task that needs to be kept simple and performed on an ad-hoc basis in the system. As mentioned above, high-level policies built with the Ponder policy specification language must be translated into the internal policy evaluation model comprised of TFFSTs. The translation process follows the principles presented in Subsection 5.4.3.

The main challenge of the deployment was the implementation of the TFs associated with the policy conditions. Considering the fact that the PROTON policy model is based on the tools provided by Ponder, the approach chosen was to keep its object-oriented approach using target and subject methods to compute TFs.

Thus, when target or subject methods are called to check a *when* clause, a corresponding method is called at the same time to assign a TF value instead of the boolean value that Ponder assigns to the condition. This method should be developed explicitly, enabling the design of different TF computations depending on a specific parameter (for conditions represented by logical combinations of simple conditions, the TF algebra remains valid).

5.5.2 Conflict resolution

An advantage of using transducers to model policies is the rich set of operations available. It is possible to join, intersect, complement, compose and determinise transducers under certain conditions. To build a TFFST that models a set of positive obligation policies, one for each policy needs to be built and join them using the *union* operation. However, the union of TFFSTs maintains ambiguities and contradictions. Therefore, *determinisation* and *composition* operations must be performed to eliminate these problems.

Determinisation transforms a TFFST into its deterministic and unambiguous version, in fact it also eliminates *static conflicts* between policies.

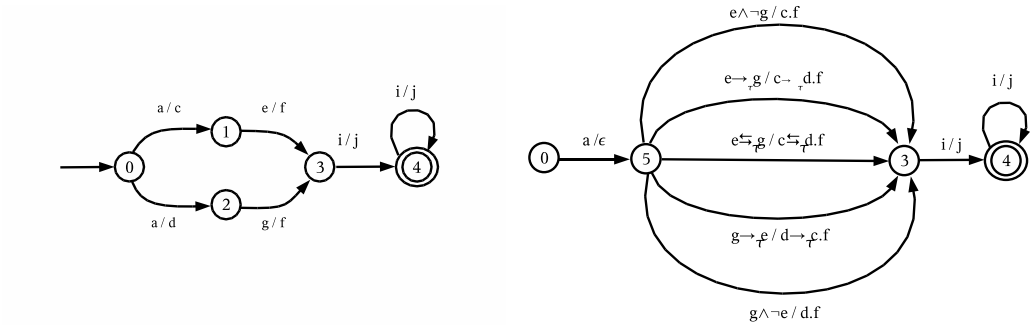
A TFFST M is *deterministic* if M has a single starting state, if there are no states $p, q \in Q$ such that $(p, \epsilon, x, q, i) \in \Pi$, and if for every state p and event e there is at most one transition (p, τ_d, x, q, i) such that $\tau_d(e)$ is positive.

If a TFFST is deterministic then the process of computing the output actions for a given stream of events ω, can be implemented efficiently. This process is linear in ω, and independent of the size of the TFFST. The determinisation algorithm has two main stages:

1. *Eliminating apparent local conflicts:* Local ambiguity may not be such if by analysing the whole transducer, it is realised that only one path is possible until the final state. This is the case of the ambiguity shown in *state 0* (see Figure 5.9(a)). Therefore, outputs are delayed as much as possible.

2. *Resolving static conflicts:* If it is not possible to delay output labels further, the second stage begins. A transition is created for each possible combination between potentially conflicting conditions applying the following criterion: although an event satisfies two conditions, one of these conditions fits more *tautly* than the other. The idea of tautness is represented by the Tautness Functions defined in Subsection 5.4.2, which can be used to compare orthogonal conditions.

In the output part of the transition, actions and events are arranged following the order given by operators on the input. These operators are in fact part of the output. Later in the process these operators will be eliminated by the *composition* of transducers to apply the given constraints in the system. Figure 5.9(b) shows the transducer after determinisation.



(a) Before determinisation (b) After determinisation

Figure 5.9: Determinisation process.

Composition eliminates semantic contradictions (i.e. dynamic conflicts) between the actions. This operation between transducers is equivalent to the composition of any other binary relation: $R_1 \circ R_2 = \{(x, z) \mid (x, y) \in R_1, (y, z) \in R_2\}$.

Thus, the process can be understood as a chain of events where the events and actions in the output of the first transducer are considered to be the input of the second one. The advantage is that the chain process is performed analytically in the network and not on the mobile device.

If a TFFST is created, which replicates all input actions on the output except for those patterns of actions not allowed on the system, and then the *composition* of that transducer with the TFFST policy model is computed, and a transducer that enforces actions without dynamic conflicts is obtained. This means actions that must not be performed at the same time, for example two handovers, each one to a different network.

Consequently, conflict resolution is intrinsic to the model. This process not only builds the transducer that models the policies, but also eliminates ambiguities and contradictions between those rules. The main steps are:

1. Compute the union of all transducers representing *rights* and *obligations*.
2. Substract the transducers representing *prohibitions* and *dispensations*.
3. Compose the resulting transducer for each constraint transducer.
4. Determinise the resulting transducer to solve conflicts.

Determinisation and composition operations are extensions to the algorithms developed for predicate-augmented FSTs [98]. Baliosian et al. present a detailed explanation of these extensions in [6], and they were also included in Appendix B.

5.5.3 Model distribution

After policy translation, conflict resolution and TFFSTs composition, the final set of TFFSTs for every mobility profile is built (everything so far happens on the network side). Thereafter, the TFFSTs need to be sent to the repository in the mobile host. The Model Distribution Module together with the Policy Master, are responsible for the installation process.

Context fragment	Component	Event
Layer 1 connectivity	AttachedSentinel	PhysicalConnection(nic) PhysicalDisconnection(nic)
Signal strength	SignalSentinel	FadingSignal(nic)
Layer 2 connectivity	LinkedSentinel	LinkConnection(nic) LinkDisconnection(nic)
Layer 3 connectivity	RouterAdsSentinel	NetworkConnection(nic) NetworkDisconnection(nic)
Handover latency	HandoverRetriever	no generated events
Logical position	LogicPositionSentinel	ChangeLogicPosition(address)
Physical position	PositionSentinel	ChangePosition(position) ContextChangeTransition()
Velocity	VelocitySentinel	PedestrianVelocity() LowAutomobileVelocity() HighAutomobileVelocity() HighSpeedVelocity()
Direction	DirectionSentinel	ChangeDirection(direction)
Network traffic	TrafficSentinel	ChangeTraffic(nic)
User preferences	UserRetriever	ChangePreference(preference)
Ongoing applications	FlowsSentinel	NewDataFlow(trafficType)
Network charac.	NetworkRetriever	no generated events
App. characteristics	ApplicationRetriever	no generated events
Network structure	InfrastructureSentinel	NearbyAccess(positionArray)

Table 5.2: CML components, context fragments, and generated events.

The TFFSTs are kept in a repository, and loaded jointly with the mobility profile according to the reception of a macro-event, e.g., LowAutomobileVelocity(), see Figure 5.8 on page 92.

5.5.4 Context gathering

Table 5.2 shows the relation between context fragments and the correspondent CML component responsible for polling or retrieving the data. The process of gathering the Networking Context has three steps. The first task is done by Sentinels and Retrievers in the CML, and consists of polling context data (i.e. Collected dataset). Then, the resulting information is filtered according to local rules—the Aggregated dataset is the result of this step. Finally, the CML components maintain a snapshot (i.e. Networking Context) of the context fragments to evaluate policies and generated events (see Table 5.2) when a relevant change in this information occurs. For example, *User preferences* context fragment represents aspects that users can control, e.g., cost, velocity, etc. The retriever *UserRetriever* gathers the corresponding data when a user modifies his priorities.

5.5.5 Policy evaluation

Policy evaluation occurs in the TFFST model. As mentioned, the computational load of deterministic transducers does not depend on the size of the transducer but rather only on the length of the input. This is possible because the computation consists of following the only possible path corresponding to the input represented by an *epoch* or window of events, which are considered simultaneous for the purpose of detecting dynamic conflicts.

When evaluating the epoch, the transducer performs two tasks: it checks the current epoch and decides if it contains a relevant event pattern in order to decide whether or not to accept it; then it produces a sequence of actions for every accepted epoch, which is sent to the Policy Enforcement component.

5.5.6 Tautness function computation

A fundamental process in the distribution of TFFST models is the appropriate computation of tautness functions. The prototype handles each parameter individually with the common idea of expressing the probability of a condition. For example when computing a condition (related to bandwidth) such as:

...
when t.effectiveBW([nic A]) <
t.effectiveBW([nic B]);
...

If *nic A* is connected to a hotspot and *nic B* uses Vodafone's GSM/GPRS network, considering the maximum data rates presented in Table 3.1, and assuming their values have uniform distributions, when the condition to *true* is evaluated, its tautness function is:

$$\frac{144kb/s}{11Mb/s \times 1000} \times 0.5 = 0.00654$$

This function shows the conditional probability of occurrence for a handover policy, based on bandwidth. The probability to handoff from a cellular system to a hotspot is one half. However, this value is affected by the difference in bandwidth. The function shows that the probability of the maximum bandwidth of a GPRS network (144 kb/s) being bigger than the bandwidth of a hotspot (11 Mb/s) is very small. In the cases where this happens, the mobile node would handoff from the hotspot (without loosing connectivity) to the cellular system.

A value as close to zero as this one means a *very strong condition*. Hence, it is *very unlikely* that this situation will occur and the manager must have had a very good reason to specify a policy with this condition. Therefore, during the determinisation process this condition will have a high priority. Nevertheless, at runtime each TF value will be evaluated according to the user profile.

5.5.7 Policy enforcement

As the Policy Master moves through the selected path in the TFFST, it evaluates conditions and generates actions to be enforced by the *executors*. The executors can play the role of *subject* or *target* in the policy [24]. For example, the executor *HandoverExecutor* plays the role of target, and is responsible for executing methods to evaluate conditions and perform actions. The interface of this executor is shown.

```
public interface NetworkingContext extends Remote {
  public String getStatus() throws RemoteException;
  public void quit() throws RemoteException;
  /*Returns the phy connectivity status of specific nic*/
  public boolean isAttached(String nicID) throws
    java.rmi.RemoteException;
  /*Returns the link connectivity status of specific nic*/
  public boolean isLinked(String nicID) throws
    java.rmi.RemoteException;
  /*Returns the network connectivity status of specific nic*/
  public boolean isReceivedRA(String nicID) throws
    java.rmi.RemoteException;
  /*Sends a message to initiate selection process*/
  public void networkSelectionEvent(String nicID)
    throws java.rmi.RemoteException;
  /*Executes upward handover*/
  public void executeUpwardHandover(String nicID)
    throws java.rmi.RemoteException;
}
```

There are two types of actions: internal and external. The former (e.g. *networkSelectionEvent*) are performed within PROTON. The latter, for example *executeUpwardHandover*, occurs between PROTON and the mobile host (see Figure 5.10), and these are executed by the Control Interface that lies between the network layer and the mobility management sub-layer (i.e. MIPv6 module). The interface controls the incoming router advertisements from different access networks and it executes the corresponding actions (received from the Handover Executor, according to the Networking Context and the TFFST).

After receiving packets from the network (ipv6_rcv.c routine) including RAs, these are placed in the input queue (input.c) where they wait to be processed. In the case of RAs, these are sent to the neighbour discovery process (ndisc.c) to collect the appropriate information about available routers in the network. However, in the case shown in Figure 5.10, router advertisements are intercepted by the PROTON support system and sent to the corresponding module. PROTON filters some RAs (using IPv6tables filters) and re-inserts the appropriate ones into the stack to control the handover process.

Actions are associated with the different stages of the handover process. The Control Interface runs scripts based on IPv6tables [46], as a packet filtering tool, to build the appropriate rules that inhibit automatic handovers (by filtering router advertisements) and control them according to the Networking Context. These scripts also set timers considering context (e.g., mobile host velocity) and execute the most convenient handover mechanism.

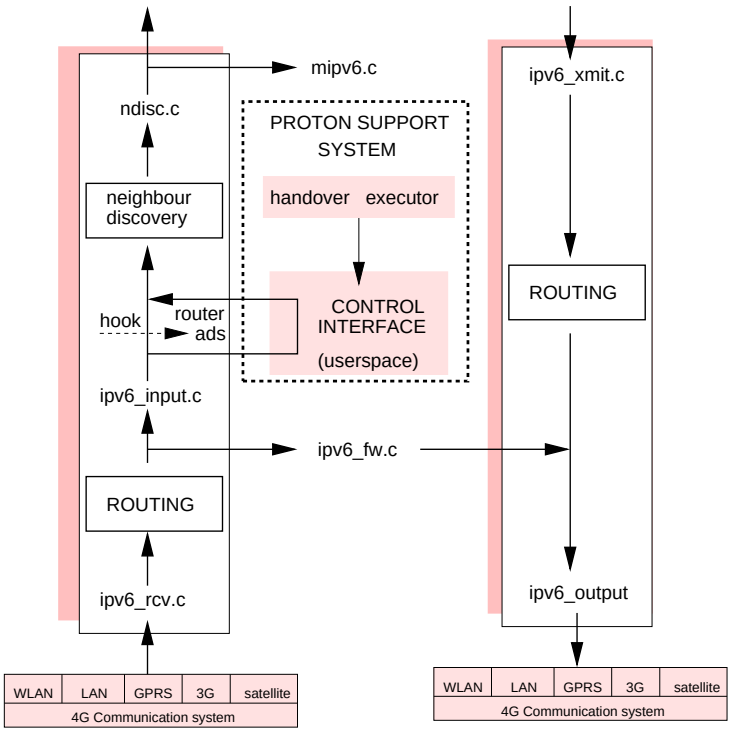


Figure 5.10: Handover Executor implementation.

5.6 Remarks

In this Chapter, PROTON was presented, a policy-based system to support multi-mode devices. Motivation behind PROTON stems from the fact that future devices will have multi-mode capability for connecting to different wireless networks. It is demonstrated how PROTON can address several issues of future networking, and how it can cope with complexity and dynamics intrinsic to future environments.

As far as I know, PROTON is the first policy-based system that attempts to offer complete mobility support for 4G mobile users. These heterogeneous environments pose challenges that remain open. Using a policy model based on TFFSTs, PROTON helps users in many decisions while hiding the added complexities.

It is also shown that concepts from autonomic computing can be applied to the design of novel solutions that brings us closer to the answer of open networking challenges such as seamless roaming among heterogeneous technologies.

This project consolidates the idea of building the policy evaluation model on the network, to enable devices with the capability to deal with complexities while keeping a powerful light-weight solution.

The Networking Context dataset presented is rich enough to allow well-informed decisions while roaming. However, the possibility of using a rich set in such a dynamic environment is empowered by the idea of having three levels of information according to the dynamics of data elements.

PROTON's architecture also reflects the concern of dealing with constraint devices, particularly in a constantly-changing environment. This is the main drive behind dividing PROTON into network- and host-side components. Every module that demands intensive computation work or high storage capacity is located in the network.

The mobile device deals, exclusively, with the evaluation of TFFSTs, a task that does not require much processing. Via the application of novel algorithms for the specification and translation of policies, conflict resolution, and TFFST distribution and installation, I have implemented a complete system that supports seamless roaming in upcoming pervasive networking environments, while representing a light-weight solution that is easy to deploy.

Powerful and more intelligent solutions are needed to support inter-networking in future communication systems. However, mobile devices and wireless environments will always exhibit strong limitations in terms of memory capacity, processing power, and stability. Hence, the prior resolution of conflicts and TFFST distribution is a very appropriate approach to overcome these constraints.

PROTON has demonstrated the potential of merging concepts of autonomic computing with the design and implementation of a policy-based system, together with a novel evaluation model and efficient conflict-resolution algorithms. The result: a solution that offers full mobility support, hides complexities, enables smart decision-making while roaming, and deals with the intrinsic constraints of 4G environments.

Chapter 6

Evaluation

The deployment of a policy-based system such as PROTON in constrained devices such as those used in mobile environments, represented a huge challenge with clear trade-offs between costs added into the network side, the link, and the end devices. Hence, a novel approach was proposed to extend the advantages of policy-based solutions beyond the wired world; it includes all the benefits of complex policy-based systems, but within a simple abstraction deployable in the wireless world.

To evaluate this work, its most relevant aspects were examined. First, using the LCE-CL testbed, a test case was simulated to demonstrate the usefulness of the system. PROTON's main goal is to improve users' connectivity and thus augment their productivity. It is important to ensure that the system takes appropriate decisions according to its purpose. Therefore, this was evaluated during the experiment; the outcome is reported in Section 6.1.

Second, the departure from a complex policy-based system, centralised in powerful computing devices, to a simple decision-making structure, distributable to mobile terminals, mandates low resource usage and overheads in terms of computational power, memory, and bandwidth. These were measured during the experiments, and the results are presented in Section 6.2.

Third, Section 6.3 discusses how feasible it is to deploy PROTON in a typical mobile phone examining the main constraints: the out-degree of TFFSTs and the number of transitions (which determine the TFFSTs' size) and the number of events.

Fourth, Section 6.4 shows the scalability of the system when increasing the three elements that bound the complexity of a policy: number of events, context fragments (conditions), and actions.

Finally, a taxonomy of mobility management systems derived from a comparison between PROTON and other similar systems is presented in Section 6.5, emphasising the characteristics that lead to a good solution for mobility management. In addition, the practical advantages of PROTON are summarised at the end of this chapter.

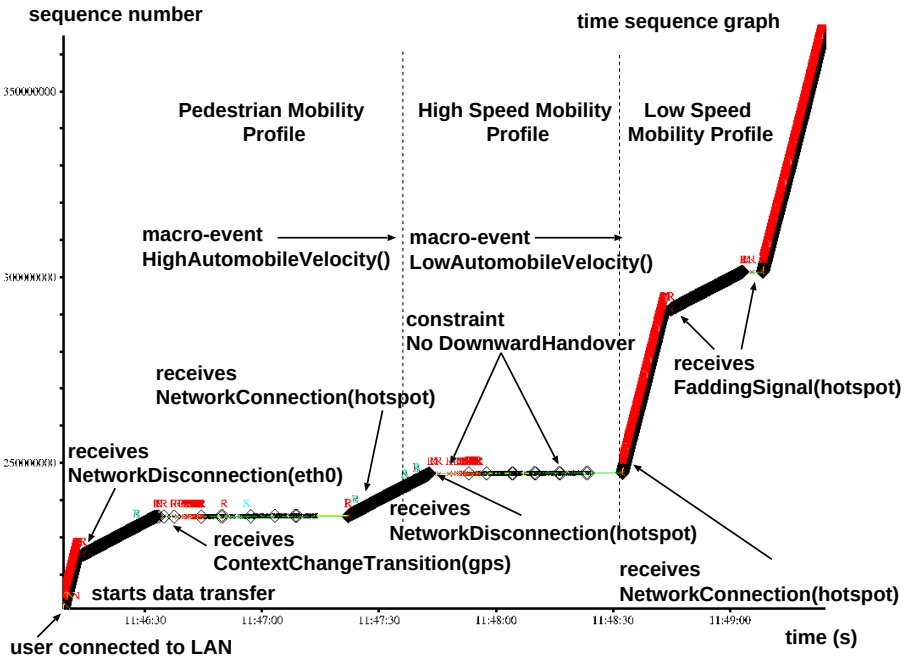


Figure 6.1: Testing PROTON in a simulated scenario.

6.1 Test case

Imagine Alice in her office using her PROTON-enabled laptop; she starts downloading a large amount of data that she needs for an important business lunch in London's financial district. She decides to leave her office immediately and continue downloading the data on-the-move. When Alice disconnects her laptop from the local network, the first PROTON event is generated: *NetworkDisconnection(eth0)*. This initiates the policy evaluation, the laptop connects to the Wi-Fi network available in the building and continues downloading the data without any disruptions to the connection.

When she leaves the building, the *PositionSentinel* cannot read the data from the indoor location system—e.g., the bat system [105]—hence generating the second event: *ContextChangeTransition(gps)*. The mobile device begins determining its location using its GPS receiver, and connects to an available cellular system (e.g., Vodafone's GSM/GPRS network). As Alice approaches her car, PROTON detects a nearby hotspot and generates the event *NetworkConnection(hotspot)* and evaluates the appropriate set of policies to decide how to use this broadband network.

She starts driving on the highway towards the city centre, and as she drives acceleration is detected by GPS-receiver. This is monitored by PROTON that generates a macro-event: *HighAutomobileVelocity()*—this triggers the selection of a different mobility profile that is loaded in the device. PROTON connects to the GPRS network when hotspot connectivity is lost, and while Alice is driving on the highway no downward handovers are allowed. This is because of the constraint *NoDownwardHandover* specified in the *HighAutomobileVelocity* mobility profile and built into the corresponding TFFST. Thus, she stays connected to the GPRS system.

She reduces speed as she reaches congested traffic areas in the city centre. This situation is detected by the *VelocitySentinel* and the macro-event *LowAutomobileVelocity()* is sent. Another mobility profile is loaded. The *NetworkConnection(hotspot)* event is received: PROTON evaluates the TFFST and decides to continue with the data transfer using an available hotspot. A few minutes later, the signal from the current access point starts fading and the event *FadingSignal(hotspot)* is generated.

PROTON changes its attachment point without any disruption; it uses the most appropriate execution method and the optimal initiation time, and adapts itself to the new QoS conditions exploiting its policy model and Networking Context dataset. Alice arrives at her final destination, and PROTON connects to the restaurant's hotspot to download the last few bits of data, while Alice starts her meeting.

PROTON's usefulness was tested through simulated cases such as the example described; these show how the system, autonomously, executes appropriate actions to enhance users' mobile experience. For example, Figure 6.1 shows how typical network activities would be efficiently performed in a real 4G system when using PROTON.

The system was installed in a multi-mode device (a Toshiba Satellite laptop) that can access multiple access networks, and a sequence of events was produced. This sequence triggered the evaluation of a subset of policies and generated the execution of the corresponding actions.

6.1.1 Test case inputs

For the test cases, several event sequences were generated. Due to the lack of experimental data to define the probability of each event occurring, these values were decided considering the following assumptions:

- Macro-events are "special" events generated when there is a significant variation in an specific (marked) context fragment (e.g., velocity). Thus, these events are less likely to occur than normal events.
- The network-dependent events (e.g., NetworkDisconnection(nic), FadingSignal(nic), etc.) are most likely to occur because of the huge variety of access networks expected to make up forthcoming 4G systems, and these represent the most probable of events.
- Changes in physical context such as position and velocity (collected dataset) are filtered (aggregated dataset) and despite the high frequency of variation, these changes do not generate events particularly often.

Based on these principles, three probability levels are used during the generation of event sequences: low, medium, and high. The probability of a network-dependent event is $P(E_{nd}) = 5/10$ (high probability).

For a macro-event $P(E_m) = 3/10$ (medium probability), and for any other event $P(E) = 2/10$ (low probability). Then, depending on the event generated, two different processes can be triggered: when a macro-event occurs the model distribution process is started and for any other event the policy evaluation process is performed.

6.1.2 Case discussion

During the above test case a sequence of events was generated and the events sent from the CML to the PML. For every situation, the event generation was successful and the PM evaluated the corresponding TFFST. To finish the process, the appropriate actions were triggered and executed by the Handover Executor. The model distribution process was also tested by generating a pair of macro-events. The first macro-event triggered the replacement of the current set of policies with a new TFFST, and the second caused the original TFFST to be reloaded—the "old" TFFST was already in the mobile device while the "new" transducer was requested from the network side.

The above scenario was simulated using the LCE-CL testbed and the results obtained from an analysis of the collected traces. The estimated average throughput shows that user experience improved when using PROTON. Moreover, when devices are not enhanced with this system, transparent mobility is not possible, terminals would be subject of black outs in connectivity, and users would not be capable of continuing their tasks without suffering network disconnections.

System performance is enhanced when using PROTON because mobile devices can exploit "ubiquitous" network access. In addition, seamless roaming between heterogeneous networks is enabled and the impact of mobility on the networking stack minimised.

6.2 Resource usage and overheads

To transform a typical network-centric policy-based system into a terminal-based support solution represents a clear trade off between costs in the network and in the end device. The complete knowledge about associated costs for each and every process is fundamental to evaluate the applicability, feasibility, and scalability of the system. This section calculates these costs and overheads in terms of computing resources and time. All the experiments were performed on the LCE-CL testbed for a complete description of the experimental platform and the computational equipment used, refer to Chapter 3.

The decision that determined the location of each process (network or device) was made in the early stages of the system's design. The main motivation behind such an architecture lies in the distinction between light policy-related processes such as TFFST evaluation and policy enforcement, and heavy processes such as TFFST creation, that do not need to be co-located in the mobile device.

In order to deploy a policy-based solution in a resource-limited environment such as a 4G system, it is necessary to shift processes with high computational cost from the mobile device to the network.

Figure 6.2 shows the location for each process and its computational costs. The architecture supports the dissociation of the individual processes: complexity on the network side and complexity involved in policy evaluation and enforcement on the host side. The former is of a considerably higher degree and involves policy selection and translation,

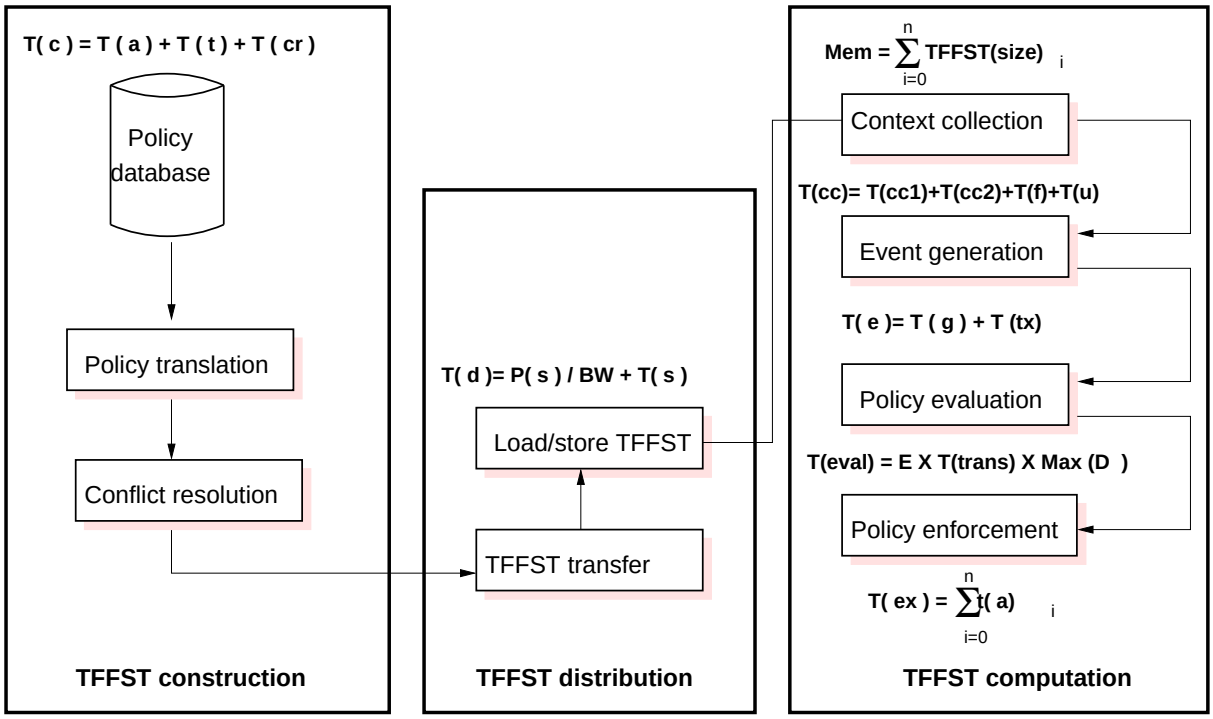


Figure 6.2: Resource usage.

and conflict resolution algorithms. As stated previously, these tasks are accomplished by powerful, centralised devices in the network, thus, they are not seen as an applicability issue.

6.2.1 TFFST construction

The construction of deployable abstractions of the policy model (TFFSTs) is done in the network side. The total processing time required to build one transducer (T_c) depends on the capacities of the computing device and includes: selecting and loading the policies from the central policy database (T_a), translating the policy set to an internal representation or transducer (T_t), and resolving potential conflicts between the chosen policies (T_{cr}). The application of these algorithms has a high computational cost and it is done *a priori*, in the network, to the TFFST distribution.

6.2.2 TFFST distribution

The first deployment issue arises from the TFFST distribution process. After the completion of the TFFST construction process, the created transducers need to be transferred to the mobile terminals. The cost of this process is twofold. The first overhead is caused by the time needed to transfer the TFFSTs to the MN using a particular access technology (P_s/BW, where P_s represents policy size and BW access technology bandwidth). Additionally, the TFFST management in the mobile terminal causes the second overhead (T_s).

Description	$P_s[kB]$	BW[kb/s]	$T_s[s]$ (Mem%)	$T_d[s]$
Nokia 3300 connected to a GPRS access network, and storing the mobility profiles in internal memory (4.5 MB).	466	52.8	0(10%)	70.6
Nokia 3300 connected to a WLAN access network, and storing the mobility profiles in internal memory (4.5 MB).	466	600	0(10%)	6.2
Nokia 6670 connected to a GPRS access network, and storing the mobility profiles in internal memory (8M B).	466	52.8	0(6%)	70.6
Nokia 3300 connected to a GPRS access network, and storing the mobility profiles in external memory (64M B).	466	52.8	2.1(1%)	72.7

Table 6.1: Aggressive distribution policy.

The main constraints in this process are: limited memory in mobile devices, limited bandwidth in wireless technologies, and limited time to operate the system. After observing the costs and constraints in the TFFST distribution, three different strategies are proposed to control the process.

One strategy could be to download all the appropriate TFFSTs when the mobile device is powered on. For the scenario under consideration it means transferring 466 KB (see Table 6.1), and depending on the technology used it could take 6 s if the MN is connected to a WLAN, and up to 70 s for the case of a "4+1" GPRS link. [1] In addition, a delay of approximately 3 s needs to be assumed to manage the TFFSTs in the mobile device (load and store the transducers). [2]

This strategy may be convenient when storage space is not an issue in the mobile device. The main disadvantages of this technique are the intensive use of the link during initialisation and the delay in the beginning to start operating. From the additional examples shown in Table 6.1 it is evident that this aggressive strategy would effectively work when the mobile device has enough memory and it is connected to a fast access network.

However, this is not always the case, as some mobile devices have less memory and mobility management support is only one of many activities demanding storage space in modern phones. Hence, a conservative strategy could be applied in these situations; the MN downloads TFFSTs on-demand and only in response to a macro-event.

This second strategy does not depend on the total number of TFFSTs (total size). However, it is limited by the frequency with which macro-events occur and the individual TFFST's size. Table 6.2 shows the worst case: a large-size TFFST(pedestrian mobility profile) downloaded using a slow access network(GPRS).

The conservative technique could be seen as appropriate for small-memory devices, very common in mobile environments. However there would be delays of approximately 5s and up to 15 seconds every time that a macro-event occurs, during which the MN would not be able to function properly. Therefore, the downloading of TFFSTs should be minimised by modifying the conservative strategy.

[1]The average effective bandwidth considered for this analysis was the AirCard750 "4+1" phonecard, resulting in 52.8 kb/s for the downlink connection.

[2]For a Nokia mobile phone, the read/write latency to external memory is 1400 b/s.

Description	$P_s[kB]$	BW[kb/s]	$T_s[s]$ (Mem%)	$T_d[s]$
Nokia 3300 connected to a GPRS access network, and storing the mobility profiles in internal memory (4.5 MB).	375 KB(pedestrian)	52.8	0(8%)	56.8
Nokia 6670 connected to a GPRS access network, and storing the mobility profiles in internal memory (8 MB).	83 KB(automobile)	52.8	0(2%)	12.6
Nokia 3300 connected to a WLAN access network, and storing the mobility profiles in internal memory (4.5 MB).	466 KB(pedestrian)	600	0(8%)	5.0
Nokia 3300 connected to a GPRS access network, and storing the mobility profiles in external memory (64 MB).	4 KB(high_speed)	52.8	0(0%)	0.6

Table 6.2: Conservative distribution policy.

A predictive version could be the third option; the MN downloads the correspondent TFFST during operation. However, some modifications are done to the plain technique: certain memory space is transformed into a TFFST cache; it is not possible to store all the TFFSTs, however some TFFST can be kept in the mobile device for a useful period of time.

Memory capacity is limited and the mobile device needs to perform "smart" TFFSTs swapping to minimise downloading and optimise the use of memory space. The device could predict the following TFFST by using data from its current context. For example, if the current mobility profile is *automobile*, the next profile is more likely to be the *pedestrian* than the *high speed*—this last profile is used when travelling in trains. This information could be used to enhance the replacement of TFFSTs in the device. If the profile is not in the device, then the MN could remain using the "old" profile until it receives the "new" profile, minimising the impact of downloading the TFFST; the terminal would not attain its optimal performance, but it would not stop operating.

6.2.3 TFFST computation

There are some processes that need to run on the mobile device; this is the case for *context collection*. Gathering context data is mainly the responsibility of two types of components: sentinels and retrievers. The main overhead in the solution is caused by sentinels and retrievers collecting context data. Hence, the CML was divided into three sub-layers to optimise cost reductions; despite the optimisations, context collection remains the most computationally intensive task executed in the end device.

In this scope, the context collection process includes: context data polling (T_{cc1}), information retrieval (T_{cc2}), application of simple filters to the collected data (T_f), and updating of the Networking Context dataset (T_u).

Sentinels and retrievers are very small objects, which perform an extremely simple task. Each of them has approximately 150 lines of Java code and an average execution time of between 10 ms and 15 ms (Table 6.3). The processing time to execute the ten Networking Context components is around 110 milliseconds (T_{cc1}), and the information retrieval process takes approximately 80 ms (T_{cc2}); the five retrievers perform tasks in response to a sentinel's request.

Context data	T_{cc}[ms]	(polling)T_{cc1}[ms]
Layer 1 connectivity	27	5
Signal strength	32	14
Layer 2 connectivity	29	5
Layer 3 connectivity	38	14
Handover latency	32	10
Logical position	43	16
Physical position	73	19
Velocity	9	4
Direction	22	4
Network traffic	45	17
User profile	5	N/A
Ongoing applications	14	N/A
Network characteristics	22	N/A
Application characteristics	16	N/A
Network structure	18	N/A

Table 6.3: Processing time for three-level and polling tasks.

Additionally, several filters are applied to the collected data to form the aggregated dataset (T_f) and the Networking Context dataset is updated; these two tasks can take up to 200 ms. The total time taken for the context collection process to complete is between 250 ms and 300 ms (left column in Figure 6.3, T_{cc}). It is critical for performance to define how often the CML updates the Networking Context, and this frequency also depends on the velocity of the terminal itself.

As updating the Networking Context dataset takes between 200 ms and 300 ms, it cannot be refreshed at a frequency higher than 5Hz. This is enough for most of the mobility scenarios (e.g., pedestrians, cars, trains, and aeroplanes) under the assumption that as the MN velocity increases, less context data is relevant.

Figure 6.3 shows the distances that a mobile would have travelled between two updates, varying velocity and update frequency. For example, when a pedestrian carrying a laptop moves, context changes will be detected every 4 m (for an update frequency of 0.25 Hz). A MN travelling at 120 km/h, and spending 200 ms collecting data, would observe context changes approximately every 33 m.

The complete execution of these tasks can consume a lot of processing time in such resource constrained devices, and it will not be possible to have high update frequencies, therefore high-speed nodes would miss important context changes. However, as the velocity increases the amount of relevant context data decreases—fewer sentinels and retrievers are examined—and small variations in the context fragments lose relevance for the policy model (fewer events are generated).

Along with context collection, there are other processes taking place on the host: event generation, policy evaluation, and policy execution. As explained previously, in Section 5.3, not all the context changes generate events, in order to reduce processing load. However, if an event is generated, then the process initiates when the correspondent sentinel or retriever constructs a valid message [32], which is sent to the event-consumer, i.e. the Policy Master. This whole process takes around 150 ms, from the construction of

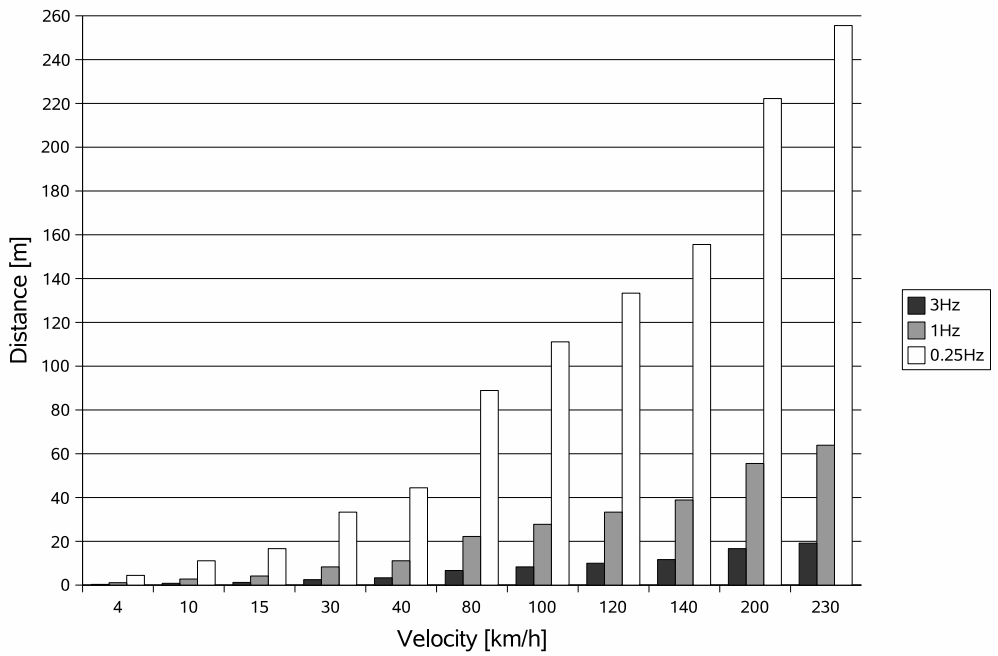


Figure 6.3: Travelled distances between updates.

Mobility profile	Max. out-degree	Evaluation time (ms)
Pedestrian	3165	396
Automobile	316	108
High Automobile	39	24
High Speed	39	24

Table 6.4: Run-time performance for policy evaluation.

the message to the instant it is consumed by the Policy Master.

Following the reception of an event, the Policy Master initiates the policy evaluation process. The evaluation time is linear with input length and it does not depend on the size of the transducer. However, the final evaluation time does depend on the maximum number of outgoing transitions belonging to any state in the transducer (transducer's out-degree. [3])

In each stage of the deterministic transducer evaluation, the valid transition must be selected from among all the transitions associated with the state. Thus, the maximum evaluation time for a set of events is: $E \times T_{trans} \times Max_i\,(D_i)$, where E is the number of events inputed, T_{trans} is the time of evaluating one transition's TF and D_i the number of outgoing transitions for the state i.

Table 6.4 shows the evaluation time for an input of two events using each of the mobility profiles. The out-degree of the mobility-profile's TFFST is also shown.

The outcome of the policy evaluation process (actions) must occur during the handover period. Table 4.1 shows latencies measurements for the most common inter-network

[3]Maximum out-degree among all states in the TFFST. By definition, the number of outward directed graph edges from a given graph vertex in a directed graph [60].

handovers. The policy enforcement process and corresponding actions should take place within this interval. According to the experiments, the Handover Executor needs to perform an average of five actions to control the complete roaming process. The possible actions include: IPv6tables rule instantiation, modifications to the routing table, network interface control, and triggering other policies.

6.3 Feasibility of deployment

A possible disadvantage, during deployment, of using TFFSTs is the high complexity of their algorithms and the size of the final transducer. By utilising heuristics in PROTON strategies, the internal TFFST model size and complexity have been kept within acceptable limits.

As described in Section 5.3, not every context fragment matters in every situation. In PROTON, important fragments are selected according to *mobility profiles*. Hence, a transducer is built for each profile, reducing the maximum size of each TFFST, avoiding processing overhead and minimising storage space in the mobile device.

The TFFST selection process, on the host side, can be seen as hot-reprogramming of the device to optimise its behaviour dependent on the scenario. The current version implements four different mobility profiles according to the *velocity context fragment* that produces the following macro-events: *PedestrianVelocity*, *LowAutomobileVelocity*, *HighAutomobileVelocity*, and *HighSpeedVelocity*. Every time that one of these macro-events is generated, the corresponding mobility profile is loaded.

Experiments in different scenarios showed that the number of transitions for each mobility profile's TFFST was dependent on the quantity of relevant context fragments, possible events, and applied constraints. These variations correspond to the following facts:

- At lower velocities more context fragments can be considered in taking decisions, increasing the number of transitions.
- At the highest velocities more constraints can be applied to the policy model, reducing the number of transitions.
- At higher velocities, fewer events are relevant to making decisions, decreasing the number of transitions.

The number of transitions required for TFFSTs representing per-mobility-profile policies was significantly smaller than per-device policies, because the conditional part of the policy is more specific (fewer context fragments are considered) and the number of combinations is lower.

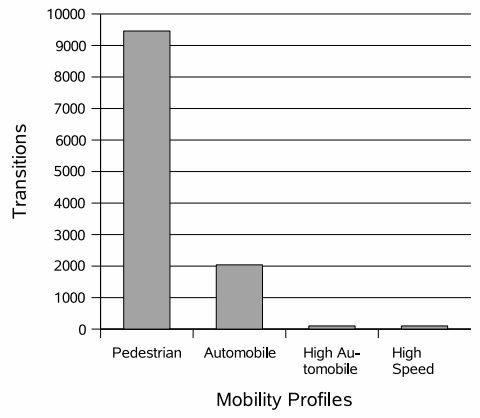


Figure 6.4: Maximum number of transition for each mobility profile.

The maximum size of a single, per-mobility-profile, TFFST is reduced by several constraints [28]. Firstly, in FST terminology, the size of an input string is limited by not allowing symbols, i.e. events, to repeat. Thus a maximum number of strings for an alphabet of m symbols is given by:

$$\sum_{i=1}^{m} C_i^m$$

where C_i^m represents all the possible combinations for a particular set of conditions. As the maximum length of each of the strings is m, an upper bound on the number of transitions in a TFFST is given by:

$$m \times \sum_{i=1}^{m} C_i^m$$

For example, the full size of a moderately simple policy such as the TFFST presented in Figure 5.6 was 15 transitions and 8 states. Figure 6.4 shows the maximum number of transitions for each mobility profile, which are formed by multiple policies. The experiments showed that the number of transitions for pedestrian velocities is much higher (around 9000 transitions per TFFST) compared to other mobility profiles: for automobile velocity the number is around 2000 and approximately 100 for high speed profiles (Figure 6.4). However, these numbers do not increase computational complexity, only the size of the TFFST.

Computation complexity of evaluation in deterministic FSTs, and thus TFFSTs, does not depend on the size of the transducer but on the length of the input (m). Hence, in terms of computing cost and processing time the most relevant dimension of a TFFST is its *out-degree*, as a valid transition has to be selected out of all available transitions. The out-degree is the maximum number of outgoing transitions belonging to one particular state of the transducer. Figure 6.5 shows the maximum out-degree for different number of conditions, which is the same as the order of the determinisation algorithm. An upper bound on the magnitude of the cost of evaluating a node i.e. state, is given by:

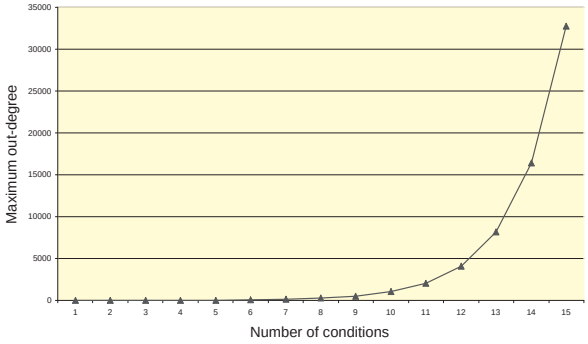


Figure 6.5: Upper bound on the out-degree for different numbers of conditions.

$$O\left(\sum_{i=1}^{n} C_i^n\right)$$

where n is the number of conditions that may be fulfilled at the same time, i.e. in a single event. Conditions as used in this context correspond to "when" statements in Ponder. This maximum bound obeys the fact that after applying the determinisation algorithm every exclusive conditions' combination must be computed.

The lower bound is simply linear with the number of conditions:

$$O\left(n\right)$$

The number of conditions was significantly reduced by grouping the policies in mobility profiles and defining macro-events. Other than any static conflicts, it is expected that TFFSTs will be highly deterministic, inherently bringing the order of their states close to $O(n)$ in the average case.

TFFST evaluation complexity will be impacted by complexity of the tautness functions used to discriminate between actions, which are deployment-specific and should be tailored to specific resource capacities of client devices.

Also relevant to the evaluation of the feasibility of deployment is the memory capacity needed to allocate TFFSTs in mobile devices. Transducers' transitions are implemented as simple objects with an average size of 40 bytes each. A TFFST can be seen as an array of transitions that in the worst scenario (pedestrian profile) would require 332 KB of memory space. Table 6.5 shows the approximate size for each of the mobility profiles built for the evaluation of the system.

This analysis discusses two important factors that affect PROTON's applicability in mobile environments: out-degree and transitions. It is better to keep these two dimensions as low as possible. Not all context fragments, i.e. conditions, are relevant at the same time, thus the number of conditions can well be well below a critical number that increases the TFFST out-degree beyond mobile device capacities.

Profile	Transitions	Size[KB]
Pedestrian	9000	375
Automobile	2000	83
High automobile	100	4
High speed	100	4

Table 6.5: Size of each mobility profile.

The state with the maximum out-degree may also be split into more than one state, thus the these subsidiary states would have a lower out-degree. This method represents a clear trade off between input string length and out-degree. The out-degree may be smaller than before the event replication, however the number of transitions can increase significantly.

Finally, the manner in which macro-events reduce TFFST complexity is twofold. First, these special events force the application of constraints to lower the number of valid policies in a particular context (mobility profile). Second, mobility profiles allow the creation of more specific policies and, as a consequence, the reduction in the number of conditions for each rule.

6.4 Scalability

The scalability of the system was analysed taking into consideration the costs and limitations mentioned in sections 6.2 and 6.3, respectively. It was observed in two operational scenarios: when an average PROTON-enabled terminal (Nokia 3300) roams from a cellular network (GPRS) to a wireless LAN (hotspot) and vice versa. The policy model can be seen as a group of condition-action rules triggered by events. Hence, to evaluate scalability the varying dimensions are: the number of events, conditions, and actions.

PROTON is a mobile-based solution, thus scalability in terms of the number of clients is not considered a problem; each terminal functions independently of others. Scalability issues arise when the MN's capacities are stressed by changes in size or volume in either of the dimensions mentioned in the previous paragraph in order to meet a user need. The main problem is that the time required by the device—considering the constraint capacities—must be acceptable for the dynamics of the environment. For example, scalability depends on how many decisions can be supported by PROTON in a 100 ms interval or how many operations can be triggered by PROTON in a 200 ms time-frame, considering a certain number of events and conditions.

- **Events:** Events affect the policy evaluation process, not by augmenting the TFFST's out-degree, but by increasing the total time needed for the evaluation. Thus, the number of events that occur within an epoch make the evaluation process last for longer by adding more steps, including other events, to the longest path of the solution. PROTON has been tested with a two-event epoch, two conditions, and one action per policy. However, following the values shown in Section 6.2, it is possible to scale the number of events without much impact on the size or evaluation time of a particular TFFST. The number of events is really limited by the duration of the epoch, which should be defined in relation to the MN's velocity.

- **Conditions:** The number of conditions directly affects the out-degree of the TFFST, which in fact gives the maximum bound for the evaluation time. If the conditional part of the policy has eight or less elements, the evaluation time stays below 100 ms. Hence, for real-time applications this decision-making delay would be tolerable. For the opposite case, where the number of conditions exceeds eight, the out-degree is over 255 transitions, therefore the evaluation time goes beyond real-time requirements. For example if the conditional part of the policy has 12 elements, the TFFST out-degree is 4095 transitions, and the navigation of the transducer would take approximately 400 ms.

- **Actions:** The duration of an action varies depending on its type and the available computing resources. However, to address the scalability of the system in this sense, some example actions are mentioned. In a typical scenario the Handover Executor may perform between five and eight actions to manage mobility; most of these related to the network stack. Policy enforcement usually includes the execution of isolated actions like the activation of IPv6tables filtering rules, or the execution of scripts (i.e. sets of actions) such as the example shown in Section 5.5.7, which have an average execution time between 60 ms and 100 ms. The number of actions and their type affects the policy enforcement process, however, not every action needs to be executed in real time or with the same priority. Experiments have shown that it is feasible to manage mobility and cope with the dynamics of the environment, and that the number of actions can scale to satisfy potential needs of 4G users.

Another scalability issue is the TFFST size. If the terminal does not have enough memory, then the transducers need to be downloaded from the network during operation (conservative strategy). The biggest TFFST size obtained during the evaluation was 400 KB, which can be downloaded in less than 60 s using a 52.8 kb/s link (GPRS). For bigger TFFST sizes, optimisation methods can be applied such as TFFST decomposition. Furthermore, if memory space is not limited, then the aggressive strategy can be used and downloading would occur only during the terminal's initialisation phase. Thus, TFFST size can scale when more complex policy models need to be deployed in order to fulfil user needs.

6.5 Qualitative analysis

A wide body of research in mobile computing has resulted in a number of solutions to support mobility management. This section presents a taxonomy that derives from the comparison between PROTON and some of these approaches.

In this discussion the following aspects are examined: scheme used by the solution and its compatibility with current standards, system's context awareness, quality of the support for decision-making, and other fundamental tasks such as network selection and handover initiation and execution.

Every mobility management system should aim to fulfil as many of these aspects as possible. Due to the popularity of IP-based services, the completeness of the solution must be inherently compatible with this protocol. Additionally, a complete solution should be pro-active and use context data to enhance decision-making and mobility management support.

Finally, in any design, applicability and deployment constraints should be considered. In mobile environments, these limitations are closely related to the location of the entities involved in the implementation—the mobile terminal, the network, or both. Following these criteria, PROTON is compared to some similar solutions. Table 6.5 shows the taxonomy obtained from this analysis.

Murray et al. [99] initiated the development of a network-assisted policy management system for hybrid networks, not specially focused on 4G systems in the beginning. However, this work continued as part of the CONTEXT IST project [47], in which they describe a context-aware system to control handover initiation in next generation networks. The main difference between PROTON and these proposals is that they are network-based and they affect the network infrastructure. Furthermore, as they require network data, they are not as dynamic as the mobile-based approach—in which decisions are made just considering immediate context.

An extension of this work was published by Yang et al. [110], which introduces mobile agents to enable service delivery between the network and its clients. The deployment of mobile agents in a wireless environment poses scalability problems due to an active exchange of code that can overload the network.

In [4] a more similar solution to PROTON is presented, Fikouras et al. describe POLIMAND, a policy-based MIP handover decision method. The policy model in POLIMAND considers only link layer data (essentially signal strength), which prevents the solution from offering full handover support. It does not assist users during the network selection or adaptation processes, mainly because of the lack of inputs from other layers or even physical context.

Other proposals such as Murray et al., [71] Makela et al., [57], and Chan et al., [14] explore the use of other decision methods. A policy-based approach could be sufficient to handle complexities in 4G systems. Consequently, other schemes such as fuzzy logic and neural networks are over complex and add undesired overhead to the decision process.

Most situations in the handover process can be modelled using a linear system that receives precise inputs—in this scenario the use of fuzzy logic becomes excessive. Finally, the dynamics of 4G environments demand agile and appropriate decision-making, not necessarily the optimal solution. Thus, complex decision models are not always the best approach to enable mobility support in 4G networks.

PROTON uses a policy-based decision scheme following the IETF PCIM specification [69]. The policy evaluation model builds on the concept of FSTs and it is intended to provide both a fast evaluation model and effective conflict resolution algorithms. Algorithms developed for natural language processing [98] were used and extended. However, these methods [6] were adapted to mimic strategies that emerged from previous research on static conflicts [56] and dynamic conflicts [16]. Additionally, a new metric called the Tautness Function is used to abstract away from technology-dependent conditions and context variables [6].

The most common method of resolving conflicts is to explicitly assign priorities to policies and decide on the one with higher value. A more complex method is using a *goal-oriented strategy*, which consists of assigning priorities to every possible system state and moving to the state with a higher priority [51].

Authors	Scheme	IP-based	Context	Decision	Initiation	Selec.	Exec.	Adapt.
H. Wang et al.	Policy	YES	NO	terminal	NO	YES	YES	NO
J. Makela et al.	Neural nets.	NO	NO	terminal	YES	NO	NO	NO
P. Chan et al.	Fuzzy logic	NO	NO	terminal	YES	YES	NO	NO
K. Jean et al.	Policy	NO	YES	network	YES	NO	NO	NO
K. Yang et al.	Policy	NO	YES	network	YES	NO	NO	NO
N. Fikouras et al.	Policy	YES	NO	terminal	YES	NO	YES	NO
K. Murray et al.	Policy	NO	NO	terminal	NO	YES	NO	NO
K. Murray et al.	Fuzzy logic	NO	NO	terminal	YES	NO	NO	NO
P. Vidales et al.	Policy	YES	YES	terminal	YES	YES	YES	YES

Table 6.6: Qualitative analysis.

The proposed conflict resolution module prioritises conditions automatically. This strategy resolves policy conflicts in a simple manner and is powerful enough to solve most situations without human intervention.

Dunlop et al. [31] use *conflicts databases*. Their work is related to PROTON in the sense that both solutions consider every possible conflict beforehand. They only detect conflicts while PROTON has algorithms to perform detection and resolution. Furthermore, conflicts databases can prove to be unsuitable for mobile environments whereas Finite State Machines represent a light-weight solution that is more feasible to deploy in mobile devices.

In summary, PROTON differs from previous schemes in the following concepts:

- PROTON is designed with highly-dynamic and complex 4G environments in mind. It is a mobile-based solution, keeping most of the computational load in the network.
- PROTON is a a context-aware system that considers not only network conditions but also other context fragments (e.g., physical environment and user preferences), which are equally important to properly solving handover-related problems.
- PROTON offers complete mobility support, which is a key advantage in 4G mobile systems. Decisions before, during, and after handover execution will improve mobile users' experience.
- PROTON is entirely mobile-based; however, network knowledge can be transmitted to the wireless device through the model distribution process.
- PROTON exploits a novel policy model, based on TFFSTs, to adapt policy-based systems to resource-constraint environments.

6.5.1 System comparison

This Section discusses how PROTON would function in various potential scenarios, contrasting the system with similar solutions. However, previous approaches considered only the very limited capacities of old mobile devices and less complex environments. Therefore benchmarking such solutions that were designed within different operational frameworks is not possible.

Intense roaming and diversity: PROTON was designed to offer full mobility support; it assists users in the whole roaming process and not only during one particular task. In a mobile world where continuity in services is not possible unless MNs are capable of roaming between different wireless technologies; the system surpasses all other solutions. In previous approaches the lack of context data leads to the specification of poor policies. To maintain a simple solution some systems such as POLIMAND [4], have a very limited policy set—based on insufficient data—that turns to be inefficient in handling 4G systems. PROTON has a complex policy model based on a rich networking context that can potentially manage every situation. Nevertheless, it has the disadvantage of adding much more functional overhead compared to POLIMAND. Overhead can be reduced by introducing the concept of TFFSTs, but a highly optimised implementation is needed to overcome this issue.

Moderate roaming and diversity: If roaming is not taking place frequently, then the usefulness of the system lessens. However, the overhead of collecting context information continues and it becomes a problem. When there are not various technologies available, a complex mobility support system like PROTON may be excessive. Roaming and heterogeneity are two prerequisites for PROTON to be fully effective. Otherwise, in environments where these conditions are not met, lighter solutions such as [104] may be more convenient.

Soft roaming: PROTON can have a particular set of TFFSTs for less demanding environments, where roaming does not usually occur or horizontal handovers are more likely to take place. The system may be adjusted by using less complex transducers and reducing the quantity of context data collected. However, there are processes that need to be executed and that overhead cannot be avoided. For this kind of environment, more traditional link-layer based mechanisms [74] are sufficiently smart and more efficient than PROTON. There is no real niche for a broader system, which only increases the operational cost.

No inter-domain roaming: In scenarios where inter-domain mobility does not occur (for example, cellular systems) or this is less frequent than in other environments, approaches such as [9] and [88] are much more appropriate than PROTON. In this case, small modifications to MIPv6 can solve most of the mobility issues, without the deployment of a completely additional system on top of it.

There is not a unique solution for every possible scenario and each environment poses specific challenges. However, PROTON has proved to be flexible enough to cover many of the possible scenarios, maybe not showing the same level of usefulness and efficiency, but coping with the different complexities. The idea of building TFFSTs based on a subset of policies, constraints, and conditions leads to a highly flexible system.

Chapter 7

Conclusion

In this dissertation I have introduced a number of methods to facilitate transparent roaming between two independent and heterogeneous wireless access technologies. Experimental evaluations demonstrate that my design enables roaming in integrated networks, reduces latency, and retains transparency by hiding complexities from users. In this chapter, I summarise my contributions and describe potential avenues for future research.

7.1 Summary

In Chapter 1, I began by motivating the need for new solutions to support seamless mobility in future integrated disparate networks. I then presented my thesis that the tools and mechanisms in this dissertation allow transparent migration of data flows between two access points belonging to independent heterogeneous technologies. Furthermore, it has been proved that these practical mechanisms can be used in forthcoming 4G systems to improve user experience.

In Chapter 2, I discussed terminology and related work in the fields of mobile computing and terminal mobility. The existing protocols described in this chapter are still inadequate to handle macro-mobility between two disparate systems. Moreover, previous mobility support systems either do not focus on handling networking resources in 4G systems, or do not handle all the inherent complexities of future mobile environments.

In Chapter 3, I discussed the issues encountered when implementing a testbed that integrates independent disparate technologies (e.g., IEEE 802.11a/g, GPRS, and Ethernet). These issues have been frequently underestimated in previous projects. However, they must be solved to produce useful architectures that support transparent mobility and ubiquitous access.

In Chapter 4, I presented a suite of optimisations for the current specification of Mobile IP version 6. These modifications were tested using the architecture presented in Chapter 3, and the results demonstrated an improvement in handover delays.

In Chapter 5, I presented a further major contribution of this dissertation: a novel solution to support transparent mobility in 4G systems named PROTON. This system makes it possible to minimise disruptions when the mobile node changes its attachment point to the Internet, and offers ubiquitous access via an integrated platform, while coping with drastic variations in link-layer characteristics.

Finally, in Chapter 6, I described how I evaluated PROTON using the testbed as a simulation environment. The system feasibility and deployment is discussed, which is important when dealing with mobile environments and the constraints of wireless devices. Costs and overheads were measured, including scalability and run-time performance, by running PROTON in a constrained environment. These results demonstrate the practicality of the solution. This chapter ended with a qualitative evaluation of PROTON, comparing it with similar approaches.

In conclusion, my thesis—that the transparent migration of data flows between two access points belonging to independent heterogeneous technologies is achievable—is justified as follows. I presented the design and deployment of an integration architecture that enables vertical handovers between the most popular wireless technologies. This convergence poses challenging research issues such as the two topics that I tackled: (a) how to minimise mobility disruptions when roaming and (b) how to manage transparency in the operation of future mobile devices. To address the former, a suite of Mobile IPv6 optimisation mechanisms is proposed and thoroughly evaluated. I therefore demonstrated that it is possible to minimise the latency during heterogeneous handovers, enabling transparent mobility. To address the latter, I described a software solution to enable complete mobility support for nomadic users in 4G systems. The performance results show that the proposed design surpasses today's insufficient policy-based approaches, handling the complexities whilst maintaining scalability, performance, and transparency. Therefore, using the solutions that I have presented and evaluated in this dissertation, it is practical to deploy a suitable architecture that supports seamless mobility in forthcoming 4G systems.

7.2 Future research

This dissertation has raised various issues that have yet to be addressed. Several of the most interesting problems are discussed below, as well as potential avenues for further research.

- Many groups are working on the specification of new standards to cover particular needs in the wireless world and new technologies are being added to the spectrum. It is challenging to design an integration architecture that could be extended to incorporate more access technologies. This architecture should consider the conditions posed by the addition of different networks, and it should have the appropriate mechanism to support them. The LCE-CL testbed can be regarded as an initial stage of a ubiquitous networking platform.

- The optimisations included in Chapter 4 suggest that a unified network-layer solution, based on Mobile IPv6, would need further support in order to sufficiently hide the impact of handovers and mobility on overall performance. An open issue is how to add QoS support to mobility management protocols, rather than keeping these features (QoS and mobility) dissociated. To enable seamless mobility, it is important to secure QoS when the MN roams to a new access router. Chaskar and Koodli [15] described a solution to perform QoS signalling along the new path in

the network, when an MN, using MIPv6, acquires a new CoA. This solution adds overhead to the mobility management protocol. However, it minimises the mobility impact by assuring a particular level of service across heterogeneous networks. Further investigation of cross-layer solutions and deeper analysis of the interaction between the layers of the network stack is required on this topic.

- Although the period of time when the mobile node is unable to receive packets both due to link-layer switching and the network-layer handover has an important impact on performance, this delay is not the only problem when dealing with mobility between disparate technologies. Cottingham and I have made further progress in this area: we have studied how the length of time a mobile terminal needs to adapt to the new link—coming from an extremely disparate technology, in terms of link characteristics—affects overall performance [21].

- An interesting aspect of PROTON is the concept of deploying TFFSTs considering other aspects such as the operator's business model, strategies, or even mobile device characteristics, and not only mobility aspects as evaluated in Chapter 5. The behaviour of the mobile device is driven by the TFFST evaluation, thus by implementing different automata we can explore more complex system responses. The PROTON prototype is an early implementation and there are many performance issues that need to be solved. For example, the communication protocol used to install TFFSTs in the mobile device needs to be improved. The internal representation of TFFSTs can be smaller, and faster evaluation algorithms can be deployed.

- PROTON supports the aggregation of new sets of policies to assist users in other tasks. For example, we have considered the implementation of a policy set for security in such ubiquitous environments [29, 28]. Policies for data adaptation [77] are essential to achieve seamless roaming; these are interesting research topics that needs further work. I propose the inclusion of autonomic computing concepts into the design of the software to support upcoming networking. The deployment of a complete autonomic solution for mobile communications stands as a challenge in the current research, and it represents a main avenue for future research.

Mobile communications are rapidly becoming the main method of personal remote interaction in society. Network operators talk about the vision "any service, via any device, over any network". However, the prerequisites to this vision are clearly not in place. Despite the progress in wireless networks genealogy—2G, 2.5G, and 3G—it has not been possible to enable full-transparent networking for nomadic users. The so-called Fourth Generation, as described in this dissertation, faces and proposes solutions for many of the problems in previous generations. Forthcoming communication systems are in a better position to offer seamless access to a broad variety of services. Important steps have been taken: an integrated architecture that overcomes the limitation in isolated wireless networks, posed by the trade-off between coverage and bandwidth; an improved mobility management protocol that eliminates disruptions in the connection, and an autonomic system to offer the appropriate assistance to users in such a complex model. Nevertheless, these contributions represent a small step towards the fundamental goal of ubiquitous networking.

Appendix A

Glossary

A.1 Definition of terms and concepts

In order to understand the content of this dissertation, it is pertinent to define the most important concepts related to mobile computing and networking. With this purpose, the basic terminology is included in this appendix. The presented definitions are based on the following sources: IETF-RFC Mobility support for IPv6 [48], IEEE 802.20 Working Group documents [45], IETF Network Working Group Internet Drafts [58] and [91], and the RFC [107].

Access router
An access network router residing on the edge of an access network and connected to one or more access points. An access router offers IP connectivity to mobile hosts. The access router may include intelligence beyond a simple forwarding service offered by ordinary IP routers.

Base station
Also called *access point*, it is the point of attachment of a mobile node to the Internet.

Binding
The association of the home address of a mobile node with the care-of address of that mobile node, along with the remaining lifetime of that association.

Break-before-make handover
During a break-before-make handover the mobile host does not communicate simultaneously with the old and the new access router.

Care-of address
A unicast routable address associated with a mobile node visiting a foreign link; the subnet of this IP address is a foreign subnet prefix.

Context-aware handover
A handover that is governed by a certain specific requirement to be fulfilled while handing the connection between two access routers.

Correspondent node
A peer node with which a mobile node is communicating.

Eager cell switching
Node should switch to the new access router as early as possible, or as soon as thee mobile node receives a router advertisement from the new access router.

Fast handover
A handover that aims primarily to minimise delay, with no explicit interest in packet loss.

Foreign network prefix
Any IP subnet prefix other than the mobile node's home subnet.

Foreign link
Any link other than the home link.

Handover
The act of changing the attachment point of a mobile node, switching the communications from one access point to another access point, also know as *handoff*.

Handover latency
Handover latency is the time difference between when a mobile host is last able to send and/or receive an IP packet by way of the old access router, until when the mobile host is able to send and/or receive an IP packet through the new access router.

Hard handover
A hard handover is required where a mobile host is not able to receive or send traffic to two access points simultaneously. In order to move the traffic channel from the old to the new access point the mobile host abruptly changes the frequency/time-slot/code on which it is transmitting and listening to new values associated with a new access point.

Home address
A unicast routable address assigned to a mobile node, used as the permanent address of the terminal.

Home agent
A router on the mobile node's home link with which the mobile node has registered its current care-of address.

Home link
The link on which a mobile node's home subnet prefix is defined.

Home subnet prefix
The IP subnet prefix corresponding to a mobile node's home network.

Horizontal handover
Also know as *intra-technology handover*, a handover between two cells (or access points) employing the same air interface technology.

Lazy cell switching
Node should stay connected to the same access router as long as possible.

Make-before-break handover
During a make-before-break handover the terminal can communicate simultaneously with the old and new access routers. This should not be confused with "soft handover" which relies on macro diversity.

Mobile node
A node that can change its point of attachment from one link to another, while still being reachable via its home address.

Movement
A change in a mobile node's point of attachment to the Internet.

Network domain
A grouping of network objects, such as computers, that simplifies the naming of network services. Within a domain, all the names must be unique.

Policy action
It is the changing of the configuration of one or more network elements in order to achieve a desire policy state.

Policy agent
It is a software component that generates and responds to policy events, evaluates policies, and enforces policies.

Policy condition
It consists of two parts, a policy condition type and a policy condition element. This structure is aimed at satisfying the need for a canonical representation of a policy condition.

Policy conflict
Occurs when the actions of two rules (that are both satisfied simultaneously) contradict each other. The entity implementing the policy would not be able to determine which action to perform.

Policy core information model
An information model describing the basic concepts of policy groups, rules, conditions, actions, repositories and their relationships.

Policy decision point
The component responsible for the policy decision process. Policy decision is the abstraction of activating and evaluating one or more policy rules. Each policy rule is interpreted in the context of a specific request for accessing and/or using one or more resources.

Policy enforcement point
The component responsible for the policy enforcement process. Policy enforcement is the action of placing the network (or a part of the network) in a desired policy state using a set of management commands.

Policy rule
A policy rule is comprised of a set of conditions and a corresponding set of actions. This combination in effect defines a sequence of actions to be initiated when the corresponding set of conditions is either satisfied or not satisfied.

Policy translation
The transformation of a policy from a representation and/or level of abstraction, to another representation or level of abstraction.

Radio access technology
The radio access technology (i.e. air interface) is the radio-frequency portion of the transmission path between the wireless terminal (usually portable or mobile) and the active base station or access point.

Registration
The process during which a mobile node sends a binding update to its home agent or a correspondent node, causing a binding for the mobile node to be registered.

Return routability procedure
The return routability procedure authorises binding procedures by the use of cryptographic token exchange.

Roaming
The use of a communication device outside a specified administrative domain (home domain) defined by the service provider.

Seamless handover
A handover that is both smooth and fast, thus provides fast lossless handover between two access routers.

Smooth handover
A handover that aims primarily to minimise packet loss, with no explicit concern for additional delays in packet forwarding.

Soft handover
Support for soft handover (in a single mode terminal) is characteristic of radio interfaces which also require macro diversity (bicasting) for interference limitation but the two concepts are logically independent.

System
A collection of elements or components that are organised for a common purpose. In the scope of this work, a communication system consists of hardware and software components that have been carefully selected so that they work well together.

Vertical handover
Also called *inter-technology handover*, a handover between two cells employing different air interface technologies.

A.2 Nomenclature

3G	Third Generation wireless communication systems
3GPP	3rd. Generation Partnership Project
4G	Fourth Generation wireless communication systems
6Bone	The 6Bone is an IPv6 testbed that is an outgrowth of the IETF IPng project
AAA	Authentication, Authorisation and Accounting
ACK	Abbreviation for ACKnowledge in a TCP-connection
AR	Access Router
ATM	Asynchronous Transfer Mode
BARWAN	Bay Area Research Wireless Access Network
Bluetooth	A standard for short-range wireless communication between computing devices and associated peripherals, including laptop and mobile computers, personal digital assistants, and mobile phones
BTExact	Advanced communications technologies, part of British Telecom
CGSN	Combined GPRS Support Node
CL	Computer Laboratory

CMI	Cambridge MIT Institute
DAD	Duplicated Address Detection
DHCP	A protocol by which a server automatically assigns IP addresses to clients so users do not have to configure them manually. DHCP stands for Dynamic Host Configuration Protocol.
DNS	Domain Name Service
DVB-T	Digital Video Broadcasting-Terrestrial
EDGE	Enhanced Data GSM Environment
EURESCOM	European Institute for Research and Strategic Studies in Telecommunications
FMIPv6	Fast handovers for Mobile IP version 6
FSA	Finite State Automata
FSM	Finite State Machine
FSTs	Finite State Transducers
FTP	File Transfer Protocol
G711	Codec: Encode linear, pulse code modulation (PCM) narrowband speech signals using A-law or mu-law encoders. Decode index values into quantised output values using A-law or mu-law decoders. Convert between A-law and mu-law index values
G723.1	ITU-T speech coding and decoding through a combination of optimised C code
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HIPERLAN/2	HIPERLAN/ 2 is a flexible Radio LAN standard designed to provide high speed access (up to 54 Mb/s) to a variety of networks including 3G mobile core networks, ATM networks and IP based networks, and also for private use as a wireless local area network system
HMIPv6	Hierarchical Mobile IP version 6
HTTP	HTTP stands for Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
ICMPv6	Internet Control Message Protocol version 6
IEEE 802.11a	This is an extension to 802.11 that uses an orthogonal frequency division multiplexing encoding scheme
IEEE 802.11b	802.11b (also referred to as 802.11 High Rate or Wi-Fi)—an extension to 802.11 that applies to wireless LANs and provides 11 Mb/s transmission (with a fall-back to 5.5, 2 and 1 Mb/s) in the 2.4 GHz band. It was a 1999 ratification to the original 802.11 standard, allowing wireless functionality comparable to Ethernet
IEEE 802.11g	It applies to 802.11 wireless local area networks and provides 20+ Mb/s in the 2.4 GHz band

IEEE 802.11x	Refers to any type of wireless local area network technology (family)
IEEE 802.16a	IEEE 802.16 is working group number 16 of IEEE 802, specialising in point-to-multipoint broadband wireless access (it also is known as Wi-Max)
IEEE 802.20	This standard is under development. The goal is to provide fully mobile broadband wireless access for native Internet Protocol traffic
IEEE 802.3	The IEEE 802.3 Working Group develops standards for CSMA/CD (Ethernet) based local area networks
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSec	One of two protocols (with PPTP) used for virtual private networks. IPsec stands for IP security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IST	Information Society Technologies
L3	Layer three (also know as network layer)
LAN	Local Area Network
LCE	Laboratory for Communication Engineering
LCE-CL	Laboratory for Communication Engineering and Computer Laboratory
LPC10	Codec used for narrow bandwidth connections. The voice signal is clear but sounds robotic
M-DVB	Mobile Digital Video Broadcasting
MGEN	The Multi-Generator (MGEN) is open source software by the Naval Research Laboratory (NRL) PROTocol Engineering Advanced Networking (PROTEAN) Research Group. MGEN provides the ability to perform IP network performance tests and measurements using UDP/IP traffic (TCP is currently being developed)
MHz	Megahertz
MIP	Mobile IP
MIPL	Mobile IP for Linux
MIPv4	Mobile IP version 4
MIPv6	Mobile IP version 6
MTU	Maximum Transmission Unit
NAT	Network Address Translator
NIC	Network Interface Card
OSI	Open System Interconnection
P2P	Peer-to-Peer
PAN	Personal Area Network
PCI	Peripheral Component Interconnect
PCMCIA	Personal Computer Memory Card International Association
PDA	Personal Digital Assistant
PDP	Policy Decision Point

PEP	Policy Enforcement Point
P-II	Pentium II processor
P-III	Pentium III processor
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunnelling Protocol
PROTON	Policy-based system for ROaming Transparently among Overlay Networks
QoS	Quality of Service
RADIUS	AAA server within a GPRS network
RADVD	Router ADVertisement Daemon
RAM	Random Access Memory
RATs	Radio Access Technologies
RFC	Request for Comments
Richonet	Metricom Richonet technology for wireless wide area networks
RNCs	Radio Network Controllers
RTO	Retransmission Time Out
RTP	Real-Time Protocol for real-time data transmission over the Internet
RTT	Round Trip Time
SACKs	Selected ACKnowledgements
Seamoby	IETF Seamless Mobility working group
SGSN	Serving GPRS Support Node
SIP	Session Initiation Protocol
SIT	Simple Internet Translation
SND.NXT	Send Sequence abbreviation in the TCP protocol
SND.UNA	Send Unacknowledged abbreviation in the TCP protocol
SND.WND	The send-window abbreviation in the TCP protocol
std	standard deviation
TCP	Transmission Control Protocol
TCPDUMP	Networking tool to collect TCP-traces
TCPTRACE	Networking tool to obtain graphs from TCP-traces
TD-CDMA	Time Division Code Division Multiple Access
TDD	Time Division Duplexing
TF	Tautness Function
TFFSTs	Finite State Transducers with Tautness Function
TRPR	TRace Plot Real-time tool
TSG	EURESCOM Technical Specification Group
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications Services
UWB	Ultra Wideband
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
WANs	Wide Area Networks
WaveLAN	see WLAN
WCDMA	Wideband Code Division Multiple Access
WG	Working Group

Wi-Fi	Wi-Fi is short for wireless fidelity and is another name for IEEE 802.11b. It is a trade term promulgated by the Wireless Ethernet Compatibility Alliance (WECA).
Wi-Max	Wi-Max is an acronym that stands for Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

Appendix B

Policy Evaluation and Conflict Resolution

This appendix includes a description of the policy representation model based on Finite State Transducers (FSTs) used in Chapter 5, and it is a short version of the work included in [6]. The main focus of this appendix is to give a detailed introduction on the formalities of the model used to represent PROTON's policy model in an unambiguous way.

In Section B.1, two important concepts are defined: transducers and the tautness function as an abstraction of condition specificity. It is shown how these functions are placed into classical transducers to give them the expressiveness needed to cope with 4G environments. Section B.2 describes the basic algebra related to the tautness function. Then, Section B.3 defines two new structures by merging the concepts of transducers and tautness function. Section B.4 summarises operations that apply to transducers extended with tautness functions, and explains the semantics of those operations. Finally, Section B.5 details more about the conflict resolution method used in the PROTON policy model.

B.1 Transducers and Tautness functions

B.1.1 Recognisers

Finite State Recognisers (FSR) can be seen as a class of graphs and also as defining languages [81]. In the context of this work, the word "language" should be understood as a set of ordered sets of events triggered by changes in context.

Graphically, a recogniser is an oriented graph with a set of start states (nodes), a set of final states and labelled transitions (arcs) between nodes. A string is consumed symbol by symbol, and in each state, the process follows the transitions labelled with a matching symbol. The common terminology of automata theory uses the word "symbol" to refer to what automata consume. However, in this field the word "event" it is more accurate—in

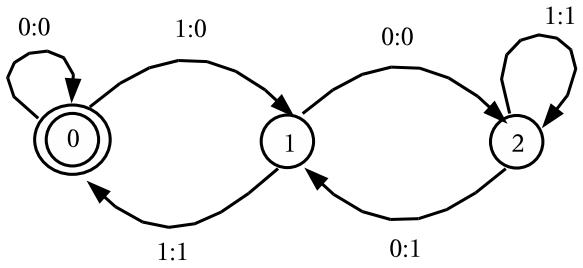


Figure B.1: Transducer representing division by 3 of binaries numbers

this dissertation, these words are used as synonyms. Common conventions when describing an FSR graphically are as follows: final states are depicted by two concentric circles; ϵ represents the empty string; and, unless otherwise specified, the initial state (usually labelled 0) is the leftmost state appearing in the figure. In-depth information about finite state machines can be found in [41, 81].

B.1.2 Transducers

Finite State Transducers (FSTs) can be interpreted as a class of graphs or a class of relations on strings. Under the first interpretation, an FST can be seen as an oriented graph with a pair of symbols in each arc, for example Figure B.1 shows a transducer that performs a division by three of binaries numbers.

Deterministic transducers are a category of transducers that present no ambiguities. This means that at any state of such transducers, at most one outgoing arc has a label with a given element of the alphabet. Output labels might be strings, including the empty string ϵ. However, the empty string is not allowed in input labels.

Deterministic transducers are computationally very interesting, because their computation load does not depend on the size of the transducer but rather only on the length of the input since that computation consists of following the only possible path corresponding to the input string and in writing consecutive output labels along this path. More information on transducers can be found in [81].

B.1.3 Tautness functions

Some of the main strategies used for solving what are known as "modality conflicts" [56] are:

- priority is always given to negative policies;
- the policy with highest priority is the one that is at the shortest distance from the managed object;
- the more specific a policy's domain is, the more priority the policy has;
- explicit priorities are directly assigned.

The work presented in Section 5.5.2 covers the first three strategies. The idea of calculating distance between a rule or policy, and the objects on which it is being applied, has already been used in the network management context and object oriented databases. In the latter, priority is given to the policy applying to the closer class in the inheritance hierarchy when evaluating access to an object referenced in a query.

One particular case of distance to the rule is the specificity related to *domain nesting* (or in a more general way to *condition nesting*) as explained in [56]. A sub-domain of objects is created for a specific management purpose to specify a policy that differs from those applying to the objects in the parent domain. This work makes use of this idea, generalising it to any condition as follows:

A policy has a condition delimiting a region where a given event can or cannot lay in. When such an event is inside two or more overlapping regions, a modality conflict may arise. The main concern is how tautly a condition fits on an event instead of how far from the border it is. Then, the preferred condition will be that which is the most "taut" around the event under consideration.

In order to quantitatively represent the aforementioned "tautness," a metric called *tautness function* (TF) was defined, so that the more taut a condition is, the closer its TF is to zero–the formal description of TF appears in Definition 1. This follows the *event-condition-action* (ECA) paradigm used, for example, in [16].

B.2 An Algebra for Tautness Functions

The logic used for tautness functions is inspired by the algebra used with **fuzzy sets**. Nevertheless, it does not mean that any kind of fuzzy logic is used to solve a policy conflict. However, the same logic behind fuzzy sets can be useful to decide how taut the condition of a certain combination of policies is for an event.

Definition 3 *Disjunction. The TF of the* disjunction *of two policy conditions A and B with TFs τ_A and τ_B respectively is defined as the maximum of the two individual TFs: $\tau_{A \vee B} = max(\tau_A, \tau_B)$. This operation is the equivalent of the OR operation in boolean algebra.*

Definition 4 *Conjunction. The TF of the* conjunction *of two policy conditions A and B with TFs $\tau_A$and τ_B respectively is defined as the minimum of the two individual TFs: $\tau_{A \wedge B} = min(\tau_A, \tau_B)$. This operation is the equivalent of the AND operation in boolean algebra. Please note that with this definition, when the event is inside the intersection of both conditions, its TF has the value of the tautest, and if the event is outside the intersection, the event is just as far outside as the one that is furthest outside.*

Definition 5 *Negation. The TF of the* negation *of a policy condition A with TFs* τ_A*, is defined changing the sign of the specified TF:* $\tau_{\neg A} = -\tau_A$*. This operation is the equivalent of the NOT operation in boolean algebra.*

Definition 6 *Tauter-than. The* tauter-than *operation* ($\rightarrow_\tau$) *is an ad hoc operation created to simplify the notation involved in the determinisation process which will be explained in B.5.2. The TF of the* tauter-than *of two policy conditions A and B with TFs* τ_A *and* τ_B *respectively is defined as:*

$$\tau_{A \rightarrow_\tau B} = \begin{cases} \tau_A, & if\ \tau_A < \tau_B \\ -1, & else \end{cases}$$

This operation is intended especially for the distance concept behind the TFs. It tells when A fits more tautly around an event than B.

Definition 7 *As-taut-as. The* as-taut-as *operation* ($\rightleftarrows_\tau$)*, like* tauter-than, *was created to simplify the notation involved in the determinisation process. The TF of the* as-taut-as *of two policy conditions A and B with TFs* τ_A *and* τ_B *respectively is defined as:*

$$\tau_{A \rightleftarrows_\tau B} = \begin{cases} \tau_A, & if\ \tau_A = \tau_B \\ -1, & else \end{cases}$$

B.3 Transducers with Tautness Functions and Identities

The main idea is to replace classic symbol labels on FSTs for TF labels. Thus, instead of trying to match an incoming symbol with a label on a transition, a recogniser with TFs will evaluate the TF on a transition for the incoming event.

Therefore, the mechanics of processing an incoming stream of events with a TFFSR is as follows: starting at an initial state, **an event is consumed if there is a transition with a positive (or zero) tautness function value evaluated for that event**. The new current state will be the one at the end of the chosen transition. Then the process is repeated.

The following two definitions (TFFSR and TFFST) are extensions of the definitions for predicate-augmented finite state recognisers (PFSR) and predicate-augmented finite state transducers (PFST) introduced in [98].

Definition 8 *A finite state recogniser with tautness functions (TFFSR) M is a tuple (Q, E, T, Π, S, F) where:*

- *Q is a finite set of states,*
- *E is a set of events,*
- *T is a set of tautness functions over E.*
- *Π is a finite set of transitions $Q \times T \cup \{\epsilon\} \times Q$.*
- *$S \subseteq Q$ is a set of start states,*
- *$F \subseteq Q$ is a set of final states.*

The relation $\widehat{\Pi} \subseteq Q \times E^* \times Q$ is defined inductively:

- *for all $q \in Q$, $(q, \epsilon, q) \in \widehat{\Pi}$.*
- *for all $(p, \epsilon, q) \in \Pi, (p, \epsilon, q) \in \widehat{\Pi}$.*
- *for all $(q_0, \tau, q) \in \Pi$ and for all $e \in E, if\ \tau(e) \geq 0$ then $(q_0, e, q) \in \widehat{\Pi}$.*
- *if $(q_o, x_1, q_1) \in \widehat{\Pi}$ and $(q_1, x_2, q) \in \widehat{\Pi}$ then $(q_0, x_1 x_2, q) \in \widehat{\Pi}$.*

The "language" of events $L(M)$ accepted by M is defined to be $\{w \in E^* \mid q_s \in S, q_f \in F, (q_s, w, q_f) \in \widehat{\Pi}\}$

An extension could be defined to allow the recogniser to deal with strings of events in each transition, in order to make some operations easier and produce more compact transducers.

A TFFSR is called ϵ-free if there are no $(p, \epsilon, q) \in \Pi$. For any given TFFSR there is an equivalent ϵ-free TFFSR. It is straightforward to extend the corresponding algorithm for classical automata to TFFSRs—this aspect is assumed.

Analogous substitution between symbol-labels and TF-labels is performed on finite state transducers. Thus, the FSTs outgoing label is also substituted by a TF. This means that, following a transition produces an event with a positive (or zero) value for the TF in the outgoing label of the transition.

Definition 9 *A finite state transducer* [1] *with tautness functions and identities (TFFST) M is a tuple (Q, E, T, Π, S, F) where:*

- *Q is a finite set of states,*
- *E is a set of symbols,*
- *T is a set of tautness functions over E.*
- *Π is a finite set of transitions $Q \times (T \cup \{\epsilon\}) \times (T \cup \{\epsilon\}) \times Q \times \{-1, 0, 1\}$. The final component of a transition is a sort of "identity flag" used to indicate when an incoming event must be replicated in the output*[2]*.*
- *$S \subseteq Q$ is a set of start states.*
- *$F \subseteq Q$ is a set of final states.*
- *For all transitions $(p, d, r, q, 1)$ it must be the case that $d = r \neq \epsilon$.*

It is assumed that the input and output sets of tautness functions are the same, and the same for input and output set of events.

The definition for the function str *from $T \cup \{\epsilon\}$ to 2^{E^*} follows.*

$$str(x) = \begin{cases} \{\epsilon\} & if\ x = \epsilon \\ \{s \mid s \in E, x(s) \geq 0\} & if\ x \in T \end{cases}$$

If $\tau \in T$ and $str(x)$ is a singleton set, then the transitions (p, τ, τ, q, i) where $i \in \{-1, 0, 1\}$ are equivalent.

The relation $\widehat{\Pi} \subseteq Q \times E^* \times E^* \times Q$ is defined inductively:

- *for all p, $(p, \epsilon, \epsilon, p) \in \widehat{\Pi}$.*
- *for all $(p, d, r, q, 0), x \in str(d), y \in str(r)$,*
  $(p, x, y, q) \in \widehat{\Pi}$.

[1] A simplify version of this definition is included in Chapter 5.

[2] The negative identity value is to express the obligatory difference between input and output, this is needed to compute the complement of a TFFST.

- *for all* $(p, d, r, q, -1), x \in str(d), y \in str(r)$, $x \neq y, (p, x, y, q) \in \widehat{\Pi}$.

- *for all* $(p, \tau, \tau, q, 1)$ *and* $x \in str(\tau), (p, x, x, q) \in \widehat{\Pi}$.

- *if* $(q_o, x_1, y_1, q_1) \in \widehat{\Pi}$ *and* $(q_1, x_2, y_2, q) \in \widehat{\Pi}$ *then* $(q_0, x_1x_2, y_1y_2, q) \in \widehat{\Pi}$.

The relation $R(M)$ *accepted by a TFFST* M *is defined to be* $\{(w_d, w_r) \mid q_s \in S, q_f \in F, (q_s, w_d, w_r, q_f) \in \widehat{\Pi}\}$.

An extension could be defined to let the transducer deal with strings of events in each transition. TFFSTs will be the base model for a policy rule. In the ECA paradigm the incoming label will model the condition part of a policy, and the outgoing label will model the triggered action.

B.4 Operations on TFFST

In several cases the operations on TFFSTs could be easily generalised from FSTs or from PFSTs. In this section, the main operations and their new algorithms on TFFSTs are presented, with special emphasis on the determinisation algorithm.

B.4.1 Identity

The identity relation for a given language of events L is $id(L) = \{(w, w) \mid w \in L\}$. Further on, it is shown that a *right* will be a identity relation applied to the incoming authorised event. For a given TFFSR $M = (Q, E, T, \Pi, S, F)$, the identity relation is given by the TFFST $M' = (Q, E, T, \Pi', S, F)$ where $\Pi' = \{(p, \tau, \tau, q, 1) \mid (p, \tau, q) \in \Pi\}$. Note that several events could be positive under τ so the "identity flag" is set to 1 in order to force the event produced to be the same that the one that entered. A more in-depth explanation of this can be found in [98] but one difference from the identity defined for PFSTs, is that a -1 value is define for the identity flag.

Symbols "¡" and "¿" around TFs in the labels express identity between the input and the output, and symbols "[" and "]" express difference.

B.4.2 Union

The union of two transducers results in a transducer that defines the relation which is the union of the relations defined by each of the two original transducers. This means that the new transducer Hill recognises the union of sets of "words" of events recognised for both original transducers and produces the same events that the original ones would produce. The union algorithm for TFFSTs is simple and analogous to the one for classical

transducers. No ϵ labels should remain in the input part of transitions, thus an algorithm such as the one in [68] should be used to avoid those labels. That algorithm does not ensure that the final transducer will be deterministic. It will be shown below how to solve this using determinisation algorithm. Typically, adding a new policy will mean computing the union of the TFFST modelling the new policy.

B.4.3 Intersection

The intersection of two transducers results in a transducer that defines the relation resulting from the intersection of the relations of the two original transducers. It is one of the most important and powerful operations on automata. As TFFSTs are not always closed under intersection, it is necessary to start thinking about recognisers instead of transducers.

In the classic case, the intersection of two given automata M_1 and M_2 is constructed by considering the cross product of states of M_1 and M_2. A transition $((p_1, p_2), \sigma, (q_1, q_2))$ exists in the intersection iff the corresponding transition (p_1, σ, q_1) exists in M_1 and (p_2, σ, q_2) exists in M_2. In the case of TFFSR, instead of requiring that the symbol σ occur in the corresponding transitions of M_1 and M_2, the resulting tautness function must be the conjunction of the corresponding tautness functions in M_1 and M_2.

Given two ϵ-free TFFSRs $M_1 = (Q_1, E, T, \Pi_1, S_1F_1)$ and $M_2 = (Q_2, E, T, \Pi_2, S_2F_2)$, the intersection $L(M_1) \cap L(M_2)$ is the language accepted by $M = (Q_1 \times Q_2, E, T, \Pi, S_1 \times S_2, F_1 \times F_2)$ and $\Pi = \{((p_1, q_1), \tau_1 \wedge \tau_2, (p, q)) \mid (p_1, \tau_1, p) \in \Pi_1, (q_1, \tau_2, q) \in \Pi_2\}$.

The transducers used in the model will always be analogous to the letter transducers presented in [81]. Then it is possible to use this intersection definition on the underlying TFFSRs, which will be defined below, to calculate the intersection of the transducers themselves.

Definition 10 *Underlying TFFSR. If $M = (Q, E, T, \Pi, S, F)$ is an TFFST, its underlying TFFSR is $M' = (Q, E, T, \Pi', S, F)$ where:*

$$\Pi' = \{(p, (x, y), q) \mid (p, x, y, p, i) \in \Pi\}$$

All properties of finite state automata apply to the underlying automaton of a transducer. For example, the intersection algorithm could be applied to the underlying TFFSR and in these conditions, properly interpreted as the intersection of two TFFSTs.

B.4.4 Complement

This is the *complement* of the relation defined by a transducer. This operation will be useful to compute a subtraction between relations or sets of rules. As in basic sets theory, $A - B$ will be expressed as $A \cap \overline{B}$.

Although it is true that transducers are not closed under *complementation* in all conditions, the "axiomatic" transducers will never accept ϵ as a valid input string nor ϵ symbols in the input part of loops. Therefore the complement of a policy/transducer can

be computed considering the transducer as a recogniser that consumes pairs x:y in the set $E \times E$ and computing the complementation algorithm on it as seen in [98]. To make this possible it is required to include a "non-equal" flag into TFFST definition, although it makes no direct sense in expressing a policy.

B.5 Conflict resolution

This section aims to extend the description included in Section 5.5.2 of the conflict resolution process. The main operations related to this process are: *Composition* and *Determinasion*.

B.5.1 Composition

The meaning of composition here is the same as for any other binary relations: $R_1 \circ R_2 = \{(x, z) \mid (x, y) \in R_1, (y, z) \in R_2\}$. This can be seen as a chain of events processing: the events outgoing from the first transducer are taken as input of the second one.

In the classic case, the composition of two transducers M_1 and M_2 is built considering the cross product between the states of M_1 and M_2. A transition $((p_1, p_2, \sigma_i, \sigma_o, (q_1, q_2))$ exists iff there is some σ such that $(p_1, \sigma_i, \sigma, q_1)$ exists in M_1 and $(p_2, \sigma, \sigma_o, q_2)$ exists in M_2. The case of TFFSTs is equal to that of PFSTs plus an extra condition for non-identity. It is required that the conjunction of both TFs be positive. Identity is required in the new transition only when it is required in both, and the same with the non-identity mark; it will remain in the new transition only if it existed in both original ones. Transitions with ϵ in the input or output should be handled differently.

Given two TFFST $M_1 = (Q_1, E, T, \Pi_1, S_1, F_1)$ and $M_2 = (Q_2, E, T, \Pi_2, S_2, F_2)$, the relation $R_1 \circ R_2$ is defined by $M = (Q_1 \times Q_2, E, T, \Pi, S_1 \times S_2, F_1 \times F_2)$ where:

$$
\begin{aligned}
\Pi = \{((p_1, p_2), d, r, (q_1, q_2), i) \mid \\
&(p_1, d, \tau_1, q_1, i_1) \in \Pi_1, \\
&(p_2, \tau_2, r, q_2, i_2) \in \Pi_2, \\
&\tau_1 \wedge \tau_2, \\
&i = 0 \; if \; i_1 = 0 \; or \; i_2 = 0 \; or \; i_1 \neq i_2, \\
&i = 1 \; if \; i_1 = 1 \; and \; i_2 = 1, \\
&i = -1 \; if \; i_1 = -1 \; and \; i_2 = -1\} \\
\cup \{((p_1, p_2), \epsilon, d, (q_1, q_2), 0) \mid \\
&(p_1, \epsilon, d', q_1, i_1) \in \Pi_1, (p_2, d', d, q_2, i_2) \in \Pi_2\} \\
\cup \{((p_1, p_2), r, \epsilon, (q_1, q_2), 0) \mid \\
&(p_1, r, d', q_1, i_1) \in \Pi_1, (p_2, d', \epsilon, q_2, i_2) \in \Pi_2\}
\end{aligned}
$$

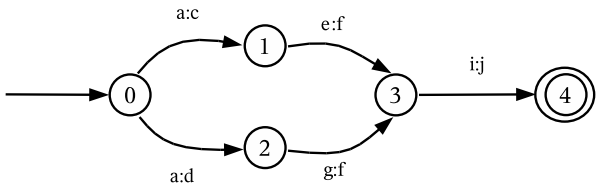


Figure B.2: Before determinisation

B.5.2 Determinisation

This is the most important operation for the objectives of the methodology, but also the one that exhibits the highest computational cost. To solve the conflicts between a set of rules, the transducer representing that set must be determinised in order to eliminate ambiguities.

A TFFST M is *deterministic* if M has a single start state, if there are no states $p, q \in Q$ such that $(p, \epsilon, x, q, i) \in \Pi$, and if for all states p and events e there is at most one transition (p, τ_d, x, q, i) such that $\tau_d(e)$ is positive. If a TFFST M is deterministic then the process of computing the output events for a given stream of events ω as defined by M, can be implemented efficiently. This process is linear in ω, and independent of the size of M.

In order to extend the determinisation algorithm of PFST [98] to TFFST, the case where, although an event satisfies two conditions, one of these conditions fits more tautly than the other must be considered. Additionally, the case where both conditions have the same specificity needs to be considered. In this way, all possible combinations between TFs are computed as with PFSTs using the following operations: *conjunction, tauter-than* and *as-taut-as*.

It is assumed that the output part of a transition contains a sequence of TFs; this implies an extension to the above-defined TFFSTs. It is also assumed that there are no ϵ input functions, because an equivalent transducer without ϵ input functions could be computed for a transducer which has these functions [81].

In this algorithm, outputs are delayed as much as possible. This is because a local ambiguity may not be such if the whole transducer is considered, and realise that only one path would be possible until the final state. This is the case of the ambiguity shown in state 0 in Figure B.2.

The algorithm maintains sets of pairs $Q \times T^*$. Each of these sets corresponds to a state in the determinised transducer. In the example of Figure B.2, once an event that satisfies condition a is read, states 1 and 2 are possible, with pending outputs c and d, then $P = \{(1, c), (2, d)\}$. In order to compute the transitions leaving such a set of pairs P, computed (as in PFSTs) $Trans^P(\tau_d) = \{(q, xy) \mid (p, x) \in P, (p, \tau_d : y, q) \in \Pi\}$. In the example, $Trans^P(e) = \{(3, cf)\}$ and $Trans^P(g) = \{(3, df)\}$. Let T' be the TFs in the domain of $Trans^p$. For each split of T' into $\tau_1 ... \tau_i$ and $\neg\tau_{i+1} ... \neg\tau_n$, and each possible order of $\tau_1 ... \tau_i$ we have a proto-transition with the operator $\rightarrow_\tau$ and another with the operator $\rightleftarrows_\tau$ between two consecutive TFs τ_j, τ_{j+1}:

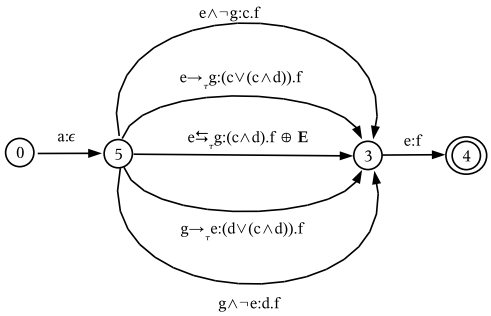


Figure B.3: After determinisation procedure

$$(P, \tau_1 ... \tau_j \rightarrow_\tau \tau_{j+1} ... \tau_i \wedge \neg\tau_{i+1} \wedge ... \neg\tau_n, A)$$
$$(P, \tau_1 ... \tau_j \rightleftarrows_\tau \tau_{j+1} ... \tau_i \wedge \neg\tau_{i+1} \wedge ... \neg\tau_n, A)$$

where A is $[Trans^P(\tau_1), ..., Trans^P(\tau_i)]$. In the example the following proto-transitions are derived:

$$(P, e \wedge \neg g, [(3, cf)])$$
$$(P, g \wedge \neg e, [(3, df)])$$
$$(P, e \rightarrow_\tau g, [(3, cf), (3, df)])$$
$$(P, g \rightarrow_\tau e, [(3, df), (3, cf)])$$
$$(P, e \leftrightarrows_\tau g, [(3, cf), (3, df)])$$
$$(P, g \leftrightarrows_\tau e, [(3, cf), (3, df)])$$

The last two proto-transitions are equivalent and one should be erased.

A transition is created from proto-transitions putting in their output the longest common prefix of TFs in the target pairs (ϵ in the example). Before removing the longest common prefix, the sequences of output TFs should be packed, putting together the sequences associated with the same target state. This is done using a combination of conjunction and disjunction in the case of a *tauter-than* relation or including an **xor** with an **error** event in the case of a *as-taut-as*. This means that, in the case that no decision can be made, an **error** is marked [3]. Thus, two pairs of target states and TF sequences (p, s_1) and (p, s_2) can be combined into a single pair (p, s) iff (p, s_1) is in $A[j]$ and (p, s_1) is in $A[j+1]$ and $s_1 = \tau_1 ... \tau_i ... \tau_n$, $s_2 = \tau_1 ... \tau'_i ... \tau_n$ and $s = \tau_1 ... (\tau_i \vee (\tau_i \wedge \tau'_i)) ... \tau_n$ if the operator between τ_j and τ_{j+1} is $(\rightarrow_\tau)$, or $s = \tau_1 ... (Error \oplus (\tau_i \wedge \tau'_i)) ... \tau_n$ if the operator between τ_j and τ_{j+1} is $(\leftrightarrows_\tau)$. In the example the third proto-transition is packed in

$$(P, e \rightarrow_\tau g, \{(3, (c \vee (c \wedge d))f)\})$$

and the fourth

$$(P, e \leftrightarrows_\tau g, \{(3, (Error \oplus (c \wedge d))f)\})$$

as can be seen in B.3.

Despite the fact that $a \vee (a \wedge b) \equiv a$, the expression on the left is chosen due to the use of events that must sometimes be replicated in the output. This expression is closer to the behaviour of the actual implementation when computing identities.

[3] This would be useful to detect non resolvable conflicts and report them to an operator.

Bibliography

[1] 3GPP. 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects. `http://www.3gpp.org/TB/home.htm`.

[2] 3GPP/TSG(2003). Feasibility Study on 3GPP System to Wireless Local Area Network (WLAN) Interworking (Release 6).

[3] 6BONE. Testbed for the Deployment of IPv6. `http://www.6bone.net`.

[4] S. Aust, N. A. Fikouras, D. Prtel, C. Grg, and C. Pampu. Policy Based Mobile IP Handoff Decision (POLIMAND). Internet Draft, `draft-iponair-dna-polimand-00.txt`, work in progress, IETF, October 2003.

[5] M. Bagnulo, I. Soto, A. Garcia-Martinez, and A. Azcorra. Avoiding DAD for Improving Real-Time Communication in MIPv6 Environments. In *Proceedings of the Joint International Workshops on Interactive Distributed Multimedia Systems and Protocols for Multimedia Systems*, pages 73–79, October 2002.

[6] J. Baliosian and J. Serrat. Finite State Transducers for Policy Evaluation and Conflict Resolution . In *Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2004)*, pages 250–259, June 2004.

[7] C. J. Bernardos, I. Soto, J. I. Moreno, T. Melia, M. Liebsch, and R. Schmitz. Mobile Networks Experimental Evaluation of a Handover Optimization Solution for Multimedia Applications in a Mobile IPv6 Network. *European Transactions on Telecommunications*, 2004.

[8] M. Buddhikot, G. Chandranmenon, S. Han, Y. W. Lee, S. Miller, and L. Salgarelli. Integration of 802.11 and Third-Generation Wireless Data Networks. In *Proceedings of Twenty-Second IEEE Conference on Computer and Communications (INFOCOM 2003)*, pages 503–512, March 2003.

[9] R. Caceres and V. N. Padmanabhan. Fast and Scalable Handoffs for Wireless Internetworks. In *Proceedings of the Second ACM International Conference on Mobile Computing and Networking (MobiCom 1996)*, pages 55–66, 1996.

[10] A.T. Campbell, J. Gomez-Castellanos, S. Kim, A. Valko, C. Wan, and Z. Turanyi. Design, Implementation, and Evaluation of Cellular IP. *IEEE Personal Communications Magazine*, 7(4):42–49, August 2000.

[11] A.T. Campbell, J. Gomez-Castellanos, C. Wan, S. Kim, Z. Turanyi, and A. Valko. Cellular IP. Internet Draft, `draft-ietf-mobileip-cellularip-00.txt`, work in progress, IETF, January 2000.

[12] V. Cerf. The Catenet Model for Internetworking. DARPA Information Processing Techniques Office, IEN 48, July 1978.

[13] R. Chakravorty, P. Vidales, K. Subramanian, I. Pratt, and J. Crowcoft. Performance Issues with Vertical Handovers—Experiences from GPRS Cellular and WLAN Hot-Spots Integration. In *Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom 2004)*, pages 155–164, March 2004.

[14] P.M.L. Chan, S. Re, and H. YF. Mobility Management Incorporating Fuzzy Logic to Heterogeneous IP Environment. *IEEE Communications Magazine*, 39(12):42–51, December 2001.

[15] H. Chaskar and R. Koodli. QoS Support in Mobile IP Version 6. In *Proceedings of the IEEE Broadband Wireless Summit (Networld+Interop)*, May 2001.

[16] J. Chomicki, J. Lobo, and S. Naqvi. Conflict Resolution Using Logic Programming. *IEEE Transactions on Knowledge and Data Engineering*, 15(1):245–250, January/February 2003.

[17] A. Corradi, E. Lupu, R. Montanari, and C. Stefanelli. Policy Controlled Mobility. In *the Workshop on Software Engineering and Mobility (ICSE)*. IEEE Computer Society, 2001.

[18] A. Corradi, R. Montanari, G. Tonti, and C. Stefanelli. How to Support Adaptive Mobile Applications. In *Proceedings of the Workshop from Objects to Agents (WOA 2001)*, pages 42–47, September 2001.

[19] X. P. Costa and H. Hartenstein. A Simulation Study on the Performance of Mobile IPv6 in a WLAN-based Cellular Network. *Computer Networks*, 40(1):191–204, September 2002.

[20] X. P. Costa, R. Schmitz, H. Hartenstein, and M. Liebsch. A MIPv6, FMIPv6 and HMIPv6 handover Latency Study: Analytical Approach. In *Proceedings of the IST Mobile and Wireless Telecommunications Summit 2002*, pages 100–105, June 2002.

[21] D. Cottingham and P. Vidales. Is Latency the Real Enemy in Next Generation Networks? In *Proceedings of the First International Workshop on Convergence of Heterogeneous Wireless Networks (ConWin 2005)*, July 2005. submitted.

[22] A. Cuevas, P. Serrano, C. J. Bernardos, J. I. Moreno, J. Jaehnert, K. Hyung-Woo, J. Zhou, D. Gomes, P. Gonalves, and R. Aguiar. Field Evaluation of a 4G True-IP Network. In *Proceedings of the IST Mobile Summit 2004*, 2004. Lyon, France.

[23] Daidalos(2004). EU-FP6 project: Designing Advanced Network Interfaces for the Delivery and Administration of Location Independent, Optimised Personal Services.

[24] N. Damianou, N. Dulay, E. Lupu, and M. Sloman. The Ponder Policy Specification Language. In *Proceedings of the Second IEEE International Workshop on Policies for Distributed Systems (POLICY 2001)*, pages 18–39, January 2001.

[25] S. Dar, A. McAuley, A. Dutta, A. Misra, K. Chakraborty, and S. K. Das. IDMP: An Intradomain Mobility Management Protocol for Next Generation Wireless Networks. *IEEE Wireless Communications Magazine*, 9(3):38–45, June 2002.

[26] S. Das, A. Misra, P. Agrawal, and S. K. Das. TeleMIP: Telecommunication-Enhanced Mobile IP Architecture for Fast Intra-Domain Mobility. *IEEE Personal Communications Magazine*, 7(4):50–58, August 2000.

[27] S. Deering and R. Hinden. Internet Protocol Version 6 (IPv6) Specification. RFC(2640), IETF, July 1999.

[28] B. Dragovic, J. Baliosian, P. Vidales, and J. Crowcroft. Autonomic System for Context-Aware Security in Ubiquitous Computing Environments. In *Proceedings of the Tenth European Symposium on Research in Computer Security (ESORICS 2005)*, Milan, Italy, September 2005. Lecture Notes in Computer Science (submitted).

[29] B. Dragovic and J. Crowcroft. Context-Adaptive Information Security for UbiComp Environments. In *Website of the Second UK-Ubinet Workshop: Security, trust, privacy and theory for ubiquitous computing*, May 2004.

[30] R. Droms, J. Bound, B. Volz, T. Lemon, C. E. Perkins, and M. Carney. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC(3315), IETF, July 2003.

[31] N. Dunlop, J. Indulska, and K. Raymond. Dynamic Conflict Detection in Policy-Based Management Systems . In *Proceedings of the Enterprise Distributed Object Computing Conference (EDOC 2002)*, pages 15–26, September 2002.

[32] Elvin. Elvin for the Impatient. `http://www.elvin.dstc.edu/doc/impatient.html`.

[33] EURESCOM. European Institute for Research and Strategic Studies in Telecommunications. `http://www.eurescom.de`.

[34] EURESCOM(P1013). First Steps Towards UMTS: Mobile IP Services, an European Testbed, 2003[4].

[35] EURESCOM(P1013-D1). Definition of Terminology for Mobile IP Definition of IP Services in the Mobility Context, September 2002[5].

[36] M. P. Fernandez, A. de C. P. Pedroza, and J. F. Rezende. Converting QoS Policy Specification into Fuzzy Logic Parameters. Technical report, Departamento de Electronica, Universidad Federal Rio de Janeiro (UFRJ), 2003.

[4] `http://www.eurescom.de/~ftproot/web-deliverables/public/P1000-series/P1013/`

[5] `http://www.eurescom.de/public/projects/P1000-series/p1013/default.asp`

[37] N. A. Fikouras, A. Udugama, C. Grg, W. Zirwas, and J. M. Eichinger. Experimental Evaluation of Load Balancing for Mobile Internet Real-Time Communications. In *Proceedings of the Sixth International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 173–179, October 2003.

[38] G. Fitzpatrick, S. Kaplan, T. Mansfield, D. Arnold, and B. Segal. Supporting Public Availability and Accessibility with Elvin: Experiences and Reflections. *Computer Supported Collaborative Work*, 11(3–4):444–474, 2002.

[39] S. Gai. *Internetworking IPv6 with Cisco Routers*. The McGraw-Hill Companies, February 1998.

[40] P. H. Hartel, P. van Eck, S. Etalle, and R. J. Wieringa. Modelling Mobility Aspects of Security Policies. In *Proceedings of Construction and Analysis of Safe, Secure and Interoperable Smart cards (CASSIS)*, pages 172–191, March 2004.

[41] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* . Addison Wesley, 1979.

[42] R. Hsieh and A. Seneviratne. A Comparison of Mechanisms for Improving Mobile IP Handoff Latency for End-to-End TCP. In *Proceedings of the Nineth ACM International Conference on Mobile Computing and Networking (MobiCom 2003)*, pages 29–41, September 2003.

[43] R. Hsieh, Z. G. Zhou, and A. Seneviratne. S-MIP: A Seamless Handoff Architecture for Mobile IP. In *Proceedings of Twenty-Second IEEE Conference on Computer and Communications (INFOCOM 2003)*, volume 3, pages 1774–1784, March 2003.

[44] IBM Research Headquarters (manifesto). Autonomic Computing: IBM's Perspective on the State of Information Technology, October 2001. `http://www.research.ibm.com/autonomic/overview/elements.html`.

[45] IEEE 802.20. Mobile Broadband Wireless Access (MBWA). `http://grouper.ieee.org/groups/802/20/`.

[46] IP6tables. Manipulation of the IPv6 Netfilter. `http://www.redhat.com`.

[47] K. Jean, K. Yang, and A. Galis. A Policy Based Context-Aware Service for Next Generation Networks. In *Proceedings of the Eight IEE London Communications Symposium*, October 2003.

[48] D. B. Johnson, C. E. Perkins, and J. Arkko. Mobility Support in IPv6. RFC(3775), IETF, June 2004.

[49] R. H. Katz. Adaptation and Mobility in Wireless Information Systems. *IEEE Personal Communications*, 1:6–17, May 1994.

[50] J. Kempf. Problem Description: Reasons for Doing Context Transfers Between Nodes in an IP Access Network. RFC(3374), IETF, September 2002.

[51] J.O. Kephart and W.E. Walsh. An Artificial Intelligence Perspective on Autonomic Computing Policies . In *Proceeding of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2004)*, pages 3–12, June 2004.

[52] S. King, R. Fax, D. Haskin, W. Ling, T. Meehan, R. Fink, and S. Perkins. The Case for IPv6. Internet Draft, `draft-iab-case-for-ipv6-06.txt`, work in progress, IETF, December 1999.

[53] R. Koodli. Fast Handovers for Mobile IPv6. Internet Draft, `draft-ietf-mipshop-fast-mipv6-06.txt`, work in progress, IETF, March 2003.

[54] K. Kuladinithi, A. Konsgen, S. Aust, and N. A. Fikouras. Mobility Management for an Integrated Network Platform. In *Proceedings of the Fourth IEEE International Workshop on Mobile and Wireless Communication Networks (MWCN 2002)*, pages 621–625, September 2002.

[55] K. Lai, M. Roussopoulos, D. Tang, X. Zhao, and M. Baker. Experiences with a Mobile Testbed. In *Proceedings of The Second International Conference on Worldwide Computing and its Applications (WWCA 1998)*, pages 222–237, March 1998.

[56] E.C. Lupu and M. Sloman. Conflicts in Policy-Based Distributed Systems Management . *IEEE Transactions on Software Engineering*, 25(6):852–869, November/December 1999.

[57] J. Makela. Handoff Decision in Multi-Service Networks. In *Proceedings of the Eleventh IEEE International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC 2000)*, pages 655–659, September 2000.

[58] J. Manner, M. Kojo, T. Suihko, P. Eardley, D. Wisely, R. Hancock, and N. Georganopoulos. Mobility Related Terminology. Internet Draft, `draft-manner-seamoby-terms-00.txt`, work in progress, IETF, January 2001.

[59] V. Marques, R.L. Aguiar, C. Garcia, J.I. Moreno, C. Beaujean, E. Melin, and M. Liebsch. An IP-Based QoS Architecture for 4G Operator Scenarios. *IEEE Wireless Communications Magazine*, 10(3):54–62, June 2003.

[60] Mathworld. Mathematics Internet Resource. `http://www.mathworld.worlfram.com/Outdegree.html`.

[61] Y. Matsushita, T. Matsuda, and M. Yamamoto. TCP Congestion Control with ACK-Pacing for Vertical Handover. In *Proceedings of the IEEE Wireless Communications and Networking Conference 2005*, March 2005.

[62] S. McCreary and K. Claffy. Trends in Wide Area IP Traffic Patterns—A View from Ames Internet Exchange. Technical report, CAIDA, 2000. `http://www.caida.org`.

[63] MGEN. The Multi-Generator Toolset. `http://mgen.pf.itd.nrl.navy.mil/`.

[64] MIND. Trials Final Report. Project Deliverable, IST, November 2002.

[65] Mind(2000). IST project: Mobile IP based Network Developments.

[66] MIPL. Mobile IP for Linux (MIPL). Developed by HUT Laboratory for Theoretical Computer Science—GO/Core project. `http://www.mobile-ipv6.org`.

[67] A. Misra, S. Das, A. Dutta, and S. K. Das. IDMP-Based Fast Handoffs and Paging in IP-Based 4G Mobile Networks. *IEEE Wireless Communications Magazine*, 40(3):138–145, March 2002.

[68] Mehryar Mohri. Finite-State Transducers in Language and Speech Processing . *Computational Linguistics*, 23(2):269–311, 1997.

[69] B. Moore, E. Ellesson, J. Strassner, and A. Westerinen. Policy Core Information Model. RFC(3060), IETF, February 2001.

[70] N. Moore. Optimistic Duplicated Address Detection for IPv6. Internet Draft, `draft-ietf-ipv6-optimistic-dad-05.txt`, work in progress, IETF, February 2005.

[71] K. Murray, R. Mathur, and D. Pesch. Intelligent Access and Mobility Management in Heterogeneous Wireless Networks Using Policy. In *Proceedings of the First ACM International Workshop on Information and Communication technologies*, pages 181–186, May 2003.

[72] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). RFC(2461), IETF, December 1998.

[73] Nomad(2002). IST project: Integrated Networks for Seamless and Transparent Service Discovery.

[74] L. Patanapongpibul and G. Mapp. A Client-Based Handoff Mechanism for Mobile IPv6 Wireless Networks. In *Proceedings of the Eighth IEEE Symposium on Computers and Communications (ISCC)*, pages 563–568, July 2003.

[75] C. E. Perkins. IP Mobility Support for IPv4. RFC(3344), IETF, August 2002.

[76] P. Philippopoulos, P. Fournogerakis, I. Fikouras, N. Fikouras, and C. Grg. NOMAD: Integrated Networks for Seamless and Transparent Service Discovery. In *Proceedings of the IST Mobile Summit 2002*, 2002.

[77] C. Policroniades, R. Chakravorty, and P. Vidales. A Data Repository for Fine-Grained Adaptation in Heterogeneous Environments. In *Proceedings of the Third ACM international workshop on Data engineering for wireless and mobile access*, pages 51–55, June 2003.

[78] Ponder. Policy Research Group. `http://www-dse.doc.ic.ac.uk/Research/policies/ponder.shtml`.

[79] RADVD. Router ADVertisement Daemon. `http://v6web.litech.org/radvd/`.

[80] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, and S. Y Wang. HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-Area Wireless Networks. In *Proceedings of the Seventh International Conference on Network Protocols*, pages 283–292, October 1999.

[81] E. Roche and Y. Schabes. Finite-State Language Processing . Technical report, MIT Press, Cambridge, Massachusetts, 1997.

[82] P. Ruiz, E. Mitjana, and L. Burness. Advanced Services over Future Wireless and Mobile Networks in the Framework of the MIND Project. In *Proceedings of the Terena Networking Conference (TNC 2002)*, June 2002.

[83] A. Sanmateu, L. Morand, E. Bustos, S. Tessier, F. Paint, and A.M. Sollund. Using Mobile IP for Provision of Seamless Handoff between Heterogeneous Access Networks, or How a Network can Support the Always-On Concept. In *Proceedings of the EURESCOM Summit 2001*, November 2001.

[84] M. Sloman and E. Lupu. Policy Specification for Programmable Networks. In *Proceedings of the First International Working Conference on Active Networks (IWAN 1999)*, pages 73–84, July 1999.

[85] SNIA. Storage Networking Industry Association. `http://www.snia.org/tech_activities/workgroups/policy`.

[86] A. C. Snoeren and H. Balakrishnan. An End-to-End Approach to Host Mobility. In *Proceedings of the Sixth ACM International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 155–166, August 2000.

[87] A. C. Snoeren, H. Balakrishnan, and M. F. Kaashoek. Reconsidering Internet Mobility. In *Proceedings of the Eighth Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, pages 41–46, May 2001.

[88] H. Soliman, C. Castelluccia, K. E. Malki, and L. Bellier. Hierarchical Mobile IPv6 Mobility Management (HMIPv6). Internet Draft, `draft-ietf-mipshop-hmipv6-04.txt`, work in progress, IETF, December 2004.

[89] Frank Stajano. *Security for Ubiquitous Computing*. John Wiley and Sons, February 2002.

[90] M. Stemm and R. H. Katz. Vertical Handoffs in Wireless Overlay Networks. *Mobile Networks and Applications*, 3(4):335–350, 1998.

[91] J. Strassner and E. Ellesson. Terminology for Describing Network Policy and Services. Internet Draft, `draft-strassner-policy-terms-00.txt`, work in progress, IETF, August 1998.

[92] W. Simpson T. Narten, E. Nordmark. Neighbor Discovery for IP Version 6 (IPv6). RFC(2461), IETF, August 1998.

[93] TCPDUMP. Sniffer to Collect Network Traffic. `http://www.tcpdump.org`.

[94] TCP/IP. Transmission Control Protocol and Internet Protocol Specifications. RFC(791/93), IETF, September 1981.

[95] TCPTRACE. Tool to Analyse TCP Dump Files. `http://www.tcptrace.org`.

[96] N. Tripathi, J. Reed, and H. Vanlandingham. Handoff in Cellular Systems. *IEEE Personal Communications Magazine*, 5(6):26–37, December 1998.

[97] TRPR. TRace Plot Real-Time. `http://proteantools.pf.itd.nrl.navy.mil/trpr.html`.

[98] G. van Noord and D. Gerdemann. Finite State Transducers with Predicates and Identities . *Grammars*, 4(3):263–286, December 2001.

[99] N. Vardalachos, J. Rubio, A. Galis, and J. Serrat. A Policy Management System for Hybrid Networks. In *Proceedings of The Seventh IEE London Communications Symposium*, September 2002.

[100] P. Vidales, J. Baliosian, J. Serrat, G. Mapp, F. Stajano, and A. Hopper. Autonomic Systems for Mobility Support in 4G Networks. *Journal on Selected Areas in Communications (J-SAC)*, 23(12), 2005.

[101] P. Vidales, R. Chakravorty, and C. Policroniades. PROTON: A Policy–Based Solution for Future 4G devices. In *Proceedings of Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2004)*, pages 219–222, June 2004.

[102] P. Vidales and F. Stajano. The Sentient Car: Context-Aware Automotive Telematics. In *Proceedings of the First IEE European Workshop on Location Based Services (LBS-2002)*, September 2002.

[103] R. Wakikawa, K. Uehara, and Jun Murai. Multiple Network Interfaces Support by Policy-Based Routing on Mobile IPv6. In *Proceedings of the 2002 International Conference on Wireless Networks(ICWN)*, July 2002.

[104] H. J. Wang. Policy-Enabled Handoffs Across Heterogeneous Wireless Networks. Technical Report CSD-98-1027, UC BerKeley, 1998.

[105] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications Magazine*, 4(5):42–47, October 1997.

[106] E. Wedlund and H. Schulzrinne. Mobility Support Using SIP. In *Proceedings of the Second ACM International Workshop on Wireless Mobile Multimedia (WOWMOM 1999)*, pages 76–82, August 1999.

[107] A. Westerinen, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser. Terminology for Policy-Based Management. RFC(3198), IETF, November 2001.

[108] E. Wohlstadter, S. Tai, T. A. Mikalsen, I. Rouvellou, and P. T. Devanbu. GlueQoS: Middleware to Sweeten Quality-of-Service Policy Interactions. In *Proceedings of the Twenty-Sixth International Conference on Software Engineering (ICSE)*, pages 189–199, May 2004.

[109] G. Wu, P. Havinga, and M. Mizuno. Wireless Internet on Heterogeneous Networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2001)*, pages 1759–1765, November 2001.

[110] K. Yang, A. Galis, and C. Todd. Policy-Driven Mobile Agents for Context-Aware Service in Next Generation Networks. In *Proceedings of IFIP Fifth International Conference on Mobile Agents for Telecommunications (MATA 2003)*, pages 111–120, October 2003.

[111] M. Ylianttila, M. Pande, J. Makela, and P. Mahonen. Optimization Scheme for Mobile Users Performing Vertical Handoffs between IEEE 802.11 and GPRS/EDGE Networks. In *Proceedings of the Thirteenth IEEE International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC 2002)*, pages 64–68, September 2002.

[112] Jukka Ylitalo, Tony Jokikyyny, Tero Kauppinen, Antti J. Tuominen, and Jaakko Laine. Dynamic Network Interface Selection in Multihomed Mobile Hosts. In *Proceedings of the Thirty-Sixth Hawaii International Conference on System Sciences (HICSS-36)*, Hawaii, United States, January 2003. University of Hawaii.

[113] T.B. Zahariadis, K.G. Vaxevanakis, C.P. Tsantilas, N.A. Zervos, and N.A. Nikolaou. Global Roaming in Next-Generation Networks. *IEEE Communications Magazine*, 40(2):145–151, February 2002.