UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

TRANSPARENT AND EFFICIENT IP MOBILITY

(MOBILIDADE IP TRANSPARENTE E EFICIENTE)

Pedro Miguel Santos Reis Vale Estrela

(Mestre)

Dissertação para obtenção do Grau de Doutor em
Engenharia Informática e de Computadores

Orientador:	Doutora Teresa Maria Sá Ferreira Vazão Vasques
Co-Orientador:	Doutor Mário Serafim dos Santos Nunes

Júri

Presidente:	Reitor da Universidade Técnica de Lisboa
Vogais:	Doutor Hamid Aghvami
	Doutor Mário Serafim dos Santos Nunes,
	Doutor Manuel Alberto Pereira Ricardo
	Doutor Rui Luís Andrade Aguiar
	Doutora Teresa Maria Sá Vazão Vasques
	Doutor Carlos Nuno da Cruz Ribeiro

Dezembro 2007

Acknowledgements

I would like to thank to my advisor, Prof. Teresa Vazão, for her friendship, for the scientific guidance, for the useful advice and for the research freedom that was given to me permanently from the very first day. In particular, I wish to thank for the substantial time and effort that was invested in reviewing this thesis, namely the careful verification of the text, of the formal specification, and of the results the thesis, and of the submitted papers and the conceded industrial patent. On top of that, I wish to deeply thank the permanent and unconditional encouragement, without which this thesis would not have been possible. I would like also to thank my co-advisor, Prof. Mário Nunes, for his friendship and encouragement since my undergraduate studies, for the scientific guidance and the reviewing of the thesis, papers and patent that he performed.

In second place, I'm deeply grateful to Dr. Patricia Lima, my INESC-ID research group assistant, for her extremely careful and interested spelling, grammar and style reviewing of this thesis text. Undoubtfully, Patricia's help, that far exceeded the typical group assistant work assignments, was an essential factor to greatly improve the legibility, flow and overall quality of this thesis text, the submitted scientific papers and the conceded industrial patent.

I would like to thank my colleagues for their partnership and useful input at periodic presentations, namely Ricardo Pereira, António Varela, João Oliveira and Ricardo Seabra.

I would like also to mention and thank both INESC-ID TagusPark and IST Taguspark, for providing me the resources I needed to perform this research work, and FCT, for the three and a half year PhD scholarship I was given.

To conclude, I would like to thank my family, especially my Parents, Sisters and brothers-in-laws for their encouragement and support, and to my friends Nuno Baptista, Pedro Virote and Bruno Espadinha for their friendship and interest. Last but not definitely the least, I would like to thank Silvia Gomes for all the support given since we first met.

Abstract:

This PhD research work will develop original contributions to the field of IP mobility, focusing on efficiency and transparency support, by proposing a mobility solution that efficiently supports any kind of terminals and networks. A full state-of-the-art review and evaluation via classificative frameworks is performed in order to evaluate the possible contributions of previous research work on the solution to be proposed in this PhD Thesis, being chosen to be extended a previous global proposal that already featured a terminal independent solution – TIMIP/sMIP. Taking this previous work as a base, this PhD work proposes the "enhanced TIMIP" micro-mobility protocol, which uses a new overlay network design to enable efficiency and transparency gains, suitable for a flexible application of the mobility service in particular deployment scenarios of both existing and future all-IP networks.

This generic architecture is applicable to both IP versions, and supports mobility-unaware legacy terminals and legacy routers, covering both immediate and future utilizations, through the use of terminal and network independence. The solution is composed of a base protocol that provides better transparency with similar efficiency levels as the best alternative solutions, by featuring fast handovers and tree-optimal routing. Then, a set of modular extensions can be combined to form the full protocol, which provides better efficiency and better transparency than the best alternative solutions, by additionally featuring seamless handovers and optimal routing. In addition, idle terminals and operator-centric scenarios also supported by specific eTIMIP extensions.

The proposed solution was formally specified with state machines, and has been evaluated and compared to alternative solutions via simulation studies in the NS2 simulator. This evaluation process has shown the applicability of the proposed solution for this PhD thesis objectives, by having better combined transparency and efficiency features than any of the other studied state-of-the-art solutions.

Keywords: eTIMIP, IP Mobility, Micro-Mobility, Seamless Handovers, Mobility Control, Transparency, Surrogate MIP, 802.11, NS2.

Resumo:

Este trabalho de Doutoramento propõem contribuições originais no ambito da Mobilidade IP, sendo focado no suporte de Transparência e Eficiência, ao propor uma solução de mobilidade que suporta eficientemente qualquer tipo de terminais e redes. Uma análise completa ao estado da arte desta área foi efectuada de forma a identificar as contribuições possíveis do trabalho de investigação anterior para a solução a ser proposta nesta Tese de Doutoramento, tendo sido escolhida uma proposta anterior que já apresentava uma solução de mobilidade independente dos terminais - TIMIP/sMIP. Tendo por base este trabalho anterior, este trabalho de investigação propõe o protocolo de micro-mobilidade "enhanced TIMIP", que utiliza um desenho "duplo-plano", que permite ganhos de eficiência e transparência, e que permitem a aplicação flexível do serviço de mobilidade em casos particulares de aplicação de redes existentes e futuras baseadas no protocolo IP.

Esta arquitectura genérica é aplicável a ambas as versões do protocolo IP, e suporta tanto terminais legados como encaminhadores legados sem suporte de mobilidade, utilizando a independência de terminais e redes. A solução está inicialmente divida num protocolo base, que combina um nível de transparência superior com um nível de eficiência semelhante às melhores soluções alternativas, ao suportar *handovers* rápidos e encaminhamento óptimo numa árvore. Adicionalmente, são definidos um conjunto de extensões modulares que, quando utilizadas em conjunto, combinam um nível de transparência e eficiência superior às melhores soluções alternativas, ao suportar *handovers seamless* e encaminhamento óptimo, alem de suportarem terminais inactivos e cenários de operadores por via de extensões especificas.

A solução proposta foi especificada formalmente utilizado maquinas de estado, e foi avaliada e comparada com alternativas existentes por via de estudos de simulação no simulador de redes NS2. Esta avaliação mostra a aplicabilidade da solução proposta em relação aos objectivos da tese, por ser mais transparente e eficiente que qualquer das outras soluções consideradas do estado da arte.

Palavras Chave: eTIMIP, Mobilidade IP, Micro-Mobilidade, Seamless Handovers, Suporte de mobilidade Transparência, Surrogate MIP, 802.11, NS2.

Table of Contents

Table of Appendixes

Table of Figures

Tables List

Table of Acronyms

4G	Fourth Generation Mobile Data Networks
802.x	Set of IEEE Standards for LAN Protocols
802.1x	IEEE standard for Port Based Authentication
802.3	IEEE standard for Ethernet
802.11	IEEE standard for Wireless LAN (aka WIFI)
802.11b	802.11 update for 2.4GHz band operation (11Mbits)
802.16	IEEE Standard for Wireless Metropolitan Area Networks (aka WiMAX)
802.16e	Mobile version of 802.16
802.21	IEEE Standard for Media Independent Handover Services
AAA	Authentication Authorization and Accounting
AC	Access Controllers
AIMD	Additive Increase / Multiplicative Decrease
ANG	Access Network Gateway
ANP	Anchor Point
AP	Access Point
AR	Access Router
ARP	Address Resolution Protocol
ARS	Access Routers
BCMP	Brain Candidate Mobility Protocol
BGP	Border Gateway Protocol
BS	Base Stations
CBR	Constant Bit Rate
CGA	Cryptographically Generated Addresses
CIDR	Classless Inter Domain Routing
CIMS	Columbia IP Micro-Mobility Suite
CIP	Cellular IP
CN	Correspondent Node
CWND	Congestion Window
DAD	Duplicate Address Detection
DCCP	Datagram Congestion Control Protocol
DHCP	Dynamic Host Configuration Protocol
DIFFSERV	Differentiated Services Model
DNA	Detecting Network Attachment
eTIMIP	Enhanced TIMIP
eTIMIPv4	eTIMIP for IP Version 4
eTIMIPv6	eTIMIP for IP Version 6
FA	Foreign Agent
FMIP	Fast MIP
FSM	Finite State Machines
FTP	File Transfer Protocol
GDA	Generic Detection Algorithm
GFA	Generalized Foreign Agent
GPRS	General Packet Radio Service
GW	Gateway
HA	Home Agent
HAWAII	Handoff-Aware Wireless Access Internet Infrastructure
HIP	Host Identity Protocol
HMIP	Hierarchical MIP
IDMP	Intra-Domain Mobility Management Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IMS	IP Multimedia Subsystem
INTSERV	Integrated Service
IPIP	IP in IP tunnel
IS-IS	Intermediate System To Intermediate System
ISI	Information Sciences Institute
L2	Layer 2

L3	Layer 3
L3AP	Layer 3 Access Points
LMN	Legacy Mobile Node
LR	Legacy Router
MAP	Mobility Anchor Point
MEHROOM	Micro Mobility Support with Efficient Handoff and Route Optimization Mechanisms
MIB	Management Information Bases
MIH	Media Independent Handovers (802.21)
MIHF	Media Independent Handovers Function
MIP	Mobile IP
MN	Mobile Node
MPLS	Multi-Protocol Label Switching
MSF	Hawaii Multiple-Stream-Forwarding
NAI	Network Access Identifier
NAT	Network Address Translation
ND	Neighbour Discovery
NE	Network Elements
NGI	Next Generation Internet
NI	Network Independence
NOAH	No-AdHoc Agent
NS	Neighbour Solicitation
NS2	Network Simulator v2
OLSR	Optimized Link State Routing
OO	Object-Oriented
OOO	Out-Of-Order Packets
OSPF	Open Shortest Path First
PDP	Packet Data Protocol
PEAP	Protected Extensible Authentication Protocol
PFANE	Previous FA
PI	Peer Independence
QoS	Quality of Service
R&D	Research and Development
RIP	Routing Information Protocol
RO	Route Optimisation
RS	Router Solicitation
RTP	Real Time Protocol
RTT	Round Trip Times
S-MIP	Seamless MIP
SEND	Secure Neighbour Detection
sFA	Surrogate Foreign Agents
sHA	Surrogate Home Agent
SLAAC	Stateless Address Auto-Configuration
sMIP	Surrogate MIP
SIP	Session Initiation Protocol
SPI	Security Parameter Index
TI	Terminal Independence
TIMIP	Terminal Independent Mobile IP
tMIP	Transparent MIP
TR	eTIMIP Router
TTLS	Tunnelled Transport Layer Security
UMTS	Universal Mobile Telecommunication System
VA	Visitor Agent
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
WMN	Wireless Mesh Networks

1 Introduction

1.1 Future All-IP Networks

Currently, major efforts are being made towards the unification of all existing networks into a common one, named the Next Generation Internet (NGI), which is expected to evolve from the current Internet and its heterogeneous IP protocol. This NGI will be increasingly important and complex, as users may use it to connect all types of equipment and access a broad range of services [82] [88].

One aspect of this trend is the design of the future All-IP mobile data networks, which is expected to be a fully multi-access, multi-service, technology-heterogeneous network that offers global roaming mobility and service portability [89] [84]. Such unified network will support multiple forms of communication - namely voice but also multimedia and data - with higher quality, capacity, coverage, reliability and much lower costs and higher productivity gains than today's current options [81] [182] [83]. These advantages will be made possible through a number of technological advances and architectural shifts, namely the adoption of a fully IP-based, digital, packet-switched network, coupled with new transmission technologies and coding schemes that support much higher user data rates than the ones currently available, at the wireless access networks [77] [78] [79].

Major cost savings will be possible through the use of a single heterogeneous network, common to all kinds of utilizations, instead of the present situation, where multiple networks are needed to access a more restricted set of services. Cost reduction is possible mainly due to the use of a unified and scalable infrastructure, which integrates management and services; other cost reductions are made possible through the use of a packet switching architecture that allows an efficient sharing of the network resources among the users, possibly using unlicensed frequencies [76].

Such next-generation mobile data networks will be fully merged with the current Internet, by using the All-IP architecture [94] [85] [89]. This future architecture will be the result of a series of evolutions of the present IP stack, which is currently characterized by its best-effort, secure-less and non-mobile properties. This IP shift in mobile networks is already causing a major increase in the number and variety of devices connected to the global Internet, such as the low-battery, low-complexity and small mobile devices [84]. Currently, this increase is feeding the exponential Internet growth in terms of size, users and information quantity [82]. Using these aspects together, new forms of business will be made possible, the successor of the current "e-business" already being called "m-business" [183].

1.1.1 All-IP Networks Technology requirements

Today, significant consensus exists for some key aspects that must be present in this future All-IP architecture [87]. Besides being based on IPv6 protocol [80] [86], which supports the necessary scalability features and address space to accommodate all the envisioned entities, the future IP architecture must also cover the essential features of Quality of Service (QoS), Security and Mobility Management [78]. All these aspects are under major research and development (R&D) activity at the moment, having already been proposed new protocols and schemes that address these issues, as described in different surveys [130] [139] [47].

Regarding **Quality of Service**, the Internet Engineering Task Force (IETF) proposed several solutions to provide its support in the current IPv4-based Internet. The Differentiated Services (DiffServ) model [131] provides heterogeneous QoS through a limited set of service classes, achieving a scalable but non-optimal QoS solution. The Integrated Services (IntServ) model [132] provides strict QoS guarantees, but does not scale well, and hybrid scenarios may be envisaged, comprising both of them, as described in reference [133]. Multi-Protocol Label Switching (MPLS) is another technique that offers traffic engineering capabilities [134]; merged with DiffServ, it can be used to provide efficient QoS support, as stated in reference [135]. From a QoS point of view, all these technologies - DiffServ, IntServ and MPLS – are expected to coexist, playing a complementary role on the NGI [136]. These IP layer heterogeneous QoS support features are also expected to be complemented with specific Layer 2 (L2) QoS support mechanisms on the access networks, fairly close to the users [137].

Concerning **Security**, several solutions have also been proposed in different layers to provide authentication, authorization and encryption of mobile users and their data flows. For this, the Institute of Electrical and Electronics Engineers (IEEE) has proposed the 802.1x solution [141], which is an L2 authentication system, accepting multiple forms of authentication. This standard has been followed by stronger, although non-standard variations (Tunnelled Transport Layer Security (TTLS) [145] and Protected Extensible Authentication Protocol (PEAP) [146]). Recently, the 802.11i standard was proposed to offer better security features to 802.11 WLANs [143], replacing the initial weak 802.11 Wired Equivalent Privacy (WEP) scheme [125]. On the other hand, the IETF has been standardizing heterogeneous higher-level approaches, using IPsec tunnels [140] to build Virtual Private Networks (VPNs) [147]. This IP extension is provided natively in IPv6 networks, and has been backward-introduced in current IPv4 networks. The VPNs are then introduced with the aid of auxiliary Authentication, Authorization and Accounting (AAA) protocols, such as RADIUS [142] and DIAMETER [148].

Concerning **Mobility Management**, Mobile IP (MIP) [1] is currently the standard network layer macro-mobility solution, providing heterogeneous and transparent mobility support to all technologies and applications. MIP features scalable mechanisms, essential for wide area mobility, where handover performance is not a major issue. The corresponding protocol was also introduced as the macro mobility solution for IPv6 networks (MIPv6) [2]. Concerning micro-mobility, quick and smooth routing changes are required in order to achieve seamless handovers, providing both low latency and low loss handovers, and a better optimisation of network's resources. Work in this area has resulted in multiple proposals over the last few years, but up to now, there is no sufficient consensus, neither in the scientific community nor in the industry, regarding the adoption of one of the existing state-of-the-art proposals as the standard solution.

Other solutions exist that provide mobility services in other layers of the protocol stack, although with different perceived efficiency and transparency features. Of these, some representative protocols are heterogeneous mobility support in the transport layer provided by the TCP-Migrate protocol [68], or the Session Initiation Protocol (SIP) for application-layer mobility [66]. The Host Identity Protocol (HIP) [70] is still another heterogeneous solution, where a new namespace layer is introduced to decouple the host identification from the location information that IP addresses inherently have. All these protocols can be aided by the Media Independent Handovers function of the emerging 802.21 standard [69]. Non-heterogeneous access mobility may be also provided in General Packet Radio Service (GPRS) or Universal Mobile Telecommunication System (UMTS) for L2 mobility [67].

However, the solutions that operate at the Network layer are considered by various authors as the most suitable for the Next Generation Internet [47] [76] [80] [90] [89]. Generically, such is mostly the result of the combination of a better **efficiency to transparency relation**. For efficiency, such generically happens through the usage of efficient micro-mobility technologies that optimize the handovers and the network resources [43] [44]. For transparency, such generically happens because IP layer mobility solutions are the ones that more naturally extend the current Internet into the envisaged NGI, by providing a transparent service that supports all existing applications, fixed nodes, transports and other IP protocols [1] [177] [87].

1.2 IP Mobility State-of-the-Art Overview

1.2.1 IP Macro-mobility

Today, in the research field, Mobile IP is the standard mobility solution for IPv4 networks, which enables global roaming through the whole Internet, in a heterogeneous approach. The protocol is best suited to providing mobility in wide area networks, typical of nomadic computing, where scalability is more critical than handover performance. This protocol has been designed to keep the compatibility with the existing fixed infra-structure and routing, only requiring modifications to the Mobile Node's (MN) protocol stack and the deployment of mobility agents in the home and, generally, in the visited sub-networks. Currently, this protocol is already identified as the base mobility protocol for the future Fourth generation (4G) core networks, which is a significant step for a future All-IP solution [80] [81] [43], and is already being used or studied for current 3G solutions [71] [95] [128].

With the development of IPv6, the original MIPv4 protocol was upgraded to a corresponding v6 version – MIPv6 – which natively addressed some of MIPv4's open issues, by taking advantage of the new IPv6 features. The new version also improved the original's applicability, by dropping the Foreign Agents' requirements in visited networks, replacing these functionalities with new IPv6 capabilities. On the other hand, if the MN's peers, called correspondent nodes (CN), also support MIP, then optimized communication is possible between them, with a fairly good level of security [14]. Thus, with the new routing and security mechanisms, this optimized version is able to properly support scalable macro-mobility scenarios for future networks. In both IP versions, work has already been done concerning the integration of MIP's scalable macro-mobility with the other appointed all-IP networks requirements, namely the QoS and Security, these integration aspects being covered in recent research work [156]-[161].

1.2.2 IP Micro-mobility

A complementary research field of this topic is micro-mobility, focusing on efficient and local mobility support in limited scenarios, up to IP domains. This mobility type has several important differences comparing to macro-mobility, being expected a much larger number of mobile users in movement simultaneously, each of them moving inside a set of high-bandwidth pico cells [104], and/or at high speed [43]. For supporting seamless handovers in these scenarios, fast and smooth routing changes are required at each movement, in order to reduce the handover latency and the handover packet losses [44].

For this task, a scalability-to-efficiency compromise is introduced, where more efficient mobility operations are used, in loss of only being scalable to limited portions of the Inter-

net [42] [44]. To achieve global roaming, MIP's macro-mobility operations are used to address the nomadic infrequent movements between domains [21]. By keeping such mobility management on a local level, handover signalling can be kept outside the core networks for the majority of the movements, which further improves the scalability of the global mobility service [42].

The research work in this area has led to the production of several proposals, which complement MIP for the handling of the majority of movements within a local scope. Examples of these are: the work of Cellular IP (CIP) [20], HAWAII [21], Hierarchical MIP (hMIP) [22], Fast MIP (fMIP) [24], Network-based Localized Mobility Management (netLMM) [105] and Terminal Independent Mobile IP (TIMIP) [26] [27] protocols, among other proposals. However, up to now, there is no sufficient consensus, neither in the scientific community nor in the industry, regarding the adoption of one of the existing state-of-the-art proposals as the standard solution. This way, it is equally viable to extend, merge or even propose entirely new solutions to address micro-mobility.

1.2.3 IP Micro-mobility comparison

One of the key factors used to distinguish and classify the mobility protocols is their relative **Efficiency**, which results from their internal mobility processes, algorithms and architectures [43]. Concerning IP mobility protocols, the efficiency metric is mostly influenced by the support of seamless handovers and mobility overhead [62]. The handover latency is improved by detecting the movement as soon as possible, while keeping the registration updates as close as possible to the MN (fast handovers); the handover packet loss is controlled by retransmitting, duplicating or buffering the handover lost packets, and by avoiding the reordering of packets in the established flows (smooth handovers). The resources utilization is improved by routing the data packets through the shortest paths inside the network, with the simplest encapsulation, and by using the MN's own data packets to minimize state maintenance overhead; the resources are also optimized when the signalling messages are kept off the core networks, concerning both the handover propagation and the periodic state maintenance operations.

Another metric that distinguishes the mobility protocols is their **Transparency** level, which measures the amount of modifications necessary to introduce the mobility service in existing IP fixed networks. As such, it can range from an immediate utilization without any changes, up to a complete redesign of network entities, equipment, topologies or clients; the intermediate case consists in a smooth upgrade option that supports backward compatibility with current fixed network's entities - in particular the legacy mobile terminals to which the mobility service is provided. This backward-compatibility notion is one of the most important design points of the Internet's evolution [104], with proven success in many well-known scenarios where the introduction of new support features didn't preclude the already deployed infrastructure [99] or protocols [100]. Despite its importance, this metric has received relatively lower attention than efficiency in the IP mobility research work. Very recently, the observed lack of either local or global mobility standardized deployment has contrasted with the successful adoption of proprietary localised management mechanisms in WLAN switches, which do not impose changes in the terminals [104]. This has essentially resulted in the re-identification [103], by the IETF, of the previously identified problem in reference [97]. This decrease of importance of efficiency as the key factor in protocol design, here

exemplified with transparency, is also present in other areas of the Internet evolution [177] [102][1].

Regarding the mobility protocols, their transparency level can vary according to the required support at the mobile nodes, at the fixed network, and at the MN's peers. The terminal's transparency is improved by maintaining the MN mobility-aware support modifications as concentrated as possible, via the addition of a single standard IP mobility client, without any transport or application modifications. However, immediate utilisation is reached only when existing legacy terminals with regular IP stacks are supported without any modifications at all [97]. The network's transparency is achieved when the introduction of the mobility service is not disruptive to the pre-existing network's legacy routers and topologies. In this scope, the network transparency is improved as less nodes are required to have mobility-aware capabilities in the global roaming support. The peer's transparency requires that pre-existing mobility-unaware fixed nodes are to able to communicate transparently to the MNs without any modification, as to support the immediate utilization of the MN with all the existing fixed infra-structure. However, if mobility-aware functions are required to be introduced in the peers, then these modifications should be as limited as possible, using a single mobility client residing at the IP layer (similar to the terminal's transparency).

1.3 Objectives and Contributions of this PHD work

1.3.1 Objectives

From what was described previously, the future heterogeneous NGI will be the result of a series of improvements to the existing Internet, particularly with technologies that enable the support of QoS, Security, and Mobility, all of which are being the subject of major research work at present. Regarding Mobility, the IP layer solutions currently emerge as the ones that are most suitable to provide mobility support to future heterogeneous mobile data networks, capable of featuring both **scalability** and **efficiency** in the mobility management mechanisms, coupled with **transparency** in regard to the currently deployed Internet infrastructure.

As the **scalability** problem has been essentially solved through the macro and micro mobility separation [44] [21] [42], and as the standard-less IP micro-mobility topic has the majority of identified open issues [103] [43] [94], this thesis will develop contributions in the area of IP Micro-Mobility, introducing innovations that provide increased efficiency and transparency aspects simultaneously and in a modular way, suitable for a flexible application of the mobility service in particular deployment scenarios of heterogeneous all-IP networks.

As such, from the set of open issues identified in the current state-of-the-art IP mobility solutions, this research work will consider the following objectives:

- Definition of a **heterogeneous mobility protocol** addressing global IP mobility support, compatible with this area's present standards, and applicable to both the current (IPv4) and the future IP networks (IPv6);

[1] Quoting from Clark et al. [102], page 467, "…solutions that are less efficient from a technical perspective may do a better job of isolating the collateral damage of tussle…"

- Support of mobility **efficiency**, capable of enabling a low latency, low loss, low signalling and low data overhead mobility service, through the support of seamless handovers and efficient resource utilisation;
- Support of mobility **transparency**, capable of enabling the immediate utilisation of the mobility service and smooth incremental upgrades of the existing infrastructure, through the support of mobility-unaware legacy terminals, legacy routers, and fixed legacy correspondent nodes.
- Support of protocol **modularity**, capable of enabling a flexible trade-off between the above mentioned efficiency and transparency objectives, but also capable of enabling further scalability, reliability and control gains, suitable for a flexible application of the mobility service in particular deployment scenarios.

1.3.2 Contributions

For the accomplishment of the above proposed objectives, this thesis presents the following set of contributions. For each contribution, the specific chapter where it is located is identified:

- A **literature review** of the current IP macro and micro mobility state of the art, focusing on the efficiency and transparency capabilities of the protocols, and complemented with the description of above-IP layer mobility solutions (Chapter 2).
- An **evaluation** of the state-of-the-art options for the fulfilment of this PhD thesis' objectives, through the use of a pair of classificative taxonomic generic frameworks, covering the efficiency and transparency aspects both separately and in combination, in order to support the choice of the previous micro-mobility research work to be extended in this PhD thesis (Chapter 3).
- The development of a generic solution - **enhanced TIMIP** (eTIMIP) - a backwards compatible secure modular micro-mobility protocol that supports flexible efficiency and transparency using an overlay network and mobility subnet mechanisms. The **base protocol**, which features better transparency with similar efficiency levels as the best alternative solutions, is also designed to support **complementary extensions** that provide further efficiency, scalability, reliability and control gains (Chapter 4).
- The formal **specification** of the eTIMIP protocol using concurrent finite state machines, and the informal specification through natural language and illustrative examples, coupled with the adaptation of the generic protocol for usage in IPv4 and IPv6 networks, through the definition of the required control and data packet formats for each IP version (Chapter 4).
- An **evaluation** and comparison with alternative proposals of the efficiency and transparency features of the proposed base solution, via simulation studies, in multiple sets of common deployment scenarios and using multiple metrics (Chapter 5).
- The formal specification using concurrent finite state machines, and the informal specification through natural language and illustrative examples, of four **eTIMIP extensions** that can be optionally and modularly combined to provide better efficiency than both the base protocol and the best alternative solutions (through the support of low latency, zero-loss handovers, low signalling and low data overhead mobility ser-

vices), and can be used to provide more scalability, reliability and control gains than the base protocol (Chapter 6).

- An **evaluation** of the efficiency and transparency features of the proposed extensions, those being compared to the base protocol in increasing levels of combination, using the same simulation studies scenarios as the initial evaluation of the base protocol (Chapter 7).

Besides the mentioned contributions, this PhD thesis also contains further minor contributions that will be presented in the first section of each individual chapter.

1.4 Organisation of this Thesis

This thesis consists of eight chapters and seven Appendixes.

This chapter places the thesis research in the context of All-IP networks, presents a preliminary analysis of the IP mobility state-of-the-art protocols, introduces the thesis' objectives and the obtained contributions.

The second chapter presents the current state-of-the-art developments in this area, relating each selected proposal with this thesis' objectives.

The third chapter presents the in-depth study of both the efficiency and the transparency topics, through taxonomic frameworks that classify and compare the selected state-of-the-art proposals for the fulfilment of this PhD thesis' objectives.

The fourth chapter describes the generic eTIMIP architecture through both formal and informal specification forms, and is complemented with the adaptation mechanisms description for IPv6 and IPv4 networks.

The fifth chapter presents performance results via simulation studies, of the base protocol in comparison with other alternatives, covering seamless handover evaluation, routing efficiency, link failure reliability and transparency-to-efficiency trade-off evaluation.

The sixth chapter describes the eTIMIP extensions that extend the base protocol with further efficiency, scalability, reliability and control gains, those being specified with the same methodology as the base protocol.

The seventh chapter presents performance results via simulation studies, of the base protocol in comparison with increasing levels of combination of the proposed extensions, in the same scenarios as the fifth chapter.

The last chapter summarizes the main results and contributions of this thesis and provides directions for further work.

Finally, the main text is complemented with a series of appendixes that cover in more detail additional topics that support the research contributions, namely the full finite state machines specification, the full eTIMIP packet formats, the software description and the formal analysis of the soft de-triangulation algorithms.

2 Related Work

This chapter will describe in more depth the current state of the art of the IP mobility field. For this study, the existing IP standards will be described, alongside the selected proposals that are most related to this thesis' objectives. Some proposals, which are essentially compositions or minor extensions of the base protocols, are also briefly mentioned. In each case, only a brief survey of the protocol's architecture, operations, advantages and weaknesses will be given. More specific details of the protocols can be found in the references given. The IP state-of-the-art study is then complemented with a description of representative above-IP mobility protocols, which can also be used to support heterogeneous mobility support at the higher layers of the network stack, and of 802.21, which can aid those with standardised L2 triggers and control.

The rest of this chapter is organized as follows: the first section will be focused on IP macro-mobility protocols; the second will present the state of the art of the IP micro-mobility protocols, and the final section will describe selected heterogeneous mobility protocols that operate above the IP layer.

2.1 IP Macro-Mobility Protocols

This section describes mobility solutions that are suitable to large scale movements, between domains. This type of mobility has been extensively studied in the past [21] [49], namely their scalability-to-efficiency compromise design. Although such protocols are scalable enough to cover the entire Internet, they do not give a fair efficiency, for the generality of situations and movements. As such, these protocols are only suitable to address infrequent or nomadic mobility.

2.1.1 MIPv4

The MIPv4 [1] protocol is the classical solution proposed by the IETF, being suitable for large movements, typical of nomadic computing. The protocol considers that hosts are reached using two global complementary addresses, to solve the classical "identifier" vs. "locator" problem that IP addresses have [14] [15]: the Home Address is a unique IP address used by the correspondent nodes to contact the MN in all locations, serving as a constant identifier; the CareOf address is a second IP address that reflects the actual MN's localization in the visited networks, changing each time it moves between networks and being used by MIP as a temporary MN locator [14]. Thus, the main objective of the MIP protocol is to redirect the packets received in the Home Network to the current Visited Network, by keeping the MN's CareOf Address updated as the MN moves between networks.

To provide this mobility service, the MN's stack must be upgraded with the MIP protocol, and specialized mobility agents must be added to the home and visited sub-networks: the Home Agent (HA) and Foreign Agent (FA) respectively. Every time the MN moves, the terminal detects this event by the FA's generated MIP beacons, starting a registration process

needed to inform the HA of the MN´s current CareOf address[2]. Based on this information, the HA can collect the packets destined to the HomeAddress and forward them to the current FA encapsulated in a tunnel. Thus, an high degree of transparency is attained, as the CNs always use the global identifier (HomeAddress) to contact the MN, without having any knowledge of the MN's mobility. On the other hand, by using tunnelling encapsulation facilities, compatibility with the existing routers infra-structure is achieved.

Although the basic MIP protocol presents a scalable solution to wide area mobility, it also suffers from several inefficiencies and other faults, which have been addressed in further research work. As all MN movements force a registration at the HA, and are detected with MIP beacons only, long latencies in each handover are expected which can result in packet losses, additional handover latency and throughput degradation. Generically, the solution to these problems has been relegated to the macro / micro mobility approach, enabling MIP to handle the rare inter-domain movements only. Handover efficiency improvements have been achieved by aiding the movement detection process with lower-layer triggers, and using local registration processes and buffering techniques.

Another problem, designated by "triangulation", relates to the data packets routing inefficiency that the transparency imposes, as these packets always pass through the MN's home domain. This problem has been addressed by a route optimization (RO) feature, where the CNs can send their data packets directly to the MN's current location [10]. In this scheme, every time the MN moves, it notifies both the HA and the CNs of the new location (e.g., its new CareOf Address). Using this information, the CNs can then associate the CareOf Address to the Home Address of the MN, and thus send the data packets directly to it, completely bypassing the HA. While this feature can greatly increase efficiency, by improving end-to-end delay and the resources utilization, it also incurs in a loss of transparency, as it additionally requires mobility-aware functionality in the CNs.

On the macro mobility context, other open issues of the original standard have also been addressed in further research works, namely the problems associated with the mobile users identification [5], authentication [4] [6], the fault tolerance of mobility agents [8] [9] and the integration with Virtual Private Network / IPsec scenarios [11]. As such, the major identified MIPv4 problems are those which have been relegated to micro-mobility scope, and the lack of transparency concerning MNs and CNs (for route optimization scenarios).

2.1.2 MIPv6

With the development of IPv6, the original MIPv4 protocol was upgraded to a correspondent v6 version – MIPv6 [2]. This support was needed because while the new IPv6 protocol introduces larger addressing, scalability and optimization features than the IPv4 protocol, it continues to not offer mobility services in the base protocol. The MIPv6 protocol is fairly similar to the previous version, being essentially the same protocol as MIPv4. However, it is more efficient and modular, by using the new IPv6 options, and by incorporating in its base version the most MIPv4 mature extensions, especially the previously described route optimization component. Thus, with the handover, routing optimizations and the security mechanisms that were introduced, this version is able to properly support macro-mobility scenarios

[2] The CareOf Address can either be the FA's own IP address, or another local address acquired by other means, like Dynamic Host Configuration Protocol (DHCP); the former case is preferable as it does not require additional IP addresses allocation per MN.

for future IPv6 networks, being for this way considered the standard solution for these scenarios.

The new version also improved the protocol applicability, by dropping the FA requirement at each visited network. For this, the MNs use new optimized Layer 3 (L3) algorithms to detect their movements, which depend on the standard IPv6 router advertisements only. At each movement, the MN generates a new CareOf Address using the stateless auto-configuration method [114], and informs the HA and its peers of the new location, using binding update messages encapsulated in the new MIPv6 mobility header, which could be piggy-backed in regular data packets [2]. However, this transparency improvement can result in lower robustness, in the case that MN and CN are both mobile and perform a handover at approximate instants; this simultaneous mobility problem has been addressed in further research work [50].

Concerning its HA, a standard IPsec authentication header is used, requiring the pre-existence of strong shared secrets, as in MIPv4; however, broader applicability is attained to the general MIPv6-enabled CNs, as a weaker authentication method is used. This method doesn't require shared secrets between the end nodes, relying only on the MN-HA authentication and on the routing consistency of the regular, non-mobile, Internet fixed routing [14]. Using this information, a MIPv6-enabled host can map the MN's IP identifier to the current locator (Route Optimization mechanism). Unlike MIPv4 that uses full IP-in-IP (IPIP) tunnels in all cases, MIP-aware CNs can send data packets efficiently directly to the MN's CareOf Address, identifying the Home Address by the aid of the new IPv6 routing header (routing header type 2 [2]).

As before, non-MIP aware CNs default to use regular MIP routing triangulated via the HA; however, another difference of MIPv6 regarding the legacy CNs is that the data packets destined to those nodes are additionally triangulated through the HA, instead of being sent directly as in MIPv4 (bi-directional tunnelling). In this sense, the triangulation problem of MIPv4 has become the quadrilateral routing problem of MIPv6 if the route optimization option is not supported [50].

On the macro mobility context, other remaining open issues of MIPv6 are centred essentially on small improvements to the already strong authentication and security processes [12] [13] [16] or other minor topics like the MN's automatic bootstrapping [17], MN's identification [19], or integration with the higher layers [18]. As such, the major identified MIPv6 problems are the ones that, as in MIPv4, have been relegated to micro-mobility scope; again, the lack of transparency for MNs and CNs (in route optimization scenarios) have not been addressed by MIP.

2.1.3 Transparent MIPv4

The transparent MIP (tMIP) protocol [101] was a first approach for the support of a transparent mobility service in the Internet, although without special efficiency or standards compliance considerations. As it was already described, such need exists because the MIP standard only provides backwards compatibility to the CNs and the networks, which forces the long and difficult MN upgrade migration cycle already mentioned. This protocol addresses exactly such problem, by presenting a mobility mechanism which does not require any changes to the MN's IP stack. Such is possible by designing the network with the capability to automatically track and change its own routing to reflect the current Legacy Mobile Nodes (LMN) locations, maintaining their connectivity.

Although based in similar principles of the standard, tMIP is not compatible with MIP, as it only uses data tunnelling mechanisms without signalling procedures. Each visited network contains a mobility Visitor Agent (VA) that detects LMN arrivals with their foreign Adrress Resolution Protocol (ARP) requests[3]. In such case, the VA responds with its own address, and tunnels all LMN traffic to the corresponding HA, which is identified via a global centralized server. When the HA receives the tunnelled packets, it learns the new LMN location by the outer tunnel's IP addresses, enabling it to subsequently collect packets destined to this LMN, and tunnels them to the current VA. As such, these operations are sufficient to provide a transparent macro-mobility service.

Although achieving one of the goals previously mentioned – transparent mobility service to MNs – this proposal suffers from several important drawbacks, which make it an inadequate global mobility solution. Hence, the most important disadvantages are: the lack of compatibility with the MIP standard, the limited detection scheme, which depends on the transmission of data in each new MN's location; the requirement of a centralized Home Agent locator entity for the entire Internet; the inexistence of IPv6 support and the lack of micro-mobility optimizations.

2.2 IP Micro-Mobility Protocols

This section describes mobility solutions that are suitable to small scale movements, inside domains only [43]. Such mobility type has several important differences comparing to macro-mobility, being expected a much larger number of mobile users in movement simultaneously, each moving inside a set of high-bandwidth dense pico-cells [104], or at a high speed. For supporting seamless handovers in these scenarios, fast and smooth routing changes are required at each movement, in order to control the handover latency and packet losses that result from each handover [73] [44].

For this task, a scalability-to-efficiency compromise was introduced, where more efficient mobility operations are used, but which may only be applicable inside limited portions of the Internet, e.g. administrative domains. To achieve global roaming, MIP's macro-mobility operations are used to address the nomadic infrequent movements between these domains [21]. By keeping such mobility management on a local level, some mobility-related signalling can also be kept outside the core networks, which also aids scalability of the mobility service as a whole [53] [42]. Some definitions of the relation between the two classes of mobility protocols have already been proposed (tight / loose coupling) in reference [46].

2.2.1 hMIPv4/v6

A recent MIPv4 extension called hierarchical MIP (hMIP) [22] was proposed to extend the MIPv4 protocol with micro-mobility capabilities, enabling faster handovers and better scalability. For this, the single HA-FA tunnel is extended to a hierarchy of FAs and the MNs will manage multiple hierarchical CareOf addresses, one per hierarchical level.

A major discussion topic within the IETF standardisation body was performed to decide if this FA hierarchy should be limited to two levels, or if it should be allowed to have more

[3] Quoting from the tMIP specification [101], page 1862, "the VA must implement a proxy ARP server with the capability of answering all ARP request that are not intrinsic to the LAN".

levels. While initial discussions tended towards the former, there were equally strong proposals that featured the latter. Finally, in order to maintain robustness and to not increase the data forwarding delays to the MN, the latest specifications opted for the former option, by limiting this generalization to two levels only [34][4].

When the MN enters a domain for the first time, it will associate its FA CareOf address with the domain's Generalized Foreign Agent (GFA), and the address of the GFA (called regional CareOf address) with the HA. Later, when the MN moves inside a domain, it only has to notify both the new FA and the local GFA of the current domain. At this node, the local tunnel is redirected to the new FA, as the HA-GFA tunnel remains valid. This optimization is able to remove the HA from the critical path registration, and to hide the local movements from the HA and the core networks, resulting in smaller registration latencies and better scalability in comparison with the regular MIP processes. The remaining operations are similar to standard MIP, as hMIP only optimizes the registration process. It should be noted that, although hMIP hides the local handover operations from the HA and the other core networks, it still requires the periodic state maintenance of the HA-GFA soft-state tunnel.

The hMIP extension was also been ported to the MIPv6 protocol, where it essentially shares the same objectives – the support of local mobility to support both fast handovers and increased scalability [33] [34]. Unlike the previous version, the hMIPv6 protocol does not extend the FA agents, as such entities have been simplified; instead, to give similar support, the HA entity is extended to a generalized agent with regional HA support. Each domain can now have a mobility anchor point (MAP), to be used as an anchor point to it. The existing FAs adjacent to the MNs need also some minor modifications related to the local HA discovery by the MNs.

When the MN enters the domain, the IPv6 router advertisements can be extended to contain also the MAP information for this domain. Using this additional information contained in the MIP beacons, the hMIPv6-aware terminal can send binding updates messages to both its HA and this local MAP. Then, subsequent handovers inside the same domain only require that terminal performs a binding update to the current domain's MAP, avoiding the costly HA binding updates.

Latest research on this protocol was performed to integrate hMIP with the MIP Route optimization mechanism. Here, instead of the MN sending RO binding updates to the CNs at all handovers, it can limit these for the macro-mobility handover events only, by associating its regional CareOf address (e.g., the MAP address) to the CNs, similarly as it does to the HA. Then, at each micro-mobility local handover, the CN→MAP direct tunnel will remain valid, enabling fast handovers. However, interestingly, even with route optimization, this scheme results in a local triangulation phenomenon that is similar, at a smaller scale, to the one present in standard MIP without RO.

In conclusion, both hMIP versions are able to improve efficiency by providing local handovers, which results in smaller registration latencies. However, further efficiency gains could be achieved, as hMIP only uses a single, possibly far from the MN, anchor point for the whole domain, and does not propose mechanisms to improve the detection efficiency and to prevent the loss of in-flight packets. Regarding transparency, hMIP has similar benefits as in

[4] Quoting from the HMIPv6 specification [34], page 18, "(…) There are no benefits foreseen in selecting more than one MAP and forcing packets to be sent (...) through a hierarchy of MAPs. (...) (thus) it is prohibited by this specification."

MIP, by supporting existing topologies and legacy routers, resulting in improved applicability. However, just as MIP, the protocol requires mobility-aware terminals, precluding immediate utilisation of existing legacy terminals.

2.2.2 Low Latency MIPv4 extension

The extension proposed in reference [23] – low latency handovers – adds MIPv4 extensions to improve the handover latency, by extending the existing MIPv4 FAs without creating any other architectural entities. The major contribution of this extension is the introduction of L2 triggers, which aid the IP handover process with hints of the MN's physical movement. These triggers can then be used to minimize the total handover latency, by reducing both the detection and registration latency. The former trigger enables the anticipation of the L3 operations, while the latter trigger enables the removal of the HA from the critical registration process, to redirect established flows to the new MN location.

In this model, the MIP layer, of either the MN or the involved FAs, is notified of the handover events either sufficiently before or immediately after the link layer transition. In both cases, the normalized triggers should provide the MIP layer the IP addresses of the involved entities (old FA / new FA / MN), in a procedure that was not yet standardized.

The first class of triggers, named "pre-registration", enables the MN to anticipate the L3 handover before the L2 handover occurs. In this model the MN communicates with the new FA while it is still connected to the old FA, enabling it to pre-build its registration state at the new location before the actual movement takes place ("make-before-break operation"). For this, after receiving the L2 trigger, the MN exchanges with the old FA a pair of proxy router advert messages to know the IP address of the new FA. Then it sends the regular MIP registration message to the HA, with the CareOf of the new FA; however, this message is handled specially, by passing through both old and new FAs before being sent to the HA. When the new FA receives such message, it prepares the necessary operations for supporting the MN, namely address allocation.

The second class of triggers, named "post-registration", enables the MN to remove the HA from the critical path during an handover, by continuing to use the old FA as an anchor while already connected to the new FA. This model enables the MN to redirect the packets received at the old FA to the new location, establishing a temporary tunnel between the involved FAs, even before performing the formal MIP registration on the HA to the new location. For this, after receiving the L2 trigger, the MN exchanges with the new FA a pair of proxy router advert messages concerning the old FA; then the FAs exchange a pair of handover request messages to establish the bidirectional tunnel. This tunnel is used until the MN sends the regular MIP registration message to the HA, with the CareOf address to the new FA, or another temporary local tunnel is built using the same operations to a third FA.

2.2.3 Fast handovers for MIPv6/v4

A similar extension to low-latency handovers, with the same objectives, was developed at the same time by other authors in the context of MIPv6 [34], being recently back-ported to MIPv4 [24]. Although this extension does not define any new architectural entities other than those present in MIPv6, it requires substantial mobility-aware support from the existing access routers (ARs), which, as previously described, were highly simplified by MIPv6. Similar to the low latency extension, this also uses a local tunnel between the involved ARs, but includes a buffering facility at the new AR to smoothly decouple the regular HA registration

process from the reception of data packets, by redirecting the in-transit packets to the MN location. This proposal consists of three phases: handover initiation, tunnel establishment and packet forwarding.

At the first phase, the MN sends a router solicitation proxy for the old AR, informing it about the identification of the new AR. This process could be initiated by L2 triggers, although no details on this are provided by fMIP (in contrast with the previous proposal). After this, the MN sends a fast binding update to the old AR, leading to both ARs to communicate by a pair of handover initiate / handover acknowledgement messages. This prepares the packet forwarding at the old AR, and the temporary buffering at the new AR. After the L2 movement, the MN arrives at the new AR, which is already expecting it, and has it's packets already buffered ready for delivery. The process ends when the MN sends a fast neighbour advertisement to the new AR, which uses this to deliver the queued packets.

It should be noted that, in opposition to hMIP, fMIP does not actually hide the mobility events from outside the domain, because all handovers must be propagated to the HA and CNs using regular MIPv6 registration processes. However, the main fMIP contribution is to enable such regular registrations to be performed in parallel with the fast local handovers between the involved ARs, which are able to quickly redirect the in-flight data packets to the new MN location.

A recent proposal details the usage of this MIP extension in the particular case of 802.11 networks, where the 802.11 L2 messages can be used trigger the L3 handovers [36]. Similarly, a recent proposal adapts fMIP to WiMax networks [37]. However, this utilization currently depends on the deployment of special RADIUS servers, thus not being completely self-contained. Recently, this MIP extension was also back ported to MIPv4, where it introduces its functionalities in FAs instead of ARs [30]. Currently, the most criticized aspect of fMIP resides in the difficulty of accurately predicting the handover movement in a heterogeneous way [52], as a too early or too late prediction increases both handover latency and losses [40], and the actual deployment feasibility of the proposed combination of triggers [56].

2.2.4 CIPv4/v6

The Cellular IP (CIP) protocol [20] was one of the first approaches to provide a mobility support more efficient than the one provided by MIP, complementing it in an independent way. This protocol is based on fairly different principles than MIP, being limited to single IP domains, and having the network elements organized on a strict tree structure topology (e.g., no support for pre-existing topologies and legacy routers is available). Inside the tree, the nodes contain direct soft-state routing entries that reference only the next hop that is closer to the terminal; thus, this routing chain is able to identify the MN's location inside the network. At the top of the tree, a special Gateway (GW) node contains routing entries for all network's MNs, being also the unique point of attachment to the outside (thus excluding multiple additional border routers). At the tree leaves, the APs provide connectivity to the MNs, emitting special CIP beacons to the wireless medium.

The MNs have their stack augmented with a CIP client, which detect their own movements using the mentioned CIP beacons. At each movement, CIP uses a simple handover scheme, named "hard handoff", to provide a simple way of achieving a low latency handovers that does not prevent packet losses due to handovers. For this, the MN generates a CIP update message that will be propagated inside the tree, from the AP up to the GW, updating each node with the new next-hop information for the MN. Starting from the AP, the tree's

nodes will create a new next-hop routing entry for the MN, until the message reaches the first common node from the old and the new routing paths, called “crossover node”. At this node, the pre-existing routing entry is changed to the new location, and the message is propagated up the tree to the GW, to refresh the remaining routing entries that will remain valid. In opposition to MIP and its extensions, the CIP signalling doesn’t require acknowledge packets, as the entries have a soft-state nature.

The inter-domain traffic, originated outside the network, is firstly received by the GW, and then propagated through the tree using the previously defined routing paths, directly node-by-node without encapsulation. CIP routing requires that intra-domain traffic also pass through the GW, being then properly routed to the MN, as in the previous case. This requirement is related to the CIP’s simplified soft-state routing entries management, where the previous outdated routing entries are removed automatically by the soft-state timer, after ceasing to be used some time after a handover.

With the necessary L2 cooperation and control, CIP also supports an alternative soft handover that aims to reduce packet loss during the handover. In this scheme, the MN first starts a semi-soft update at the new AP, and then reverts to the old AP frequency to receive data packets forwarded to it. When the semi-soft update message reaches the crossover node, packets are sent to both APs at the same time (bicasting), resulting in lower packet drops due to the L3 handover[5]. To reduce the state maintenance overhead, the routing path refreshment can be made using the data packets emitted by active terminals, which serve as a proof-of-life of the MN. Finally, an innovation introduced by CIP is the support of idle terminals (MNs that are not receiving or emitting IP packets) with a special paging support. For this, the protocol is able to downgrade the mobility service to idle mobile nodes, by tracking only their approximate location. When the MNs become active, the location is automatically searched on demand through an operation known as paging. As it has been shown, such operation is able to improve scalability and power consumption [20].

Concerning the integration of CIP with MIP, while the terminals are located inside the domain, the protocol enables local mobility by requiring only a tunnel from the HA to the GW, which has the role of a single FA for this domain. Each intra-domain movement only requires the update messages to reach the GW, in the worst case.

The initial CIP protocol was also upgraded to an IPv6 version [32], being essentially the same protocol apart from some differences directly related to IPv6 itself, namely the support of IPv6 terminal auto-configuration and the usage of the IPv6 hop-by-hop header extensions to carry the signalling messages. One novel CIPv6 contribution is the introduction of an indirect soft handover that has similar low-loss characteristics to the regular soft handover, but requires less control of the lower layers by IP.

Considering both CIP versions, this protocol is able to improve efficiency comparing to a pure MIP scenario, at the cost of restricting transparency and robustness. The former happens because special support at both the MN and the network is required, precluding the use of legacy terminals and legacy routers. By requiring such a strict tree topology, the protocol’s robustness is fairly limited, as it has already been identified in the past [63] [64].

---

[5] However, it should be noted that this mechanism can incur in significant increase of the L2 handovers latency, as the terminal may need to switch several times between the AP’s frequencies.

2.2.5 HAWAII

The Handoff-Aware Wireless Access Internet Infrastructure (HAWAII) protocol is an alternative proposal that transparently extends MIP with micro-mobility support [21]. An important difference from CIP is that the terminals are only required to implement a modified MIP client, as the protocol provides micro-mobility transparently. For this, a similar division of domains and hierarchy of nodes is established, forming a logical tree with a Root Gateway taking as the sole point of attachment to the outside, which again precludes support for multiple mobile-aware border routers. The tree leaves contain the APs, which implement a MIP-compatible interface to the wireless interface on the form of a FA. Although HAWAII forces a node tree structure for the arrangement of the routers, it may use extra links between the nodes to create the mobile paths.

Whenever the terminal moves within its domain, the client detects and generates the normal update signalling, as each AP is seen by the MN as a regular FA. Upon receiving such MIP signalling, the AP converts it to a corresponding HAWAII update message, and can choose among four different handover schemes that feature different requirements on the type of routing tables supported by the routers. In the most general, called Multiple Stream Forwarding (MSF) scheme, the AP forwards the update message to the previous MN location, identified by the MN in a MIP extension header called “previous FA” (PFANE). At this node, new forwarding entries are added to reflect the new MN location, and the message is propagated to the new AP, modifying each router’s MN entries in the process. Depending on the domain topology and the specific MN movements, the traffic paths may either be augmented to the new location, or redirected at a tree crossover node [45]. For data transmission, each node always uses the existing paths to forward packets, or up the sub-tree by default, without encapsulation. This means that the intra domain-traffic must pass through the domain’s GW only in the worst case [48].

When the terminal switches between domains, a slightly modified MIP procedure is used to assign a new unique CareOf Address to the MN and to redirect the MIP tunnel to this location. After this process, subsequent movements inside the domain are dealt with the HAWAII protocol only. After the first version, further work has been done to introduce a paging facility to the protocol, similar to the corresponding CIP facility [28], Up to now, no further work has been identified concerning future IPv6 networks support of this protocol.

Comparing to MIP, although the protocol is able to improve efficiency, it again does not improve transparency, by requiring MIP clients and a tree of HAWAII nodes to be used. Again, such design precludes the usage of both legacy terminals and legacy routers. By requiring such a strict tree topology, the protocol’s robustness is fairly limited, as it was already identified in the past [63] [64] [60]. Finally, even tough its handover design typically achieves low latency handovers by being directed to the previous AR, the routing paths that are produced are highly subject to the entry point and subsequent MN movements, resulting in variable routing efficiency for the MN locations inside the domain [45] [60].

2.2.6 TIMIP/sMIP

The TIMIP [27] protocol was proposed to give efficient mobility support to all existing terminals in micro-mobility contexts, by introducing a terminal independent mobility architecture feature. This model, originally proposed in reference [26], identified the key mobility operations that must be performed by the network in order to achieve this independence. Thus, the network is the sole responsible for the mobility actions that are typically executed

by the terminals while roaming, implementing a “surrogate behaviour”, which was defined originally in reference [97]. Using this feature, TIMIP can support any terminal, and thus does not impose the client migration cycle that MIP and other protocols require. Unlike other proposals, TIMIP featured strong LMN support from its base, as it only uses mechanisms that do not imply modifications to the terminals.

In this architecture, mobile portions of the Internet are split into multiple domains. Inside each TIMIP domain, the network elements are organized on a strict tree structure topology, being the IP routers that execute the TIMIP mobile protocol and cooperate among themselves to provide intra-domain mobility support to the terminals. At the top of the tree, a special Gateway is used to centralize the management functions, while presenting to the Internet the domain as a classical IP network. On the other hand, the tree leaves contain the Access Points (APs), being connected via the Access Routers. The remaining components of a TIMIP domain are the LMNs, which transparently roam between the APs.

The terminals only have to be able to connect to the network via a suitable wireless interface, as the APs will track the arrival and movements of the LMNs among them. For this task, the protocol uses a single primitive that signals the attachment of the LMN to the AP. This primitive uses all available L2 information for accurate localization of terminal. If the required L2 information is not available, or is not enough, then a passive detection model is used, via a Generic Detection Algorithm (GDA) [27].

After detection, the MN’s location is dynamically updated inside the domain, using update messages generated transparently by the current AP of the MN. When a MN arrives at a TIMIP domain (power-up), the TIMIP’s signalling messages travel up to the domain’s GW. For subsequent roaming operations they are directed to the old AP, being confined to the local sub tree that connects the old and new APs. As no information of the old AP is available at the network-side, it is the network itself that has to infer about its location, using the previous outdated routing paths to pass the messages in a guaranteed way. This registration process is initiated by the new AP, when it detects a new MN and generates signalling information, ending when reaching the old AP. During this process, the routing tables are updated, hop-by-hop, starting from the new AP. Reliability mechanisms were embedded into TIMIP signalling messages, by using acknowledge packets and clock synchronization procedures.

Regarding the forwarding of data packets, like HAWAII, TIMIP is able always to use the routing entries at all nodes inside the tree, meaning that packets are always forwarded using the shortest path in the network tree, up to the crossover node, without necessarily reaching the domain’s GW. Similarly to CIP, TIMIP also optimizes state refreshment, by using IP data packets to refresh the routing entries on the network. Only for idle terminals, the protocol generates explicit signalling information, subject to a back off mechanism.

Concerning macro-mobility support, a TIMIP-integrated extension to the MIPv4 architecture was presented, which introduced the same terminal independence concepts into macro-mobility scenarios. This extension, named surrogate MIP (sMIP) [27], extends the standard MIP agents to perform surrogate mobility mechanisms for macro-mobility operations. Here, the surrogate home and foreign agents (sHA, sFA) will be able to detect LMN movements and generate standard MIP signalling automatically, on behalf of the LMNs that can now roam transparently between TIMIP domains.

The sMIP detection is based on the TIMIP mechanisms. For this, at the GW, the TIMIP power-up operation triggers the sMIP’s detection. After that, the sFA uses standard MIP registration procedures, but generates and receives standard MIP messages on behalf of the

LMN. MIP security procedures may be assured by the sFA or by the LMN, with help of a special authentication application. Thus, as no changes are required at the HA side, the protocol is fully interoperable with standard MIP.

2.2.7 BCMP

The BRAIN Candidate Mobility Protocol (BCMP) is another micro-mobility proposal for IPv6 [25]. Like fMIP, BCMP requires mobility-aware functionalities at the ARs, and like hMIP introduces a new mobility-aware agent, called Anchor point (ANP), which is placed somewhere inside the domain. As in hMIP, the handovers will be limited between the ARs and the ANP, by changing the local tunnel. However, the protocol specifically considers that multiple ANPs can be used inside a domain, being used macro-mobility MIP handovers to change ANPs, when the current ANP is too is far from the MN.

At first, when a MN enters a domain, it sends a registration message to its AR, which chooses an ANP for it. Like hMIP, two tunnels are made, one global HA-ANP, and other local ANP-AR. Afterwards, before a handover takes place, the BCMP protocol supports a planned handover which is similar to fMIP pre-registration scheme, or falls-back to a regular unplanned handover scheme if the handover could not been predicted in sufficient advance.

The planed handover is similar to fMIP: at first, the MN sends a handover preparation message to the old AR, which forwards it to the new AR. This node will then reply with a handover preparation acknowledgement message, enabling the old AR to create a temporary tunnel to forward the in-transit packets to the new AR. Then, after the L2 handover, the MN sends a handover message to the new AR, which is then forwarded to the current ANP and the old AR, and starts delivering the buffered packets if the handover was planned. When the update message is received at the ANP, this node simply redirects the local tunnel to the new AR. If the handover could not be planned in advance (unplanned handover), then only the second part of the described operations are performed, where the MN sends an update directly to the new AR.

While the MN moves inside the domain, the current ANP can be located far from the terminal, resulting in a local triangulation phenomenon. This is addressed in BCMP by reverting to macro-mobility event, where a regular MIP registration changes the global tunnel at HA to the new ANP. Thus, unlike hMIP and several other micro-mobility protocols, BCMP does not completely hide the local MN’s movements inside a domain. This results in BCMP having a lower scalability than those proposals, by requiring extra signalling messages to reach the core networks in certain MN handovers.

2.2.8 netLMM

The Network-based Localized Mobility Management (netLMM) protocol is a current IETF research effort that aims to provide a radical new IETF approach to the problem of mobility support. Here, it was noted that even after scalable (e.g. MIP) and efficient solutions were researched and standardized (namely hMIP and fMIP), these efforts still haven’t resulted in the effective deployment of either local or global IP standardized mobility services. On the other hand, this lack of deployment contrasts with the very successful adoption of proprietary localized management mechanisms in the WLAN infrastructure market [104]. These proprietary L2 and L3 mechanisms all share a common feature of not requiring any changes on the terminals, by providing proprietary transparent mobility services on the local network [104]. Inspired from this success, netLMM introduces a paradigm shift in the IETF

standardization processes, by having explicitly the goal of supporting mobility-unaware MNs [103][6]. As such, by fairly increasing the importance of transparency in a micro-mobility scenario, the netLMM protocol is aimed to fulfil the same transparency requirements as the TIMIP protocol.

The initial version of the protocol, still in the very early standardisation phases, was developed in the IPv6 scope, being later expected to be back ported to IPv4. The protocol uses an architecture similar to the one proposed in hMIP, but which introduces separate mechanisms for managing the new LMN↔AR interface, and the more usual AR↔MAP interface.

Concerning the LMN↔AR interface [105], which is the main distinguishing factor of this proposal in comparison to the previous research work, the ARs present a transparent interface to the MNs which can support mobility-unaware mobile nodes. However, this support currently only applies for the latest generation of IPv6 terminals, which besides standard IPv6 neighbour detection capabilities, also have to support very recent IPv6 additions of Secure Neighbour Detection (SEND) [149], Cryptographically Generated Addresses (CGA) [150], and Detecting Network Attachment (DNA) [151]. All these requirements are related to detecting and authenticating the terminals by the network at each handover, in order to avoid a plethora of attacks if such authentication were to be skipped [152].

The LMN↔AR interface works as follows: when the LMN is powered up, it uses the stateless address auto-configuration mechanism to generate a private link-layer address derived from its public key, using the CGA mechanism. Then, as all standard IPv6 nodes, the LMN will confirm that this address is unique, by performing a Duplicate Address Detection (DAD) process on its current location [114]. This results on the emission of a broadcast neighbour solicitation (NS) message to that link-address, which is to be signed with the LMN's public key using the SEND neighbour discovery extension. When the AR receives this message, it contacts the central MAP node to bind the LMN's public key to the CGA link-layer address. If by chance the terminal has chosen an address already in use by another LMN, then the MAP will reply back with an error so that the AR can send a message to defend the address, cancelling the original DAD process.

Then, the LMN will proceed with the global address auto-configuration, by soliciting the subnet information to the routers, using a broadcast Router Solicitation (RS) message. Here, all ARs of the same domain will respond with the MAP's network values, to give the impression to the LMN that is always located in the same subnet adjacent to the MAP. Using the common subnet information sent by the ARs, the LMN will then perform similar steps for the global address: the LMN generates a new global address using the CGA mechanism, and performs DAD using its public key using SEND. In return, the ARs ask the MAP to add new address binding to the same LMN, which is possible because both cryptographic addresses were generated from the same public key [149].

After an L2 handover, the DNA procedures will be triggered at either the LMN or the new AR. In the first case, the LMN will confirm the default router reachability by sending a new router solicitation message, signed by its public key using SEND. The new AR will then use this message as before to contact the central MAP, but now it only modifies all the existing

---

6 Quoting from the netLMM specification [103], page 8, "2.8 Support for Unmodified Mobile Nodes (Goal #7): (…) (a LMM) solution should be able to support any mobile node that joins the link and that has an interface that can communicate with the network, without requiring localized mobility management software on the mobile node".

bindings from the previous AR to the new AR, as the LMN is identified by its public key. In the second case, when the DNA procedures are triggered in the AR, this node will use a NS message to contact the LMN, which will eventually force the same operations described above.

Traffic destined to the LMN always received in the MAP, as all LMNs feature global IP addresses from the MAP subnet, similar to hMIP. Then, the data is tunnelled to the current AR to be delivered to the LMN. The data packets delivery from the LMN to the AR in still under study, as the described handover operations does not change the LMN's default router (which is by default the first AR where the LMN was power-up in the network).

The AR↔MAP interface is still in study, being expected to be used a protocol similar to hMIP, being decoupled from the terminal independence support that is handled by the described LMN↔AR interface protocol.

Regarding this thesis objectives, this solution has the merits of specifically addressing the terminal independence transparency problem, providing an IPv6 native approach to it. However, such IPv6 LMN support is still fairly complex, by only supporting the latest generation of IPv6 mobile nodes with the latest components (SEND / CGA / DNA), which may not be deployed on current IPv6 networks and MNs. On the other hand, IPv4 transparency, an essential feature for supporting transparent mobility to the vast majority of existing Internet LMNs, has not been studied yet; currently, it is expected that the netLMMv6 protocol will eventually be back-ported to IPv4. Apart from this these transparency problems, all the other common efficiency optimizations are specifically outside the scope of netLMM.

2.2.9 Other recent IP micro-mobility proposals

This section provides a very brief description of some of the other recent IP micro-mobility proposals, being essentially used to relate these proposals with aspects of the already studied protocols.

- MEHROM: The Micro mobility support with Efficient Handoff and Route Optimization Mechanisms (MEHROOM) [51] is a proposal with an architecture similar to HAWAII, which presents two handover phases: in the first phase, an efficient handoff is performed where the update message is sent directly towards the old AR, and traffic is redirected in a crossover node with an entry for the terminal; In the second phase, the possible triangulations are removed, where the crossover node informs the node(s) up the tree about the non-optimal routing.
- Seamless MIP: The seamless MIP (s-MIP) [52] proposal combines the hMIP and fMIP methods to provide a fast handover, and introduces to it smooth loss-less handover capabilities that specifically avoids out-of-order packets. This is done by separately marking the IP data packets forwarded by the old AR in the temporary tunnel, and the ones that are bicasted by the MAP to both ARs. These two sets of packets are then collected and ordered in separate buffers at the new AR, being delivered in sequence after the L2 handover to the MN.
- IDMP: the Intra-Domain Mobility Management Protocol (IDMP) [31] is a two-level hierarchical approach similar to hMIP which has the interesting possibility of being independent of the specific macro-mobility protocol in use (either MIP or SIP). Like hMIP, IDMP introduces an anchor agent in the middle of the visited domain, and manages a regional and global CareOf addresses. A fast handover mechanism is built

using single L2 triggers that only hint the movement, without managing addresses, and the utilization of bicasting techniques using temporary multicast to expected ARs. Lastly, IDMP also supports a paging facility also based on multicast transmission, and has support for MIP-RO usage where the CNs forward data directly to the local anchor, being then tunnelled to the current location.

- iMesh: iMesh [98] is a layer 3 infrastructure-mode 802.11-based mesh network that features IP mobility capabilities. A key interest in this proposal is the support of terminal transparency features, needed to support the mobility of unmodified 802.11 clients. Here, the 802.11 APs also function as routers, which run between them a link state routing protocol (OLSR). When the 802.11 terminal arrives at an L3 AP, the L2 identifier contained in the 802.11 association message is converted to the configured IP address, and an immediate OLSR update is triggered to create direct, per host paths to the current location of the terminal.
- fMIP simultaneous bindings: The fMIP simultaneous bindings [40] extends the fMIP protocol with the possibility of sending the data packets to both the old AR and all the prospective ARs which the MN can make an L2 handover to. This facility aims to remove, within bounds, the timing ambiguity associated to the accuracy of the L2 handover anticipation, as a too early or too late prediction increases both handover latency and losses. This is achieved in cost of increased overhead, and the possibility of introducing duplicate packets, which disturbs the established flows.
- Proxy MIP: The proxy MIP [29] is a transparent macro-mobility proposal that is similar to the surrogate MIP and transparent MIP proposals, as it introduces terminal independence to the MIP macro-mobility architecture. Here, the MN is detected by the network by a mobility proxy agent in an implementation-defined way, and subsequently sends standard MIP signalling to the HA, featuring a sequence number to avoid race-conditions. This signalling is authenticated with the MN's security key, which must be disclosed to the proxy. Using the message, the HA creates the MIP tunnel to destined to the proxy agent. Then, a variety of ARP emulation techniques are used to give the MN the impression of being located in its home subnet always using MAC pseudo addresses.

2.3 Other Layer Mobility Protocols

This section describes mobility solutions that are located in other layers other than IP. Although also providing a heterogeneous mobility support to all access technologies, they don't generally provide the same level of efficiency and transparency, by generically using simpler methods and by requiring more modifications to the existing applications, transports and CNs. In addition, the section is complemented with the description of 802.21, which can aid the heterogeneous protocols those with standardised L2 triggers and control.

2.3.1 TCP-Migrate

TCP-migrate [68] is an example transport layer mobility protocol, which decouples the IP address identifier and locator nature at the end-hosts by extending the TCP flow connection mechanisms. As such, TCP-migrate requires the modifications on the TCP layers of both the mobile nodes and all possible correspondent nodes, even if these are fixed server nodes. However, in contrast to all other mobility solutions, no fixed infrastructure is required at

home domain, as this protocol depends on modification of the MN's home network's DNS entries.

At first, when a TCP flow is established, a special token that identifies the flow is negotiated between both ends during the initial connection establishment. Thus, a TCP connection becomes identified by a triple: <source IP address, destination IP address, token>. Later, when the MN moves to another IP address, a new Migrate TCP option is to be included in SYN segments to identify the previous established connection, rather than a request for a new connection. The previously broken TCP connection can be resumed after the token information exchange between the client and the server[7]. Meanwhile, new connections are redirected to the MN transparently by modification of its DNS entry, on the DNS server at its home network. This requires that such entries are non-cacheable, which increases both the network load and the call setup time of new connections.

This solution has the benefit of having the lowest possible network infra-structure requirements, of presenting direct optimal routing without encapsulation techniques being required, and not modifying the applications. However, it has the serious drawbacks of requiring all MNs and CNs to support this special TCP layer, of being TCP-centric and not providing mobility to other transports, and having scalability concerns regarding the DNS updating, among other problems [47].

2.3.2 HIP

The Host Identity Protocol [70] is another mobility protocol that introduces a new paradigm for solving the locator / identificator problem. Here, instead of using network or transport addresses as the unique identifier, a new namespace is introduced on a new layer between L3 and L4, and hosts are identified between them via their public key. This design option pretends to ease the integration with security protocols, and to provide the cleanest decoupling between the identifier from all the existing addresses that are already used for other ends. However, similar to TCP-migrate, this is incompatible with the existing legacy infrastructure, requiring the addition of the protocol to all mobile and fixed nodes (e.g., servers).

At first, the MN registers its current location in DNS, via a secure DNS procedure similar to the one proposed in TCP migrate. When a connection is established, a four-packet handshake is required to set up keying material for data exchange, being this a simple version of the Internet Key Exchange (IKE) [153]. After authentication, the CN associates this node's identity to its public key, and data packets are transmitted through the direct path. Although no IP encapsulation is used, specific HIP overhead is required per packet; if IPsec is used, this requirement is optimized by deriving the MN's identity with the security parameter index (SPI) carried in the IPsec-protected packets. During a handover, a secure update packet informs the CN's HIP layer of the new IP address location, thus implementing mobility support[8]. Meanwhile, new connections are redirected to the MN transparently by modification of its DNS entry, on the DNS server at its home network, which must be non-cacheable

[7] By its similarity to MIPv6-RO, TCP-M is also expected to be vulnerable to the robustness problem of simultaneous mobility [50].

[8] By its similarity to MIPv6-RO, HIP is also expected to be vulnerable to the robustness problem of simultaneous mobility [50]

(similar to one proposed in TCP-migrate); to optimize this process, a fixed rendezvous server can be used in the home network, resulting in a solution that closely resembles MIP RO's HA.

This solution has the benefit of presenting the cleanest solution for solving the locator / identifier problem of IP, solving the problem in a contained layer, which is transparent to all applications, transports and the IP protocol itself, and additionally having integration advantages with the existing IP security protocols. However, this protocol has the serious deployment barriers of requiring all MNs and CNs to support this special layer, to assume widespread IPsec and IKE deployment, to have a high overhead both for short transactions initiation (handshake requirement) and data transfer (IPsec requirement), among other problems [90].

2.3.3 SIP

The Session Initiation Protocol (SIP) is an of example application layer mobility protocol [66], which is already used in 3G solutions as a component of the IP Multimedia Subsystem (IMS) [72]. By being located in the upper part of the network stack, this protocol can be used to provide terminal mobility, like the other layer protocols, but also service and user mobility. For this support, the SIP protocol requires modification on each application that establish UDP flows which require the mobility support, being these modifications required in both the client and the server, and the introduction of an application-specific redirect proxy servers at its home location.

The operation of SIP is similar to the MIPv6-RO operations, without encapsulation being required. During an SIP session, the modified applications (both client and server) establish special SIP connections with the aid of the mentioned SIP user agents. After the detection of the MN movement, an SIP "Invite" message is exchanged at the established flows to redirect the subsequent data packets to the new MN location[9]. On the other hand, the SIP redirect server, located in the user's home domain, must also be notified about the new location, using "register" messages", in order to redirect the future SIP calls to the current MN location.

As SIP dynamically changes the MN's IP addresses of the established flows, it supports only UDP flows; in contrast, TCP flows are not supported, as these require the usage of constant IP addresses for identification. Also, even tough SIP operates in a completely different level from the L3 solutions, its basic operations and architecture are similar to the IP mobility protocols, as the main SIP entities (user agents, proxy servers and redirect servers) have direct correspondence to the IP mobility agents and clients used [47][10].

The main strength of SIP is that provides mobility for applications that require such service in a simple way, being also typically based in user-space software rather than kernel-space that all the other layers typically have. Naturally, SIP mobility is especially useful for applications that already use SIP, or to new applications yet to be developed using SIP. However, these benefits can be negated by the requirement of having to modify and custom-

9 By its similarity to MIPv6-RO, SIP is also expected to be vulnerable to the robustness problem of simultaneous mobility [50].

10 Quoting from Banerjee, Wei and Das [47], page 59, "…observe that the same entities are involved in the handoff no matter at which layer the mobility management protocol is working…"

ize all the other remaining applications that require mobility services, on both the MN and the CNs, and by the introduction of application-specific proxies for redirection.

2.3.4 MIH / 802.21

The Media Independent Handovers (MIH) of 802.21 [69] is an emerging IEEE standard that provides methods and procedures to facilitate handover between heterogeneous access networks in an efficient manner. This is done by the usage of well-defined services that can optimize the heterogeneous mobility protocols' functionality. As the MIH framework is located between the L2 and the L3 layers, it can provide standardized communication, information passing and command execution between the wireless layers and the heterogeneous mobility protocols.

In particular, 802.21 is able to assist the L3 or above heterogeneous protocols in the key aspects of Handover Initiation, Network Selection and Interface Activation in vertical handovers, provided that the specific L2 wireless technologies are enhanced to support 802.21 interaction. As MIH is able to make use of information gathered from both the mobile terminal and the network infrastructure, it can better achieve the user's Handover, QoS and Mobility requirements with a better network selection; similarly, 802.21 also enables co-operative handover decision-making, by being able to split these responsibilities between the client and the network.

The handover initiation and handover preparation scope is supported by the Media Independent Handover Function (MIHF) that provides abstracted services to higher layers with a unified interface. MIHF defines three different services: the "Event Services" deal with the abstraction of L2 smart triggers, which asynchronously inform the above layers of the dynamic changes of the link characteristics. Examples of those are the "link up/down", "link going down" and "link parameters changed" triggers.

Complementarily, the "Command Services" deal with the synchronous instantiation of the above-L2 handover decisions on the link behaviour. A novel factor is that such messages can be generated either on the client or on the network side, enabling heterogeneous network-controlled handovers. Examples of those are the "poll link", "scan link" and "switch link" commands. Finally, the "Information Services" provide an extensible and dynamic information model, enabling context-aware information exchange of the neighbouring networks and their capabilities. Examples of such information are lists of available networks, MAC addresses, required security protocols and available L1 data rates.

3 State-of-the-Art Evaluation Study

This chapter will study the two major metrics of this PhD thesis – **Efficiency** and **Transparency** – by presenting an in-depth taxonomic study in the form of generic classificative frameworks. Such taxonomies will identify the particular mobility mechanisms that have the greatest impact on the achieved efficiency and transparency levels of the mobility solutions.

After the definition of the proposed models, the frameworks are used to classify and compare the state-of-the-art proposals, concerning their suitability for this thesis' objectives. Initially, this classification is done separately within the scope of each component, by identifying the specific efficiency and transparency mechanisms that the protocols may feature. Later, these results are weighed, according to the author's perspective, and combined in a single graph, which represents a possible view of the efficiency-to-transparency trade-off characteristics of each protocol. This final graph is then used to cluster the mobility solutions into families, and to help with the evaluation of the possible contributions of previous research work on the solution to be proposed in this PhD thesis.

The first proposed framework will identify the key architectural aspects that have the most influence on the resulting efficiency level of the mobile protocols as a whole, considering the major aspects of seamless handovers and mobility overhead. The former is influenced by the support of fast and smooth handovers; the latter is influenced by the control and data packets overhead. The second framework will identify the key architectural aspects that have the most influence on the transparency level of the mobile protocols as a whole. Again, this classification will be separately differentiated for multiple aspects, being evaluated the modifications required at the mobile nodes, at the network and at the correspondent nodes. By making such detailed review of the mobility protocols, the two proposed framework studies are able to provide a different classification view from previous mobility reviews, which compared the protocols using global advantages / disadvantages lists [47], taking into account their generic architectural features [76], their high-level characteristics [94], the routing paradigms used to support mobility [41], or which focus on other important related metrics of mobility networks, like robustness, scalability and QoS support [43]. Outside the scope of these studies are the specific implementation aspects of common available implementations [184], which can also influence the achieved efficiency of the protocols.

Similarly, the final study, which combines the frameworks' results into a single graph, is also able to provide a complementary insight into previous mobility reviews, namely the ones that specifically evaluated the suitability of each network layer to address the mobility support problem [47] [90] [91] [92]. While such previous studies compared the solutions by means of textual descriptions and advantages / disadvantages lists, these relations were not actually quantified in order to help assessing the most suitable layers and protocols to deploy mobility services in existing IP networks. In contrast, this study specifically proposes a possible quantification of the protocols' efficiency and transparency trade-off for the deployment of mobility services in the typical existing legacy networks and future All-IP ones.

The rest of this chapter is organized as follows: it begins with the proposed framework studies for efficiency (section 3.1) and transparency (section 3.2), being then followed by the graph that illustrates both aspects simultaneously (section 3.3). The chapter ends with a final evaluation of the possible contributions of previous research work to the solution to be proposed in this PhD thesis (section 3.4).

3.1 Efficiency Classification Framework

3.1.1 Framework General Overview

Generally, efficiency is an essential and fairly well known metric used to compare and evaluate the mobility proposals' suitability for each given scenario [62] [47] [76]. For this, the efficiency level, as a whole, can be considered as the aggregation of simpler efficiency components, which can be combined into a single concept that enables the protocol's evaluation for each specific utilisation scenario. These components are grouped by the framework into two major orthogonal topics: seamless handovers support and mobility overhead.

Concerning the **Seamless Handovers** framework division, this topic considers the specific mechanisms that manage the routing reconfigurations derived from the MNs' physical movements and their impact on the MN's established flows. The seamless handovers capability is achieved when such reconfigurations are unnoticed by the users [44], and it is the combination of simultaneous support to both fast handovers and smooth handovers [23].

The protocols are able to support the **fast handovers** capability when the handover latency, defined as the time interval in which the MN is unable to receive data packets due to the handover, is minimized. For the MN, such latency is experienced as an interruption of the connectivity service, which results in the degradation and possible cancellation of its established flows. For mobility protocols, minimization of handover latency is achieved by detecting the movement as soon as possible (Movement Detection), while keeping the registration updates in locations as close to the MN as possible (Anchor Point Location).

Likewise, the protocols are able to support **smooth handovers** when the handover data loss, defined as the amount of data packets that are lost due to the handover, is minimized. For the MN, such losses are due to the L2 handovers, if the terminal is physically disconnected from the network and unable to receive data packets (dropped packets), and due to possible reordering or duplication phenomena introduced by the L3 handover itself (out-of-order / duplicated packets). As in the previous case, the dropped packets may be experienced by the terminal as interrupted or unreliable service, resulting in the degradation and possible cancellation of its established communications and flows. In addition, the packets received out-of-order can be dropped, depending on the receiver and flow characteristics, which causes degradation of the established flows [165] [166]. For mobility protocols, the minimization of packet losses is achieved by avoiding dropping packets during the MN's physical disconnection periods (Packet Loss Protection), and by avoiding changing the packet's flow order during the routing reconfigurations (Flow Perturbation Avoidance).

Concerning the **Mobility Overhead** framework division, this topic considers the different resource requirements of the mechanisms that manage the mobility service itself, which is primarily composed of signalling processing and data packets routing to / from the MN's current locations [53] [54]. This way, the resulting mobility overhead is divided into control and data packets-related mechanisms.

Regarding **control packets**, their classification depends on: the minimum distance required to propagate signalling information at each handover (Signalling Propagation); the way that the signalling packets are encoded by the protocol (Control Encapsulation) [47]; and the state maintenance operations required, for both active and idle terminals (Active / Idle State Maintenance). In all cases, the efficiency of the mobility mechanism is improved by achieving the same mobility results with the simplest, most geographically limited and rarest signalling operations.

Regarding **data packets**, their classification depends on: the way they are encoded, if encapsulation is required by the mobility protocol (Data Encapsulation), and the paths through which the data packets are sent in the wired part of the network either inside the MN's domain or outside it (Inter-domain / Intra-domain Forwarding). In all cases, the efficiency of the mobility mechanism is improved by using the simplest encapsulation of the data packets, and by using the most direct paths in the wired part of the network either inside and between domains.

3.1.2 Mobility Phases definition

To aid the identification and division of the mobility operations that have the most influence on the different efficiency topics previously mentioned, a simple model is introduced that groups these operations into three main protocol phases.

In most situations, the MN is connected to the network through a certain AP and has all the mobility mechanisms stable for this current location, which enables it to send and receive data packets normally. When the MN moves between different APs, it is assumed that physical and L2 connectivity are broken, being restored at the new AP using L2 handover operations. If the new AP lies behind a different IP sub-network or IP node, then mobility operations are required both to detect such event and to restore the necessary routing information consistency in the new location. After these operations, the next handover will only be required for the subsequent physical movements that modify the MN's IP point of attachment again.

Thus, the operations are grouped into three main mobility phases:

- Detection phase: This phase considers the operations needed for the mobility protocol to become securely aware that the network has outdated routing information about the MN's current location.
- Registration phase: This phase considers the signalling operations that update all the critical nodes with the new MN location information, as required to re-establish the MN connectivity as fast as possible.
- Execution phase: This phase occurs between the handovers, when the MN doesn't change its IP location and the existing routing information is used to route the MN data traffic.

3.1.3 Framework Models Characterization

This section will propose classificatory models for each framework component that was described previously.

3.1.3.1 Seamless Handovers

Regarding the Movement Detection component of the seamless handovers metric, all the layers above L2 are not aware of the MN's physical movements by being independent of all access technologies. In contrast, in most radio technologies, the L2 is fully aware of these movements, and thus is able to help the mobility protocols in their detection process. The level of integration between L2 and the mobility protocol results in different movement detection models. If a Passive model is used, no integration is performed and the mobility protocol has to detect the MN's movements using its own mechanisms, which are required to be independent of all access technologies. In the Reactive model, there is a minimal cooperation

between the two layers, where the L2 operations are simply exposed to the higher layer after its L2 handover is complete. In a Predictive model, L2 becomes responsible for predicting handovers and informing the higher layer before they actually take place. Finally, the Active model requires a total cooperation and integration of L2, as the higher layer mobility protocol takes full control of the L2 handover operations.

Regarding the Anchor Point Location component, this is differentiated by the geographical location of the minimum set of nodes that need to be informed of the MN's movement at each handover in order to re-establish connectivity at the new location by redirecting the MN's data flows. The Inter-Domain model always requires the updating of nodes outside the MN's current domain. In contrast, the Intra-Domain model only requires the notification of nodes belonging to the current MN domain, but which are located outside the shortest path between the old and new MN's locations. Finally, in the Cluster model, only the nodes belonging to the shortest possible path, between the old and new APs, must be notified.

Regarding the Packet Loss Protection component, this is differentiated by the nature of the mechanisms, if any, used to protect from packets drops due to the L2 and mobility protocol handover. The None model doesn't offer any specific loss protection – in-transit packets transmitted via the old path will be dropped if the old AP does not have L2 coverage to the MN after the L2 handover. The N-Casting model considers that handover drops are avoided by simultaneously sending data packets through the old and the new paths; the Buffering model considers that the data packets that are to be sent through the old path are buffered until the handover is complete and the new MN location is known [45].

Regarding the Flow Perturbation Avoidance component, this is differentiated according to the nature of the mechanisms, if any, that are used to protect the established flows from packet reordering phenomena due to the handover. The None model does not offer any specific protection – packets are reordered or duplicated, depending on the specific handover situations; the Packet Marking model considers that the data packets are marked to enable their subsequent sorting to their flow's natural order at the new location [164]; the Smooth Flow Redirection model achieves the same goal without packet marking, by delaying the packets sent through the new path for the minimum period of time so that they will always arrive later than the packets received through the old path.

3.1.3.2 Mobility Overhead

Concerning the mobility overhead efficiency metric, the Signalling Propagation component is differentiated by the geographical location of the nodes that must be informed of the MN movement at each handover. The Multiple Core Networks model always requires the updating of multiple nodes outside the MN's current domain; the Inter-Domain model requires a single updating to a node also outside the current domain. In contrast, the Intra-Domain model only requires the notification of nodes always belonging to the current MN domain.

Regarding the Encapsulation of control packets component, it measures the overhead associated with the signalling packets of the mobility protocol. The Large model refers to a large overhead per packet, typical of non-binary encoding of the mobility fields; the Small model refers to a small overhead per packet, typical of binary encoding of the mobility fields. Finally, the In-band is applicable when the mobility fields can be piggy backed in data packets, which has the lowest overhead of all models.

Regarding the Active State Maintenance component, it is differentiated by the way the mobility's state is maintained for active terminals. The Explicit model relies on the existence

of signalling messages for the MN's state refreshment at the necessary nodes. Its duality is the Implicit model, where the data traffic transmitted by the MNs is used to accomplish the same task. A related notion is measured by the Idle State Maintenance framework component; the Uniform model considers that the idle state is refreshed using signalling at a constant rate, and the Exponential model considers that the refresh cycle length can be dynamic, reducing the state maintenance overhead.

Regarding the data packets, the Forwarding component measures how the data packets are routed inside the fixed part of the network from the CNs up to the MNs, either outside or inside the current domain. For Inter-domain data forwarding, the Optimal model is applicable when the shortest paths are always used for this operation; the alternative Triangulated model is applicable when the data packets are always required to pass through certain fixed nodes outside the current domain. For Intra-domain data forwarding, the Optimal model always uses the shortest paths for this operation. The Tree-Optimal is a more limited model where the packets are sent through the most direct paths inside a common tree in the wired part of the network. Finally, the Triangulated model is applicable when the data packets are required to pass through certain fixed nodes of the current domain, regardless of the sender and receiver locations.

Regarding the Encapsulation of data packets component, it measures the overhead associated with the forwarding of these packets by the mobility protocol, compared to the corresponding non-mobile forwarding in equal circumstances. The Full Tunnel model refers to a large overhead per MN data packet, where the MN's data packets are encapsulated in completely new data packets for their transport; the Soft Tunnel model refers to a small overhead per MN data packet, when those are encapsulated using the original data packets. Finally, the None model is applicable when there is no difference between the data packets encapsulation compared to those managed by the fixed routing in equal circumstances.

3.1.4 Classification of existing Proposals

Figure 1 and Figure 2 represent the efficiency framework with the proposed divisions, components, models and mobility phases previously described, being separated for the two main global metrics of seamless handovers (Figure 1) and mobility overhead (Figure 2). The figures also contain the classifications of the selected state-of-the-art mobility solutions in each of the proposed models. Standard MIP has been studied for both IP versions, with specific differentiation if the MIPv6's route optimisation method leads to a different classification. The other MIP extensions and particular options of other proposals are only discriminated when those results in a framework classification that is different from the base case.

For each framework component, the majority of the protocols are grouped in the "**other alternatives**" group, with the protocols which have different classification models being explicitly identified in their appropriate models. As the base MIPv4/v6 is the most well-known and complete heterogeneous mobility protocol, its efficiency classifications will be firstly detailed thoroughly, in order to constitute a basis for comparison for all other proposals; then, the remaining framework's classifications will be detailed for each component in turn, by relating each of the protocol's features described in chapter 2 with the framework models' definitions described in section 3.1.3.

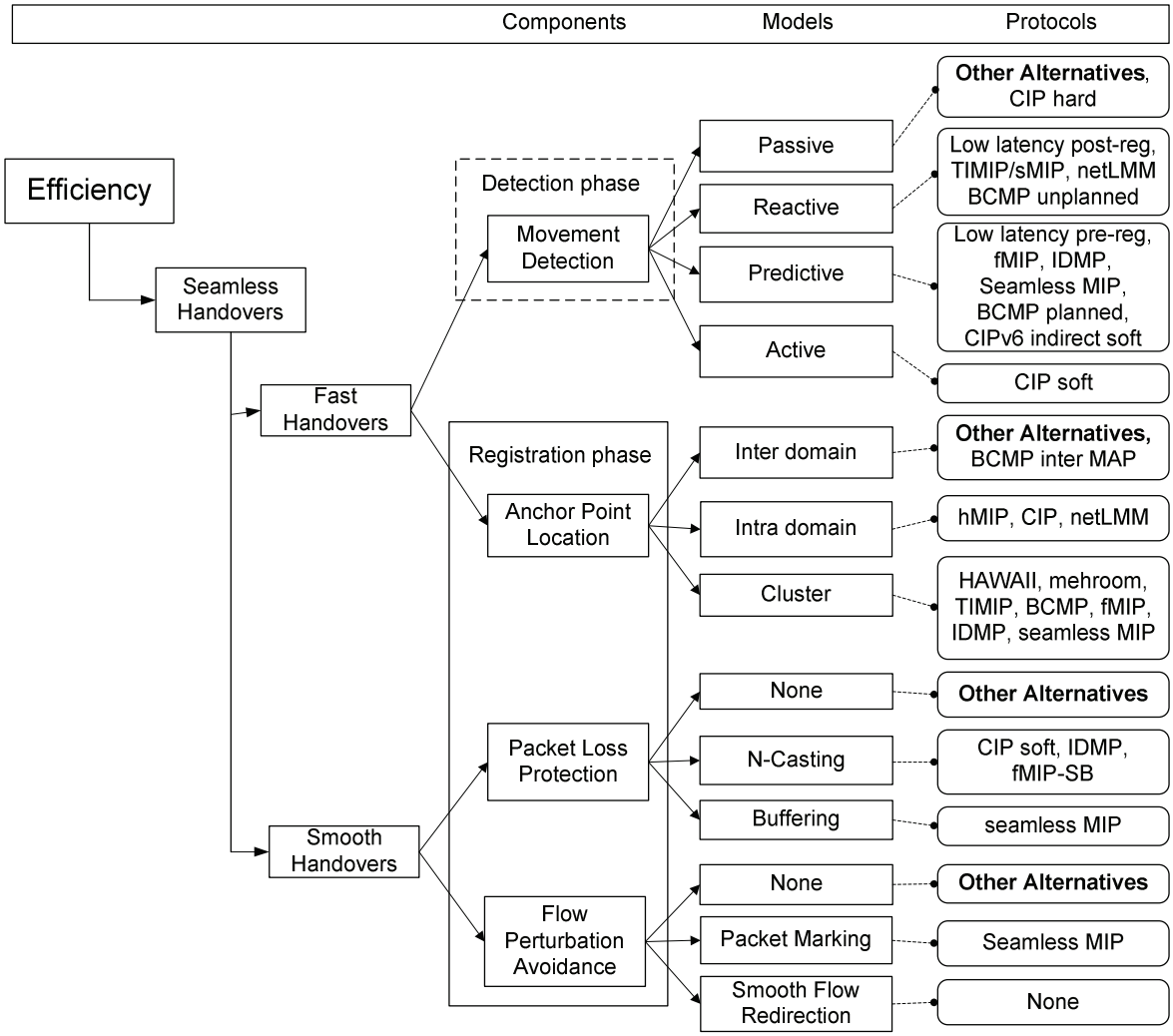


Figure 1: Efficiency Classification Framework – Seamless Handovers Components

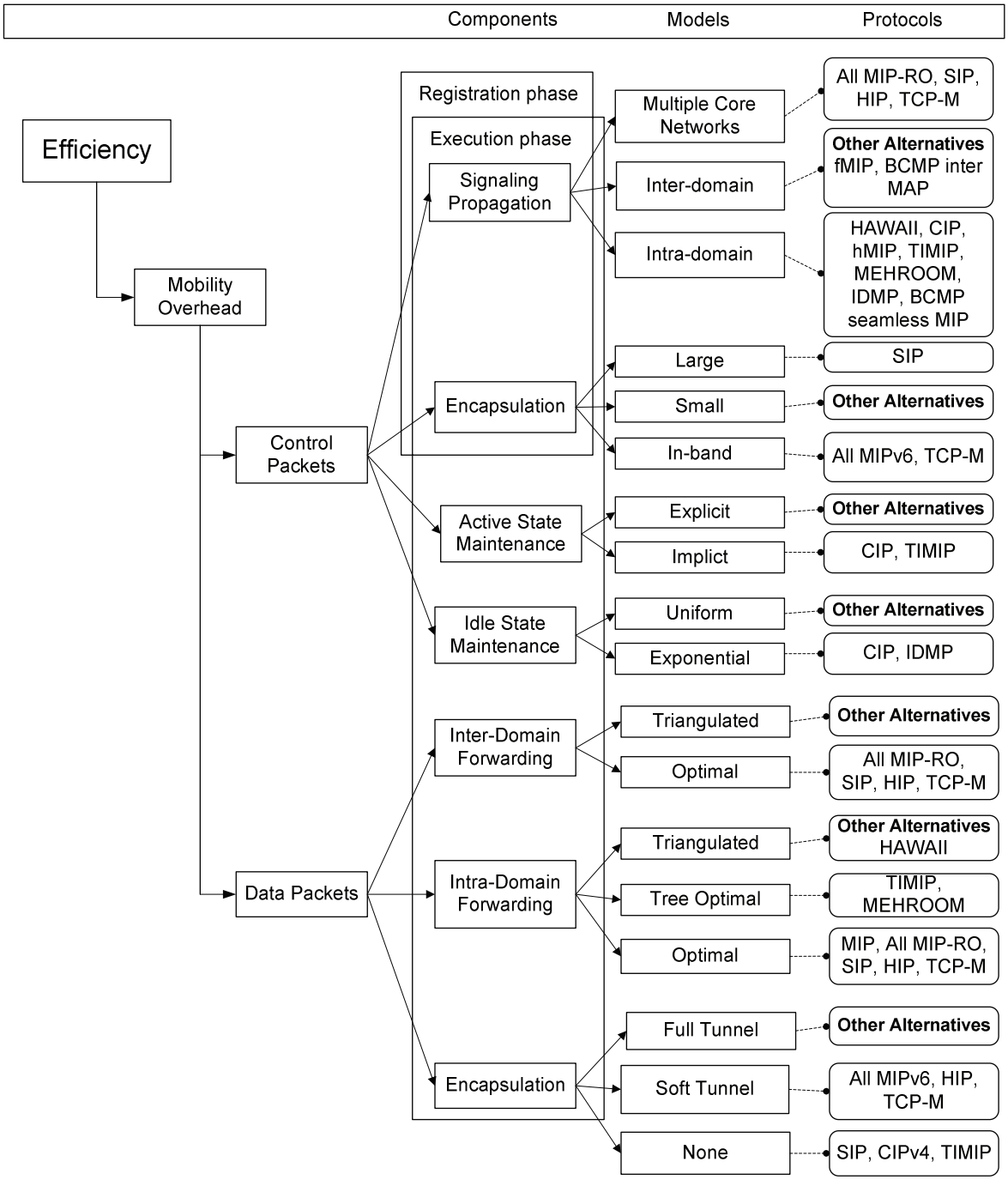


Figure 2: Efficiency Classification Framework – Mobility Overhead components

3.1.4.1 Standard MIP efficiency classification

Regarding the classic solution, the MIPv4 protocol is only suitable for large movements, typical of nomadic computing. In its basic version, every time the MN moves, the terminal detects this event using MIP beacons, and starts a registration process that always involves the HA to re-establish the full IPIP data tunnel to the new location. As such, MIP features a Passive movement detection model and an Inter-Domain Anchor Point Location. Besides this lack of fast handover support, MIP doesn't feature any other smooth handover specific mechanisms. Thus, these architecture design options explain the long latencies and high losses that are expected at each handover.

Regarding the signalling mobility overhead, standard MIPv4 has no optimisations as all movement updates, which typically need to pass through core networks, are propagated to the HA (Inter-Domain Signalling Propagation), using control packets encapsulated in UDP and binary encoded (Small Encapsulation). Standard MIP also doesn't distinguish between active and idle terminals, forcing the periodical renewal of the MN's state through periodic update messages (Uniform, Explicit State Maintenance operations). Regarding the data packets overhead, MIPv4 requires that all data packets pass through the HA in order to be encapsulated in a full IPIP destined to the MN's current FA. This results in a Triangulated Inter-Domain data forwarding classification, which however becomes Optimal inside the visited domain. Finally, MIPv4 features a Full Tunnel Data Encapsulation due to the IPIP tunnel.

Regarding MIPv6, the protocol is fairly similar to the previous version, sharing most of the same high-level operations and algorithms. Thus, MIPv6 shares the same classification as MIPv4 for most of the framework models, in particular when traffic is routed through the HA. However, by enabling the MIPv6 CNs to receive the MN binding updates (the route optimisation option which is native to MIPv6), MIPv6-RO is classified with Optimal forwarding for both inter and intra-domain, which benefits the data efficiency of the mobility protocol. This feature is achieved at the cost of a higher signalling overhead, as this optimisation requires the diffusion of binding updates to all its CNs and the HA, both at each movement and periodically for the state maintenance. Thus, MIPv6-RO is classified with the Multiple Core Networks Signalling Propagation model.

Finally, in the spirit of IPv6, MIPv6 and all the protocols based on IPv6 use simpler encapsulation methods for both control and data packets: the binding updates are now sent in the IPv6 mobility header, and thus can be piggybacked with data packets (In-band Signalling Encapsulation); the data packets are redirected to the MN's current location, by using the simpler IPv6 source routing header (Soft Tunnel Data Encapsulation).

3.1.4.2 Seamless handovers classifications of other proposals

Concerning the movement Detection proposed models, the Reactive classification is applicable to the low-latency MIPv4 extension, in its post-registration option, which defines the specific triggers and parameters that the L3 should receive immediately after the L2 handover. This model is also applicable to TIMIP/sMIP protocols, as it defines a similar L2 detection primitive to signal the LMN's arrival at the new AR, and to the BCMP unplanned handover scheme and netLMM handover operation. The Predictive model is applied to both the low-latency MIPv4 extension, in its pre-registration extension, and the fMIP extension. By initiating the L3 handover while still connected to the old AR, these protocols can parallel the L3 handover either before or during the L2 one, which further reduces the L3 handover latency. However, it should be noted that no specific triggers have actually been defined in fMIP, in contrast to the low-latency proposal, and that service disruption still occurs, for at least the period of time concerning the anticipation and the L2 handover time. By extending fMIP, the recent proposals IDMP and seamless MIP also inherit the same fMIP detection model; CIPv6 also features this model due to its indirect soft handover; such is also the case of the BCMP planned handover. Finally, the Active model is applicable to the soft-handover option of CIP, enabling the terminal to return to the previous AP frequency while initiating the handover at the new AP.

Concerning the Anchor Point Location models, the Inter-Domain classification is applicable to hMIP, netLMM and CIP, as these protocols only have to update wired nodes located in the current domain, in order to redirect the established flows to the new MN location; for

hMIP/netLMM, this is the regional MAP; for CIP, this is the crossover node, which is the last node located in the common path between the old and the new paths starting from the GW on the domain's tree. The most efficient Cluster model is applicable to all protocols that send their update message directly to the previous MN location. This includes HAWAII and its derivative MEHROOM, fMIP and its derivatives IDMP / seamless MIP / BCMP planned, and TIMIP. All these protocols use the signalling directed to the old AR to redirect the data packets received at this node to the new location; TIMIP, HAWAII and MEHROOM perform this by modifying the existing tree routing entries in the new MN location; fMIP and its derivatives perform this by creating a local temporary tunnel between the involved ARs.

Concerning the Packet Loss Protection mechanisms, the CIP, IDMP and MIP simultaneous binding extension proposes a bicasting option where the data packets are duplicated and sent to both the old and the new ARs during a handover (N-casting model). The Buffering model is adopted by the seamless MIP proposal, by explicitly buffering the data packets that would be lost during the L2 handover at the old AR.

As far as Flow Perturbation Avoidance is concerned, only seamless MIP proposes a Packet Marking mechanism to avoid introducing out of order packets. In this scheme, the packets are bicasted to both involved ARs, and the received packets at the old AR are redirected to the new one using a local tunnel; however, to avoid reordering the flows, each data packet is marked to distinguish between those that have been sent directly from the crossover node, and those that have been tunnelled from the old AR. Finally, even though CIP has a delay mechanism to try to avoid perturbing the flows by delaying the packets at the crossover node for a constant period of time, such still causes packet duplication [45] and is inefficient as the packets are always delayed by the crossover node, regardless of it being necessary or not.

3.1.4.3 Mobility Overhead classifications of other proposals

Concerning the Control Packets Overhead components, the SIP, HIP and TCP-migrate protocols present a mechanism that is similar to MIPv6-RO's, where each CN must be notified at each handover to ensure the optimal routing. Thus, these protocols are classified with the Multiple Core Networks model, which adversely increases the mobility overhead. At the opposite end of the scale, most IP micro-mobility protocols use regional IP addresses inside the visited domain to efficiently hide the MN's movements to nodes outside the domain, namely the HA or the CNs. Thus, almost all the IP micro-mobility protocols feature the Intra-domain signalling propagation, which has the best overhead requirements. A notable exception is fMIP: while this protocol is able to achieve fast handovers by using a local tunnel between the involved ARs, this is only a temporary mechanism that reduces handover latency; in all cases, a regular handover to the HA must be performed to make this update permanent (and to the CNs, when using the route optimization option). Thus, as fMIP doesn't shield the handovers and the state maintenance from the nodes outside the visited domain, it features an Inter-domain model. A similar but less pronounced case is BCMP; here, all handovers are quickly handled inside the domain using a local tunnel between the involved ARs. However, when the local tunnel to the MN's anchor point (ANP) becomes large, BCMP may make an inter-ANP handover, which is a macro-mobility event. Thus, BCMP doesn't shield all the handovers that the other protocols successfully hide from nodes outside the domain.

Concerning the Signalling Encapsulation, only SIP suffers from Large overhead per control packet, a result of encoding its fields' real time protocol (RTP) packets in ASCII. On the other hand, MIPv6 and its direct extensions benefit from a lighter signalling encoding, by be-

ing able to piggyback its mobility fields with regular data packets, using the IPv6-defined mobility header extension. Thus, MIPv6 is classified with the In-band signalling encapsulation model, to which the least overhead requirements per control packet correspond. TCP-migrate also shares this light encapsulation option, as its update message is only a modified TCP SYN sent to the new location with the previous session token.

Regarding the Active State Maintenance component, only CIP and TIMIP take advantage of the MN's own data transmissions to refresh the MN state on the network, to which a lower overhead per served active MN corresponds (Implicit model); all other protocols must explicitly maintain the MN's mobility state at the network using explicit signalling packets (Explicit model).

Regarding the Idle State Maintenance component, only CIP and IDMP give special low-overhead support to idle terminals, by means of on-demand paging facilities which greatly reduce the state maintenance requirements of the idle terminals (Exponential model); the other protocols do not offer any special support for this special class of terminals, as these continue to perform the regular state maintenance operations of the active terminals (Uniform model).

Regarding the Inter-Domain (Data) Forwarding overhead component, only the protocols that use the MIP-RO extension or that operate above IP are able to support optimal forwarding between the domains. This happens because these protocols propagate their updates up to the CNs, informing of the current location of the terminal (Optimal model); all other protocols require some form of triangulation between domains (Triangulated model).

Regarding the Intra-Domain (Data) Forwarding overhead component, again all the protocols that use the MIP-RO extension and all the higher layer protocols are able to support Optimal routing inside the domain; however, this is also supported by the base versions of the MIP macro-mobility protocols (Optimal model). Regarding the efficient micro-mobility protocols, all of them will typically resort to an inefficient Triangulated model, by making the data packets pass through certain fixed nodes in the visited domain. As it is well-known, this design option reduces the optimization of the network resources and increases the end-to-end delay. However, in the visited domain, both TIMIP and MEHROOM are able to use the direct routing paths inside the network tree (Tree-Optimal), which is particularly advantageous in case of internal intra-domain traffic. HAWAII would also theoretically be able to benefit from such tree-optimal model, but it suffers from non-optimal paths that depend on the MN's entry point to the network and subsequent movements [45] [62], which results in it being classified with the base Triangulated model.

Concerning the Data Encapsulation, MIPv6 and HIP only require a simplified encapsulation for each data packet by only introducing an extra header per packet (in comparison with the data packets forwarded to fixed nodes) (Soft Tunnel model). However, only SIP and TCP-migrate are able to route the MN's data packets completely unchanged on the end-to-end path, as these are indistinguishable from the regular data packets forwarded to fixed nodes (None model). CIP and TIMIP also share this model for forwarding inside the visited domain, as all packets are forwarded inside the domain without any encapsulation, (as the MIP tunnel ends at the FA, which is to be co-located with the domain's GW).

3.2 Transparency Classification Framework

3.2.1 Framework General Overview

In general, the transparency feature, which compares and evaluates the mobility proposal's deployment suitability in current and future Internet scenarios, is a less studied aspect than efficiency. However, very recently, the observed lack of deployment of IP mobility in current networks in contrast to the successful adoption of infra-structured WLAN market [104] [103] [178], and even the observed slow deployment of fixed IPv6 networks [99] [100], have significantly increased the importance of this aspect, which now reaches the same importance level as efficiency.

As in the previous framework, the transparency level, as a whole, can be considered the aggregation of simpler transparency metrics, summarized into a single concept that enables the evaluation of the protocol's deployment suitability in current utilisation scenarios[11]. Of these, this framework groups such metrics into three major complementary components, which correspond to the major architectural divisions of the mobility protocols: terminal independence, network independence and peer independence.

The first component addresses the type and amount of modifications needed at the mobile node for the introduction of IP mobility, being designed as **Terminal Independence** (TI). In this light, the mobility protocols can either support the mobility of regular fixed hosts, or various levels of host modifications may be required at the network stack or at the deployed applications. As it was already studied, there is consensus on pointing the IP level as the most appropriate layer for the introduction of mobility services in the IP networks [47]. This is especially adequate for future 4G networks [80], as it provides heterogeneous and transparent mobility to upper layers, avoiding any application or transport modifications and neither requires the deployment of application specific proxies at the network or the modifications to the fixed CNs. However, even if the mobility support is limited to the IP layer only, the MN IP stack's modification is highly problematic or costly, as the nodes which would most benefit from mobility, namely the laptops, PDAs and other simple devices, still use regular fixed IPv4 stacks. Thus, the support for LMNs is of major importance in the current deployment of mobility services. Taking these aspects into consideration, the mobility protocols can range from the absence of modifications in the MNs, to the introduction of single or multiple mobility-aware components that manage the mobility service for all existing and future applications.

Another component – **Network Independence** (NI) – considers the type and amount of modifications required for mobility deployment in pre-existing network topologies. Thus, the critical aspect to consider is the support for Legacy Routers (LR), which are the pre-existing IP, non-upgradeable, fixed-only routers, coupled with the amount of additions and/or modifications to the existing networks topologies for the deployment of the mobility service. Taking this aspect into consideration, the mobility protocols can range from the absence of any network modifications to the network, or the addition of mobility agents or application-specific proxies that respect the pre-existing fixed network, up to a complete replacement

[11] This mobility transparency metric is similar in nature to the "deployability" metric proposed in the context of congestion control mechanisms (sender only/ receiver only / both hosts / isolated router / all routers / hosts and routers) [168].

and redesign of pre-existing routers and topologies. Regarding this topic, there is long-time consensus that the modifications to the network should be as limited as possible, and if necessary, pushed to its borders in order to maintain scalability [176] [177]. This aspect has a major importance in the current deployment of mobility services, as the lack of support for pre-existing routers, topologies and equipment is often unacceptable. In contrast, the support for pre-existing networks is essential to enable an incremental and smooth upgrade path of the mobility features, and to use current fixed networks immediately. The Network Address Translation (NAT) is a good example of the network independence paradigm, as it is the only IPv4 evolution that has truly gained support and adherence, in spite of its multiple major problems that have been studied in the literature [99].

The final component addresses the type and amount of modifications that the pre-existing mobility-unaware fixed nodes require for communicating with the MNs, being designed as **Peer Independence** (PI). In this light, the mobility protocols can either support the communication of the MNs with the legacy fixed nodes without modifications, or various levels of modifications may be required at the network stack or at the deployed applications of the pre-existing fixed nodes. Therefore, the mobility protocols can either support such nodes transparently without any special support, or specific support options must be required at the fixed CNs (similar to the ones introduced in the MNs). It should be noted that these nodes form the most critical group where transparency support is extremely essential, as it stands for the vast majority of existing fixed Internet nodes; in particular, its total absence would result in the MNs only being able to use the applications that require mobility [93] among themselves, severely reducing the protocol's applicability and usefulness. These reasons explain why there is total consensus for the support of such unmodified peers, at least as a fall back solution in IP mobility [1] [2].

3.2.2 Framework Models Characterization

Regarding the Terminal Independence framework component, the modification level required at the network or other layers results in the different models proposed for this aspect. The simplest – No Mobility-Aware Components – does not permit any changes to the MN, at either the existing IP network stack or the deployed applications. By supporting both the fixed IP stack and all the existing applications, it is only this model that results in a completely transparent mobility service to this class of nodes. In the Single Mobility-Aware Component, the MN's network stack may be modified through the introduction of a single mobility-aware component that centralizes all mobility-related mechanisms, and provides heterogeneous and transparent mobility services for all existing applications, which cannot be modified. If a second additional mobility-aware client is required besides the standard one, that will be classified with the Double Mobility-Aware Components. Finally, if multiple components, which can either belong to the network stack or to the deployed applications, must be modified in order to become mobility-aware, this is classified with the Multiple Mobility-Aware Components framework model. This model corresponds to the poorest transparency result, as the introduction of the mobility services in the current deployed networks will require the maximum amount of modifications in the already deployed terminals and applications.

Regarding the Network Independence component, the amount and location of modifications required at the network side are the key factor for the different models proposed in this framework. The simplest – No Mobility-Aware Components – does not permit any changes at all to the pre-existing networks, resulting in a completely transparent mobility service for

the current fixed networks. In the Single Mobility-Aware Equipment Addition per Home domain, the MN's mobility service only permit the addition of a single proxy at each domain, which must be sufficient to enable global roaming. Although such a model modifies the network, compatibility with all existing fixed networks, topologies and equipment is possible because the agents are only added to the network. The following model – Single Mobility-Aware Equipment Addition per Subnet – relaxes the previous requirement to also permit the addition of an agent to each location that the MN may visit. The next model – Multiple Mobility-Aware Equipment Addition per Domain - further relaxes this requirement by enabling the addition of further entities to the network-side, but keeping the requirement to support existing LRs and topologies. The final model – Mobility-Unaware Equipment Replacement – breaks the previous model's network transparency by dropping the requirement of supporting existing LRs and topologies, which can now be replaced or modified.

Regarding the Peer Independence component, again, the modification level required at the network or other layers results in the different models proposed for this aspect. Thus, the proposed models for fixed CN's transparency closely match the corresponding ones for MN transparency. The simplest model – No Mobility-Aware Components – does not permit any changes to the fixed CNs, at either the network stack or the deployed applications, resulting in a completely transparent mobility service for this class of nodes. Thus, both the fixed IP stack and all the existing applications are supported immediately. Similarly, in the Single Mobility-Aware Components and Double Single Mobility-Aware Components the fixed CN's network stack may be modified through the introduction of a one or two mobility-aware component that centralizes all mobility-related mechanisms and provides heterogeneous and transparent mobility services for all existing applications, which cannot be modified. Finally, if multiple components, which can either belong to the network stack or to the deployed applications, must be modified in order to become mobility-aware, this is classified with the Multiple Mobility-Aware Components framework model. This model corresponds to the poorest transparency result, as the introduction of the mobility services in the currently deployed networks will require the maximum amount of modifications in the already deployed fixed nodes and applications.

3.2.3 Classification of existing Proposals

Figure 3 represents the transparency framework with the proposed components and models previously described, and the classifications of the selected state-of-the-art mobility solutions in each of the proposed models. Again, standard MIP has been studied for both IP versions, with specific differentiation if the new options lead to a different classification. As in the previous framework, the majority of the protocols are grouped in the "**other alternatives**" group, with the protocols which have different classification models being explicitly identified in their appropriate models.

As the base MIPv4/v6 is the most well-known and complete heterogeneous mobility protocol, its transparency classifications will be firstly detailed thoroughly, as this protocol constitutes a basis of comparison for all other proposals; then, the remaining framework's classifications will be detailed for each component in turn by relating each of the protocol's features described in chapter 2 with the framework models' definitions described in section 3.2.2.

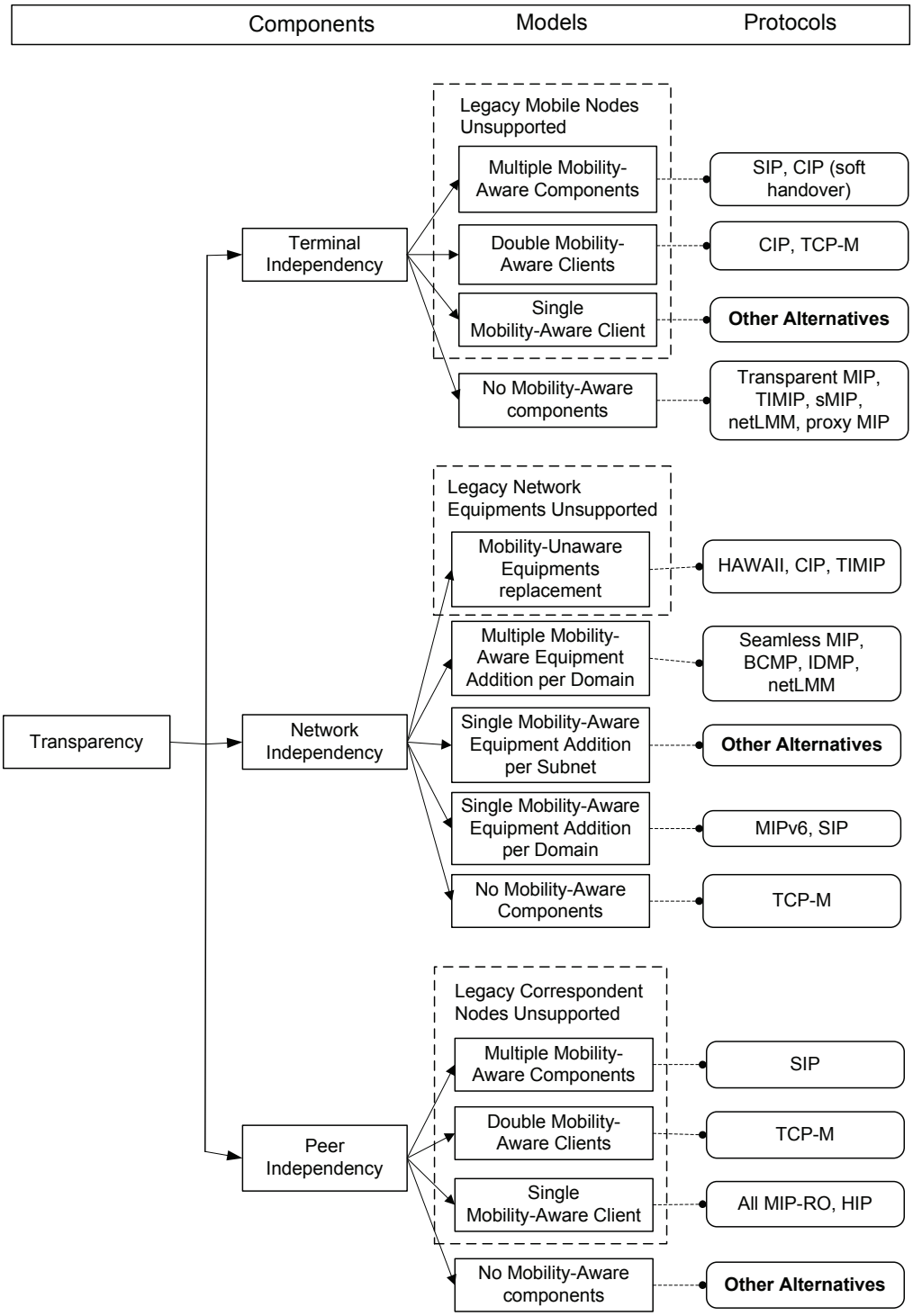


Figure 3: Transparency Classification Framework

3.2.3.1 Standard MIP efficiency classification

Regarding the classic solution, the MIPv4 protocol has been designed with great care to maintain compatibility with the pre-existing fixed infra-structure and routing, especially the existing fixed CNs and all the existing applications. As such, the MIP protocol must only be supported with a standard MIP client in the MN's protocol stack, and through the deployment of new mobility agents at the home and, generally, also in the visited domains. All the other fixed infrastructure and deployed applications are immediately supported without any

modifications. As such, MIP is classified with the Single Mobility-Aware Component regarding the MNs, with the Single Mobility-Aware Equipment Addition per Subnet regarding the network, and the No Mobility-Aware Components regarding the fixed CNs.

Concerning MIPv6, the new version greatly improved the original's applicability by dropping the foreign agents' requirement in visited domains, using new IPv6 capabilities. Using the increased address space and new ICMP router discovery options, the MNs can now use the standard IPv6 fixed routers capabilities to both detect the MN´s movement and to collect a CareOf address for its current location (the two key functions performed by MIPv4's FAs). Thus, MIPv6 is classified with the Single Mobility-Aware Equipment Addition per Domain for the network transparency. On the other hand, the optional route optimization scheme is only possible if the fixed CNs also support the protocol. Thus, MIPv6 with RO is classified with the Single Mobility-Aware Component model for the fixed CNs, while the regular, non-optimised, MIPv6 continues to be classified with the No Mobility-Aware Components, just like MIPv4, as a fallback solution for the fixed CNs that do not support MIPv6.

3.2.3.2 Transparency classification of other protocols

Concerning the Terminal Independence framework's classifications, all protocols tend to require a single mobility aware component in each MN (Single Mobility-Aware Component model), which has the responsibility of detecting their own movements and of informing the network of the current MN location. Besides this centralized mobility-aware component, all the other parts of the network stack, transports and applications remain mobility-unaware, and thus free from further modifications. For most IP layer protocols, this is a standard or a modified MIP client; an example is the HAWAII protocol, which transparently converts the standard MIP messages, extended with the previous FA information, to HAWAII messages inside the visited domain. Although HIP is located in a higher place in the network stack, it has the same framework classification, as it also requires a single MN mobility-aware client for managing all the mobility mechanisms, which doesn't modify all the transports and applications as previously described. In all these cases, the requirement of mobility support in the terminals precludes the immediate deployment of mobility services in the existing terminals.

Due to the fact that TIMIP and sMIP specifically support legacy mobile nodes, both protocols are able to overcome this problem, as the movement detection and signalling generation are performed by the network itself. This is also the case of netLMM (for the latest generation of IPv6 terminals supporting SEND and DNA), and for the macro-mobility proposals Transparent MIP and proxy MIP. Due to this, all these proposals are classified with the No Mobility-Aware Components for terminal independence, which enables immediate utilization of all the existing deployed LMNs.

On the other hand, CIP is classified with the Double Mobility-Aware Components, as it additionally requires a CIP client to manage the efficient micro-mobility, besides the regular MIP for macro-mobility. Additionally, in case soft-handovers are used, this transparency classification is further worsened, as modifications to L2 are also required, breaking IP heterogeneity.

Although in theory TCP-Migrate only requires a single mobility client, it is also classified with the Double Mobility-Aware Component. This happens because this protocol only supports the redirection of TCP flows, precluding the mobility support of all the other transport protocols in study on the IETF [47], e.g. Datagram Congestion Control Protocol (DCCP) [68]. However, to still use applications like VoIP or multimedia streaming in heterogeneous mobility scenarios, mobility solutions for UDP or other types of transport

must still be developed[12]. Thus, as TCP-migrate do not solve the general mobility problem, in practice it must be complemented with other mobility solutions at the MNs.

Finally, SIP is classified according to the MN's Multiple Mobility-Aware Components model. Even though SIP is a fairly simple mobility solution, a major handicap is that SIP only offers mobility support for SIP applications, while all the other studied solutions enable mobility support for all existing applications in the existing Internet. Thus, either the existing deployed applications must be modified in order to run on top of SIP, to become mobility-aware, or other alternative mechanisms are necessary for supporting the mobility of these legacy applications in the current Internet. Thus, comparing to all the other heterogeneous mobility protocols, which offer transparent mobility services to all existing applications without modifications, SIP features a much poorer transparency level.

Concerning the Network Independence framework component, only TCP-Migrate is able to support mobility services exclusively using the existing network entities without making those mobility-aware (No Mobility-Aware Components model). This is achieved by modifying the MN's DNS entries with the current location of the terminal, which is able to redirect the new connections (the already established connections being dynamically redirected by using the token field in the TCP SYN messages).

All the other protocols involve modifications to the network, and are mainly distinguished if the existing legacy equipment and topologies can be maintained. SIP, similar to MIPv6, is also able to support global roaming with only the addition of a single piece of equipment to the home domain. These are the application-specific proxies that are able to redirect the SIP calls to the MN's current location; no further proxies are required by SIP in the visited domains (Single Mobility-Aware Equipment Addition per Home domain model).

MIPv4 and other recent IPv6 micro-mobility protocols are divided according to the intermediate classification (Single Mobility-Aware Equipment Addition per Subnet model), by requiring mobility support at each visited subnet, through the addition of a mobility aware agent. While this problem was specifically addressed in the base MIPv6, as previously described, this transparency improvement was broken again in order to achieve further efficiency gains. For instance, the handover improvements featured in fMIPv6 require special support on network's access routers to provide fast binding updates and data buffering operations, which are inherently mobility-aware operations.

The latest IPv6 micro-mobility protocols further biases this efficiency-to-transparency trade-off in favour of efficiency, namely in the seamless-MIP, IDMP and BCMP proposals. All these protocols can be regarded as the combination of fMIP with hMIP operations, and provide more efficient mechanisms through the introduction of additional IP mobility agents in the middle of the visited domain, as well as the necessary access routers agents that are deployed adjacent to the MNs. In particular, seamless-MIP extends the fMIP+hMIP architecture with the decision engine entity. However, the key point in all these proposals is that they keep the MIP's original spirit of support for the existing LRs and topologies (Multiple Mobility-Aware Component Addition per Domain framework model).

[12] However, it should be noted that some short UDP transactions, such as DNS resolution queries, don't require mobility support due to their small timescales and recovery capabilities [93].

Finally, all the three host-specific [45] efficient micro-mobility protocols – TIMIP, HAWAII, CIP - feature a similar architecture which requires specific routers support and specific topologies, that must be physically organized in a strict tree structure. This is done to provide a very close hierarchical mobility support for the MNs, aiming to reduce handover latency to the minimum and to enable other efficiency gains like the forwarding of data packets unencapsulated inside the domain (e.g. CIP, TIMIP). However, all these protocols do not support the existing legacy routers and topologies, which preclude the immediate or smooth deployment of the mobility service in existing networks (Mobility-Unaware Equipments Replacement framework model).

Concerning the Peer Independence framework component, most IP layer mobility protocols take specific transparency attention in order to support the communication of the mobility-unaware fixed CNs with the MNs (No Mobility-Aware Components model). The exception is the route optimization support for MIPv6, which requires the inclusion of a MIP client in every possible fixed CN to provide optimized direct data routing (Single Mobility-Aware Component framework model). While this precludes immediate utilization of such optimized routing, as the vast majority of the existing fixed CNs are MIP-unaware, and thus unable to perform this optimization, this design option allows for a smooth upgrade path, as the fallback basic triangulated routing is always provided using the HA.

As pointed out previously, these transparency concerns regarding the fixed CNs are not present in the higher layer mobility protocols: in fact, by always requiring specific mobility-aware fixed CNs, without any fallback option, none of the protocols HIP, TCP-Migrate or SIP permit immediate utilization or a smooth upgrade path – thus, at first, communication is only possible between MNs running the new protocol. Like the MN component, here the key difference is the amount of modifications required at each fixed CN; HIP requires a Single Mobility-Aware Component (the client application), providing a transparent mobility service to all existing applications at the fixed CNs; finally, for the same reasons described for Terminal Independence framework's classification, TCP-Migrate is classified with the Double Mobility Component, and SIP is classified with the Multiple Mobility-Aware Components model.

3.3 Efficiency vs. Transparency Quantification

3.3.1 Overview

This section presents a possible view of the efficiency-to-transparency trade-off characteristics that each protocol inherently has, regarding their deployment of mobility services in the typical existing legacy networks, and in the future All-IP ones.

This contribution aims to complement the previous research work in this topic by summarizing the presented frameworks in simple quantifiable metrics according to the author's perspective, by quantifying and aggregating the individual frameworks' components described previously. These metrics are then subsequently used to illustrate patterns in the efficiency-to-transparency trade-off of the protocols' families, and to help comparing the protocols' stronger and weaker points.

3.3.2 Methodology description

To perform such evaluation, framework's previously described components and models are subjectively quantified according to their expected impact on the final metric, without considering the protocols themselves. For this, each set of models of each component were sorted using the same ascending order already presented in the framework's descriptions. Then, each model was compared to the others of the same component, and was associated to a certain weight, using a 5-point continuous line-scale that purposely only features the two anchor end points on the extremes (Worse / Better). This line-scale is depicted in Table 1.

Value	Subjective Comparison
1	Worse
2	
3	
4	
5	Better

Table 1: Framework's Models comparison metric (Ci)

Afterward, a similar operation was conducted that compared the framework's components among themselves; again, the framework's components were firstly sorted and compared, and then associated to a second 5-point interval scale, depicted in Table 2, which evaluated the expected impact on the final efficiency or transparency metric. In particular, this process resulted that the different components, which have non-distinguishable expected impacts on the final metric, were associated with the same values in the scale.

Value	Expected Impact On Metric
1	Very Low Impact
2	Low Impact
3	Medium Impact
4	High Impact
5	Very High Impact

Table 2: Framework's Components impact on the global metric (Mi)

The proposed values for the framework's components and the component's models are depicted in Table 3 and Table 4, for each framework, respectively.

Components	Ci	Models	Mi
Movement Detection	4	Passive	1
		Reactive	3
		Predictive	4
		Active	5
Anchor Point Location	5	Inter-Domain	1
		Intra-Domain	4
		Cluster	5
Packet Loss Protection	2	None	1
		N-Casting	4
		Buffering	5
Flow Perturbation Avoidance	1	None	1
		Packet Marking	4
		Smooth Flow Redirection	5
Signalling Propagation	2	Multiple Core Networks	1
		Inter-Domain	3
		Intra-Domain	5
Control Encapsulation	1	Large	1
		Small	2
		In-Band	5
Active State Maintenance	1	Explicit	1
		Implicit	5
Idle State Maintenance	2	Uniform	1
		Exponential	5
Inter-Domain Data Forwarding	5	Macro Triangulated	1
		Macro Optimal	5
Intra-Domain Data Forwarding	3	Micro Triangulated	1
		Micro Tree-Optimal	3
		Micro Optimal	5
Data Encapsulation	2	Full Tunnel	1
		Soft tunnel	3
		None	5

Table 3: Efficiency Framework's Components impact into Global Efficiency metric

Components	Ci	Models	Mi
Terminal Independence	4	Multiple Mobility-Aware Clients	1
		Double Mobility-Aware Clients	2
		Single Mobility-Aware Client	3
		No Mobility-Aware Components	5
Network Independence	3	Mobility-Unaware Equipments replacement	1
		Multiple Mobility-Aware Equip. Addition / Domain	2
		Single Mobility-Aware Equip. Addition per Subnet	3
		Single Mobility-Aware Equip. Addition per domain	4
		No Mobility-Aware Components	5
Peer Independence	5	Multiple Mobility-Aware Clients	1
		Double Mobility-Aware Client	2
		Single Mobility-Aware Client	3
		No Mobility-Aware Components	5

Table 4: Transparency Framework's Components impact into Global Efficiency Metric

Then, the metric value for each protocol was found, by averaging the models of each protocol weighed by the corresponding component's impact on the final metric. Then, the framework value for a given protocol $V_{protocol}$ is given by:

$$V_{protocol} = \frac{\sum_{i=1}^{Cn} \left(C_i \times M_{i,\ protocol} \right)}{\sum_{i=1}^{Cn} \left(C_i \right)} \qquad \text{(Eq. 1)}$$

where:

- C_i stands for the expected impact of each framework component,
- $M_{i,protocol}$ stands for the subjective comparison value of the component's model according to which this protocol has been classified to.

Thus, the final metrics were found by applying this formula to both frameworks and to all the studied protocols in order to find the proposed orthogonal efficiency and transparency global metrics.

3.3.3 Application to the frameworks

All IP micro-mobility protocols were combined with both the baseline MIP protocol and the macro-mobility route optimization extension (where such functionality was possible, depicted by the "+RO" tag). The resulting metric values were then related in a XY scatter graph, shown in Figure 4. Finally, the protocols were clustered into families according to their position in the graph, being separated for L3 micro-mobility solutions, L3 macro-mobility solutions, and L4-L7 above-IP mobility solutions.

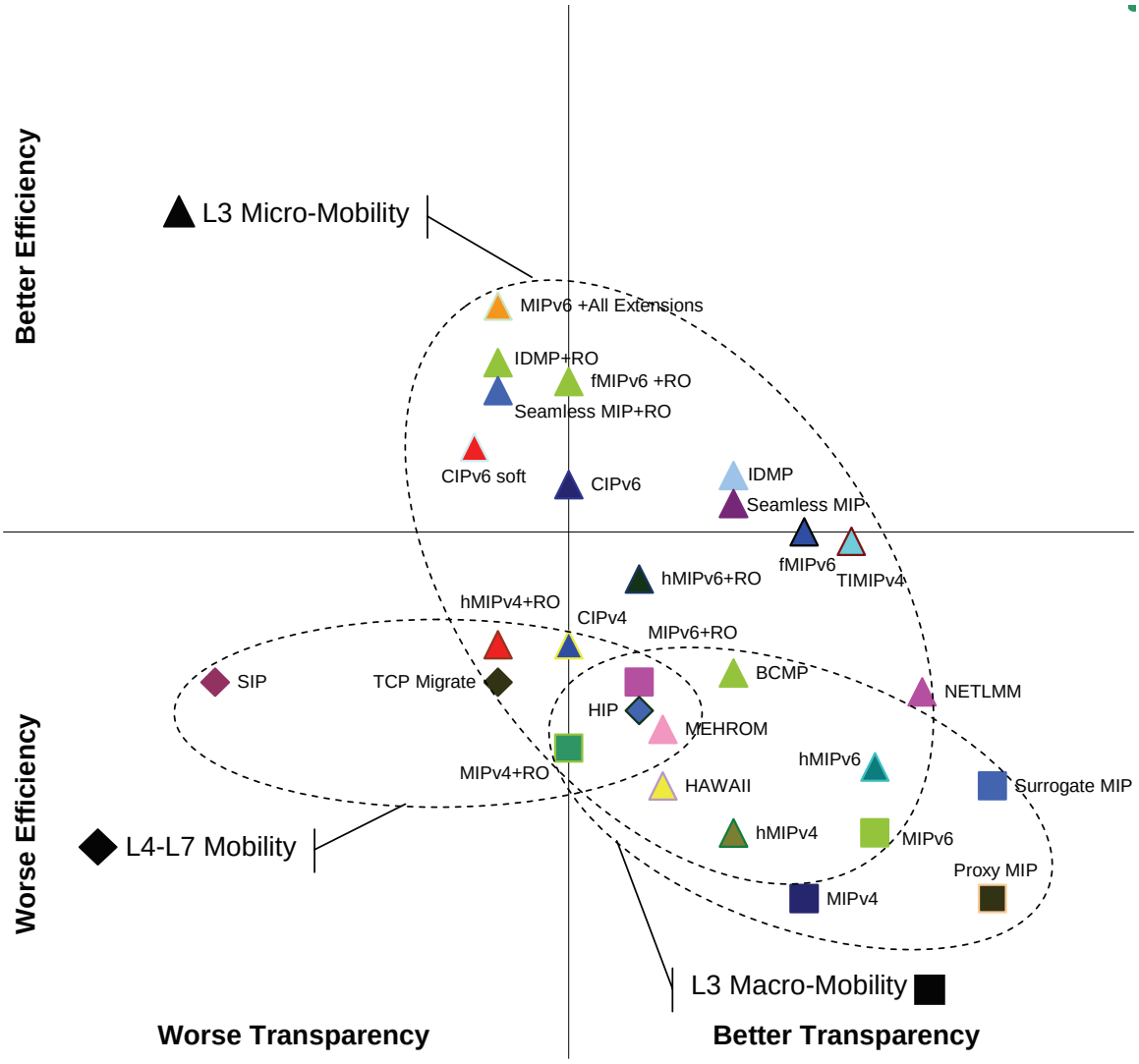


Figure 4: Proposed global efficiency and transparency metrics for state-of-the-art protocols

Here, the efficiency-to-transparency trade-off is apparent, as protocols that require the least modifications (better transparency - right side quadrants) typically only achieve the poorest efficiency level (bottom-right quadrant). On the other hand, the most efficient protocols (better efficiency - top side quadrants) typically only achieve the poorest transparency level (top-left quadrant). This trade-off is also clear by comparing the introduction of the MIP's RO option in the various IP protocols as such optimisation improves efficiency at the cost of degrading transparency, as it was previously described.

The figure illustrates that, typically, the IP macro-mobility solutions are mostly characterized by the provision of transparency to the fixed deployed Internet, even if it results in a lower efficiency value. While the above-IP mobility protocols feature less transparent mechanisms in the deployed Internet, they will achieve a higher typical efficiency. However, such efficiency levels are still far from the ones achieved by micro-mobility solutions, a sign of having a greater amount of research effort in the last years. Thus, considering the framework studies, which are illustrated in this graph, and previous research work [47] [76] [80] [90], it can be argued that the IP layer with micro-mobility extensions is the one that is best suited to handle the mobility support of mobile nodes, both efficiently and transparently, in current deployed networks and future All-IP ones.

The graph also hints that TIMIP, fMIP and its derived proposals (IDMP / Seamless MIP) are the protocols which are expected to be best suited to address the objectives defined in this PhD thesis work – efficiency and transparency – by generically featuring the highest efficiency-to-transparency relation. This motion will be more formally analysed in the following section.

3.4 Requirements of a Global Solution

This section will discuss the possible contributions of the state-of-the-art standards and previous research work to this PhD thesis. For this, a heterogeneous and transparent solution working on the IP layer is to be proposed, which must cover both immediate and future utilisations. Following the IP state-of-the-art proven design, this global solution will use the well-known dual protocol design, in which the complementary micro and macro mobility scenarios are considered separately. As it is described in the literature, this design allows for a solution which is both highly efficient and scalable, where the good scalability design of macro mobility is combined with the increased efficiency of micro mobility, merging the strong points of both approaches into a better solution [43] [44].

Regarding micro mobility, there is currently no sufficient consensus, neither in the scientific community nor in the industry, regarding the adoption of one of the existing state-of-the-art proposals as the standard solution. This way, it is equally viable to extend, merge or even propose entirely new solutions. As depicted in the proposed frameworks' studies, and illustrated in the previous section, the most efficient and transparent micro-mobility proposals are currently TIMIP, fMIP and its derived proposals (IDMP / Seamless MIP).

Taking the **Efficiency** framework study into account (Figure 1 and Figure 2) and limiting it to the most transparent micro-mobility solutions (as illustrated in Figure 4), it can be verified that the fMIP protocol and its derivations are, at the present, the best positioned protocols to provide an efficient micro-mobility solution, as a whole, within the components of seamless handovers and mobility overhead. However, further efficiency gains are still possible, namely the most efficient models that are already supported by alternative protocols (e.g. limited signalling propagation, reduced state-maintenance, etc.) or the ones which, although identified, are not supported by any micro-mobility protocol (e.g. optimal routing, smooth flow redirection).

Taking the **Transparency** framework study into account (Figure 3), and limiting it to the most efficient micro-mobility solutions (as illustrated in Figure 4), it can be verified that the TIMIP protocol is, at present, the best positioned protocol to provide a transparent micro-mobility solution, as a whole, within the components of terminal, network and peer independence. However, further transparency gains are still possible, notably by improving network independence through the support of existing legacy routers.

Nevertheless, by considering both frameworks simultaneously, it can be argued that TIMIP is, at the present, the best positioned protocol to provide an efficient and transparent micro-mobility solution. This happens because this protocol has a more flexible expected efficiency-to-transparency relation, a result of having been designed specifically with higher transparency concerns - as efficiency is a significantly more researched paradigm, it is expected that such paradigm will be more easily integrated in an already transparent solution than vice-versa.

Thus, the micro mobility component of the global mobility solution will be based on the initial TIMIP proposal. This protocol is to be generalised for IPv6 networks deployment, and enhanced to address the remaining efficiency and transparency models that it currently lacks, namely the **predictive detection**, **buffering packet loss protection**, **smooth flow redirection**, **in-band control packets encapsulation**, **exponential idle state maintenance**, **optimal intra-domain forwarding**, **and legacy network equipment support models**.

Regarding macro mobility, the current IETF standard solution is the MIP protocol, for both IPv4 and IPv6 networks. Considering that the efficiency problems of the standard solution are handled by the micro-mobility separation, the standard solution continues to suffer from the transparency issues noted previously – lack of **terminal independence** and, for route optimization scenarios, also **peer independence**. On the other hand, the described sMIP and tMIP proposals have shown progress concerning the first problem by providing a terminal independent mobility solution for IPv4 networks. However, only sMIP is actually compatible with the MIPv4 standard, as the latter uses a different, signalling-less, set of operations.

Thus, the macro mobility component of the global mobility solution will be based on the sMIP proposal, which, by being fully compatible with the TIMIP protocol, will be directly applicable to the micro-mobility solution to be researched in this thesis.

4 eTIMIP Basic Mobility Architecture

This chapter presents the new architecture that is proposed to address the previously outlined major objectives – **Efficiency** and **Transparency**. For this, the architecture features design aspects that support the frameworks' major aspects of seamless handovers, low mobility overhead, terminal independence and network independence.

The proposed architecture was derived from the previous TIMIP/sMIP model, which is now extended to address the features missing in the initial protocol. In this process, three proven design features that were already present in TIMIP/sMIP have been inherited by the new architecture:

- The terminal independence paradigm, where all mobility functions are exclusively performed by the network, without requiring support on the Legacy Mobile Nodes.
- The agent tree usage, where a hierarchical local registration enables efficient fast handovers and tree-optimal routing support.
- The micro / macro mobility paradigm, where global mobility is complementarily managed with separate mechanisms, combining the scalable and transparent macro-mobility service provided by the sMIP protocol, with the efficient and transparent micro-mobility service provided by the protocol specified in this chapter.

The micro-mobility component is addressed by an extended version of TIMIP, called **enhanced TIMIP** (eTIMIP). The eTIMIP architecture introduces a generic flexible mobility service based on an **overlay network**. Using the overlay network, the protocol is able to separate the underlying physical network that features the existing legacy network elements and non-mobility-aware fixed routing, from the added or upgraded mobility-aware agents that create, among themselves, the mobile routing that supports an efficient and transparent mobility service.

Using this architecture, a base secure mobility service is defined (**basic eTIMIP**) that, by using an agent tree and a mobile subnet on the overlay network, supports a mobility service with better transparency and similar efficiency as the best alternative solutions of the state of the art. Thus, basic eTIMIP is able to provide full transparency without sacrificing efficiency, enabling a flexible application of the mobility service in particular deployment scenarios.

In particular, eTIMIP transparency features are an advance on traditional efficient micro-mobility solutions, as the latter are tightly coupled with both the mobile nodes and the network itself, by requiring mobility-aware terminals, routers and custom hierarchical topologies. This is not the case of eTIMIP, as it enables immediate utilization and smooth deployment in existing networks, by supporting both Legacy Routers and Legacy Mobile Nodes. Initially, eTIMIP agents must only be deployed adjacent to the terminals, in a variety of wired and wireless scenarios, to support the mobility of unmodified LMNs. Later, additional agents may be introduced to improve efficiency.

On the other hand, eTIMIP efficiency features are also an advance on traditional transparent micro-mobility solutions: by using an agent tree instead of a single anchor, eTIMIP supports fast handovers and a tree-optimal routing service, as each handover must only update the routing entries located in the vicinity of the terminal, and the data packets can travel using the shortest paths inside the overlay agent tree. In addition, basic eTIMIP also supports

optimized state maintenance, by using the active terminal's own data packets to transparently refresh their state without any overhead, and the support of a fast detection security procedure, which enforces the use of the mobility services by authenticated LMNs only, via a fast local validation message exchange using session keys.

In addition, the use of an overlay network also improves reliability against wired link failures, as alternative links can now be used to route the data packets to mobile nodes, and also supports the usage of multiple mobility-aware points of attachment to the outside of the domain, enabling mobility routing at an earlier stage. Finally, eTIMIP is now defined generically, requiring only minor adaptations to be deployed in either IPv4 or IPv6 networks. These versions are called, where appropriate, "eTIMIP for IP version 4" (eTIMIPv4) and "eTIMIP for IP version 6" (eTIMIPv6), respectively. For the former, backward-compatibility options enable compatibility with the previous TIMIPv4 protocol.

The rest of this chapter is organized as follows: the first section will be focused on eTIMIP architecture which supports both the base components and the modular extensions; the second section will present the basic eTIMIP routing operations, being the individual mechanisms specified both through formal and informal methodologies; the final section will describe the adaptation of the generic architecture for usage in IPv4 and IPv6 networks.

4.1 eTIMIP Architecture

Figure 5 outlines the eTIMIP architecture. Each eTIMIP domain is logically organized in two network planes – physical and overlay – to decouple the mobility service from the underlying physical network. This duality is represented in the figure by two separate planes, with the correspondence between the entities of both network planes shown as dotted lines.

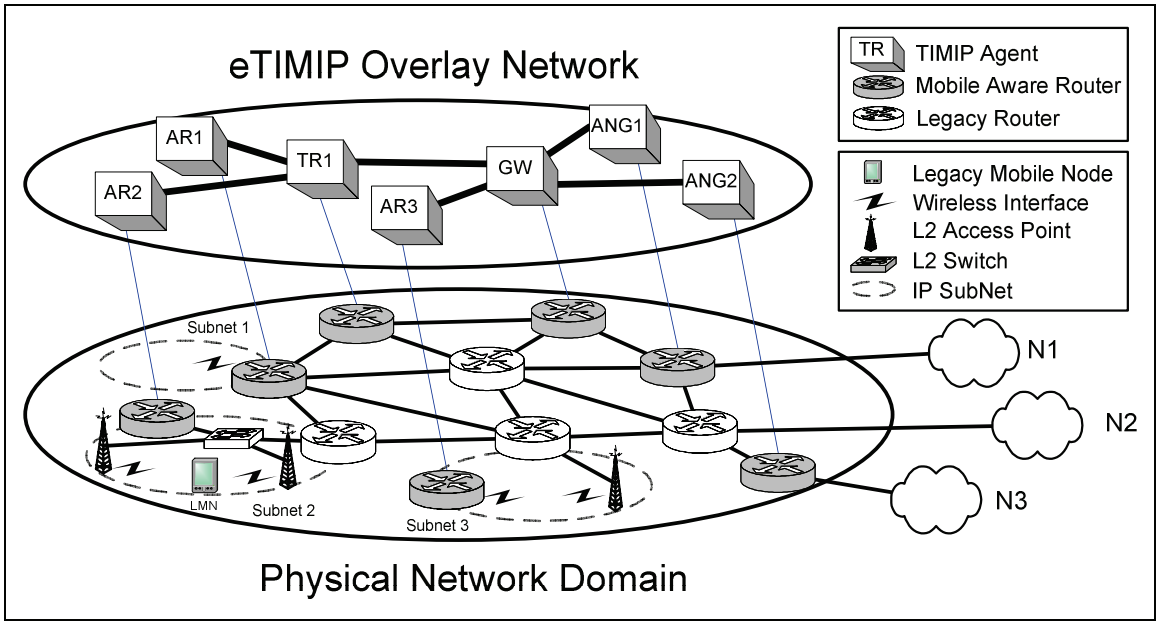


Figure 5: eTIMIP micro mobility architecture

4.1.1 Physical network architecture and components

Regarding the physical network, each eTIMIP domain can have any possible fixed topology, with any combination of routers, fixed links, wireless interfaces and Layer 2 equipment,

of any technology. To introduce mobility into the network, two options are available: upgrading or replacing existing Network Elements (NE) with eTIMIP, turning them into mobility-aware routers, or adding new mobility aware NEs to the network without modifying the deployed legacy routers. Any combination of these options is possible. In the physical domain, the figure illustrates examples of such mobility-aware equipment in grey, and such legacy mobility-unaware routers in white.

All Layer 3 elements, represented as cylinders in Figure 5, are regular IP routers and it is assumed that the routing among them is managed by an appropriate intra-domain routing protocol, like the Open Shortest Path First (OSPF) [110], the Routing Information Protocol (RIP) [109], or the Intermediate System to Intermediate System (IS-IS) [118] protocols. This routing protocol should be able to cope with the wired topology changes, namely link and node failures, and to route dynamically packets inside the domain between physical NEs. Additionally, some routers located in the edge of the domain may use inter-domain routing protocols, such as the Border Gateway Protocol (BGP) [111], to connect to the outside networks. From now on, these protocols will be designated as **fixed routing**, in opposition to eTIMIP routing, which will be called **mobile routing**. In all cases, the eTIMIP mobile architecture is independent of the fixed routing protocol, which can use any kind of metric, namely QoS metrics based on the network traffic conditions [130].

On the borders of the domain there are multiple IP subnets, represented in the figure as dashed ovals, containing the hosts, which may have a mobile or a fixed nature. Regarding the fixed terminals, they are supported by the fixed routing protocol only, without the influence of the eTIMIP mobility protocol.

Due to the heterogeneous support of eTIMIP, which is a key characteristic of any L3 solution, the users can connect to the domain using different access technologies. This is made possible as the physical network may contain any kind of L2 equipment, namely APs, Switches and Access Controllers (AC). The APs are layer 2 pieces of equipment with wireless network interfaces, of any technology, that provide network access to the MNs, being physically adjacent to them; the ACs are advanced wired L2 switches that cover a set of APs and centralise their control [108]. Support is also given to Layer 3 Access Points (L3AP) [98], which are IP routers that contain typical AP functionality, namely wireless interfaces and L2 management. This type of network equipment is particularly useful in recent network designs, namely in Wireless Mesh Networks (WMN) [119].

Without a loss of generality, some of the most common wireless and wired deployment scenarios and options are illustrated in the figure. Subnet #1 illustrates the concept of an L3AP, where an IP router also contains typical AP functionalities. Subnet #2 is the classical subnetwork design, where multiple APs are connected to a L3 router through wired infrastructure, like Ethernet switches. Subnet #3 illustrates the usage of a single AP directly connected to a Legacy IP router in a back-to-back design.

4.1.2 Overlay network architecture and components

On top of this physical network, an **overlay network** is built to provide efficient mobility support, using eTIMIP software agents that manage the domain's mobile routing information. These agents, represented as boxes in Figure 5, and linked to their mobility-aware routers in the physical plane, are deployed on the network in an incremental way, either by upgrading existing upgradeable routers or by introducing additional ones into the network. This smooth deployment feature of eTIMIP is a consequence of its network independence support, which enables quick deployment of micro-mobility and provides incremental and

smooth upgrades of existing legacy networks. An example of this smooth upgrade process is illustrated in the sequence of figures presented in Appendix A.

The eTIMIP agents constitute the overlay network by establishing and maintaining a **logical tree** between them. The generic building block of the overlay network is a generic agent designated as eTIMIP Router (**TR**), using a terminology that is adapted from the TIMIP architecture, and combined with this area's practices [43] and recommendations [3]. Inside the overlay network, control and data packets are swapped between the TRs to support the LMN's mobility. The control packets are used to distribute and maintain the LMN's routing information inside the network. This routing information is subsequently used for the forwarding of the LMN's data packets to the correct destinations inside the network.

Depending on the location of the tree, their adjacency to the LMNs and the type of connections to the outside of the domain, the TRs can also be endowed with additional, possibly multiple, roles. At the top of the tree, a special TR called eTIMIP Gateway (**GW**) is used to centralize the management functions, being unique to any given eTIMIP domain. For connecting with the networks outside the domain, enhanced TRs, named eTIMIP Access Network Gateways (**ANG**), can be used to connect the domain to the outside networks. These border nodes, which can be multiple per domain, establish the boundaries of the eTIMIP mobility-aware protocol, as recommended by the IETF in reference [3]. The ANG's dual is the eTIMIP Access Router (**AR**), which is a TR that offers IP connectivity to LMNs, being physically located in the same subnets of the LMNs. The ARs are able to detect the LMN's movements and to generate and maintain signalling on behalf of them. The final components of this overlay network are the **LMNs**, which are only required to connect to the network via a suitable wireless interface. Using terminal independence, it is the network that automatically detects this event and reconfigures itself to provide the necessary IP connectivity to the terminals.

As such, the minimum eTIMIP network is the one that contains only the AR nodes adjacent to the LMNs, which establish a minimal logical tree between themselves (Appendix A).

4.1.3 Overlay network transparency support

The provisioning of a terminal and network independent solution is directly related to the problem of how to force the packets to reach the eTIMIP agents. The problem is solved by inserting eTIMIP agents at the network's boundaries to intercept all incoming traffic.

Deploying ARs at the LMN's network boundary is always possible without modifying the pre-existing physical network, as the eTIMIP ARs can either be incorporated in pre-existing upgradeable access routers or added to the same subnets of the LMNs: In Figure 5, AR1 represents the first case, by using a mobility-aware L3AP; AR2 focuses on the second case, in the common scenario where the AR can be introduced in the sub-network through a wired connection. AR3 also illustrates the second case, where an AR has been added to the sub-network with a wireless connection without modifying the existing wired infrastructure. In all deployment cases, packets sent by a LMN are directly forwarded to its AR, as the ARs uses L2 discovery mechanisms that transparently configure themselves as the LMN's default router.

In contrast, deploying ANGs at the core network's boundary may only be possible by using pre-existing upgradeable border routers or by modifying the pre-existing topology. The former case is illustrated by ANG1, while the latter case is illustrated by ANG2; the benefit of having mobility-aware capabilities at the network's edge is that the data packets that pass

through these nodes are subject to mobility routing at an earlier stage, which can improve forwarding efficiency and be used to provide load-balancing. Regarding the edge routers that remain mobility-unaware (illustrated by the legacy router directly connected to external network N2), the LMN's packets that pass through this node are always transparently redirected to the GW; once there, they can then be routed by eTIMIP as normal. This transparent redirection is accomplished through the usage of a **mobile subnet**; this is a fixed routing's regular Classless Inter Domain Routing (CIDR) subnet [112] that is associated to the GW, but which is used to provide IP addresses to the LMNs. Thus, from the fixed routing point of view, all LMNs are virtually located in the GW, which achieves transparency without modification of the LRs.

4.1.4 Base eTIMIP components

The definition of the eTIMIP protocol is divided into a base specification, which provides a full transparent service with good efficiency, and a number of eTIMIP extensions that can be optionally and modularly combined to achieve further efficiency, scalability, reliability or control gains. This modularization design follows a number of well-known Internet best practices, namely [102] [169] [170] [171]. The protocol is described at first in a generic form, being afterwards complemented with the specific adaptations required for deployment in existing IPv4 and IPv6 networks.

The base eTIMIP specification considers a simple mobility method that provides a completely transparent service, supporting both LMNs and LRs, and a good efficient service, supporting tree-optimal data packets forwarding and an implicit state maintenance for active terminals. On the other hand, basic eTIMIP uses a simple handover scheme and local security check that is enough to provide a simple low-latency service, although without special handover loss guarantees. These features are supported in eTIMIP routing algorithms similar to the one proposed in the TIMIP architecture, being now designated as **basic routing**.

Inside the overlay network, the eTIMIP agents manage simple per-LMN routing entries, named **basic entries**, which contain the information of the next hierarchical agent towards the LMN. This enables a fast handover operation, as only the routing entries located in the vicinity of the terminal, in the sub-tree that connects the involved previous and new ARs, must be updated in each handover. The control packets that support this basic handover operation are sent agent-by-agent, following the strict path inside the logical tree, using the logical connections of the agent tree. The data packets also travel following the same shortest path inside the overlay agent tree, resulting in optimal forwarding inside the tree.

In both cases, each logical connection on the overlay network is dynamically mapped by the fixed routing to a path on the physical network, which can contain any number of physical links and routers. In addition, when legacy routers are present in the path chosen by the fixed routing to transmit data packets destined to MNs, then encapsulation or source routing techniques are needed to pass through these routers, using tunnelling operations. By using such encapsulation when necessary, the reliability of wired link failures is also improved, as alternative links can be used to route the eTIMIP control and data packets.

Regarding the state maintenance operation of the terminals in basic eTIMIP, this is optimized by using the active terminal's own data packets to transparently refresh their state without any overhead; if necessary, explicit refresh operations are used, but the refresh cycle itself is dynamic and managed by a back-off process sensible to the data forwarding operations. This enables such explicit refresh operations to be mostly focused on the routing entries that are actually being used to forward data packets, which are exactly the ones that are

most likely to be refreshed without overhead by the active terminals. This results in the support of fast departure detection of the active LMNs, while allowing a slower departure detection of the inactive ones.

Finally, basic eTIMIP also improves TIMIP's native fast handover capabilities, which mainly result from the usage of a fast hierarchical local registration, with a fast detection security authentication that is able to locally authenticate the terminal with only a local validation message exchange without the establishment of pre-shared security keys. In addition, in order to support terminal independence, eTIMIP does not require the change of the LMN's IP address at each handover, which also benefits the low-latency handover support.

4.1.5 Backward compatibility with TIMIP

Taking advantage of the fact that no special encapsulation operations are needed when direct links exist between involved TRs, the proposed architecture can be reduced to the earlier TIMIP model, providing a backward-compatibility option to the original protocol. This is possible by forcing the logical links to match the physical ones and by introducing additional TRs in the overlay network tree, if necessary. Additionally, a single ANG must be chosen to become the new domain's GW, as TIMIP does not support this new entity. Although direct extra links can be used, which increases robustness, the remaining legacy routers are ignored by this backward compatibility eTIMIP feature.

4.1.6 eTIMIP's description methodology

4.1.6.1 Components Specification methods

Each eTIMIP mechanism is described, illustrated and specified using multiple complementary forms that seek to benefit the clarity of the exposition without sacrificing correctness or non-ambiguity:

- An informal description, expressed in natural language, which focuses on the description of the algorithms, the distributed processes, and the interactions with the standard components of both the terminal's and the router's IP stack.
- An illustrative description, based on particular examples, which detail the eTIMIP operations, using time message diagrams that differentiate between control and data packets and feature a set of triggers that signal the internal communication between the eTIMIP modules.
- An authoritative formal specification, described in the form of extended mealy-type finite state machines (FSM) [172] with statecharts extensions [175], where each state transition, triggered by the arrival a single event, results in the execution of a list of actions.

4.1.6.2 Base eTIMIP specification methodology

Each eTIMIP module that constitutes the base eTIMIP specification is modelled by a single independent state machine. To form the complete system, the state machines are executed concurrently, being their communication possible either directly, via explicit triggers which generate new events, or indirectly, via updates of the global state in the main memory.

The explicit triggers are able to cause chain reactions between the state machines using a local broadcast facility [175], where the triggering state machine invokes a certain function that is received as an event in the triggered state machine. The possible triggers are described

in the next figure, being detailed the formal function name that is invoked / consumed (described in detail in Appendix B), and the informal symbol used in the application examples.

INTER MODULE TRIGGERS

PHASE		TO		
		Detection	Registration	Execution
FROM	Detection	(D) Unsecured Detection (S) Secured Detection	(C) Confirmed Detection	(U) LMN Departure Notification
	Registration			(F) Fast Handover
	Execution		(X) Power-Down (I) Init Idle	(R) Active Refresh (IR) Idle Refresh (T) Triangulation Detected

Figure 6: Trigger list between eTIMIP modules, by phase

In addition to explicit triggers, the state machine **events** can also be triggered by simple C-language-like logical constructs with the usual operators (And / Or / Not / Equality / Inequality / Precedence / Wildcard) and by the analysis of local private variables, which are assumed to be pre-initialized to zero (described in detail in Appendix C); the event processing logic is also simplified by assuming that untreated events are buffered for later treatment.

Similarly, in addition to the explicit triggers, the state machine **actions** can also modify local variables (Appendix C), and call auxiliary functions, which perform packet generation, timer scheduling, configuration information, timestamps managing and generic security functions, among others. All parameters are passed to the functions by value, being the full function list described in Appendix B.

Regarding the updates of the global state, a module can influence the operation of others by calling functions that update state variables residing in main memory (setter functions), which the other module will consume via reader functions. Both types of functions are detailed in Appendix B.

Finally, the presented illustrative time diagrams can present both the control and data packets simultaneously. The control packets are depicted using oriented arrows, which may feature some of the most important control fields. The data packets are depicted using shaded blocks, which are inclined according to its orientation (in respect to the time axis). Such data packets may also include the IP source and destination addresses information, where such information is essential to comprehend the time diagram. Finally, the time diagrams also feature a set of actions using rounded boxes, which loosely correspond to the ones performed by the formal state machines.

4.1.6.3 eTIMIP Extensions specification methodology

Regarding the optional eTIMIP extensions specification, to be presented only in Chapter 6, these are modelled through modifications and additions to the base specification, where

the base eTIMIP state machines are modified by the introduction of additional states, variables, events and functions. Thus, the eTIMIP extensions specifications will be presented as series of incremental state machines versions, ranging from the basic protocol to the full one.

In order to simplify these procedures, and to clarify the exposition, the base state machines are presented with explicit modular capabilities, in the form of additional dummy states. This technique is possible by introducing state transitions that do not involve any actions, and state transitions that immediately succeeds (i.e., by being triggered by a true() condition) [172][13], or a combination of both, to which results dummy states that will only be used in the subsequent eTIMIP extensions. Nonetheless, the presented base state machines generate the same sequence of actions for the same sequence of input events, i.e. is equivalent to any possible optimized versions.

Thus, each eTIMIP extension will refine the previous presented state machine with extra functionality, being stressed the differences to the immediately previous version in a bold typeface; for reference, the complete authoritative state machines are presented in Appendix C.

4.2 Basic eTIMIP routing operations

4.2.1 General overview

This section presents the basic algorithms and operations that constitute the base specification of eTIMIP, which constitute the minimum required set of mechanisms that enable the micro-mobility support of LMNs.

In this basic model, both control and data packets are forwarded using the shortest paths inside the overlay agent tree. Data packets are encapsulated only when legacy routers are present in the path chosen by the fixed routing. This type of routing is characterized by the hierarchical creation of per-LMN soft-state routing entries, named basic entries, which contain the next-agent information. This structure, which is similar to other micro mobility proposals [20] [21], provides a fairly good trade-off among:

- Domain scalability, as only one LMN entry per hierarchical level in the overlay tree is necessary.
- Low latency handovers, as the LMN's handover updates are restricted to the local subtree that connects the involved ARs, being thus limited to the vicinity of the terminal.
- Efficient resource utilization, as data packets always follow the shortest paths inside the tree.

The first step of the mobility service is executed when a LMN arrives at an eTIMIP domain. eTIMIP detects this initial arrival, authenticates the LMN's access to the network, and performs a **Power-Up** operation, which creates the initial LMN state in the network, by adding a new basic routing entry in all the agents that lie between the AR and the GW. The second step of the mobility service is executed when the LMN roams between ARs of the same

13 Quoting from Kurose and Ross [172], page 203, "When no action is taken on an event, or no event occurs and a action is taken, we'll use the symbol Λ below or above the horizontal line respectively to explicitly denote the lack of an action or event"

domain. eTIMIP detects this **Handover** and updates only the basic routing entries that became outdated in respect to the new LMN location. In parallel, these soft-state routing entries are periodically confirmed by the LMN's own data packets or by explicit refreshes otherwise; when the terminal is considered to have left the network, an explicit **Power-Down** operation is used to remove the corresponding basic routing entries from the network.

4.2.2 Detection phase

During the Detection phase, the LMN's location is dynamically and securely discovered by the network itself, which is responsible for tracking the LMNs' movements inside the domain. For this, the domain's ARs will continuously locate the LMNs connected to them and validate their authentication to use the mobility service. To perform this task, a single primitive, designated in the examples as "D", is used to signal the unsecured attachment of the LMN to the AR, which can use any kind of available information. After that, a subsequent primitive designated in the examples as "S" is issued after eTIMIP security procedures are executed, assuring the authenticity of the LMN and its authorization to use the mobility service. Finally, in basic eTIMIP, the AR will locally decide to start a handover process immediately, without consulting the other ARs, by triggering a final primitive designated in the examples as "C".

4.2.2.1 Generic Detection Algorithm

General description

eTIMIP uses a Generic Detection Algorithm (GDA) for detecting the presence of the LMNs in an independent form of the particular L2 technology in use. For this, the ARs use an L3 algorithm similar to those used by the "MAC learning" algorithm in L2 equipment, but which is executed at the IP layer. As such, by being independent of the L2 protocol, it provides detection support to heterogeneous access networks.

In this method, the AR continuously scans the wireless medium and listens to the received packets in order to detect the LMNs that may have arrived. The AR maintains the knowledge of the LMNs present in its own subnet, in a private local memory. When a LMN connects to an AR, the first packet it sends is interpreted by the AR as a new arrival, and an entry is created for it in this memory. Each entry is then refreshed by subsequent packets sent by the LMN, being removed from the memory when the associated lifetime timer expires, in order to assure that future arrivals of the same LMN are detected. If the entry is removed too early, the next packet sent by the LMN will simply trigger a new detection process.

This generic detection algorithm provides a transparent solution, which can be used in any kind of access technology. However, it does not provide an efficient solution as the detection is only driven by the transmission of packets by the LMN. However, it should be noted that this efficiency issue will be specifically addressed by an eTIMIP extension in the context of fast handovers. In addition, other MN-supported protocols might be used to achieve the same unsecured detection result, namely the standards-track protocols of Neighbour Discovery (ND) [115], Detecting Network Attachment [151] or Dynamic Host Configuration Protocol [155].

Application example

Figure 7 shows a temporal diagram example that represents the complete cycle of the generic detection algorithm. As stated in the figure, the first incoming IP data packet triggers the creation of a new entry in the local memory, triggering the "D" primitive that signals the

LMN attachment to this AR. Then, subsequent packets only refresh the LMN entry, without further actions. The cycle is finalized when the lifetime timer expires, by removing the LMN entry from the memory.

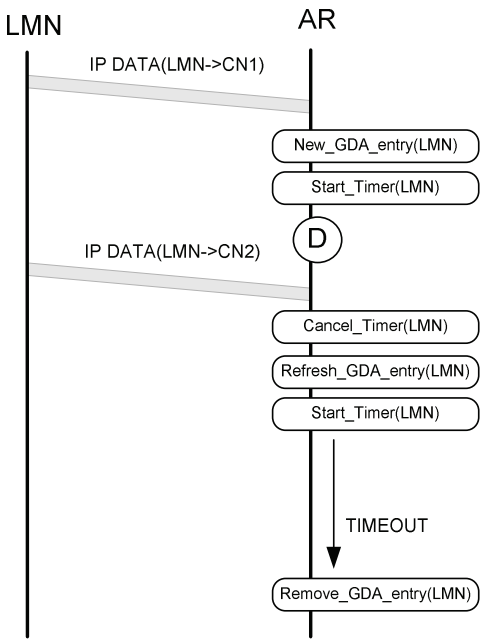


Figure 7: Generic algorithm detection

Formal Specification

The generic detection algorithm is formally described in the state machine of Figure 8, which uses the functions and triggers described in Appendix B and the variables described in Appendix C.

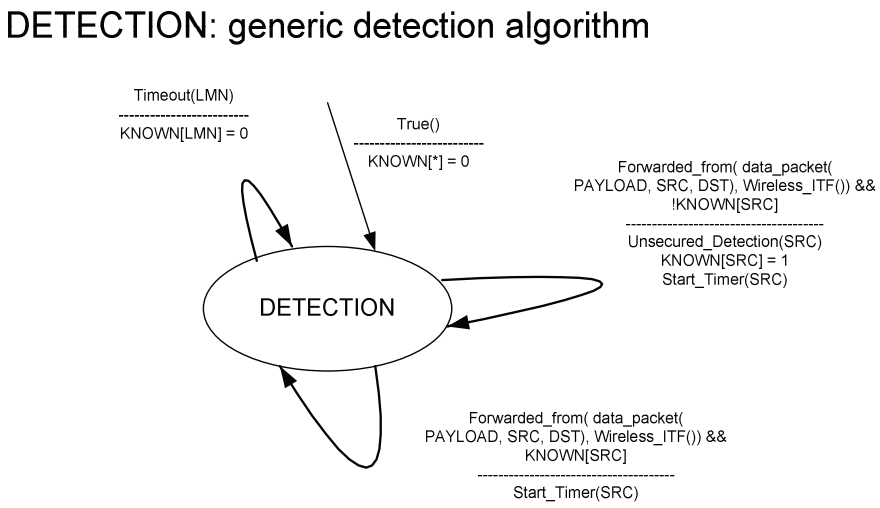


Figure 8: Generic detection algorithm formal specification - Optimized version

As explained previously (section 4.1.6), all basic eTIMIP state machine are presented in a modular form that both benefits the introduction of extra functionality and the clarity of the exposition. Essentially, this is done by the introduction of additional states and transitions that do not change the state machine behaviour, but which greatly improves its extensibility and clarity features.

In the particular case of the GDA state machine depicted above, these methods will enhance the state machine by aggregating the data packet reception event and the timer initialization action in shared transitions, and add additional dummy states that will only be used in the subsequent state machines. This results in the equivalent state machine depicted in Figure 9. The individual steps of this process are detailed in Appendix D.

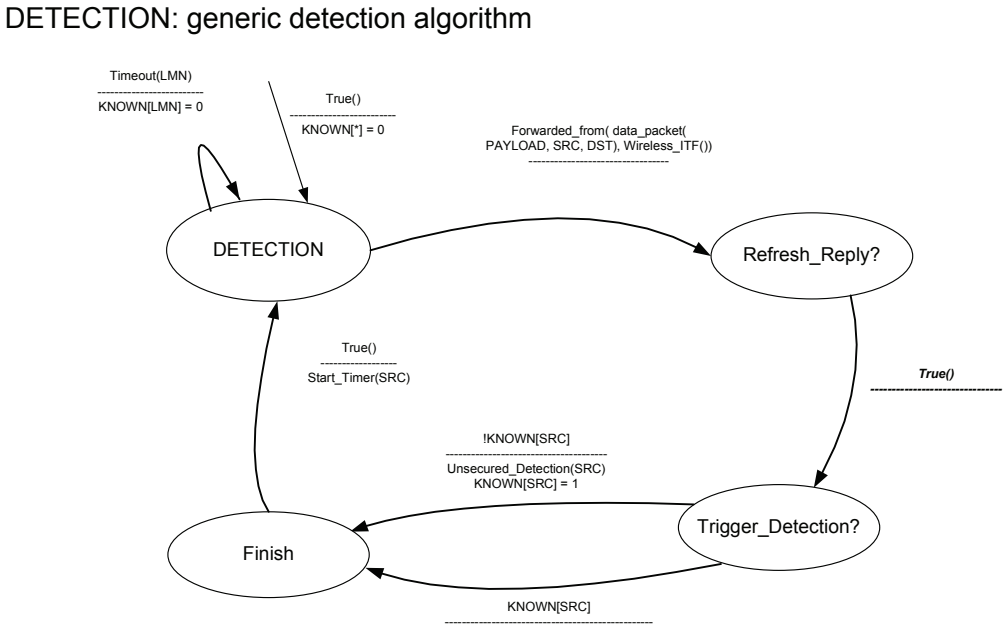


Figure 9: Generic detection algorithm formal specification - Modular version

Mapping of application example in State Machine

This section clarifies how the formal state machine generates the desired series of actions in response to a given series of events. For this, the previous application example of Figure 7 will be mapped step-by-step in the state machine of Figure 9.

Initialization

At first, when the state machine starts in the initial state (named “Detection”), the local array variable KNOWN is explicitly reset for all possible indexes, meaning that no LMNs have been detected in the state machine’s start.

First data packet

Then, a certain LMN will generate a data packet, which is detected in the AR’s wireless interface. Thus, the event Forwarded_from() will change the current state for the Refresh_reply? state, with the local variables PAYLOAD, SRC and DST having the values of the data packet’s payload, source address and destination address, respectively.

Then, after a trivial transition to state Trigger_detection?, the value of the KNOWN array for this LMN is consulted. As such variable holds “0”, the unsecured_detection() function will be invoked, which will trigger the state machine that is waiting for the generation of this event, continuing the mobility process[14]. In addition, the KNOWN variable for this LMN is updated to prevent subsequent detections of the same LMN. Then, in the last transition be-

[14] Which, in this case, is the detection security state machine, to be described in the next section.

fore returning to the initial state, the start_timer() function is called for this LMN, to install a new timer for this LMN that will enable the LMN's re-detection in the end of the example.

The previous state machine actions are mapped in the application example by the calling the "D" trigger, and by calling the new_GDA_entry and start_timer example functions.

Subsequent data packets

The application example continues by the generation of a second data packet by the LMN. Initially, the state machine performs the same transitions until it reaches the trigger_detection? state. However, as the KNOWN variable holds "1" for this LMN, no trigger is invoked (in contrast to the previous case).

In state "Finish", which is shared with the previous case, the start_timer() function is re-invoked, which cancels the previous timer and install a new one (according to the definition presented in Appendix C). Thus, this sequence of operations is mapped by the refresh_GDA_entry and cancel/start_timer functions in the application example.

Timeout

In the "Detection" state, the only considered events are the reception of a data packet and the expiration of a scheduled timer. The last part of the example deals with this latter case; after a certain time period without packet reception, the "timeout()" event is invoked for the LMN. When this happens, the KNOWN variable for this LMN is reseted, enabling the re-detection of the given LMN at a later stage (not shown in the example). These sequence of actions are mapped in the application example in the Timeout followed by the remove_GDA_entry functions.

4.2.2.2 Detection Security

General description

These simple MN detection techniques performed by the network should not be used as a definitive proof of the LMN's authenticity and/or authorization to use the network; just like in other unprotected networks, it is possible to trick the network with false MAC or IP addresses, using well-known spoof techniques [152]. To avoid them, security procedures should be employed to, at least, guarantee the users' authenticity through authentication procedures to protect the mobility service.

From the network's point of view, the LMN's mobility service security validation can only be done via a verification of the authentication and authorization credentials of the mobile terminal. As the mobile terminal is the one that must be authenticated by the network, minimal security functionality must be implemented at the LMN itself, or must be derived from existing generic security technologies / applications that the LMN may feature. In basic eTIMIP, this verification is done through the use of a security application, which is mobility unaware and does not modify the secure legacy IP protocol stack. This application manages a private / public key pair for the terminal and cooperates with the network's mobility agents for its authentication. By default, the following local efficient procedure is defined:

At each detection, the AR that detected the terminal will independently generate a session key, named PID, which is unique between the network and the terminal, and will verify the LMN's knowledge of it with a validation operation. If this operation is successful, a subsequent primitive ("S"), that signals the secure detection of LMN at its AR, is triggered; if this operation is not successful, the security procedures are initialized as follows:

First, the AR sends the LMN's information that needs to be confirmed to the GW. This information (field MNID) consists of its IP address and an opaque identifier provided by the terminal itself, which could be its Network Access Identifier (NAI) [154], which is asymmetrically encrypted with its own private key. The resulting Sec_Init_Req message is then encrypted and signed using a secret network key, with an arbitrary length, named NK, which is only known by the GW and by the remaining eTIMIP agents. When the GW receives the message, it consults an AAA infrastructure server to access the authorization of the LMN, by giving the encrypted opaque MN identifier (MNID), and securely retrieves the LMN's public key. After a positive acknowledgement of the AAA infrastructure [142], the GW informs the AR that the LMN is authorized to access the network and what is the value of its public key, using a Sec_Init_Ack message.

Then, the AR will generate the session key as follows: the network key is concatenated with the IP address of the mobile host and a one-way hash function is applied [113]. The result, concatenated with the IP address of the GW, will be the shared secret of the LMN and the network. This secret, named PID, can be defined according to the following equation:

```
PID := Concat (Hash(LMN, NK), GW)                    (Eq. 2)
```

After computing the PID, the AR encrypts the PID with the LMN's public key. This encrypted session key is sent to the LMN's security application using a Security_Init_Req message. Using the private key, the application decrypts the message and associates the PID to the IP address of the GW, and ends the operation with a Security_Init_Ack message. This initial operation results in the secure sharing of a unique secret key by the network and the LMN. The PID remains the same during each handover and can be easily computed by each AR without any communication.

Then, each time the LMN is detected, the validation process is issued to verify the LMN's knowledge of its associated PID. In this operation, the AR sends a Security_Validate_Req message to the LMN's security application. This message contains the IP address of the LMN, the IP address of the GW, a random value, named Rand, and a timestamp value, named AR_TS. The resulting quadruple is defined according to the following equation:

```
Tuple_1 := Concat (LMN, GW, Rand, AR_TS)             (Eq. 3)
```

This message is then digitally signed by the AR using the PID, and sent to the LMN.

Upon receiving this message, the application verifies its origin by validating the digital signature using the PID, increments the random value, and appends a local timestamp, named LMN_TS to it. This results in the following data, defined according to the following equation:

```
Tuple_2 := Concat(LMN, GW, Rand + 1, AR_TS, MNID, LMN_TS)    (Eq. 4)
```

This information is then digitally signed with the PID, and sent to the AR in a Security_Validate_Ack message.

Upon receiving this message, the AR verifies its origin by validating the digital signature with the PID, if the random value was incremented, and if its timestamp and the GW address information is the same as the one in the request. If these tests are successfully executed, then the AR signals the secure detection of the LMN by triggering the primitive "S", proceeding with the routing reconfiguration.

The main advantage of this method is that the PID remains the same during each handover and can be independently computed by each AR without any communication with the previous ARs or GW, resulting in a very efficient local form of authentication.

It should also be noted that these mandatory eTIMIP detection security procedures may be replaced if the LMN's authenticity can be verified by other means equally secure. Currently, this may include secure L2 technologies that explicitly provide a secure association of the terminals to the network [152], like 802.1x [141], or L3 technologies that may be present at the terminal with the same characteristics, namely SEND [149] in conjunction with CGA [150] [105]. Another L3 alternative is the use of the authentication for DHCP [155], in the same form as suggested for netLMM in reference [39], or the usage of Protocol for Carrying Authentication for Network Access (PANA) [144].

Application example: Detection security initialization

Figure 10 shows an example that represents the initialization process of the secure detection. First, the AR that detected the terminal computes the PID for this LMN and tries to validate the LMN authentication. The LMN security application will reply with an error, as is doesn't possess the PID yet, and indicates its identity encrypted with its private key (MNID).

Then the AR will send the message to the GW, which starts the LMN authentication and authorization process on a local Authentication, Authorization and Accounting server (AAA), which securely authorizes the LMN and obtains its public key. After that, the PID, encrypted by its public key, is sent to the LMN, and a new validation process is restarted.

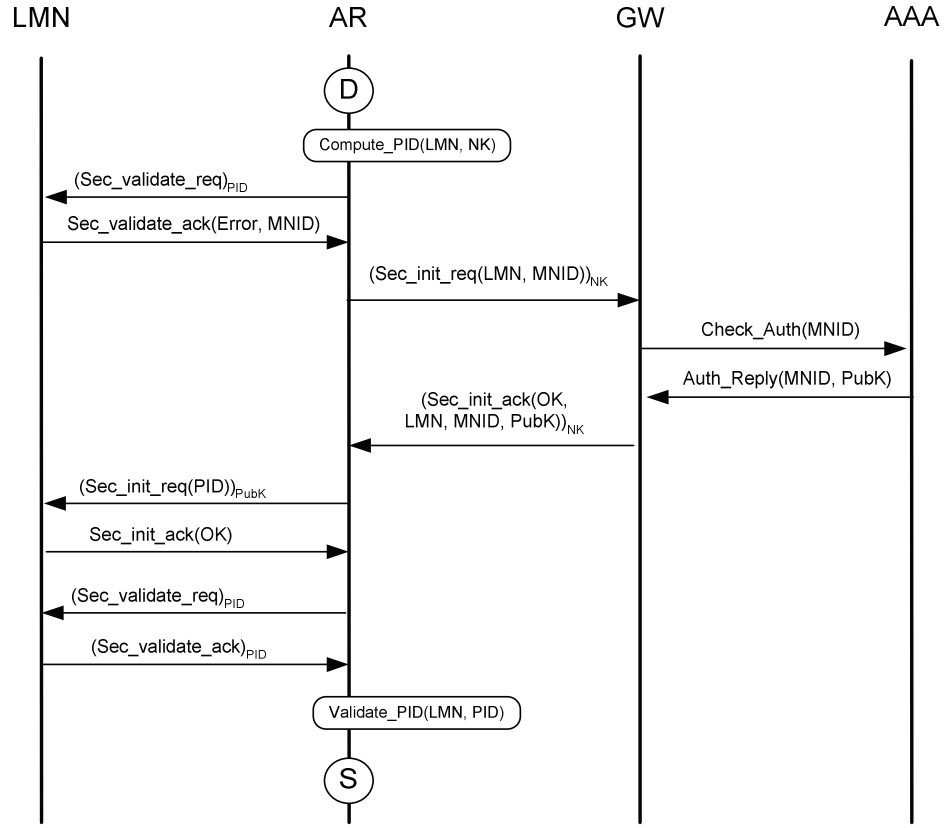


Figure 10: Detection security initialization

Application example: Detection security validation

Figure 11 shows the local validation process that is performed at each detection, after the PID has been delivered to the LMN. Here, the AR that detected the LMN computes the PID for this LMN, and tries to validate the LMN authentication. The LMN security application

will be able to use the PID to construct the appropriate reply message to the AR, which will lead to the triggering of the registration process at the AR (depicted by the trigger “S”).

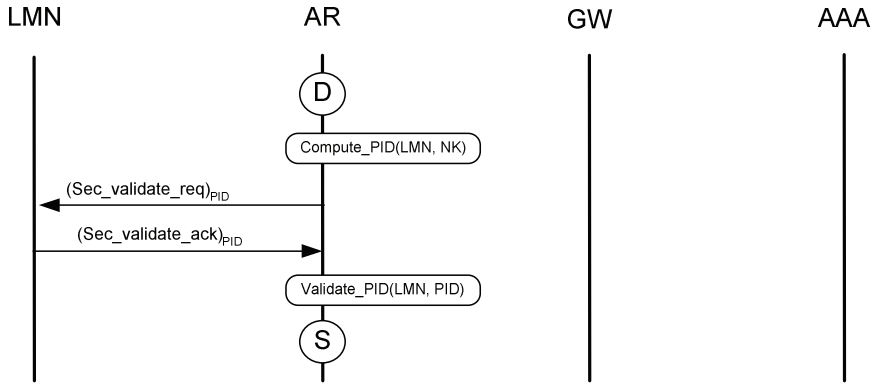


Figure 11: Detection security validation

Application example: Detection security validation through secure neighbour discovery (SEND) with Cryptographic Generated Addresses (CGA)

Figure 12 shows how the local validation process can be performed by alternative secure means that may reside on the terminal, being exemplified with the Secure ND with CGA combination. At first, the LMN will allocate IP addresses that are cryptographically related in a certain way with its public key [150]. Later, when the terminal transmits neighbour discovery messages, namely in the duplicate address detection mechanism [114], the generated messages will contain the public key in the CGA option as defined by SEND [149]. Upon reception of this NS message, the AR will validate the generated address of the LMN [105], using the Public key obtained securely in the security initialization mechanism described previously, and trigger the registration process if such test passes.

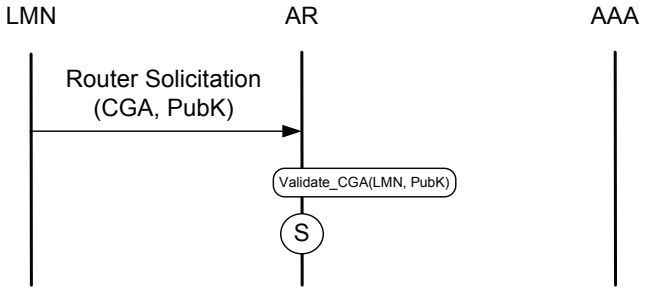


Figure 12: Detection security validation using SEND

Formal Specification

The described detection security procedures are formally described in the state machine of Figure 13, for the AR nodes, and Figure 14, for the LMN clients. All used functions are described in Appendix B, being the used variables described in Appendix C.

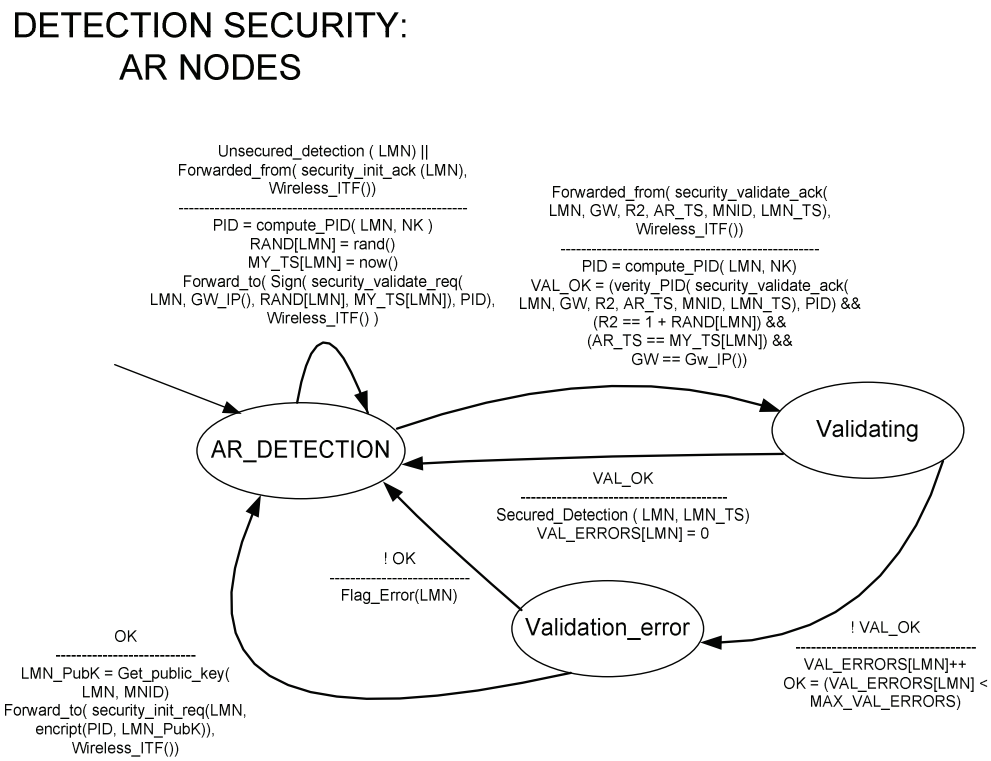


Figure 13: Detection security (AR nodes) formal specification

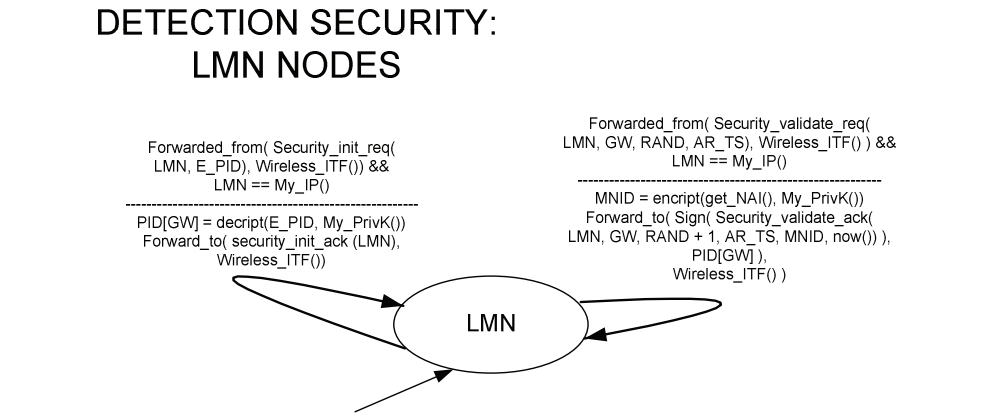


Figure 14: Detection security (LMN nodes) formal specification

4.2.2.3 Local Detection Confirmation

General description

After the successful detection security validation, the AR will proceed directly to the registration phase, by locally deciding to make a LMN handover to itself. As it will be seen later on, this component will be further expanded for the network-controlled handovers eTIMIP extension.

Application example: Local Detection Confirmation

Figure 15 shows a temporal diagram example which represents the local detection confirmation that basic eTIMIP routing uses, showing that it only performs a trivial decision.

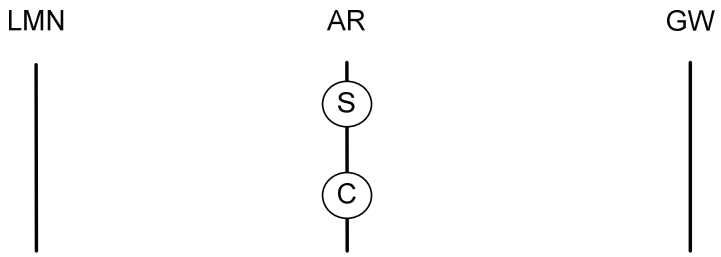


Figure 15: Local Detection Confirmation

Formal Specification

The local detection confirmation procedure is formally described in the state machine of Figure 16. All used functions are described in Appendix B.

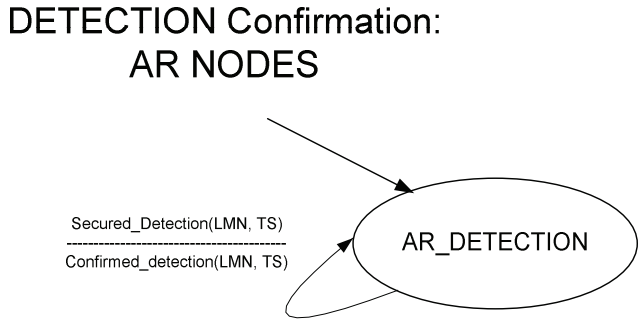


Figure 16: Local detection Confirmation formal specification

4.2.3 Registration phase

During the Registration phase, the LMN's location is dynamically updated by the network in order to maintain the routing entries consistent with the terminal's movements inside the domain. For this, the AR that detected the LMN in the previous phase is responsible for starting the registration operation and for informing the appropriate eTIMIP agents about the LMN's new location. In the initial registration process (the power-up) this update is sent to the tree agents, from the AR up to the GW. In handover operations, this update is limited to the agents in the vicinity of the terminal, which are located in the local sub-tree that connects the new and the old ARs. These update messages have specific reliability mechanisms that guarantee this process, preventing the occurrence of control packet losses or race conditions.

4.2.3.1 Unreliable distributed update procedure

General description

For both power-up and handover operations, the AR that detected the LMN starts the registration process by reconfiguring its own routing table with the information needed to describe that the LMN is located in this AR neighbourhood. Then, this AR initiates a distributed process that reconfigures the routing entries in the network. To start it, it creates an Update message which contains the IP address of the LMN, the LMN detection timestamp, whose value is taken from the detection security phase, and a flag that marks this packet as a basic update. The AR sends this update message to the previous agent that was associated to the LMN or to the parent agent if no routing entry exists.

Each TR that receives an update message for a certain LMN proceeds as follows. First, the TR updates the routing entry which describes that the LMN is accessible through the origi-

nating TR, and stores the associated timestamp information. Then the current TR sends an update message to the previous agent that was associated to the LMN, or to the parent agent if no routing entry exists. This process stops when it was the TR itself that was the previous responsible for the LMN (e.g. it contained a routing entry which states that the LMN was situated in this location), or if the update message came from the node which was already responsible for the LMN (e.g. it contained a routing entry which states that the LMN was already situated in the originating node).

Application example: Power-up operation

When the LMN arrives at an eTIMIP domain for the first time, a Power-up operation, depicted in Figure 17, is issued in the network. After detection, the AR reconfigures its own routing table with the information needed to describe that the LMN is situated in its vicinity. Then, the AR generates an eTIMIP update message to the parent agent up the tree (step 1). The TR1 agent receives this update message, reconfigures its routing table with the information that the LMN is located at AR1, and sends the message up the tree to the GW (step 2). The GW agent associates the LMN to TR1 in its routing table and ends the power-up process.

Figure 17 presents the final state of the routing tables of the agents, with entries for the LMN's next hop, and a default fallback route up the tree, represented by a dash ("-"); finally, AR1 has a LMN entry that points at its own wireless interface, represented by "ITF".

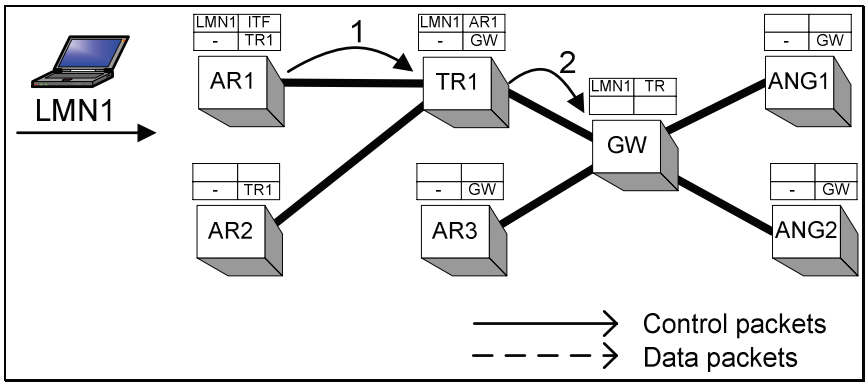


Figure 17: Basic eTIMIP routing registration: Power-Up

Application example: Handover operation

The outlined method for creating the initial routing path is the same that is used for reconfiguring it, as the LMN roams inside the domain (Figure 18). In this operation, eTIMIP's signalling is again generated by the new AR (step 1), but is directed to the old AR through the local sub-tree that connects them. As no direct information about the LMN's old AR is available at the network-side, the eTIMIP agents infer the previous location using the previous routing paths that have just become outdated. Signalling is then sent using these paths (step 2), in a process that also updates the routing entries with the new LMN's next-hop information. The process ends at the old AR, which removes the existing entry, as the LMN is now reachable via its parent agent (e.g. TR1). This last operation is illustrated by the dashed routing entry in Figure 18.

Outside this local subtree, the remaining eTIMIP agents are unaware of the local movement and their routings entries remain valid, namely the one present at the GW.

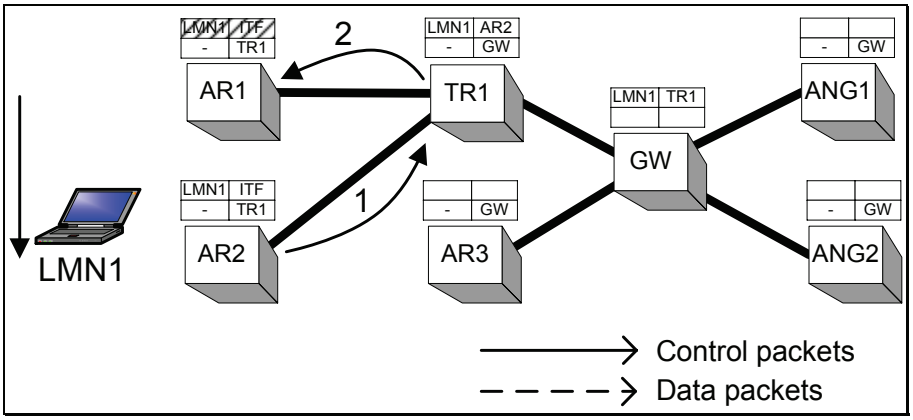


Figure 18: Basic eTIMIP routing registration: Handover reconfiguration

Formal specification

The unreliable distributed update procedures are formally described in the state machine of Figure 19, which all TRs execute in the registration phases. All used functions are described in Appendix B, being the used variables described in Appendix C. Of these, the most important are NEXT, PREV and STOP_SEND variables:

The NEXT variable holds the IP address of the agent that has sent this update message for this LMN to the current node, being updated in the routing table as the new next-hop agent for the LMN. In the example of Figure 18, when the update message reaches the TR1 node, the NEXT variable will point to the AR2 node.

The PREV variable holds the IP address of the previous agent where the LMN was located (by consulting the routing table before updating it with the value of the NEXT variable), and is the agent where the update message will be sent to. Again, in the example of Figure 18, when the update message reaches the TR1 node, the PREV variable will point to the AR1 node.

The STOP_SEND boolean variable just holds the result of the stop condition evaluation that signals the end of the handover process, having the value of “0” in node TR1 in the conditions of Figure 18.

REGISTRATION:
Basic

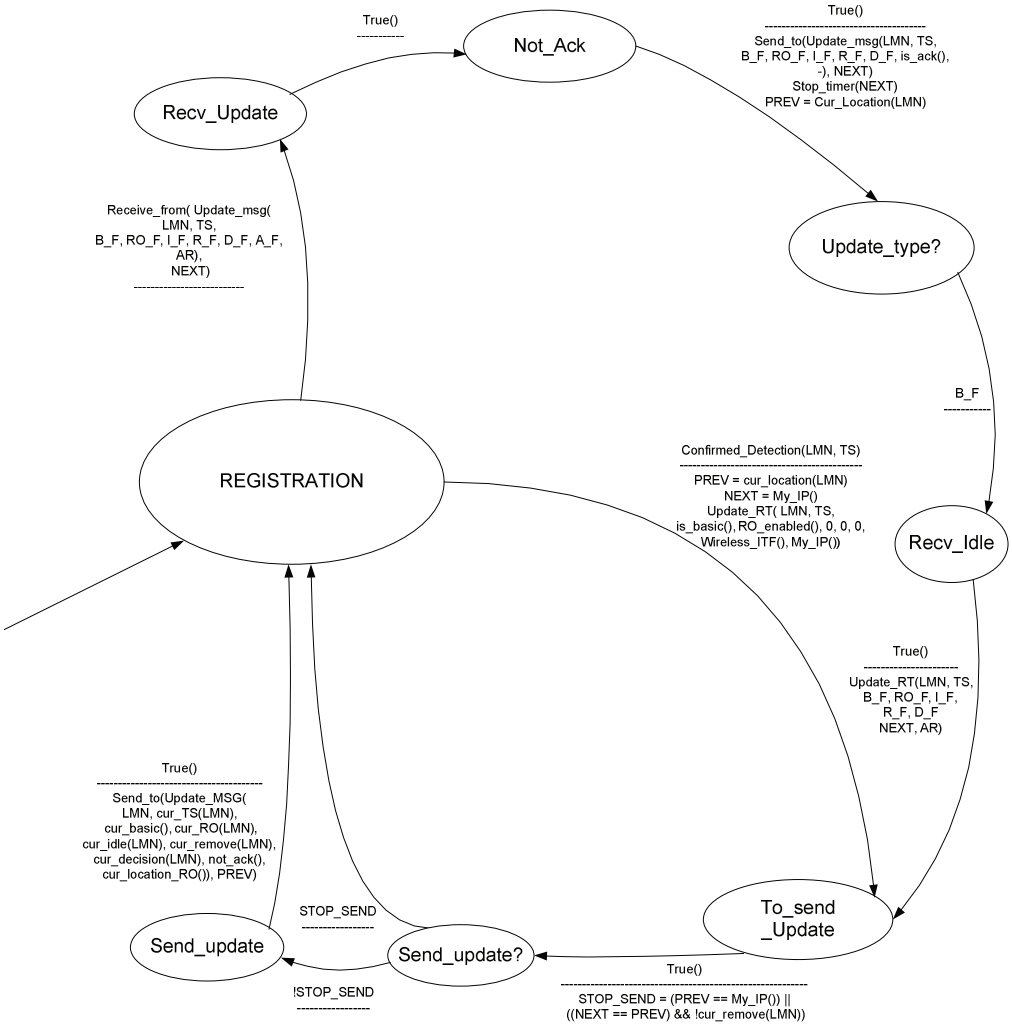


Figure 19: Unreliable distributed update formal specification

4.2.3.2 Reliable distributed update procedure

General description

The previously described distributed update procedure is unreliable, leading to possibility of loosing control packets, which may cause inconsistent entries to be created that do not reflect the LMN's current location. Due to this, reliability procedures are used for guaranteeing this process, using acknowledgement packets for control packet loss detection and timestamps for temporal sorting maintenance [116].

When a TR sends an update packet, it additionally starts a timer to prevent the possible loss of this control packet.

Each TR that receives an update message for a certain LMN, it firstly compares the timestamp contained in the packet with the current timestamp associated with this LMN. If the packet's timestamp is older, then this update message is not accepted, and the current TR replies with a new update message to the originating TR with the most updated data of the LMN. If the packet's timestamp is equal or newer, then the update message is accepted, and the current TR sends an update acknowledgement message to the originating TR and cancels

any pending timer associated with it. Then the TR proceeds as before, by updating the routing entry with the latest next hop and timestamp information.

On the other hand, if no reply is received during the timeout value by the original TR, then the update message is retransmitted, up to a configuration maximum number of tries.

Application example: Signalling reliability – Loss of update or Ack packet

Figure 20 and Figure 21 show a temporal diagram example which illustrates a reliable handover operation using guaranteed signalling. In the first part, the update message sent by AR2 to TR1 is lost, being retransmitted by AR2 when the associated timer expires. In the second part, the loss of an acknowledgement packet from AR1 to TR1 leads to the generation of a second update / acknowledgement pair after the timeout at TR1.

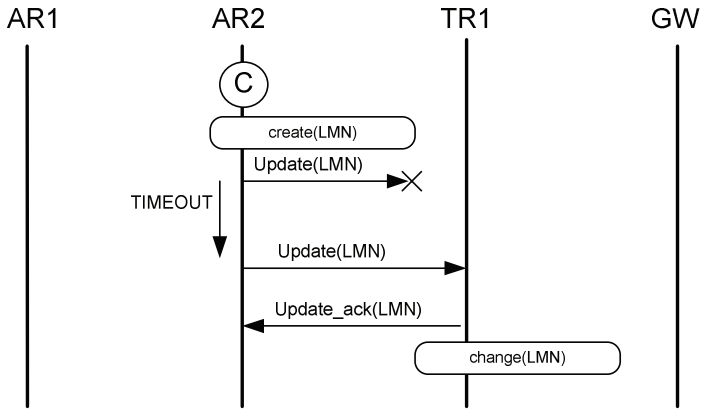


Figure 20: Guaranteed signalling: loss of update message

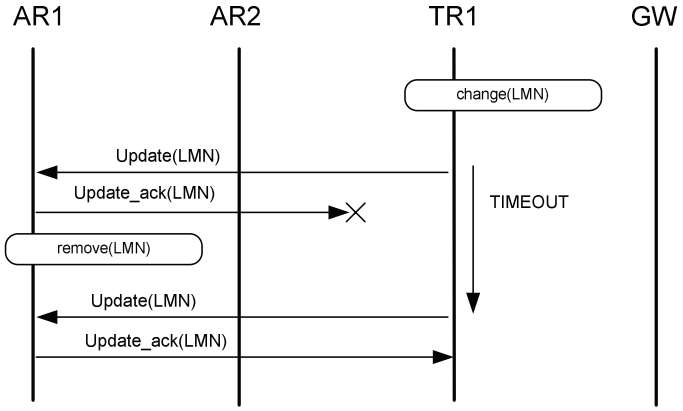


Figure 21: Guaranteed signalling: Loss of Acknowledgement message

Application example: Signalling reliability – Race Condition in Signalling packets

Figure 22 shows another temporal diagram example which illustrates a reliable power-up operation using the temporal sorting feature. Firstly, AR1 detects a LMN, but its update message is lost; later, the same LMN is detected by AR2, which successfully performs a power-up operation. When AR1 retransmits its original update packet, the TR1 agent will be able to verify that the update message is related to the previous location of the LMN, using the timestamps, and will reply to it with another update message instead of an acknowledgement. When AR1 receives this update, it uses the timestamp to cancel its pending power-up process and acknowledges the update packet, ending the process.

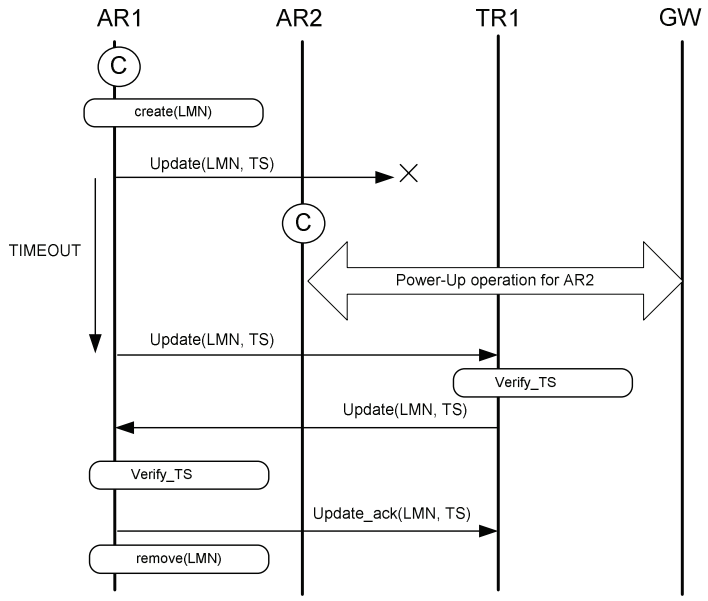


Figure 22: Temporal sorting of signalling

Formal specification

The reliable distributed update procedures are formally described in the state machine of Figure 23, which all TRs execute in the registration phase, including the AR specific functions. This formal definition expands the corresponding state machine of Figure 19 with the reliability-related mechanisms, being these extensions identified by a bold typeface. All used functions are described in Appendix B, being the used variables described in Appendix C. Of these, the most important added ones are the TS and the “retries” variables:

The TS variable holds the timestamp contained in each message, being compared with the lastest received timestamp using the function Accept_TS(). The retries variable holds the number of retransmissions already performed of the last signalling message, being used to control the retransmission timers.

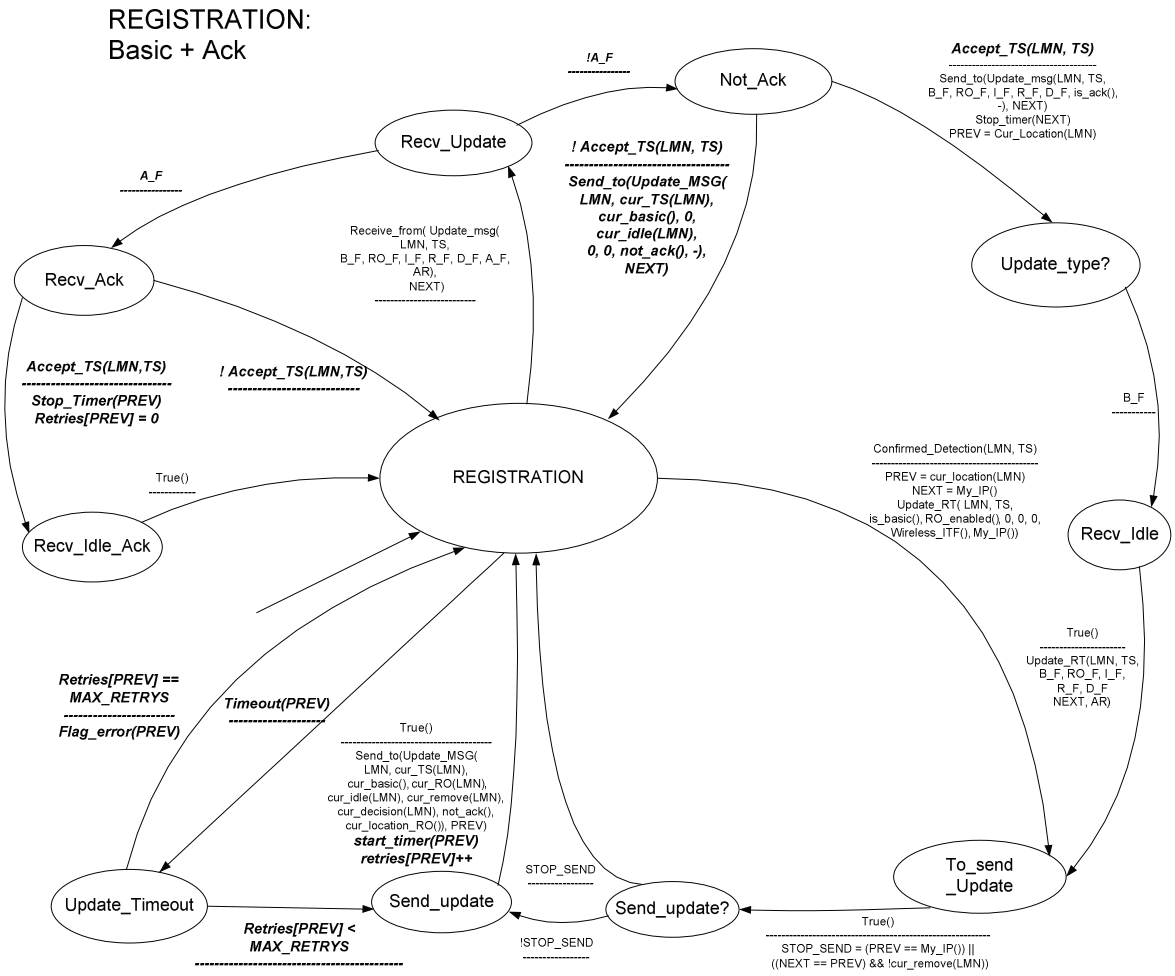


Figure 23: Reliable distributed update formal specification

4.2.3.3 Registration Security

General description

Concerning the security of the eTIMIP registration phase, all signalling messages are signed and time-stamped by the agents with the secret network key, with any known secure authentication method. This prevents malicious mobile hosts from changing location information related to other mobile hosts using a spoofed source address.

For this, each packet is concatenated with a time-stamp and a random number, and is signed using the secret network key. Each agent that receives an eTIMIP packet can then verify its origin, by checking if the digital signature was generated using the secret network key, and discards the control packets that fail such test.

4.2.4 Execution phase

The Execution phase considers the forwarding of data packets between the eTIMIP agents inside the overlay network, using the routing entries maintained by the registration phase. Other operations that occur in the execution phase are the specific transparency procedures required to force the data packets to enter and exit the overlay network, and the soft-state basic routing entries maintenance, which is optimized using the LMN's own data packets to perform their refresh.

4.2.4.1 Data forwarding

General description

When an eTIMIP agent receives a data packet, its destination is checked against the eTIMIP agent's routing table. If a match is found, the packet is forwarded to the next eTIMIP agent defined in that entry, resulting in a downlink routing procedure; in the opposite case, the packet is forwarded to the default upstream agent towards the GW, resulting in an uplink routing procedure.

In these operations, the control and data packets are swapped agent-by-agent in the overlay network by strictly following the logical paths of the agent tree. The logical connections are always used and are mapped to the physical paths as decided by the fixed routing within each pair of adjacent agents. In the case of data packets, they are also encapsulated if legacy routers are present in the chosen path.

Application example: Intra-domain routing

Figure 24 illustrates the forwarding of intra-domain data packets. At first, a correspondent node (node CN) delivers a data packet to AR1 (step 1). AR1 consults its routing table for LMN1 and, as no match is found, forwards the data packet to its parent node (TR1) using encapsulation if legacy routers are present in the path to TR1 chosen by the fixed routing (step 2). TR1 performs similar operations and forwards the data packet to AR2, as a routing entry for LMN1 exists (step 3). Finally, AR2 delivers the data packet directly to the mobile node.

Using these rules, the packets exchanged by LMNs located in the same eTIMIP domain are forwarded using the local subtree only, reaching the domain's GW only in the worst case.

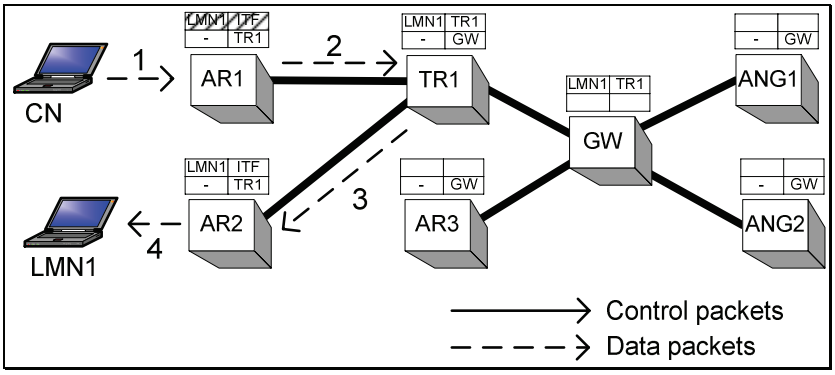


Figure 24: Basic eTIMIP routing execution: Intra-Domain

Application example: Inter-domain routing

Figure 25 illustrates the complementary case, concerning the forwarding of inter-domain data packets. The data packets sent by a CN are firstly forwarded to the eTIMIP domain by the fixed routing (step 1), being received by an ANG located at the domain's edge (ANG1). This agent will forward the data packet to the GW by default, using encapsulation when necessary (step 2). From this point on, the data packet is forwarded by the routing entries of each agent as previously described (downlink routing, steps 3 to 5).

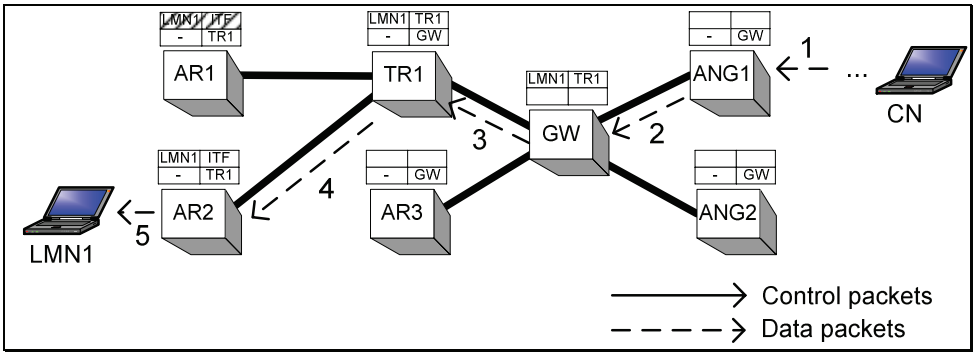


Figure 25: Basic eTIMIP routing execution: Inter-Domain routing

Formal Specification

The described forwarding methods are formally described in the state machine of Figure 26. This figure only reflects the mobile data forwarding functions, as all TRs also have fixed forwarding capabilities. All used functions are described in Appendix B, being the used variables described in Appendix C.

Of these, the most important is the DST variable, that holds the IP address of the destination of the data packet, contained in its IP header; this variable determines where the packet will be forwarded to, by consulting the routing table using cur_location(DST); all other variables do not condition this state machine in any way, being present for extending it in the latter sections only.

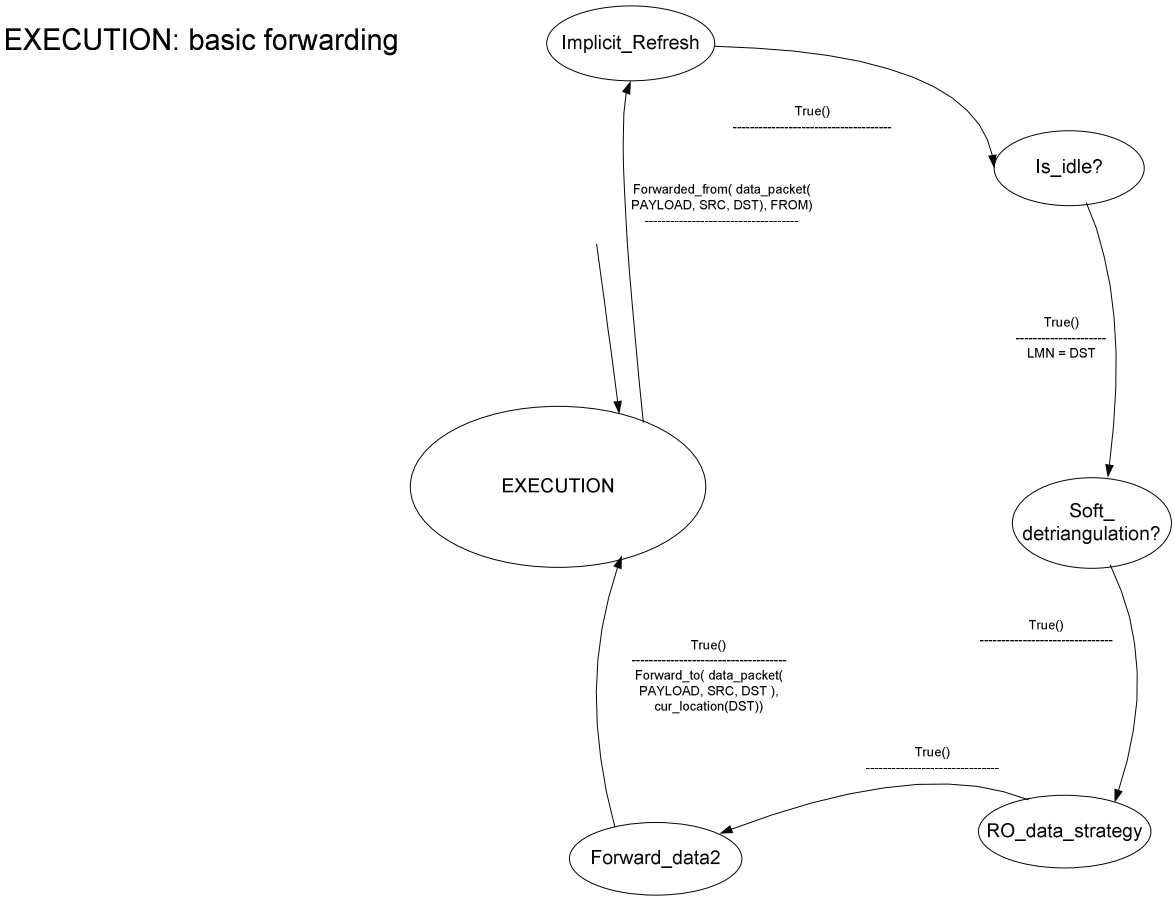


Figure 26: Basic data forwarding formal specification

4.2.4.2 Overlay Network Usage Enforcement

General description

The provisioning of a transparent mobility solution, for both terminal and network sides, is essentially related to the problem of how to ensure that the overlay network is always used to route the mobile data packets, without modifying the LMN's protocol stack, the existing network topology, and the existing legacy routers. Generically, such transparency aspects are achieved by inserting eTIMIP agents at the network's boundaries to intercept all incoming traffic, and by encapsulating data packets that need to pass through legacy routers between each pair of eTIMIP agents.

Deploying ARs at the LMN's network boundary is always possible without modifying the pre-existing physical network, as the eTIMIP ARs can either be incorporated in pre-existing upgradeable access routers or added to the same subnets of the LMNs just like any regular IP router. Then, by being adjacent to the LMNs, the ARs can perform the L2 discovery mechanisms described below to transparently configure themselves as the LMN's default router, in order to capture the LMNs' data packets.

In contrast, deploying ANGs at the core network's boundary may only be possible in particular situations without modifying the existing infra-structure. For packets arriving through LRs, a **mobile subnet**, which is a regular Classless Inter Domain Routing (CIDR) subnet [112], created and managed by the fixed routing, and associated with the GW, the ARs and the LMNs, is used to transparently redirect the data packets to the GW by default. This is made possible as the mobile subnet is used to provide IP addresses to the LMNs. Thus, from the fixed routing point of view, all LMNs are virtually located in the GW, which achieves transparency without modification of the LRs. Recently, a similar solution for this problem has started being discussed in the netLMM Working Group [105].

Concerning the LMNs' reception of data, it is necessary that these maintain a unique IP address in all locations of the domain, a requirement that is also shared by netLMM [104]. For this, the LMNs are pre-configured with IP addresses belonging to the domain's mobile subnet. This can be achieved through any suitable process, namely stateless address allocation where all ARs announce the mobile subnet, stateful IP addresses allocation using centralized IP addresses pools at the GW and DHCP relay agents at each AR [120], PPP negotiation [121] or manual configuration. As the LMNs have a constant IP address and are adjacent to the ARs, those agents can deliver the data packets directly to them, without any kind of encapsulation. Thus, eTIMIP shares the same netLMM advantage of avoiding the data packets tunnelling overhead over the air interface.

Concerning the LMNs' delivery of data, it is necessary to force the LMNs to transparently deliver all their data to the current AR, using only the facilities present in the regular IP stack. For this, in the above-mentioned configuration processes, the LMN is configured with a special network mask and a special default router. The former, named "closed netmask", contains all bits with the value "one", and forces all data to be routed through the LMN's default router [121]; the latter is a special unique IP address, designated by X_IP in the examples, which all ARs of the domain respond to with their own MAC address, using proxy and gratuitous neighbour discovery operations, and which forces the receipt of all LMN data by the AR. Such neighbour discovery techniques are the same as the ones used by MIP for capturing the packets at the HA agents [1]. Alternatively, the technique proposed by the

netLMM group may also be used to achieve the same result: if the auto-configuration process supports an on-link flag [115][15], such may be used to force all packets to be delivered to the AR transparently, by setting this flag to "0" [105].

It should be noted that in the above-mentioned stateless address auto-configuration, the standard duplicate address detection process can fail to prevent two terminals from allocating the same IP address, by being located in different ARs [173]. This problem can be solved by performing the address allocation at the first detection of the LMNs, and doing an additional check in the previously described eTIMIP initialization procedure; in this, the GW will maintain a list of the correspondence pairs MNID→IP address, and will be able to check if two different MNIDs are using the same IP address. In this case, the GW returns an error to the AR, and the AR performs a proxy neighbour advertisement [2] on behalf of the LMN that allocated the address first.

Concerning the LRs support, if legacy routers are present in the path chosen by the fixed routing, the packets will be required to be forwarded with any suitable form of encapsulation, which can be based on IP tunnelling [7] or any other suitable technique [134]. Such is required because if legacy routers receive unencapsulated data packets destined for LMNs, these routers will forward these packets towards the GW by default, which is associated with the mobile subnet. As by definition the GW is a TR node, this also serves as a late fallback entry point of the data packets in the overlay network.

Application example: LMN data reception

Figure 27 illustrates the reception of data packets by the LMN. As the LMN maintains a constant IP address inside the domain, the ARs can simply deliver the data packets to them, without any kind of encapsulation, which has the benefit of not requiring overhead in the limited wireless network resources.

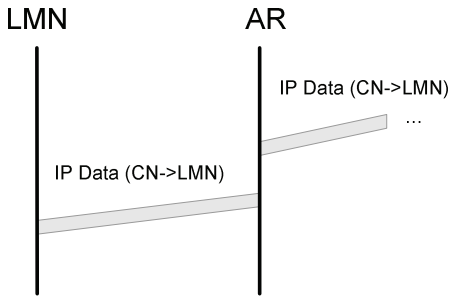


Figure 27: LMN data reception

Application example: LMN data transmission

Part a) of Figure 28 illustrates the transmission of data packets by the LMN. At first, when the LMN is discovered by an AR (trigger "D"), the AR performs a gratuitous neighbour discovery operation for the LMN's default router (IP address X_IP) using its own MAC address. Then, when the LMN has packets to transmit, these will be routed via the default configured router, regardless of the destination, because of the closed network mask. As this

[15] Quoting from the IPv6 neighbour discovery specification [115], "1 bit on-link flag: When set, indicates that this prefix can be used for on-link determination. When not set the advertisement makes no statement about on-link or off-link properties of the prefix. For instance, the prefix might be used for address configuration with some of the addresses belonging to the prefix being on-link and others being off-link".

default router is associated with the AR's MAC address, the LMN will transparently send all data packets to the current AR, the rest of the forwarding process being performed as described previously.

Part b) of Figure 28 illustrates the same situation, when the LMN lacks the necessary L2 neighbour information about its own default router. Here, the only difference is that the LMN will first issue a regular neighbour discovery request to its default router (IP address X_IP). Again, the answer is given by the AR with its own MAC address, in the form of a proxy neighbour discovery operation.

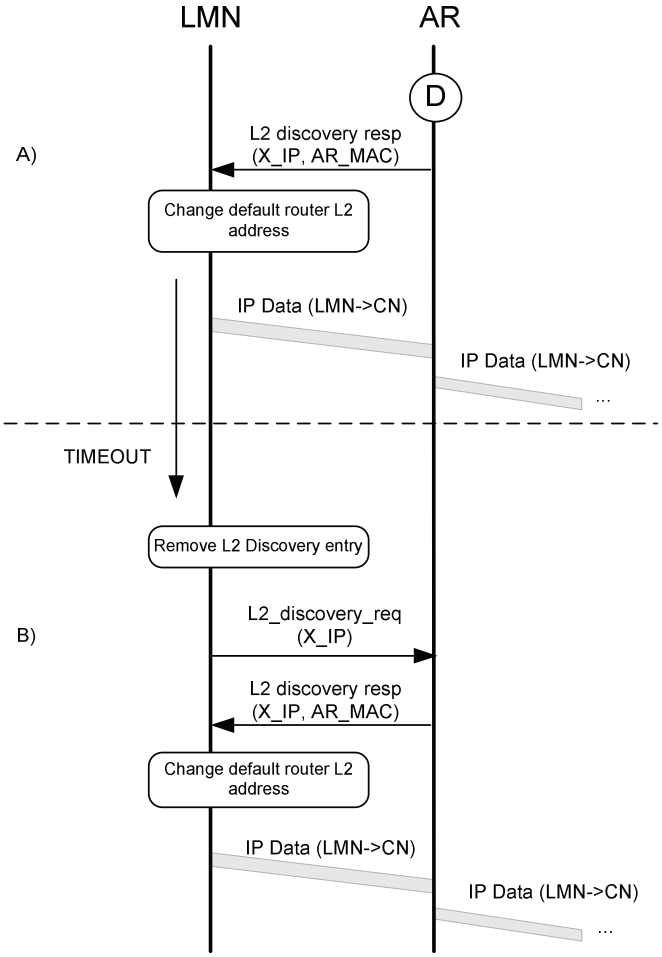


Figure 28: LMN data emission a) gratuitous neighbour discovery b) proxy neighbour discovery

Application example: Legacy Routers support

Figure 29 presents an example of the interactions between the overlay network and the domain's pre-existing legacy routers, which are supported by the mobile subnet and the encapsulation mechanisms. For this example, the data packets are represented with their IP addresses, in order to better show the encapsulation procedures and the LR interactions.

First, when a data packet from a correspondent node CN to a terminal LMN is received by a legacy router LR1, this router consults its local fixed routing table, and forwards the packet by default towards the router that contains the GW agent. This happens because the packet's destination (e.g. the LMN address) belongs to the mobile subnet, which is associated with the GW by the fixed routing (section "a" of Figure 29).

Then, the GW consults its eTIMIP mobile routing table, and finds that the next agent to this LMN is TR1. Then, the GW consults its local fixed routing table and verifies that agent TR1 is connected via legacy router LR2. Considering this, the GW encapsulates the data packet and sends it to legacy router LR2. When this router (LR2) receives the encapsulated packet, it forwards the packet to TR1 transparently, as the packet is directed to node TR1. Then, agent TR1 de-encapsulates the data packet and continues the forwarding process (section “b” of Figure 29).

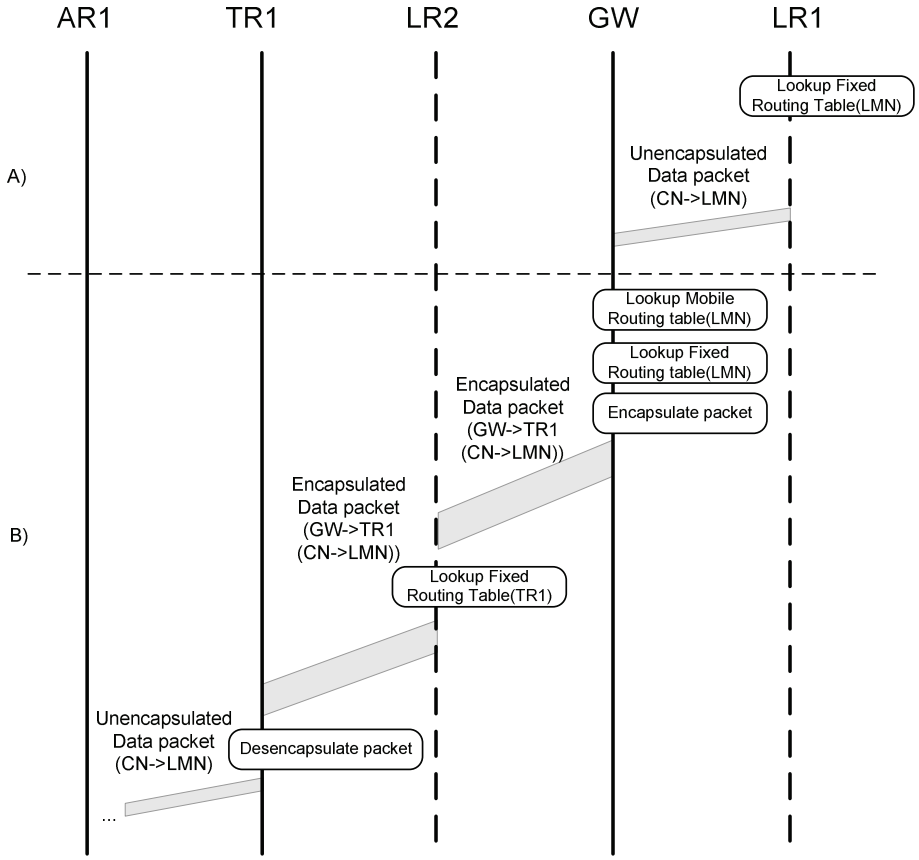


Figure 29: Legacy routers support using: a) Mobile Subnet; b) Data Encapsulation

Application example: StateFul address configuration using DHCP

Figure 30 presents an example of how stateful address configuration can be performed using the DHCP protocol [155].

For this, the GW will feature a DHCP server that manages a pool of centralized addresses of the mobile subnet, and each AR will feature a DHCP relay server to redirect the address allocation procedures initiated by the LMN to the central server. This address allocation procedure is orthogonal to the eTIMIP mobility processes, and follows the standard DHCP processes depicted in Figure 30. After the LMN receives an address from the mobile subnet and the special default router / netmask described previously, the first data packet will trigger the regular eTIMIP detection processes, resulting in the power-up operation described previously.

After the LMN moves to another AR of the same domain, the same DHCP procedure will result in the refreshment of the previously allocated IP address, and the first data packet will trigger the regular eTIMIP handover process (Figure 31).

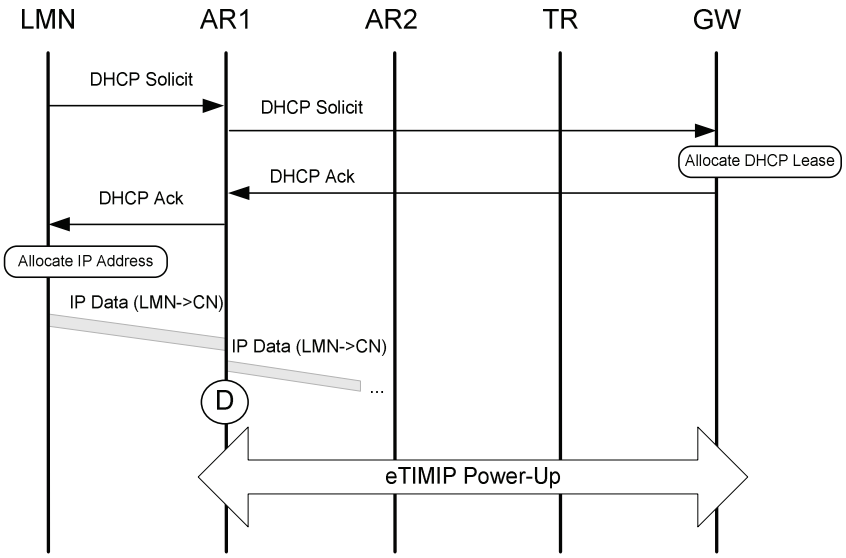


Figure 30: Power-up using DHCP Stateful address configuration

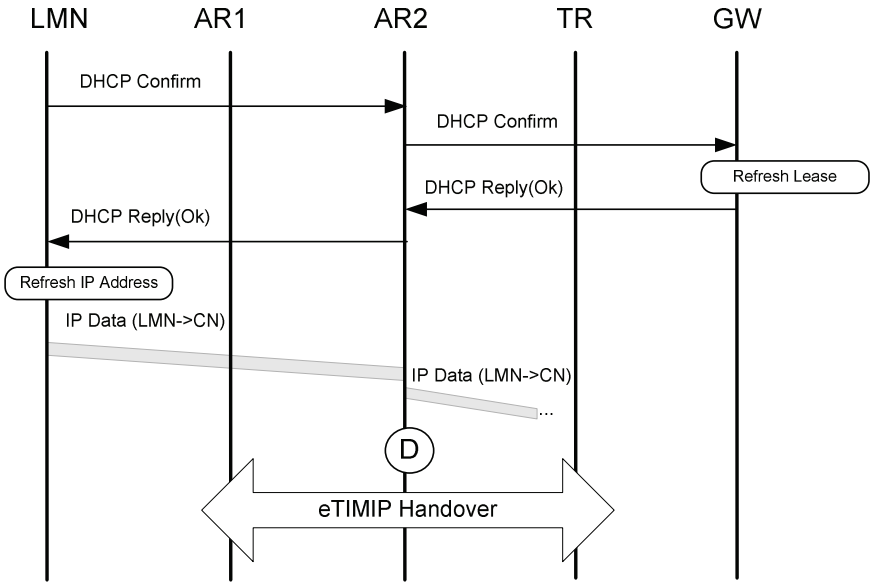


Figure 31: Handover using DHCP Stateful address configuration

Application example: Stateless Address Auto-configuration

Figure 32 presents the corresponding example of how Stateless Address Auto-Configuration (SLAAC) [114] can be performed in eTIMIP domains. For this, each AR will periodically broadcast that it manages the mobile subnet, using a router advert message that identifies the default router as X_IP, and does not have its “on-link” bit set (in order to force the terminal to send all its traffic to its default router [115]).

When the terminal receives the mobile subnet advert, it will allocate an address from the mobile subnet, and will initiate the duplicate address detection procedure, by verifying in the local link if the address chosen is not already in use, via a neighbour solicitation to its own address. After this, the previously described security and registration eTIMIP procedures are triggered to perform the power-up operation for this terminal.

After the LMN moves to another AR of the same domain, the same SLAAC procedures will trigger any address reconfiguration, as the terminal will receive a similar route adver-

tisement to the mobile subnet in the new AR, which triggers the handover process (Figure 33).

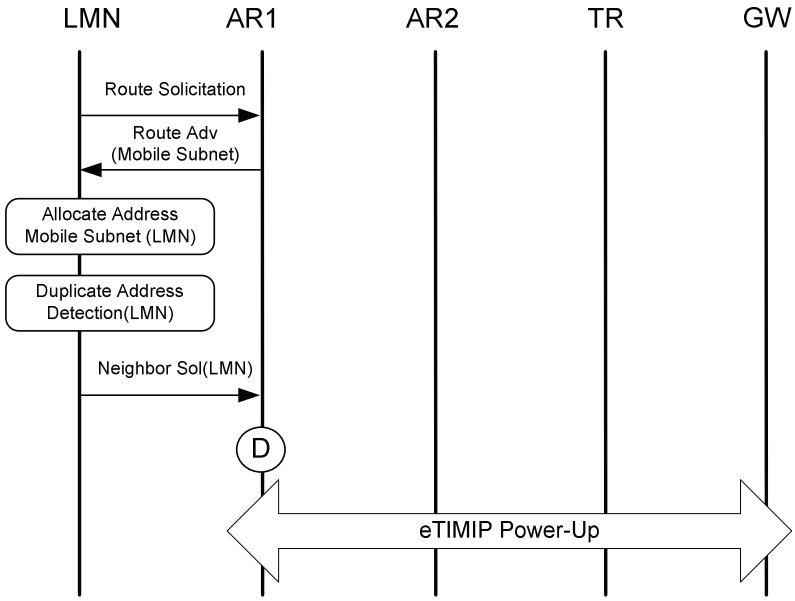


Figure 32: Power-up using Stateless address auto-configuration

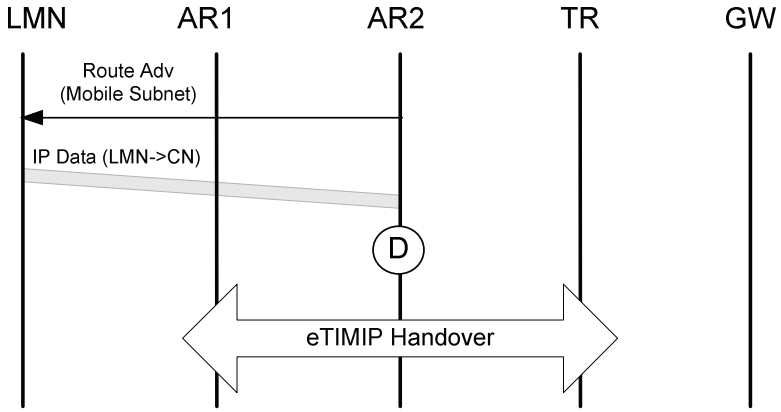


Figure 33: Handover using Stateless address auto-configuration

Figure 34 covers the case where two different terminals may allocate the same IP address using SLAAC. First (dashed box), a terminal with ID MNID2 allocates an IP address LMN1 (not shown in the figure). During the security initialization procedure, the GW associates the MNID1 to the chosen IP address LMN1. Later, a different terminal MNID2, located in a different AR, allocates the same IP address. During the security initialization procedure, the GW is able to verify that such IP address was already allocated by a previous terminal. In this case, the GW returns an error message to the AR, and the AR performs a proxy neighbour advertisement [2] on behalf of the LMN that allocated the address first. This will force the second terminal MNID2 to restart SLAAC and choose a different IP address, restarting the security initialization again.

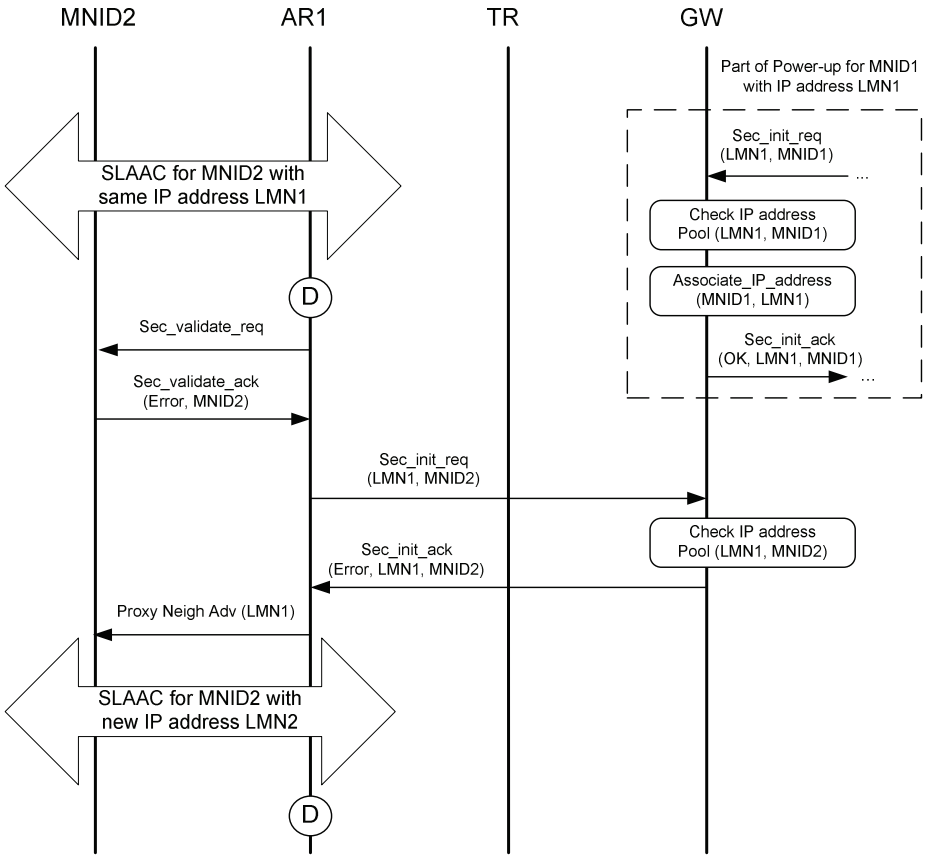


Figure 34: Centralized duplicate address detection during eTIMIP security initialization

Formal Specification

The described neighbour discovery LMN support methods are formally described in the state machine of Figure 35. All used functions are described in Appendix B.

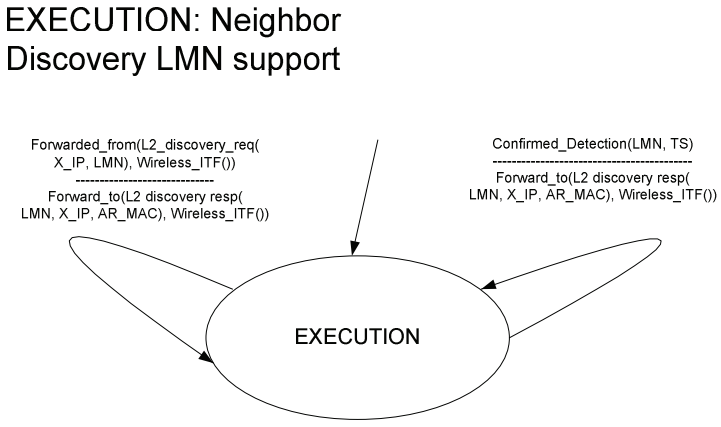


Figure 35: Neighbour discovery LMN support

4.2.4.3 State Maintenance

General description

The basic routing entries managed by eTIMIP have a soft-state nature, having to be periodically refreshed to avoid deletion. In this process, a refresh cycle is defined as the time period between two subsequent refresh operations. This refreshment is either performed implicitly, using the data packets sent by the LMNs as they pass through its AR agent, or

explicitly, using explicit refresh signalling. In this later case, the explicit refreshes are subject to a back off procedure that dynamically changes the length of the refresh cycle between preconfigured bounds. For the LMNs that fail to refresh their state on the network explicitly, after several failed explicit refreshes, their associated routing entries are removed from the network by their AR, using a “power-down” operation. This procedure uses an update message with a remove flag, which is sent up to the GW to remove the routing entries.

Each LMN routing entry at each TR has a current timeout value associated with it, named TO, which contains the length of the current refresh cycle, and a current lifetime value in seconds, named LT, used to detect the end of the current refresh cycle. The timeout value is preconfigured to a certain initial value, TO_INIT, which can be different for each level of the agent tree, and is always kept within predefined bounds (TO_MIN, TO_MAX). This timeout value is then used to generate the corresponding lifetime value, using an exponential function that has back-off capabilities.

The LMN data packets that are sent by the LMN and are received by the wireless interface, are used by the AR to refresh the corresponding routing entry, by setting the refresh cycle length to the initial value and generating a new lifetime value. On the other hand, if no data packets sent by the LMN are received at the AR agent during the current refresh cycle, then the agent sends an explicitly refresh request packet to the LMN, and reduces the duration of the following refresh cycle.

At this point, when the corresponding refresh reply is received by the TR, a new refresh cycle is started with a timeout value longer than the current one. If the reply doesn’t arrive, because the terminal didn’t respond or because it was lost, then the inverse happens, using a new refresh cycle with a timeout value shorter than the current one. During these explicit refresh operations, the arrival of a data packet sent by the LMN is sufficient to return the routing entry to the initial active state; complementary, the forwarding of a data packet to the LMN always limits the timeout to, at most, the initial value, which ensures a fast departure detection of the active LMNs that are receiving data traffic.

Finally, the lower bound of the timeout value has special implications in the explicit refresh cycle. When a timeout occurs and the timeout value is already at its minimum, which is a signal that multiple explicit refresh cycles were tried without success, then the routing entries are removed in a guaranteed fashion from the TRs, using a power-down procedure, as it is assumed that the terminal has left the domain. For this, the AR sends an update message to this LMN with a remove flag, which is propagated and acknowledged up to the GW.

Application example: Implicit refresh for active LMNs

Figure 36 presents an example of the optimized refresh for active LMNs using their own data packets. Here, when the data packets are received by the AR, the LMN entry is refreshed, and the packet is forwarded as normal. Using this facility, the state maintenance of active terminals does not incur additional overhead for active terminals.

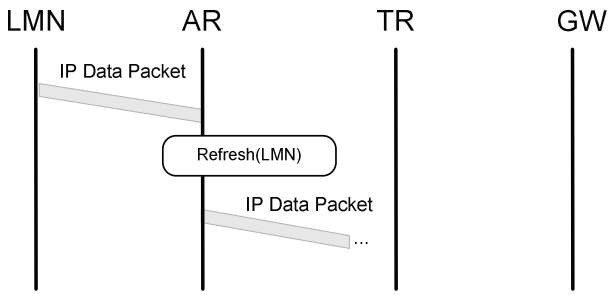


Figure 36: Implicit refresh for active LMNs

Application example: Explicit refresh for inactive LMNs

In the case that some AR is not refreshed, either because the terminal is idle or has left the domain, will get its timer expired (trigger "R") and will start to send ICMP EchoRequest messages to the terminal. This forces the LMN to reply with an EchoReply message destined to the AR. If this Reply message is received within the current refresh cycle, then the timeout value is doubled, up to a configuration predefined maximum (Figure 37); on the other hand, if no reply is received, then the value is divided by two up to a configuration predefined minimum. Using this facility, the state maintenance of inactive terminals is decreased, as long as such terminals are not receiving data packets.

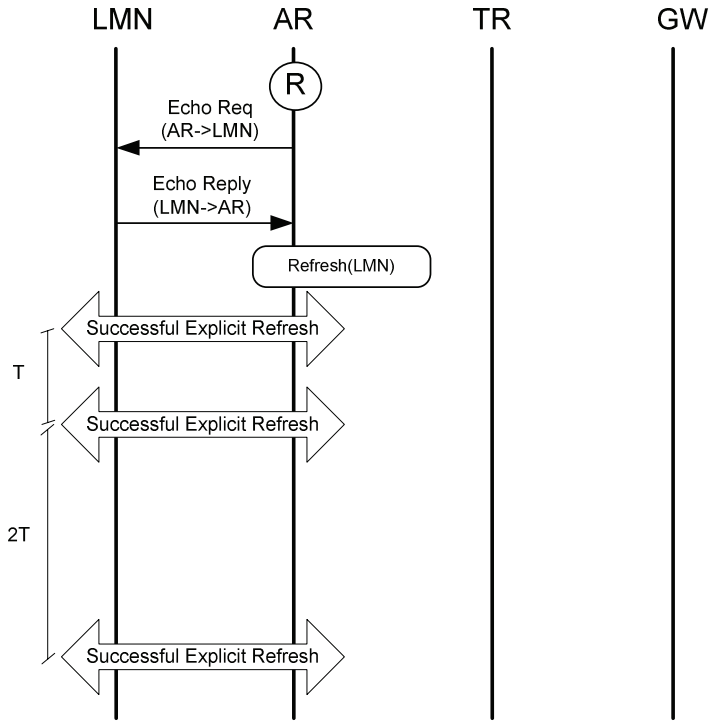


Figure 37: Explicit refresh for inactive LMNs

The relation of the possible events, the length of the refresh cycle, and the special actions taken at both the minimum and maximum values are described in graph of Figure 38, and Table 5. The graph shows the effect of each timeout and each refresh reply received on the TO value, while the table shows an configuration example of lifetimes between explicit refreshes, approximated to seconds and minutes.

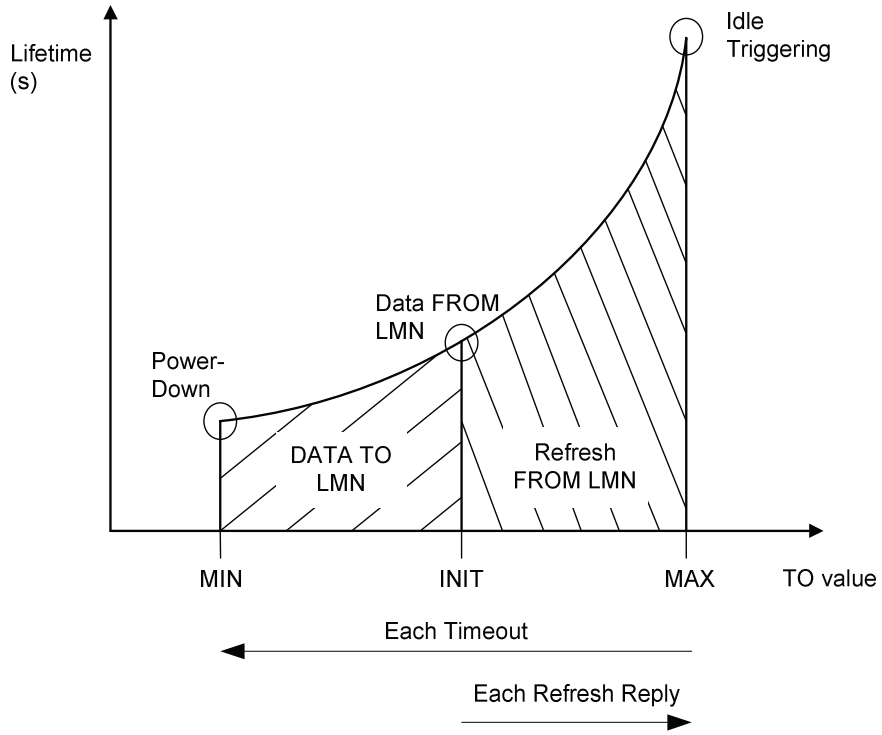


Figure 38: Evolution of the length value of refresh cycles

Constants	MIN						INIT				MAX
TO (Timeout)	0	1	2	3	4	5	6	7	8	9	10
LF (Lifetime)	1	2	4	8	16	32	64	128	256	512	1024
Time between refreshes	1s	2s	4s	8s	16s	32s	~1m	~2m	~4m	~8.5m	~17m
Refresh Involved levels	1	1	1	1	1	1	1	1	1	1	1

Table 5: Example configuration of state refresh values

Application example: Explicit refresh for listener LMNs

If the LMN is a listener-type terminal, then the network will resort to using explicit state maintenance with short refresh cycles, as each packet forwarded to the LMN by the AR automatically limits its refresh cycle to, at most, TO_INIT (Figure 39). Such behaviour is desirable for providing faster departure detection, as this LMN is involved in data communication with other peers, and thus the network is required to provide a faster departure detection service.

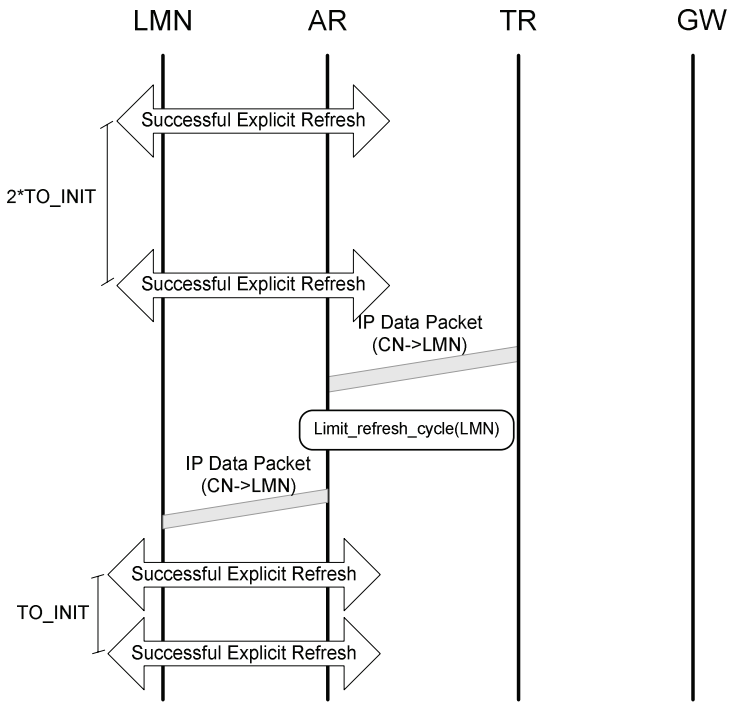


Figure 39: Explicit refresh for listener LMNs for fast departure detection

Application example: State removal for absent LMNs (Power-down)

Figure 40 illustrates how soft-state entries are removed from LMNs that have left the domain. After multiple unsuccessful explicit refresh cycles (trigger “X”), the AR will start the removal of the LMN’s entries in the involved TRs, using a power-down procedure, that has the same reliability guarantees as the regular update messages.

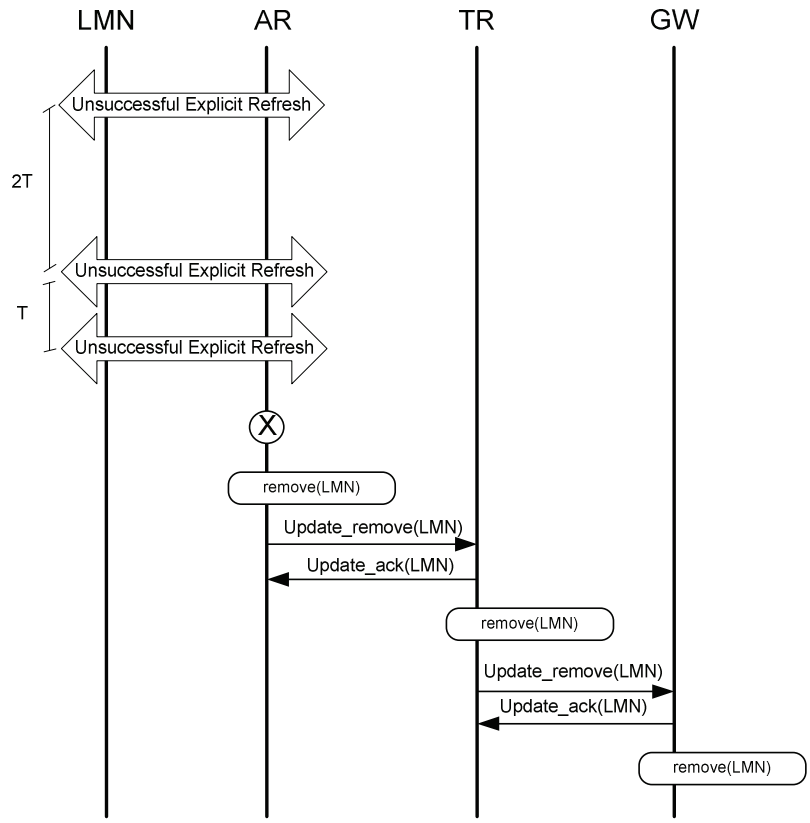


Figure 40: State removal for absent LMNs

Formal specification

The state maintenance methods are formally described in the state machine of Figure 41, which expands the corresponding state machine of Figure 26 and adds state maintenance-related mechanisms to it, being these extensions identified by a bold typeface. All used functions are described in Appendix B, being the used variables described in Appendix C. Of these, the most important added ones are the TO and the LT variables, which have the same behaviour previously explained.

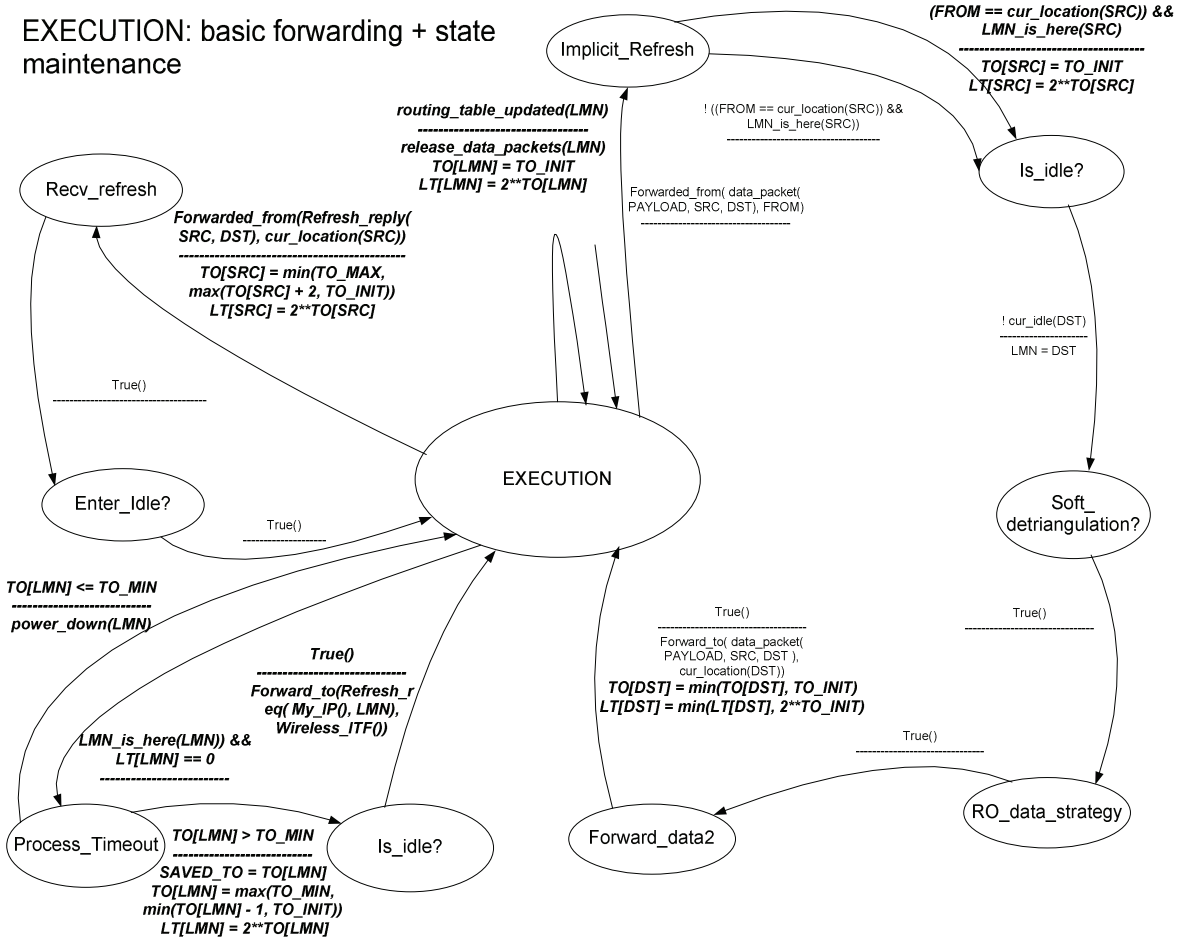


Figure 41: State Maintenance formal specification

4.2.4.4 Execution security

General description

The definition of security mechanisms that protect the user data packets falls outside the scope of the eTIMIP protocol. For this topic, any further generic encryption can be performed in addition to the control packet authentication built into eTIMIP. In particular, the shared PID maintained in the detection phase can optionally be used as a session key for any security mechanism that uses these mechanisms.

4.3 eTIMIP versions

This section outlines the specific adaptations of the generic eTIMIP architecture for IPv4 and IPv6 protocols. In both cases, there is a definition of the packet formats used in both control and data packets, and other details that are specific to the particular IP version in use.

4.3.1 eTIMIPv4 version

The eTIMIPv4 protocol is the version of the generic architecture previously described for IPv4. This version updates the previous TIMIP model with the new control messages, which are now delivered on top of the UDP protocol (previously, TIMIP messages were encapsulated in ICMP).

4.3.1.1 Control Packets

The update control packets used by the eTIMIPv4 agents are briefly outlined in Figure 42, which focuses on the key fields only, with a more precise definition being present in Appendix E. These packets are passed from agent to agent without encapsulation, containing the eTIMIP messages encapsulated in UDP to a unique, well-known port. The eTIMIP update message extends the unique codes already defined in reference [27], to identify the messages contained in the control packets. The update message contains the LMN address field that identifies the IPv4 address of the LMN, the timestamp of the LMN's arrival time at the AR coded in NTP format [116], the flags that identify the type of update (Basic / Remove).

IPv4 Header		UDP Header		eTIMIP Header			
IP Src: Agent A	IP Dst: Agent B	Src Port: eTIMIPv4	Dst Port: eTIMIPv4	Type: Update	Flags: (Basic/Remove)	LMN address	NTP timestamp

Figure 42: eTIMIPv4 update control packets

As mentioned in section 4.2.3.3, all the eTIMIPv4 control packets carry mandatory Authentication Extensions compatible with those defined by standard MIPv4, by incorporating digital signatures derived from the shared network key, in order to protect the control messages of the attacks mentioned previously. This prevents malicious mobile hosts from exchanging location information related to other mobile hosts using a spoofed source address.

4.3.1.2 Refresh Packets

eTIMIPv4 uses standard ICMPv4 Echo Request / Echo Reply control packets [96] for performing its refresh operations at the AR.

4.3.1.3 Neighbour Discovery Packets

eTIMIPv4 uses standard IPv4 ARP request / ARP reply packets [122] for performing its proxy and gratuitous neighbour discovery operations [1] which forces the delivery of all LMN data traffic to its current AR as defined in section 4.2.4.2.

4.3.1.4 Data Packets

The LMN data packets are managed by eTIMIP forwarded inside the physical network either directly, when there are direct links between the selected agents, or with full IP-in-IP encapsulation [7], which enables legacy regular routers to pass these packets without modifications.

In the latter case, each agent that receives a data packet from a LMN determines the next hop using the previously described eTIMIP routing and encapsulates the packet with an IPv4 header containing the agents' IP addresses. The resulting packet, illustrated in Figure 43, is then forwarded using regular IP forwarding mechanisms. The next agent removes the encapsulation and repeats the process. This process is repeated until the packet reaches the AR of the terminal, which delivers the original unaltered packet to the correct LMN.

IPv4 Header		IPv4 Header		Higher Headers
IP Src: Agent A	IP Dst: Agent B	IP Src: CN	IP Dst: LMN	…

Figure 43: Encapsulated IPv4 data packets

4.3.2 eTIMIPv6 version

The eTIMIPv6 protocol is the version of the generic architecture previously described for IPv6. This version is able to take advantage of the new IPv6 features, namely the new header treatment and the source routing option, which are both native to all IPv6 hosts and routers.

4.3.2.1 Control packets

The update control packets used by the eTIMIPv6 agents are briefly outlined in Figure 45, which focuses on the key fields only, with a more precise definition being present in Appendix E. These packets are transmitted by the agents without encapsulation, containing the eTIMIP messages in the Mobility Header. This header was proposed by MIPv6 to carry special mobility information usable by mobility protocols in IPv6 [2]. As usual in all MIPv6 extension headers, it features a checksum that protects both the mobility header itself and the standard IPv6 header[16], and a mobility selector field that is used to identify the specific mobility message contained in it, with specific eTIMIP codes being used for this purpose. The specific eTIMIPv6 update fields are the same as in eTIMIPv4, but extended to use IPv6 addresses.

IPv6 Header		Mobility Header			
IP Src: Agent A	IP Dst: Agent B	Type: Update	Flags: (Basic/Remove)	LMN address	NTP timestamp

Figure 44: eTIMIPv6 update control packets

As mentioned in section 4.2.3.3, all eTIMIPv6 control packets carry mandatory Authentication Headers by incorporating digital signatures derived from the shared network key in order to protect the control messages from the attacks previously mentioned. This prevents malicious mobile hosts from exchanging location information related to other mobile hosts using a spoofed source address.

4.3.2.2 Refresh Packets

eTIMIPv6 uses standard ICMPv6 Echo Request / Echo Reply control packets [123] for performing its refresh operations.

4.3.2.3 Neighbour Discovery Packets

eTIMIPv6 uses standard ICMPv6 neighbour discovery packets [115] for performing its proxy and gratuitous neighbour discovery operations [2], which forces the delivery of all LMN data traffic to its current AR; in addition to the requirements described in section 4.2.4.2, the ND messages transmitted by the ARs must have the on-link flag [115] unset, to further force the reception of all LMN packets by the AR.

4.3.2.4 Data packets

Concerning the data packets, these are forwarded inside the physical network either directly, when there are direct links between the selected eTIMIP agents, or efficiently encapsulated using the native source routing facilities of the IPv6 protocol, enabling regular routers to pass them without modifications. IPv6 natively defines a routing header extension that contains a list of nodes to be visited up to the destination [174], being limited to a single

[16] This feature is necessary as the IPv6 header doesn't have a checksum protecting field, being checksummed by extension headers only.

intermediary destination using MIPv6's type 2 routing header [2]. In the context of agent-to-agent data encapsulation inside the network, this header is especially suitable for this task because it can be logically introduced and removed between two nodes very efficiently, as there is no checksum field in the modified headers (e.g., the IPv6 standard header and the routing header)[17].

Thus, each eTIMIP agent that needs to encapsulate data packets simply inserts this header in the packet, updates the payload length and the next header fields accordingly, and swaps the final destination of the packet with the next eTIMIP Agent IP address. The resulting packet, illustrated in Figure 45, is then forwarded using regular IP forwarding mechanisms. When the encapsulated packet is received by the next eTIMIP agent, it replaces the destination IP addresses (e.g. its own address) with the LMN IP address contained in the routing header, removes the routing header and updates the payload length and the next header fields accordingly. These actions are sufficient to restore the original packet. Then the process continues, being trivial to carry out such operations only logically inside the router if the packet is to be de-encapsulated and encapsulated in succession. When the packet reaches the final AR, the original packet is restored and is directly delivered to the LMN, which will accept it as the upper headers' checksums will be valid again.

IPv6 Header		Routing Header	Higher Headers
IP Src: CN	IP Dst: Agent B	IP Final Dst: LMN	…

Figure 45: Encapsulated IPv6 data packets

[17] In IPv6 forwarding, the standard header is protected by the upper header's checksums, which consider an IPv6 pseudo-header composed of the final addresses, total length, and next header fields, being only verified by the destination node.

5 Basic eTIMIP Protocol evaluation via Simulation Studies

This chapter presents a complete performance evaluation of the eTIMIP basic protocol concerning both its efficiency and its transparency aspects. For this, the base protocol will be studied, analysed and compared in high-speed mobility scenarios to selected state-of-the-art solutions via simulation studies, modelled in the NS2 event-based simulator [162] [163] [179].

The simulations are performed in a reference scenario that features an Internet domain consisting of wireless mobile nodes and an infra-structured backbone, which support the MNs with dense pico-cells [104]. Regarding the wireless part, the MNs are connected to the network using IEEE 802.11 links, while the wired part is composed of wired ethernet links that form a mesh or tree topology. In this scenario, the mobile node's mobility will be managed either by basic eTIMIP or by the alternative studied solutions: the original TIMIP protocol, CIP, HAWAII or hMIP proposals. To complement the micro-mobility analysis, the original MIP macro-mobility protocol was also modelled and applied to the same situations, which confirms the necessity of micro-mobility technologies for high-speed, localised movement support [43].

The simulation studies were performed in the latest Network Simulator v2.31, which was extended to model both the eTIMIP and the TIMIP protocols. For comparison, implementations of HMIP, CIP, HAWAII and MIP from the CIMS v1.0 mobility suite were used [180], which were also upgraded to the latest version of NS2 (v2.31). Besides the common NS2 modules of traffic generation, links, queuing, wireless transmission, ARP, UDP/TCP agents, measurement agents and fixed routing, the simulator was extended with emulation of 802.11 hard handovers and with the detection of UDP out-of-order and late packets. The details of the full simulation pipeline, statistics calculation, load generation, scenario parameters calibration and the software implementation in C++ / oTCL / bash scripts are presented in Appendix A.

The rest of this chapter is organised as follows: it begins with the description of the simulation modelling, by detailing the reference scenario, the used topology and the evaluated metrics; then, the second section uses the chosen scenario to compare the basic version of eTIMIP to the alternative proposals using UDP data traffic; the third section will repeat the same tests with TCP traffic; finally, the final section will present a discussion part that summarizes the major pros and cons of the eTIMIP basic protocol overall, on the topics of handover efficiency, routing efficiency, transparency, reliability and scalability.

5.1 Simulation modelling

5.1.1 Reference Scenario

Figure 46 shows the chosen simulation scenario that represents an Internet domain organized according to cellular principles [45] [104] [57]. In this, mobile nodes will roam inside the network, being connected to it using IEEE 802.11 wireless links. The fixed part of the network (backbone) is constituted by a hierarchy of nodes, connected by wired Ethernet links forming a mesh or tree topology. Outside the domain, simple nodes emulate core do-

mains, home agents and correspondent nodes; to connect to the outside, the domain uses multiple border routers.

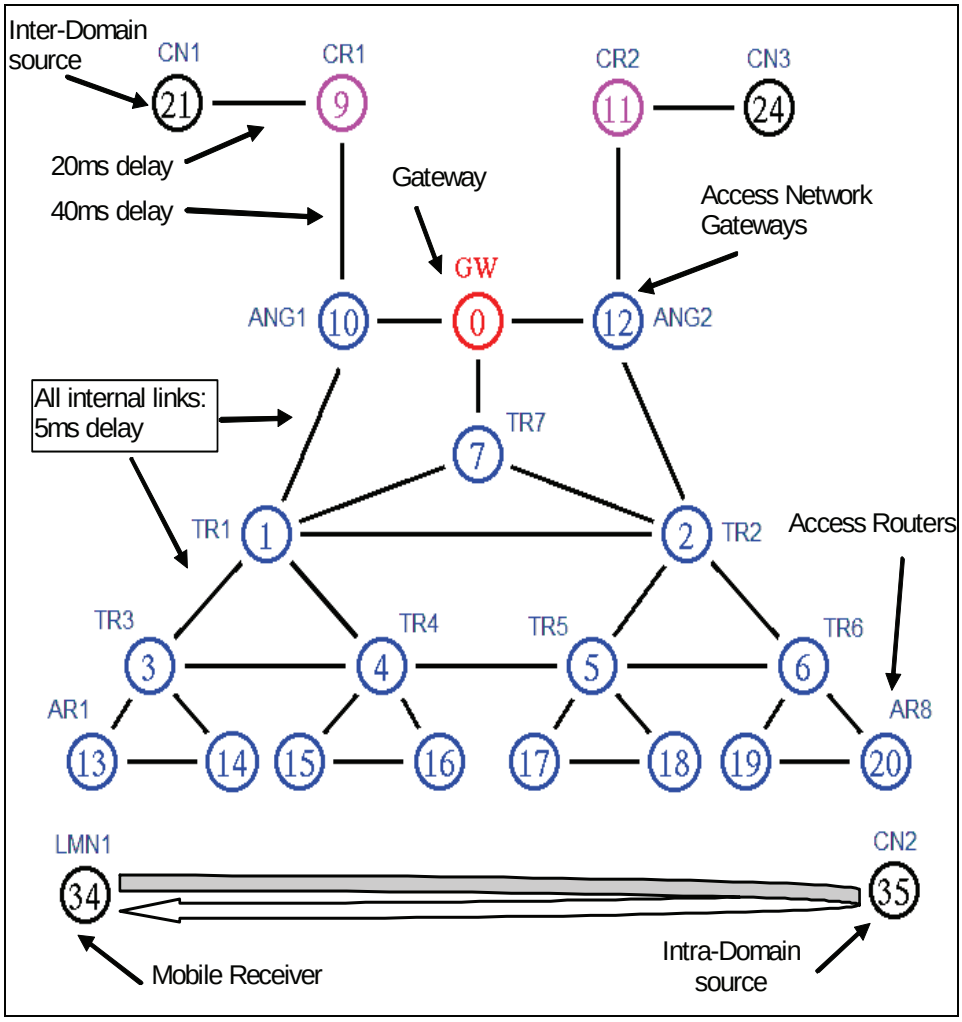


Figure 46: Simulation Scenario

The scenario features 8 NS2 BaseStations that share the capabilities of ARs and APs – i.e., they have both Layer 2 and Layer 3 capabilities. Each one of them will manage an independent 1 Mbit/s IEEE 802.11 cell with different frequencies. Thus, hard L2 handovers will be emulated, as the stations will only be able to receive data packets via one AP at a time [129]. The ARs are interconnected to the single GW by a series of agents, organized in a hierarchical tree structure of point-to-point wired links of 10 Mbit/s and sufficient buffer space to force queuing instead of packet dropping. The topology features extra redundant links that are required for testing the reliability against link failures, and the efficiency gains that some protocols may provide. All internal links feature the same constant delay of 5ms, being this value calibrated in Appendix A. The simulated backbone links can be customized by introducing link failures and/or link load, which are used to test the robustness of the protocols. If not specifically stated, all test sets assume that all internal nodes are mobility-aware by containing mobility agents, in order to test the maximum possible performance of the protocols.

The network features two MNs connected to it: the first (LMN1) will roam inside the domain, being the receiver of the test traffic, while the second (CN2) will be stationary at the last AR, being used for generating traffic for intra-domain situations. The first MN will move inside the network at an high speed (30 handovers/minute), performing back-and-forth

movements that cover the whole domain. In these movements, the MN will connect to each AR in sequence, starting from AR1 to AR8, to return from AR8 back to AR1. In addition, some tests explore the case of localized movements, where the MN performs the same number of handovers between two adjacent ARs, and stationary situations, where the MN is stable located at a given AR.

Finally, the domain also features two access network gateways (ANGs) to connect to the core networks, which are simulated using links with the higher delay values depicted in Figure 46. To simulate inter-domain traffic, a wired correspondent node (CN1) is used outside the domain that is directly connected to the Home Agent and is used to generate test traffic destined to the first MN; the intra-domain traffic is simulated by generating traffic at the stationary CN2 located at AR8.

Both UDP and TCP traffic types are considered in the performed tests. Regarding the former, the moving MN1 will be the receiver of a Constant Bit Rate (CBR) test flow of small test packets, of 100 bytes each, at a rate of 200 packets per second (the generation rate of this test stream is calibrated in Appendix A). Besides regular drop detection, the UDP receiver also features out-of-order packet detection capabilities. Regarding TCP, the MN will also be the receiver of a File Transfer Protocol (FTP) test flow of large packets, of 1500 bytes each, using the standard NS2 TCP Tahoe agent implementation. Both test streams are started long enough before the MN movements, by means of a warm-up period that stabilizes the scenario before the handover events. For the same reasons, the optional link failures and link load will start long enough before the handover events.

After this initial warm-up period, the chosen series of simulated movements will be performed, ending with the calculation of the metrics. To enable a higher level of confidence in the measured metrics, a series of multiple independent runs is performed. For this, the handover time instants are randomized in such a way that maintains the desired average MN speed. Besides the handover time instants, the random seed also randomizes other NS2 components, namely the traffic start instant and the wireless back-off random variables. Thus, each presented metric is taken from the average of all performed independent runs, coupled with its 95% confidence interval [162].

In the described scenario, the basic eTIMIP protocol will be compared with the original TIMIP model, and with the alternative micro-mobility proposals CIP, HAWAII and hMIP and the macro-mobility standard, MIP. From the set of CIMS options, a sample subset was selected: the CIP hard-handover option, due to the fact that L2 hard-handovers are the ones used in real 802.11 networks and HAWAII Multiple-Stream-Forwarding (MSF), due to usage of standard routing tables without interface information, and because the MN is able to listen/transmit to only one base station at one moment. As all protocols in this group, except HAWAII, have shown to have the exact same results in both tree and redundant meshed networks, all the results are presented for the Mesh case, and the HAWAII results in specific tree topologies is marked as HAWAII_tree.

The implementation of the simulation scenario is further described in Appendix A, which details the full simulation pipeline, statistics calculation, randomness sources, the software implementation in coded in C++, oTCL and bash scripts languages and the calibration of the UDP packet rate, the number of handovers and the internal link delays scenario values.

5.1.2 Metrics measurement

In the performed simulation studies, multiple metrics were evaluated to characterize the performance of the each protocol. The considered metrics are divided into three groups: the metrics that characterize the mobility service itself, the metrics specific for UDP transport, and the ones specific for TCP.

Mobility Service metrics

The mobility service metrics consider generic control and data forwarding metrics that deal with the mobility service itself, being independent of the data transport protocol in use by the MNs. The measured mobility service metrics are defined in Table 6.

Metric	Definition	Unit
Control Load	Average number of forwarded control packets per MN handover	%
GW Data Load	Absolute number of forwarded data packets by the GW	N
Maximum Buffer Usage	Absolute maximum number of buffered packets in all mobility queues per handover	N

Table 6: Mobility service metrics definition

UDP service Metrics

The UDP service metrics consider generic control and data forwarding metrics that are either specific to applications supported in UDP, or that are mobility-related but only measured by UDP test probes. The measured UDP service metrics are defined in Table 7.

Metric	Definition	Unit
Drop ratio	(Number of not received packets at the MN * 100) / Total number of sent packets	%
Out-of-order (OOO) ratio	(Number of received but not in sequence packets at the MN * 100) / Total number of sent packets	%
Total losses ratio	Sum of Drop and Out-of-order packet ratio	%
UDP Throughput ratio	((Number of accepted ordered bytes passed to the application * 100) / measurement time) / Theoretical maximum	%
One way delay	∑ (Reception timestamp - sender timestamp) / number of received packets	ms
Handover latency	∑ (reception timestamp first packet received via the new AR - reception timestamp last packet received via the old AR) / number of handovers	ms

Table 7: UDP service metrics definition

Figure 47 illustrates how several UDP service metrics are measured. The figure shows a time-series diagram zoomed in a handover event, and shows the generation time of each UDP packet at the sender and the corresponding reception time at the MN receiver.

In the X-axis of the figure, it can be seen that the handover latency considers the time difference between the first packet received via the new AR and the last packet received via the old AR. Also depicted is the one way delay, which measures the downstream delay experienced by the data packets, being calculated from the average differences between the reception and generation timestamps of the data packets that are received correctly.

In the Y-axis of the figure, the dropped packets metric considers the amount of packets that were never received by the MN due to the handovers, while the Out-of-order (OOO)

packets considers the amount of packets that were received at the MN, but were found to be out of sequence.

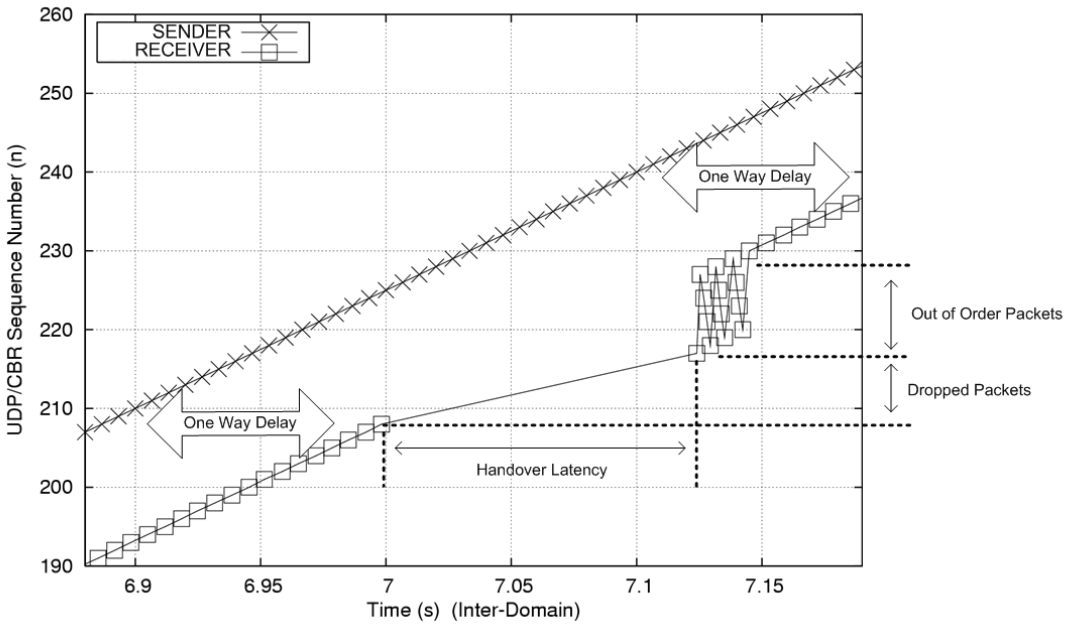


Figure 47: Example calculation method of several UDP service metrics

TCP service Metrics

The TCP service metrics consider generic control and data forwarding metrics that are specific to applications supported in TCP. The measured TCP service metrics are defined in Table 8:

Metric	Definition	Unit
TCP Throughput ratio	(Number of bytes passed to the application / measurement time) / Theoretical maximum	%
Overhead ratio	Number of retransmitted TCP segments / Number of transmitted TCP segments	%
Congestion Window	Value of the CWND TCP variable over Time	N

Table 8: TCP service metrics definition

Figure 48 illustrates how several TCP service metrics are measured. Again, the figure shows a time-series diagram that relates the time instants of the reception of TCP segments and generated TCP acknowledgements at mobile receiver, using the base NS2 TCP Tahoe agents.

At a handover, a certain number of packets will be dropped, resuming the flow reception at the receiver at a higher TCP sequence number. This will cause the MN receiver to emit multiple duplicated acknowledgements back to the sender, one per received disordered packet, to ask for the first missing packet. When the sender receives such duplicated acknowledgements, it performs two actions: first, it retransmits the dropped packets, which increases the “TCP overhead” metric; second, it lowers its own congestion window (CWND), which will cause a degradation of the “TCP throughput” metric. After the last missing packet is received at the MN, the TCP Tahoe receiver orders the received packets and passes those

to the application without duplications. At that time, the receiver issues a single final acknowledge that confirms all packets in a row, ending the handover effect in TCP.

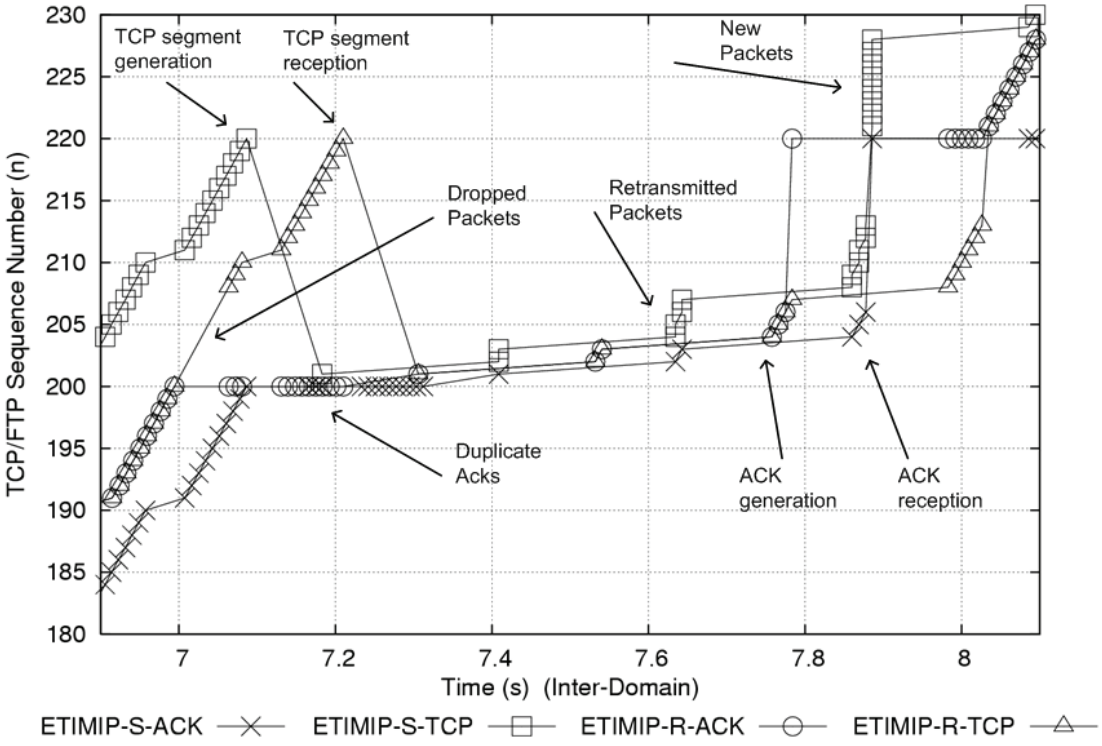


Figure 48: Example calculation method of several TCP service metrics

5.2 eTIMIP base tests using UDP data traffic

This section presents the results that compare the basic eTIMIP protocol in the reference scenario and its variations in UDP traffic types. Firstly, a discrete handover case will be presented, followed by a study of stationary MNs. Then, the handover effect will be considered for a variety of MN speeds, and in greater depth for the high-speed reference case, on a variety of situations. This will be done by varying the movement pattern, amount of background load, transparency level and amount of wired link failures.

5.2.1 Discrete Isolated handover

This test will illustrate what happens during a single handover, from the point of view of the mobile node, when being served by the eTIMIP basic protocol. For this, the MN will perform a single isolated handover in the middle of the domain, from node AR4 to node AR5, to show precisely what happens to the established data flows in each handover.

Figure 49 shows the same time diagrams that were presented in the UDP metrics measurement section, which depicts the generation time of each UDP packet at the sender and the corresponding reception time at the MN receiver, for both intra (left-side) and inter-domain (right-side) traffic types.

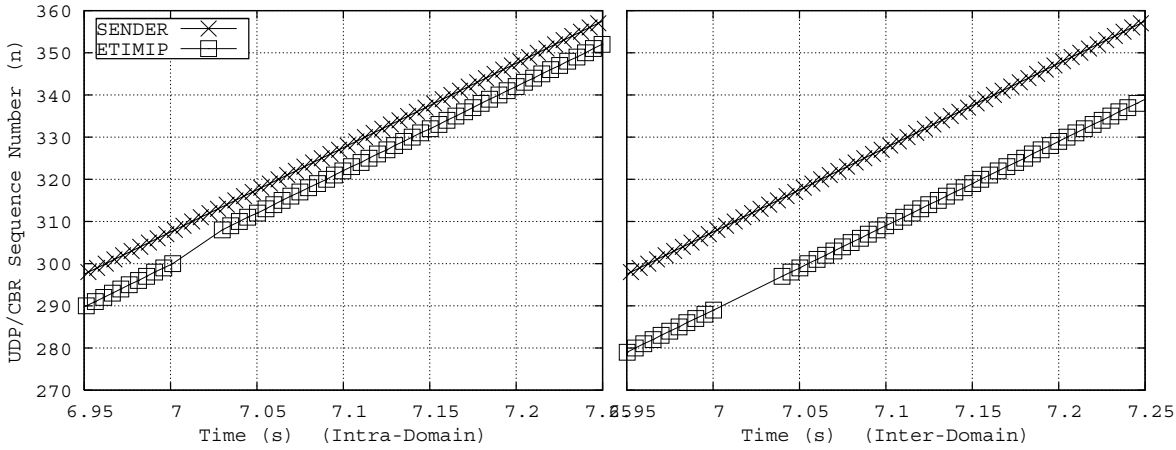


Figure 49: Isolated handover UDP time series - basic eTIMIP protocol

At first, the flow is established and packets are being received normally. Then, exactly at time instant 7.0s, a handover occurs, where the MN changes the point of connection to the network; as it changes between frequencies of the APs, it stops receiving all in-flight packets destined to the previous AR. Then, after a certain time (handover latency), the packet flow is re-established by the reception of a new packet through the new path. However, all previous packets are lost, having been dropped by the previous AR. Three major metrics can be measured from this event: the number of lost packets, the absence of out-of-order packets (y-axis), and the time latency needed to receive the first packet through the new path (x-axis).

5.2.2 Stationary Measurements

This test will characterize the routing paths used by the mobility protocols to stationary MNs, either because those are not being in movement, or if they only move inside the coverage area of its current AR.

For this, the MN will firstly move to each possible AR, starting from AR1, and remain stationary there during the metrics measurement. The one-way delay results are summarized in Figure 50, where the X-axis refers to the MN's location in the domain (AR), and the Y-axis refers to the average one-way delay. This metric is related to the number of hops, being important for the delivery of UDP real-time services.

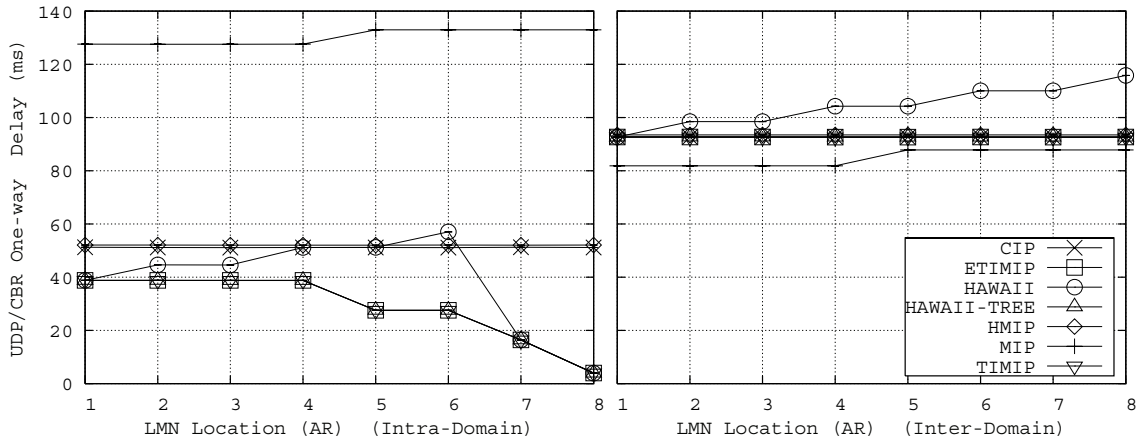


Figure 50: Stationary CBR/UDP one-way Delay per MN location

Regarding the intra-domain case, this experience shows that both eTIMIP and HAWAII protocols have varying one-way delay values while the MN approaches the sender (AR8).

While eTIMIP constantly reduces this value, HAWAII actually increases it up to AR6. At AR7 and AR8, the delay values drop to the same values as eTIMIP. TIMIP shows the exact same values as eTIMIP, while CIP and hMIP both feature a constant high delay.

eTIMIP's handovers reuse the previous paths to achieve fast handovers, but maintain those tree-optimal by always sending the signalling on the tree. As all links have the same delay at each hierarchical level, it results in a constant delay for the left side of the tree (ARs 1 to 4); on the right part of the tree, where the MN is closer to the sender, much lower delay values are measured as fewer and fewer hops are used. The eTIMIP protocol features the same latency results as the original TIMIP protocol, by featuring a similar tree-based handover and routing mechanisms.

Regarding HAWAII, this protocol uses additional links to reduce its handover time, and augments the previous routing paths with additional hops between the involved ARs. However, in the presence of meshed networks, these incremental handovers result in sub-optimal paths with higher hop counts [45] [64] [60], because the initial hops of the path are not modified by the handover procedure. In the case of this scenario, this means that packets are forced to always pass through node 3. Thus, while HAWAII starts with the same delay value of TIMIP, it increases this value each handover. Interestingly, the handover to AR7 cancels this phenomenon, as the crossover node in this case is node 6. However, if a tree scenario is considered, then HAWAII presents optimized results as eTIMIP for stationary situations.

Regarding CIP and HMIP, these protocols impose the constant high delay, because all intra-domain packets are forced to always pass through the GW. Thus, all packets must be sent upstream, only to be turned back at the GW. Finally, MIP has a very large one-way delay, due to its triangulation effect, where all packets are forced to pass through the HA, located outside the domain.

Regarding the inter-domain case, when packets are received from outside the domain, all protocols except HAWAII impose a constant delay, necessary for the 7 hops that the packets pass through. Similarly, each HAWAII handover increases the packet delay, up to the maximum value at AR8 [60].

Interestingly, MIP in this scenario has a lower one-way delay for inter-domain traffic than the micro-mobility protocols. This happens because the HA is directly located in the path from the CN, and thus MIP, in this particular scenario, is not penalized with the triangulation effect. After reaching the HA, the packets are sent directly to the correct AR, which grants an optimal direct routing inside the domain, using the extra links that connect the ANG. In contrast, the micro-mobility protocols redirect the packet flow to the GW in order to forward them as normally. Thus, this experiment shows that the micro-mobility protocols feature a local domain triangulation phenomenon, which results in the slightly higher measured delays.

5.2.3 Continuous Movement (multiple MN speeds)

This section presents simulations using continuous MN movements with different speeds, which are obtained by varying the time between the handovers. This will consider very slow movements of 60s inter-handover intervals, corresponding to 1 handover per minute, up to extremely fast 1s inter-handover intervals, corresponding to 60 handovers per minute. As the previous tests and pre-calibration have measured the absence of lost packets in stationary scenarios, all resulting packet losses will be due exclusively to the handovers. Thus these top speeds can be used to better study the differences among the protocols. The respective results

are shown in Figure 51, representing the packet loss ratio per handover rate, for both intra and inter-domain traffic sources, and Figure 52, representing the throughput for each MN speed.

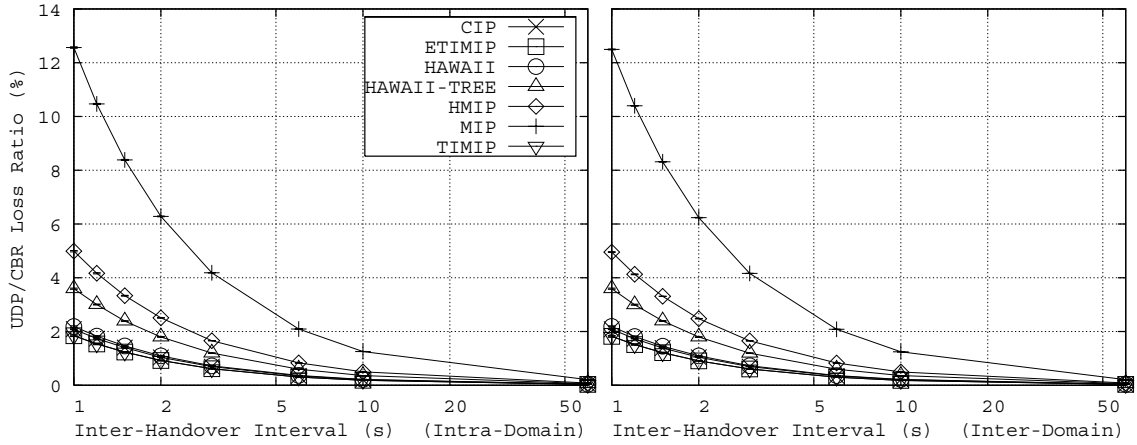


Figure 51: CBR/UDP Loss Ratio per inter-handover interval

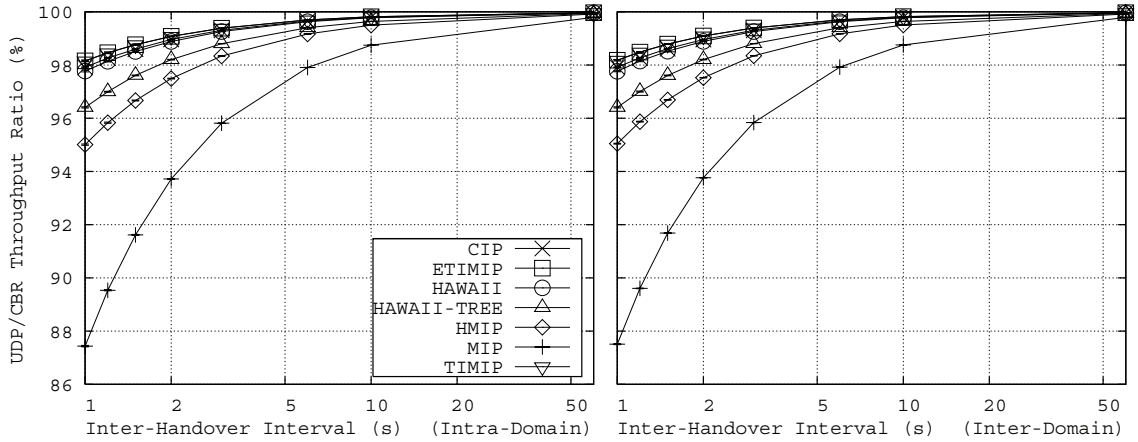


Figure 52: CBR/UDP Throughput per inter-handover interval

Regarding the intra-domain case, Figure 51 shows that the slowest speed (60s inter-handover time) results in marginal loss ratios for all protocols, even if a macro-mobility protocol is used (e.g. MIP). However, as depicted in section 5.2.1 (single handover), even such rare or isolated handovers are still noticed by the MNs, as those always impose losses and handover latency. At higher speeds, with lower inter-handover intervals, the effect of the handovers and the differences in the protocols becomes fairly clearer in these long-term measurements, which result from the protocols' different handover schemes and routing reconfigurations.

Considering the graph as a whole eTIMIP, CIP and HAWAII consistently present the lowest degradation of all alternatives, having eTIMIP a slightly better advantage over the other two. Regarding eTIMIP and CIP, such losses result as both protocols only have to update the crossover node, located in the local sub-tree, which is typically hierarchically close to the new AR. HAWAII also has a similar degradation ratio, but its losses are mainly due to the introduction of out-of-order packets in the UDP stream during a handover. However, in tree scenarios, HAWAII incurs in a much higher loss ratio, due to a much higher out-of-order phenomenon [62]. Again, TIMIP shows the exact same values as eTIMIP, using the same handover mechanisms. hMIP shows the worst performance of all micro-mobility protocols, because all handover's signalling must pass through the single GW, all packets being

sent to the previous location until the update packet reaches the GW. MIP further degrades this effect as all handovers must reach the far HA, located outside the domain.

Regarding the inter-domain case, the results are similar to the intra-domain traffic case. Such happens because as the MN movement is the same, the handovers will use similar locations to the crossover nodes, resulting in a similar amount of lost packets [60] [59].

Figure 52 shows the resulting throughput ratio, as experienced by the UDP receiver at the MN for both types of traffic. As the packets are lost or reordered by the handover, these affect the normal reception of the UDP flow at the receiver, degrading the achieved long-term throughput. The graphs show that the handover losses have a direct impact on the achieved long-term throughput, essentially being the inverted version of the previous losses graph, as the studied traffic is of CBR type. This happens because the sender does not respond to either congestion or network drops.

5.2.4 Reference Scenario (Single average high MN speed, Non-localized handovers)

This test constitutes the reference scenario that will be used to analyse the protocols' differences in greater depth. This is done by systematically analysing all the different mobility and UDP metrics defined in section 5.1.2, using an high speed scenario with an average speed of 30 handovers per minute. In this test, the MNs will perform the previously described movement that covers the entire domain, back-and-forth, resulting in an average inter-handover interval of 2 seconds.

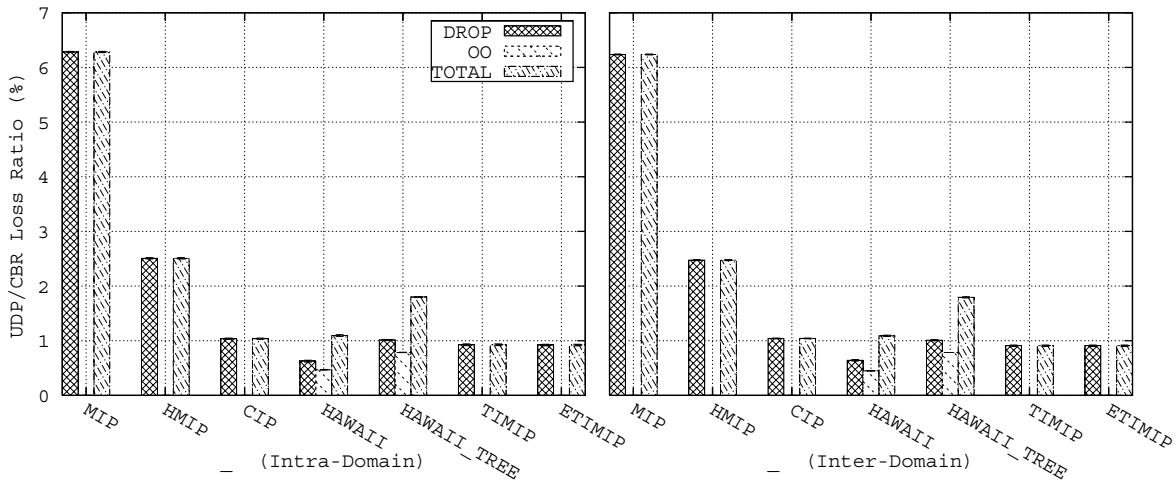


Figure 53: CBR/UDP Loss Ratio - reference case

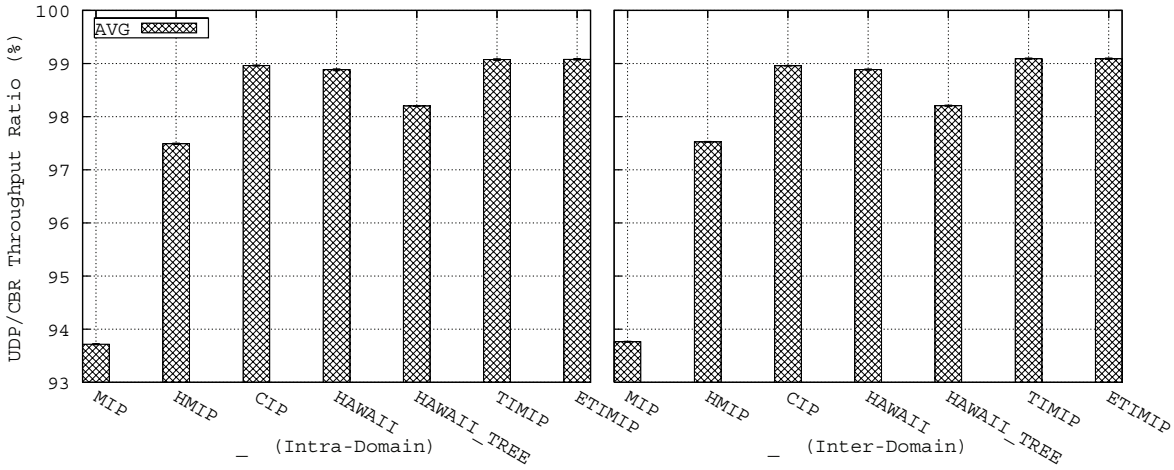


Figure 54: CBR/UDP Throughput - reference case

The graph of Figure 53 shows the total loss ratio separated by dropped packets and out-of-order packets, being clearer the differences between the protocols and the mentioned slight eTIMIP/TIMIP advantage. Besides the conclusions already mentioned in the previous test, based on the “total loss ratio” series, this graph extends the previous study by showing that HAWAII is the only protocol that introduces out-of-order packets due to handovers. As the HAWAII total losses are similar to eTIMIP and CIP, HAWAII necessarily forces a lower amount of dropped packets than in the other protocols.

Such happens because this protocol sends its update message directly to the previous AR, bypassing the tree nodes. However, by modifying the routing entries this way, it reorders the in-flight packets that are still being sent to the previous location through the tree. This effect is amplified in the case of a tree scenario, where HAWAII results in higher total loss ratio. The graph of Figure 54 shows the average throughput, confirming the previous observations.

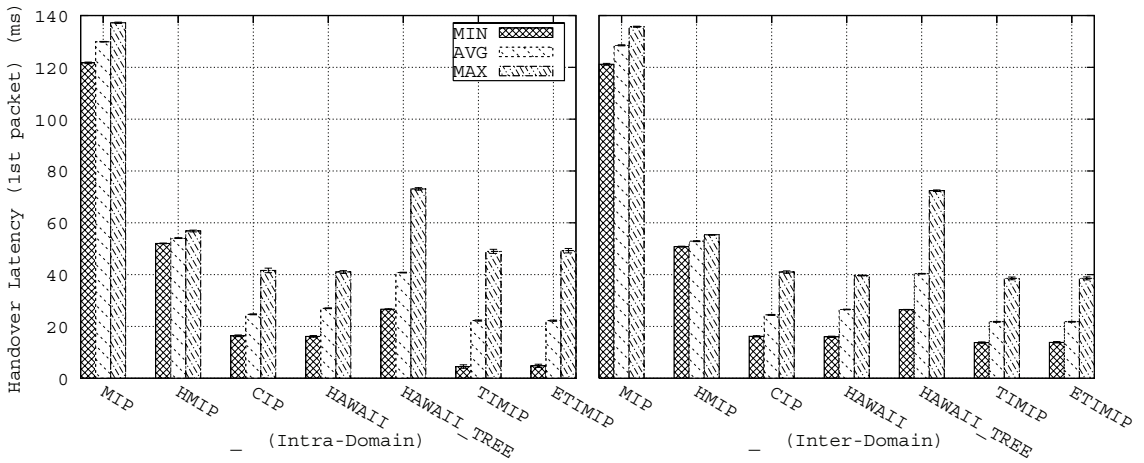


Figure 55: Handover latency - time of first packet received through the new AR

The graph of Figure 55 further reinforce the previous conclusions and characterize the HAWAII handovers, by showing the handover latency values for all protocols, separated in absolute minimum, maximum and average values.

All the efficient micro-mobility protocols have greatly varying handover delays, as this scenario considers a mixture of movements and locations, where each movement can have a close or far crossover node location. Such is not the case of HMIP and MIP, which have a

much higher and fairly constant handover latency, as the crossover node is constant and located in the GW or the HA node.

However, the average handover latency value is similar in all efficient micro-mobility protocols because the crossover node is always hierarchically close to the MN[18]. In particular, even tough HAWAII sends its update directly to the previous AR, its resulting handover latency is still similar to the other efficient protocols. Such happens because the packets take a similar amount of time to be redirected between the two ARs in HAWAII, as the amount of time to be redirected in agent tree in eTIMIP/TIMIP and CIP.

Finally, eTIMIP/TIMIP shows the greatest amplitude difference between the absolute minimum and maximum handover latency values. On one hand, eTIMIP has the lowest minimum handover latency of all cases, which happens when the MN performs the handover from AR7 to AR8 in intra-domain scenarios - as the MN moves next to the UDP sender, the in-flight packets may be immediately turned down on the shared AR8. On the other hand, eTIMIP also has the highest maximum handover latency of the efficient micro-mobility protocols, (except for HAWAII_tree), which happens when the MN performs the returning handover in the middle of its tree, from AR5 back to AR4 in intra-domain scenarios - as the MN moves for the right-side to the left-side of the tree, the crossover node will be node 2, as the traffic will be concentrated in the local right-side tree between AR8 and AR5.

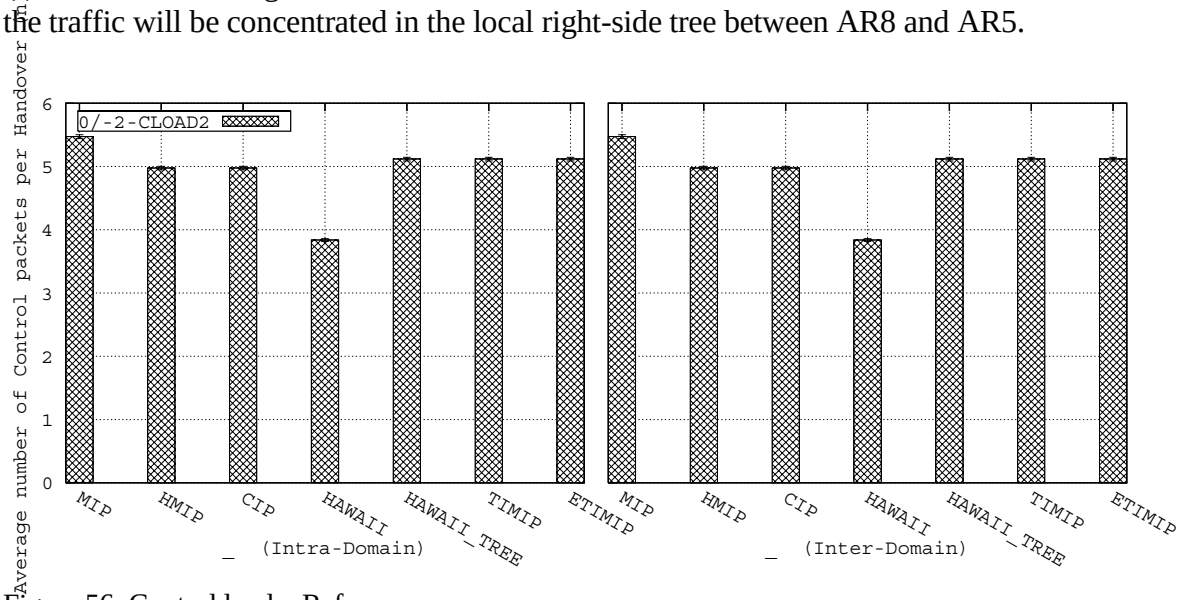


Figure 56: Control load – Reference case

Figure 56 shows the average control load that the protocols require to instantiate the handovers. All protocols have a similar number of update control packets forwarding, because all of them use a single update message that is forwarded, on average, 5 times per handover for all protocols. HAWAII has a slightly lower value, as the update packets are sent directly to the previous AR, bypassing the tree.

18 Except for HAWAII_Tree, for the same reasons described previously.

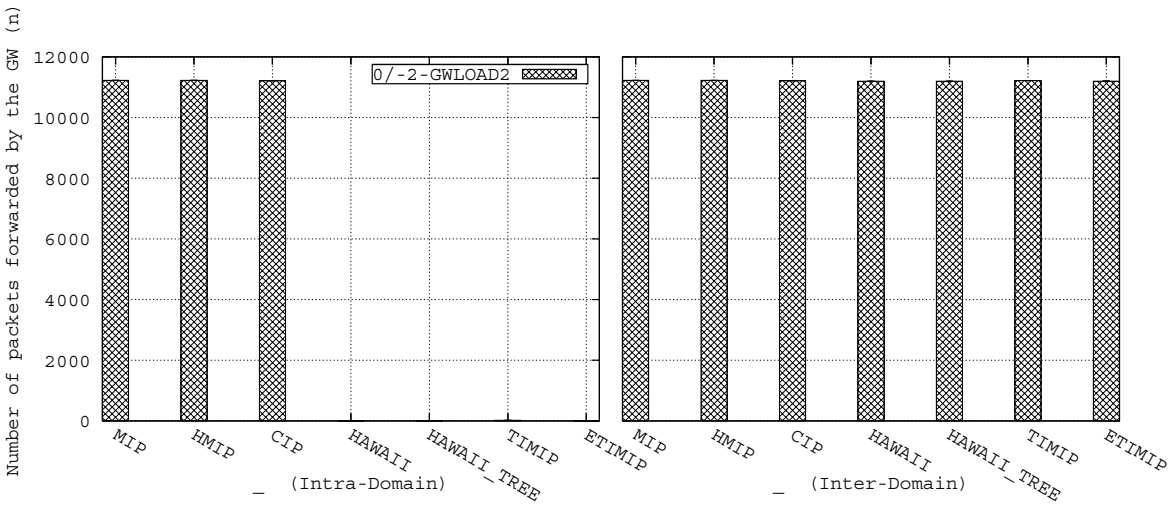


Figure 57: Data load on the GW – Reference case

Figure 57 shows the average data load on the GW, which is measured by counting the number of data packets forwarded at this central point. In general, all protocols have the same results, as the same amount of data traffic is always required to pass through the GW node. Such is not the case of eTIMIP/TIMIP and HAWAII in intra-domain cases, as these protocols can send the internal traffic directly in the lowest parts of the tree, without involving the GW. This represents a major advantage for these protocols, as the GW only has to perform control functions and inter-domain data traffic forwarding.

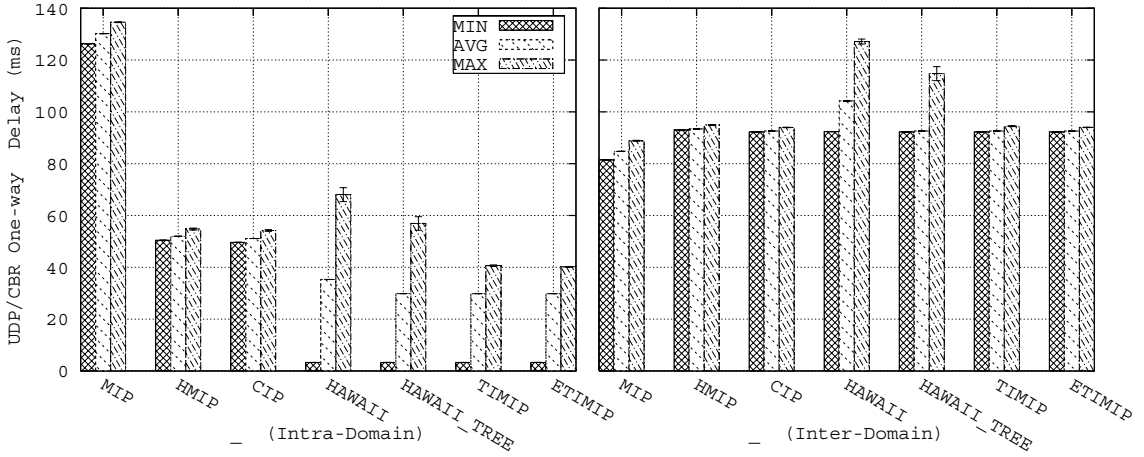


Figure 58: Averaged one-way delay for the reference scenario

Figure 58 shows the average one-way delay for the moving MN while roaming inside the domain. By covering all AR locations, this graph has essentially the same information already described in the stationary test (section 5.2.2), with the same conclusions being taken, where the values are aggregated into absolute minimum, maximum and averages.

5.2.5 Localized movement - Best and Worst cases

As it has been shown in the previous test, the eTIMIP protocol presents the greatest variation of the measured metrics, considering its minimum and maximum values. As such, this section will specifically study both the best and the worst eTIMIP cases, in order to further characterize the eTIMIP handover mechanisms.

For this, the reference scenario is modified to make the MNs perform its movements localized in a certain location of the domain, by continuously moving between two selected ARs.

In the best scenario, the MN will perform its handovers between AR 3 and AR4, which are located in the same local sub-tree, linked by crossover node 4. In the worst scenario, the MN will perform its handovers between AR4 and AR5, which are located in the opposite parts of the main tree, node 7 being the crossover node.

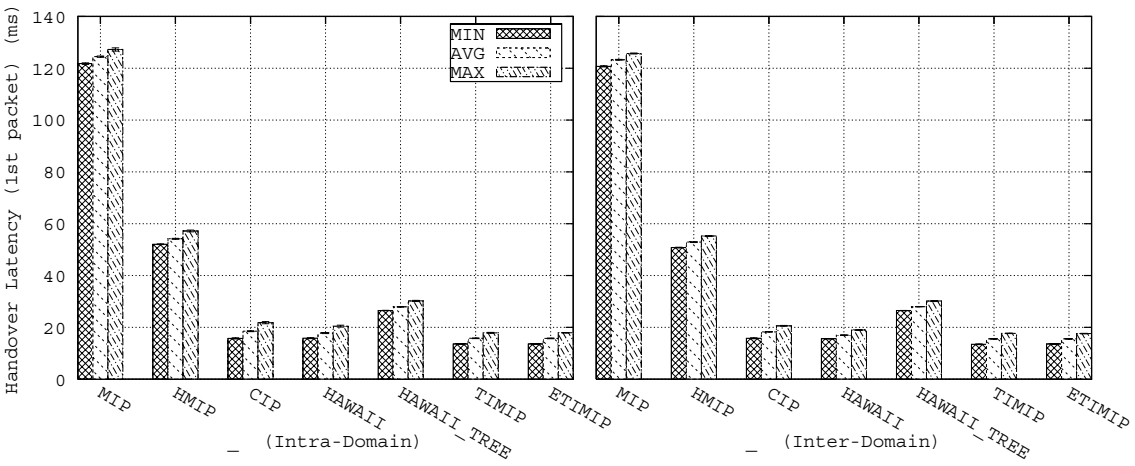


Figure 59: Localized handovers – handover latency – best case

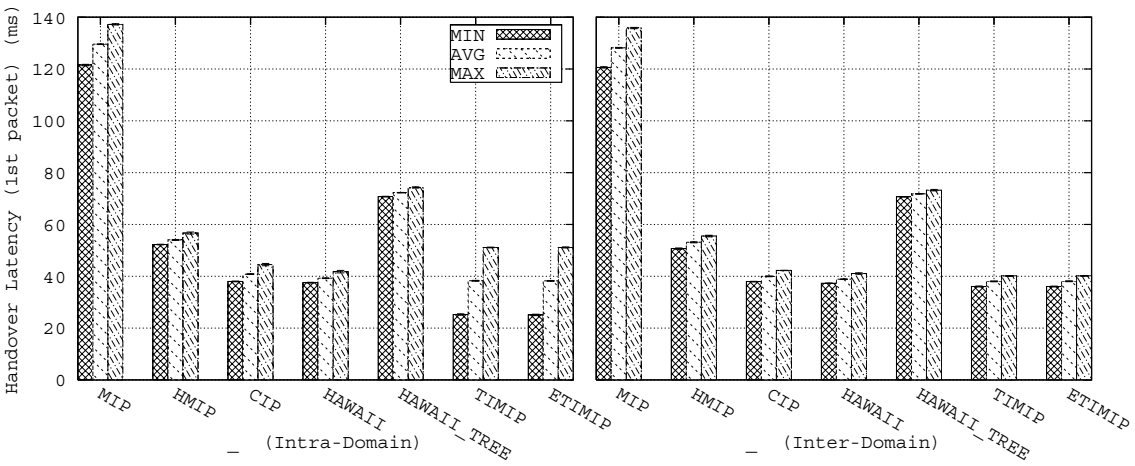


Figure 60: Localized handovers – handover latency – worst case

Figure 59 and Figure 60 show the latency results for the two extreme cases. In contrast with the previous test, now all protocols have close minimum and maximum handover latency values. The exception is eTIMIP/TIMIP in intra-domain case, where the left-to-right handover, which approaches the terminal from the sender, takes much shorter time than the complementary right-to-left handover. This happens because the former takes advantage of redirecting the packets that were already in-flight to the left-side tree at crossover node 2.

5.2.6 Wired Load Effect - Continuous movement

This test modifies the reference scenario by introducing increasing load on all wired links, in order to study the effect of the network load on the mobility protocols. For this, it was added to each link an Exponential On/Off traffic source, with the parameters were calibrated in such a way that produces bursts of queued packets that results in a variable long-term average link usage (this load calibration is described in Appendix A). In all protocols, the update packets do not have any kind of QoS priority preference, being treated like the regular data packets, which are treated under the Best-Effort traffic class.

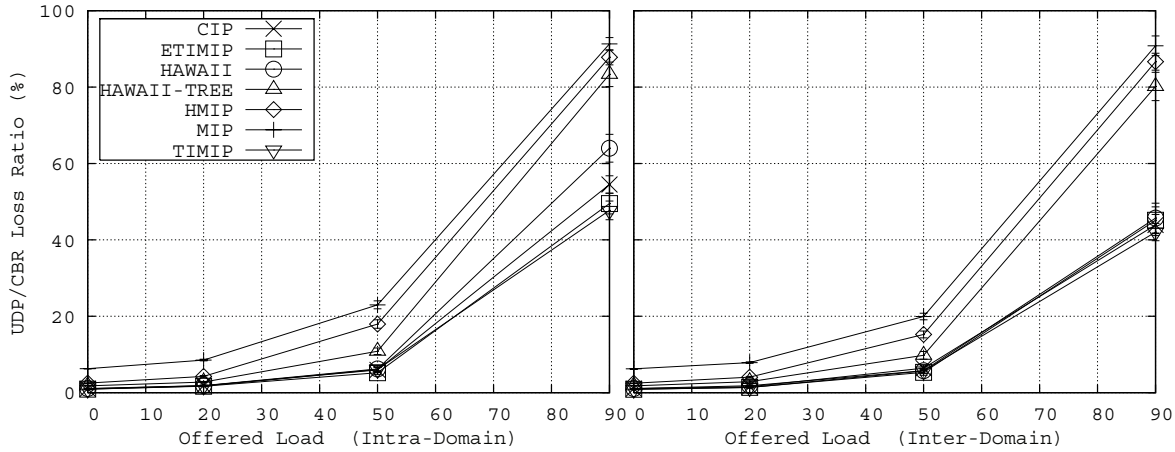


Figure 61: Wired load effect - total average loss ratio

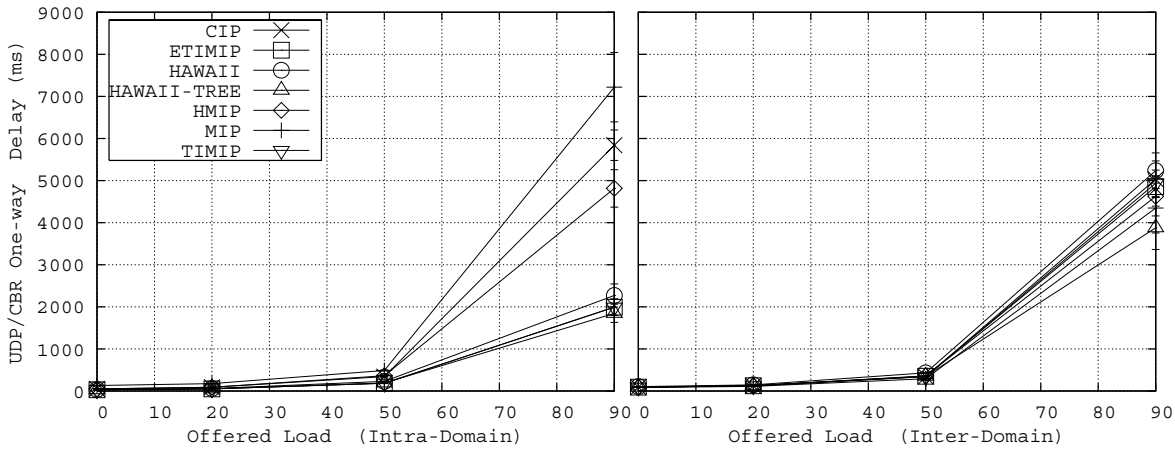


Figure 62: Wired load effect – Average one-way delay

Figure 61 show the corresponding loss results for various amounts of wired link loads, Figure 62 having the delay results. Here, it can be seen that both the handovers and the basic routing service are moderately affected by small amounts of wired link load, this effect being greatly amplified for the highest amount of link loads. This happens because, even though the routers have very large queue sizes to try to prevent packet losses, such queuing greatly delays the handover update, resulting in more packets being forwarded to the previous location where they will be dropped. For this reason, and by having simpler handover mechanisms, both eTIMIP and CIP show the lowest degradation in the highest load scenarios.

5.2.7 Transparency vs. efficiency - Number of agents

This test evaluates the trade-off between transparency and efficiency that eTIMIP provides, by measuring the impact of the number and the location of additional TRs on the handover efficiency. For this, the reference scenario is modified to have a varying number of agent levels in the tree, depicted in Figure 63. This will vary from a minimum degenerate tree-less scenario (1 agent level), where the mobility agents are only located in the single GW and all the ARs, up to the full agent tree that was studied in the reference scenario (4 agent levels).

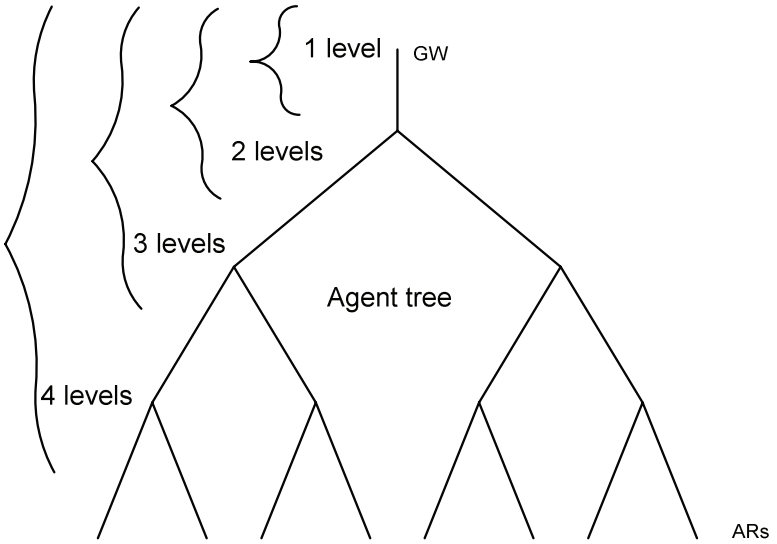


Figure 63: Number of agent levels (1 – GW + ARs only / 4 – full agent tree)

However, all protocols, except eTIMIP, only support parts of these deployment scenarios: hMIP and MIP only support the minimum deployment. In the efficient micro-mobility protocols CIP, HAWAII and TIMIP, only the full agent tree is supported, where all fixed routers must feature a mobility agent. In contrast, eTIMIP presents a choice between efficiency and transparency by being able to provide any combination between these two scenarios.

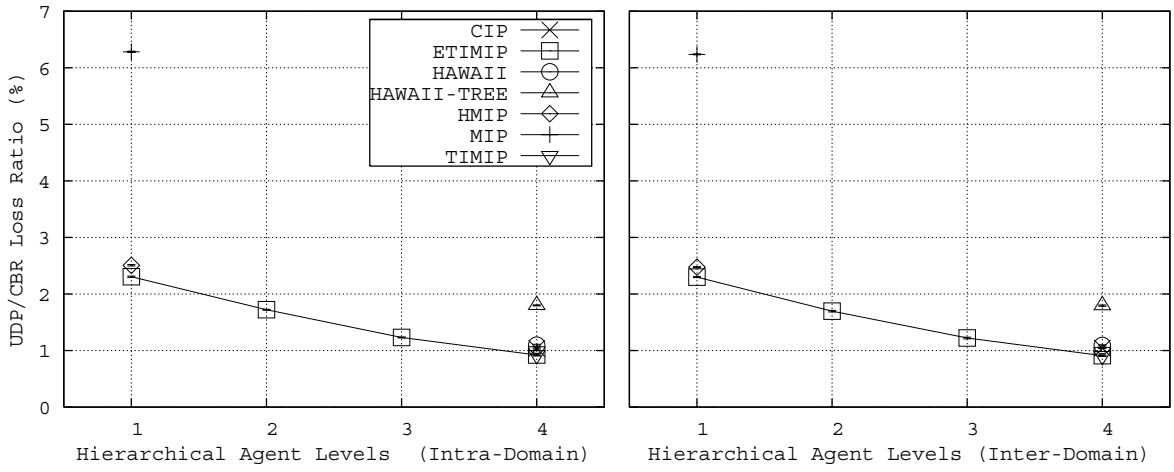


Figure 64: Number of agent levels (1 – GW + ARs only / 4 – full agent tree) – Loss ratio

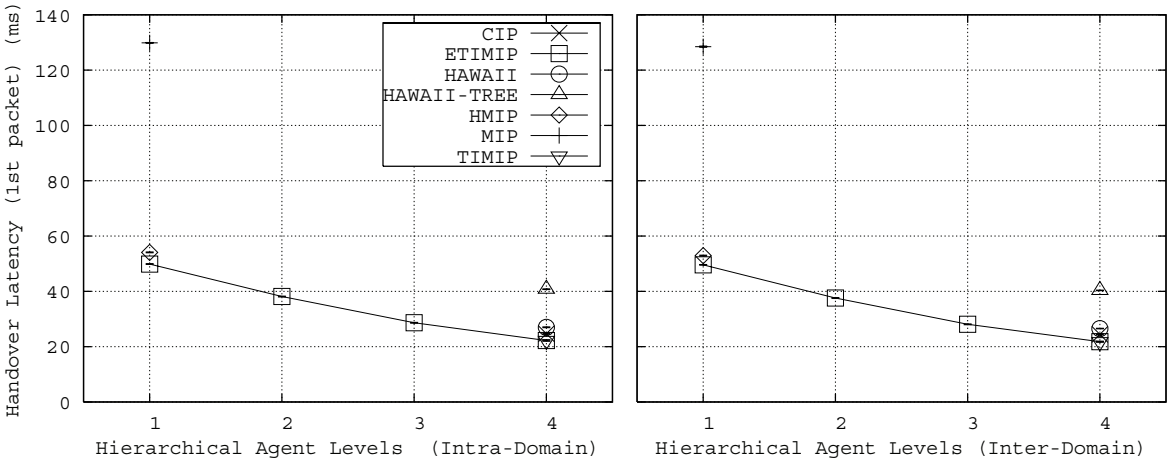


Figure 65: Number of agent levels (1 – GW + ARs only / 4 – full agent tree) – Handover Latency

Figure 64 and Figure 65 show the total handover losses and handover latency for all protocols by varying the number of agent tree levels. As hMIP only has one hierarchical level, it has the worst efficiency, as all handover updates must reach the single GW. As discussed previously, these results are amplified in MIP (featuring 8% of losses), where all updates must reach the distant HA. On the other hand, the efficient micro-mobility protocols (CIP / HAWAII / TIMIP) which require a strict hierarchical tree of mobile-aware routers, show the best performance, as the handovers are highly localized in lower parts of the tree.

In contrast, eTIMIP offers the flexibility of supporting any combination of legacy and mobility-aware routers in the network, for maximum transparency or efficiency. The results show that, as more and more agents are mobile-aware, the eTIMIP protocol efficiency increases, up to the same levels as the traditional efficient proposals when compared in the same circumstances as a full agent tree.

5.2.8 Degenerate tree test

This test further details the important case where a degenerate tree is used instead of a partial or full agent tree mesh. By using a single central GW and multiple edge agents (ARs / ANGs), eTIMIP becomes similar to other protocols that only feature a single MN point of attachment inside the domain [34] [105] [25], sharing with those the benefits of simpler agent reliability mechanism and simpler deployment requirements.

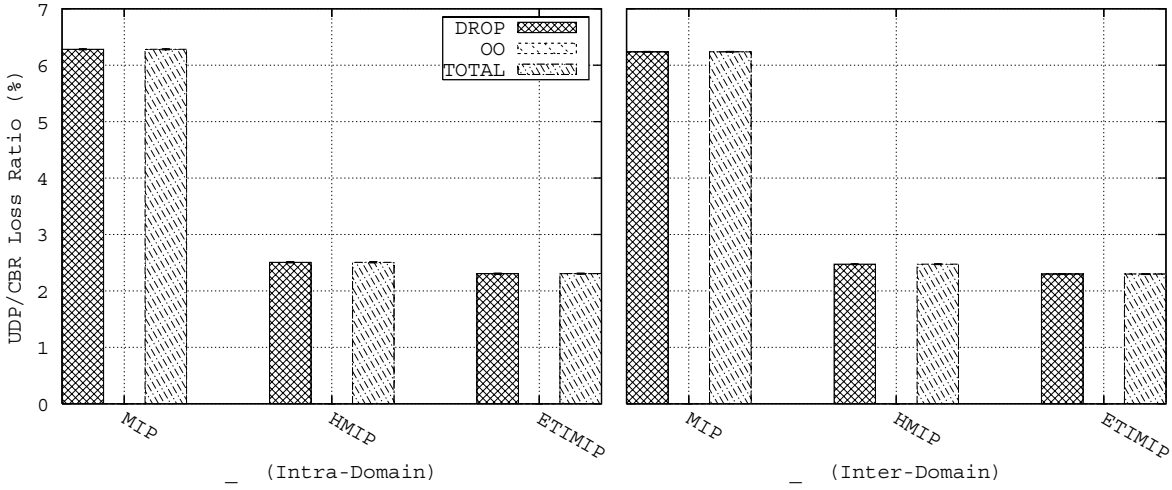


Figure 66: eTIMIP degenerate tree vs. other protocols – Loss Ratio

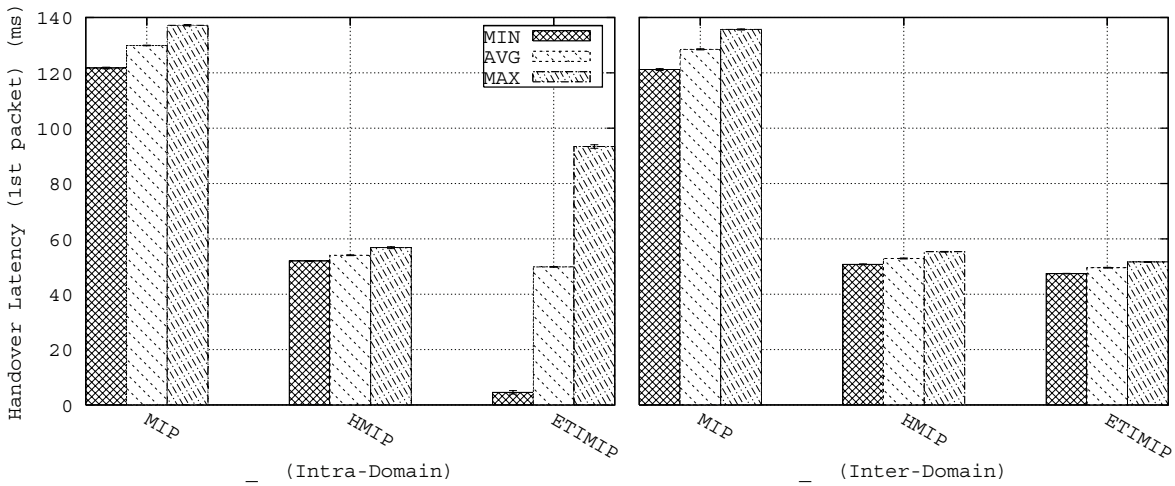


Figure 67: eTIMIP degenerate tree vs. other protocols – Handover Latency

Figure 66 and Figure 67 show the handover loss and handover latency. Here, the previous conclusions are maintained, eTIMIP having a performance that is similar to hMIP's. However, the handover performance is actually slightly better for intra-domain cases, because when the MN moves to the same AR as the sender, the handover latency is almost instantaneous (resulting in the very low minimum handover latency in Figure 67).

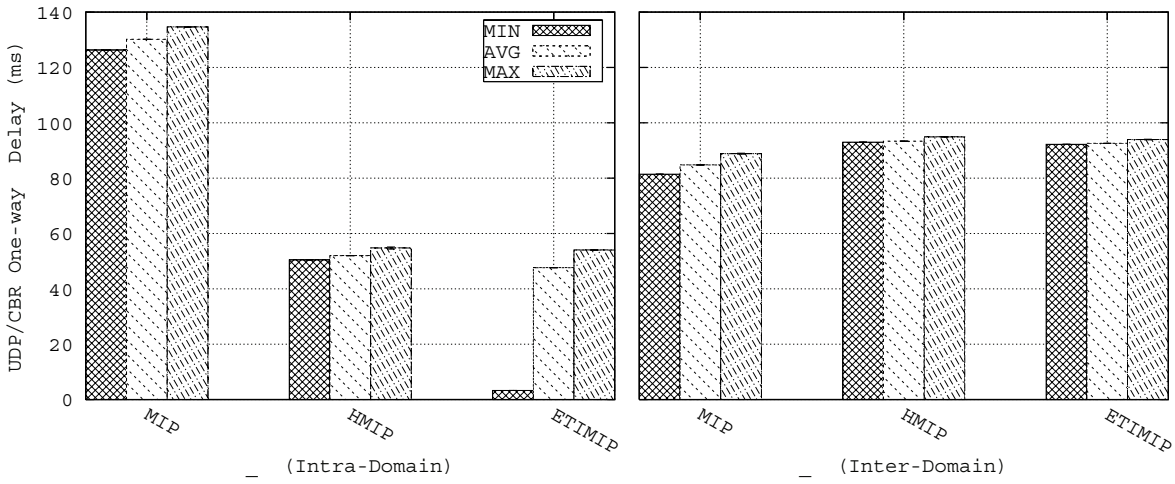


Figure 68: eTIMIP degenerate tree vs. other protocols – One Way delay

Figure 68 shows the one-way delay, which confirms the previous conclusions: eTIMIP with a degenerate tree forces all traffic to pass through the GW, just like hMIP, except when both MNs are located in the same AR (confirmed by the minimum delay value in the intra-domain case).

5.2.9 Link Failures Effect - Stationary

This section evaluates the mobility service reliability through the study of the impact of the wired link failures on the communication process for stationary MNs. As the failures on the wireless part of the network equally affect all protocols, this test set will focus on the failures that happen on the wired links. Although these failures can be considered as rare, they can result in a major impact on the mobility service that can affect **all** terminals located in a certain AR.

As such, this study will repeat the previous stationary test with the introduction of link failures in selected domain locations long enough before the measurement events. The loss ratio results are presented in two different forms: Figure 69 shows the effect of a random link failure when the LMN is stationary at each possible location; Figure 70 shows the effect of a specific hierarchical level failure (where the links are grouped as depicted in Figure 71) when the LMN is stationary at a random location.

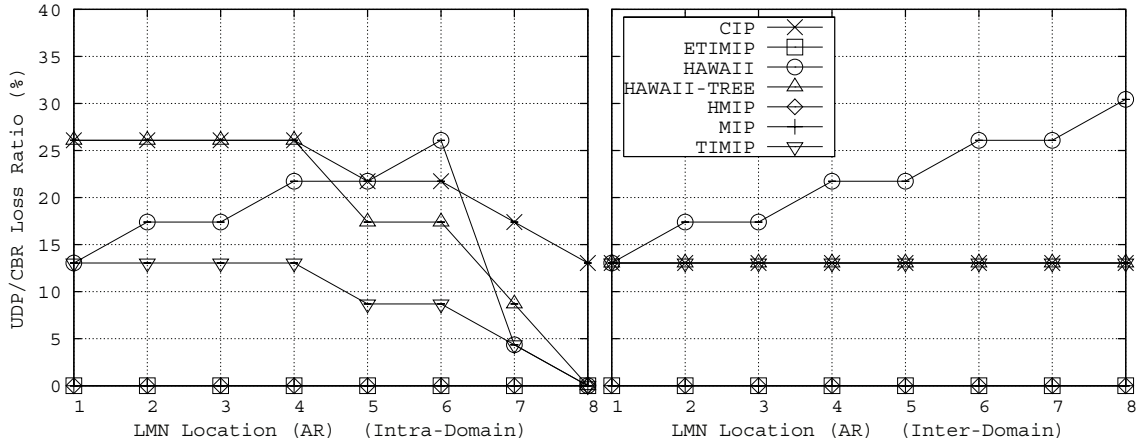


Figure 69: Packet loss ratio with random link failures

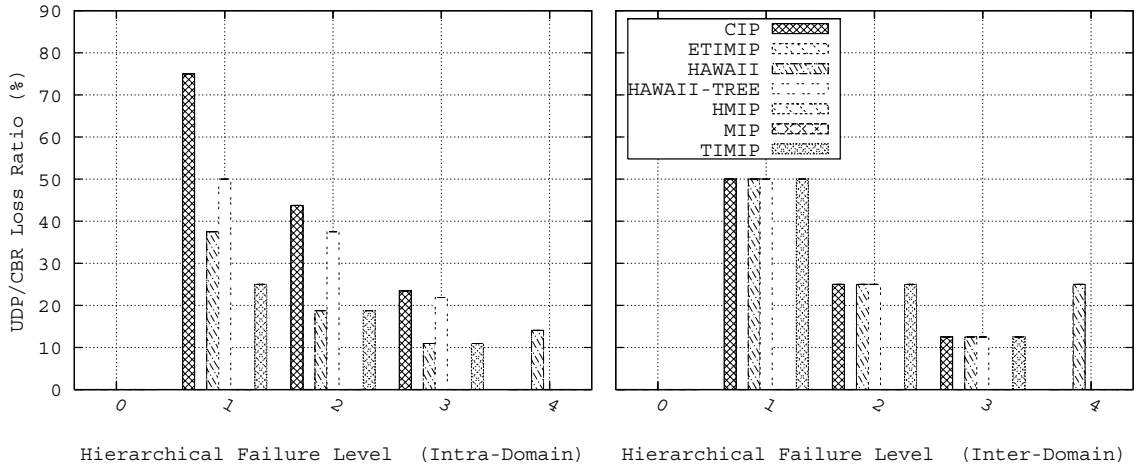


Figure 70: Packet loss ratio with hierarchical link failures (0 – no failure / 4 – mesh links failure)

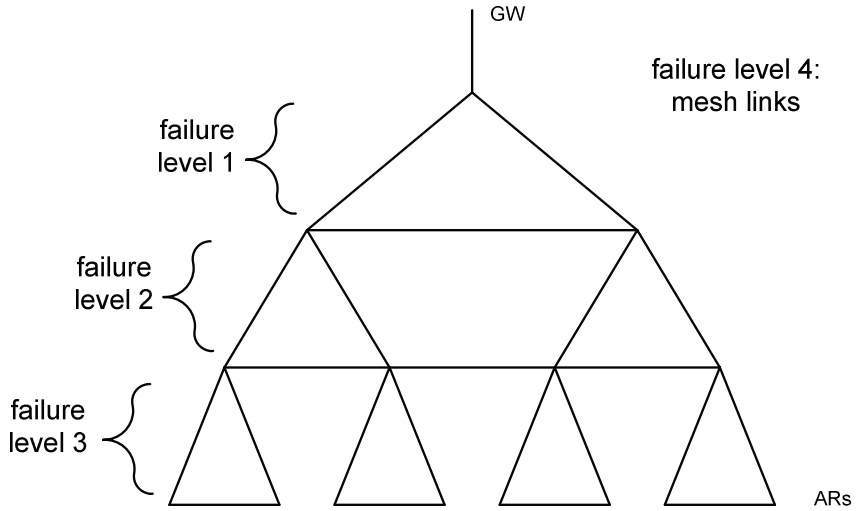


Figure 71: Packet loss ratio with random link failures

Concerning intra-domain communication, the results show that the absence of link failures results in 0% losses (series “0” in Figure 70). However, the presence of link failures results in intolerable packet losses in CIP, HAWAI and TIMIP, for all MNs that are located at the affected ARs. This is due to their direct, static, per-host, un-encapsulated routing entries; although used to provide fast handovers, these fail to address alternative means to route the packets when the path is broken by wired link failures. Interestingly, the reported behaviour is fairly different in the other protocols, TIMIP being the least affected. In contrast, eTIMIP and HMIP/MIP have marginal loss ratios in all cases, as the fixed routing is able to recover from the network impairments and can find alternative routes to send the data.

When the MN is located at the left part of the domain (AR1, Figure 69), CIP has double the loss ratio of TIMIP and HAWAII. This happens because CIP requires packets to follow a strict path on the way to the GW; this is not the case with the other protocols, which can use the fixed routing for uplink routing. Thus, CIP is the only protocol affected by failures on the right part of the domain. Although starting equal to TIMIP, HAWAII increases its loss ratio at each location, as its longer paths are more vulnerable to failures. Finally, losses drop to marginal values on AR7 and AR8, because the shorter paths in this location are less vulnerable to failures.

When the failures happen in the upper parts of the tree (Figure 70), CIP is again the most vulnerable protocol, as it forces all traffic to pass through the GW. In particular, a failure in the link that serves the GW results in the loss of all traffic for all locations and terminals. TIMIP and HAWAII have similar loss values, with TIMIP having an advantage because of the longer paths that HAWAII tends to have. As the failures are located in the lower parts of the tree, less traffic is disrupted, as more ARs become reachable again. Finally, the graph presents link failure for the mesh links, in group 4. In this case, only the HAWAII protocol is affected by these failures, confirming the earlier conclusions regarding the use of these links.

Concerning inter-domain communication, similar conclusions can be reported for the previous test. HMIP only imposes marginal loss ratios in all situations. For TIMIP and CIP, the location of the MN is independent of the loss ratio when in the presence of random link failures, resulting in a constant loss ratio (Figure 69). However, HAWAII sees its loss increase on the right part of the domain, because of the longer paths it uses. Regarding the level of failure (Figure 70), a failure in each of the levels conducts to cascaded failures of 50%, 25%

and 12.5% for each level. Finally, HAWAII is the only protocol affected by mesh-link failures.

5.2.10 Link Failures Effect - Continuous movement

This test will extend the reference scenario with the effect of wired link failures for moving MNs. By combining both handovers and wired link failures, the resulting losses and throughput degradation will be largely coincident with the addition of the two previous tests that studied each degradation component separately.

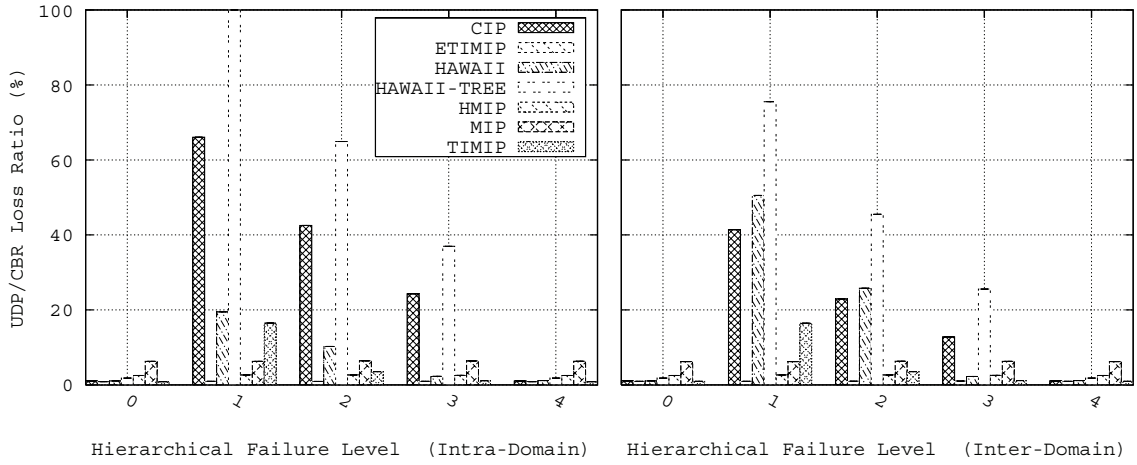


Figure 72: CBR/UDP Loss ratio for moving nodes, per level of hierarchical failure

Concerning an intra-domain traffic situation, the results show that in the absence of link failures, all losses are only due to the handovers (section 5.2.4). In the presence of link failures, eTIMIP and HMIP/MIP have the most regular behaviour, as all losses are mostly due to their handovers only, eTIMIP having the best result due to its optimised handover.

This is not the case of the other protocols, as the small handover losses are eclipsed by major losses due to link failures. In higher-level link failures, the uplink impact phenomenon is also present in CIP, reaching more than twice the number of the original TIMIP protocol or HAWAII losses. For lower-level link failures, both TIMIP and HAWAII can greatly improve their loss ratios, because some handovers can result in a different routing path unaffected by the failure. Such is possible as the signalling is directed to a fixed node (GW or old AR) that is always reachable by fixed routing (using the redundant links).

5.3 eTIMIP base tests using TCP data traffic

5.3.1 Discrete Isolated handover

This test will illustrate what happens during a handover, from the point of view of the mobile node and the sender, for the specific case of TCP traffic. This important class of traffic has the major aspect of using a closed feedback loop, in which the sender uses the receivers' acknowledgements to provide a reliable service and congestion control. Thus, the loss of packets, either of drop or flow reordering nature, will have a greater impact on this kind of flow transmission, comparing with the previous CBR UDP tests.

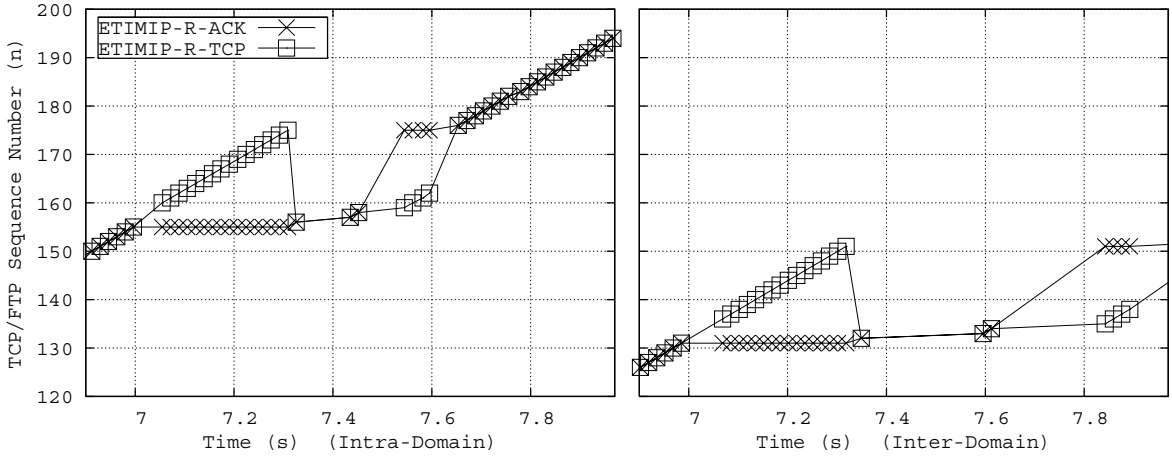


Figure 73: Isolated handover – TCP segment and acknowledgements at mobile receiver

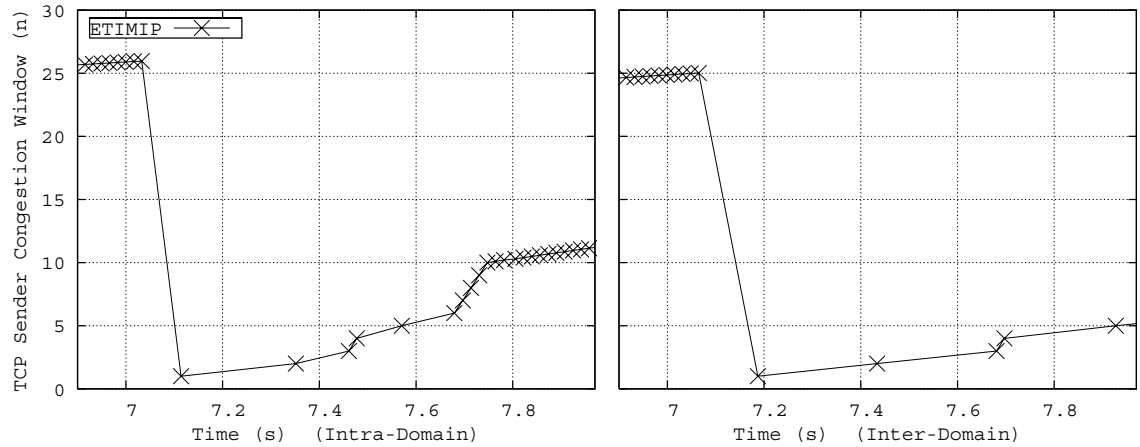


Figure 74: Isolated handover – TCP Congestion window at fixed sender

Figure 73 shows a time diagram that relates the time instants of the reception of TCP segments and generated TCP acknowledgements at mobile receiver, for both intra and inter-domain cases. Figure 74 shows the corresponding congestion window value at the fixed sender, which regulates the generation of the TCP segments.

At first, the flow is established and the packets are received normally at the MN. As no packets are missing or reordered, each received TCP segment causes the generation of an TCP acknowledgement for the same sequence number, resulting in the overlap of both packet types at the left-side of the graph [65]. When the fixed sender receives such TCP ac-

knowledgements, it will linearly increase its congestion window (CWND) as defined by TCP Tahoe standard.

Then, exactly at time instant 7.0s, a handover occurs, where the MN changes the point of connection to the network; as it changes between frequencies of the APs, it stops receiving all in-flight packets destined to the previous AR. In the meantime, the TCP sender will continue to send more TCP packets, as it features a large enough congestion window CWND TCP value.

After the handover mechanisms studied previously, the packet flow is re-established through the reception of new TCP segments via the new path. As there are missing packets, the receiver stores the received TCP segments in a buffer, and generates an ACK per received TCP segment requesting the first missed packet. Thus, this phenomenon results in the horizontal ACK line of Figure 73, one per received packet. When the TCP Tahoe sender receives three duplicate ACKs, it halves its data rate, sets the congestion window to 0, and starts retransmitting the requested packets in very small bursts (TCP slow start).

After a complete iteration of this feedback cycle, the missed packets reach the receiver, which can then acknowledge multiple sequence numbers and sort the previously received out-of-order packets. Then, the received acknowledgements will exponentially increase the CWND, in the slow start phase, which enables the sender to retransmit a larger amount of packets. The flow finally converges to a regular, pre-handover operation, when the MN receives all retransmitted packets, sorts them, and requests a new packet. However, due to the values of Round Trip Times (RTT) involved, especially in the case of the inter-domain traffic, this feedback cycle is not quick enough to avoid the unnecessary retransmission of some packets that were actually correctly received and buffered for subsequent sorting at the MN (e.g., around time7.6).

5.3.2 Continuous Movement (multiple MN speeds)

This section presents simulations using continuous MN movements at varying speeds, by varying the time between the handovers, in the same conditions of the previous tests of section 5.2.3. The respective results are shown in Figure 75, representing the average throughput for each MN speed, and Figure 76, representing the average TCP overhead for each MN, required for supporting a reliable transport service.

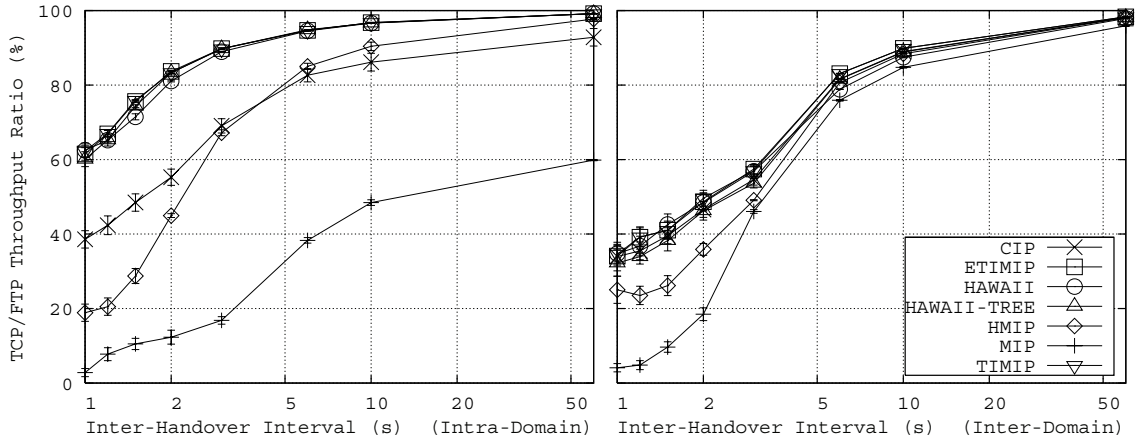


Figure 75: FTP/TCP Average Throughput per inter-handover interval

Regarding the intra-domain case, the graph of Figure 75 shows that the slowest speed (60s inter handover time) results in a small throughput degradation for all protocols except MIP.

In contrast with the previous UDP test, here MIP is highly penalized by its long RTT in intra-domain scenarios (section 5.2.2) which results in this protocol failing to achieve high throughput values. In higher speeds, with lower inter-handover intervals, the effect of the handovers and the differences in the protocols become fairly clearer in these long-term measurements, which result from the protocols' different proposed handover schemes and routing reconfigurations.

Considering the graph as a whole, all protocols impose a much higher amount of throughput degradation than the corresponding UDP tests (section 5.2.3). Such happens because the existence of dropped or out-of-order packets in all protocols will likely cause a slow start phase in the TCP sender, which severely limits the throughput performance. Thus, to solve this problem the handover process must deliver **all** packets to the TCP receiver without any gaps or out-of-orders, by supporting a loss-less / reordered-less handover service [61]. If such is done, the receiver will fail to notice the handover, and thus will not send the series of duplicate ACKs that reduce the TCP sender's congestion window.

Even tough all protocols eTIMIP/TIMIP, HAWAII and CIP have a similar number of lost packets per handover (per the UDP tests in section 5.2.3), not all result in the same TCP degradation in high-speed intra-domain conventions. This extra CIP throughput degradation is explained by its much higher RTT: as all TCP packets must always reach the GW, CIP will react slower to the handover and have a lower throughput. Thus, if the handover process does not avoid the occurrence of packet drops or reorders, then the mobility protocol should support lower RTTs to enable a faster retransmission of the missing packets.

Even though HMIP has an RTT performance which is similar to CIP's, its overall performance is still lower than the latter's. This is explained by the higher amount of packets dropped per handover, which require more retransmissions, leading to a longer slow start phase due to its already high RTT. Thus, if the handover process does not both avoid the occurrence of packet drops or reorders and does not support a lower RTT, then the mobility protocol should support a lower amount of dropped or reordered packets, by using a faster handover scheme[19].

Finally, the macro-mobility MIP protocol has the worst results for similar reasons as hMIP, but even more amplified by its higher RTTs and higher amount of dropped packets. These graphs clearly show the inability of macro-mobility protocols without micro-mobility technologies (e.g. standard MIP) for supporting very high-speed movements, by essentially not being able to support any connectivity at all.

Regarding the inter-domain case (right side of Figure 75), all protocols have a much lower performance than the intra-domain case, especially on the fastest MN speeds. As concluded previously, this happens because the handovers always involve packet drops and the long RTTs in inter-domain traffic lead to a substantial performance impact on all protocols. Even so, the best performance continues to be given by eTIMIP/TIMIP and HAWAII, being now joined by CIP as it features RTTs that are similar to the ones in these protocols. hMIP and MIP have the highest degradation of all protocols, by reasons which are similar to the previous case – high RTTS and a higher amount of dropped packets.

---

19 This fast handover requirement should not be confused with the above-mentioned loss-less/reordered-less handover requirement.

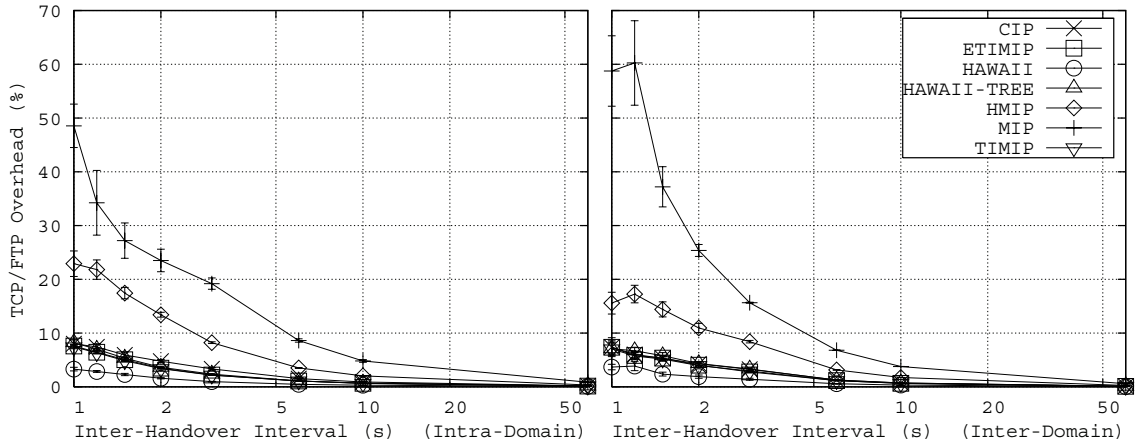


Figure 76: FTP/TCP Average Overhead per inter-handover interval

Figure 76 shows the resulting overhead ratio needed to ensure a reliable TCP transfer, which shows a different view on the same experiment. While the throughput graph was mainly determined by both the amount of lost packets and the RTT time, the overhead graph is mainly determined by the amount of lost packets only, which need to be retransmitted. Thus, the results are consistent with the ones verified with the previous graph, as all efficient micro-mobility protocols, including CIP, show a low overhead, a sign of a low amount of dropped packets. Also as expected from the previous UDP measurements, the inter-domain part of the overhead graph presents very similar values as the intra-domain part of the graph, as the amount of lost packets is similar in both cases.

5.3.3 Reference Scenario (single average high MN speed, non-localized handovers)

This test recreates the reference scenario with TCP traffic, in order to better study the impact of the handovers in this type of traffic in high speed scenarios (30 handovers/minute).

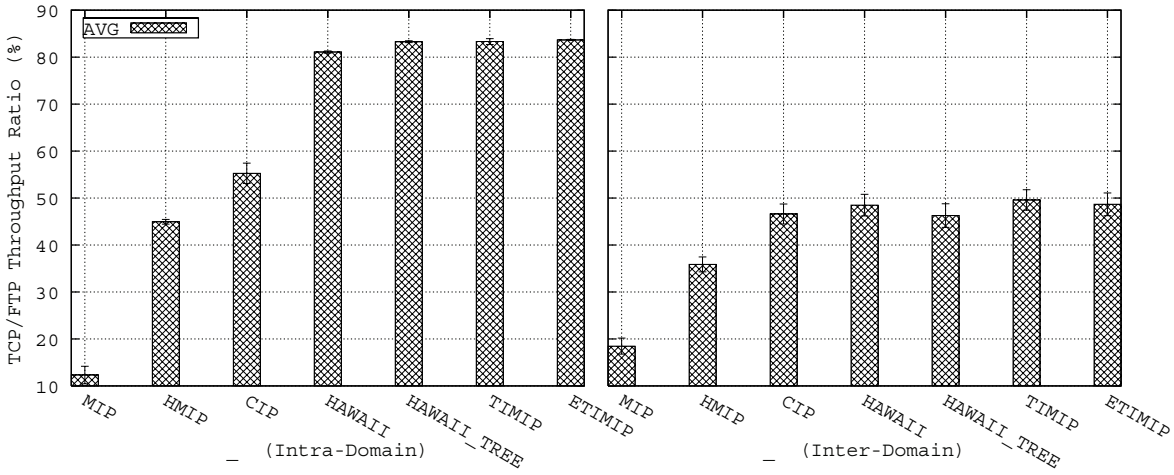


Figure 77: FTP/TCP Average Throughput – reference case

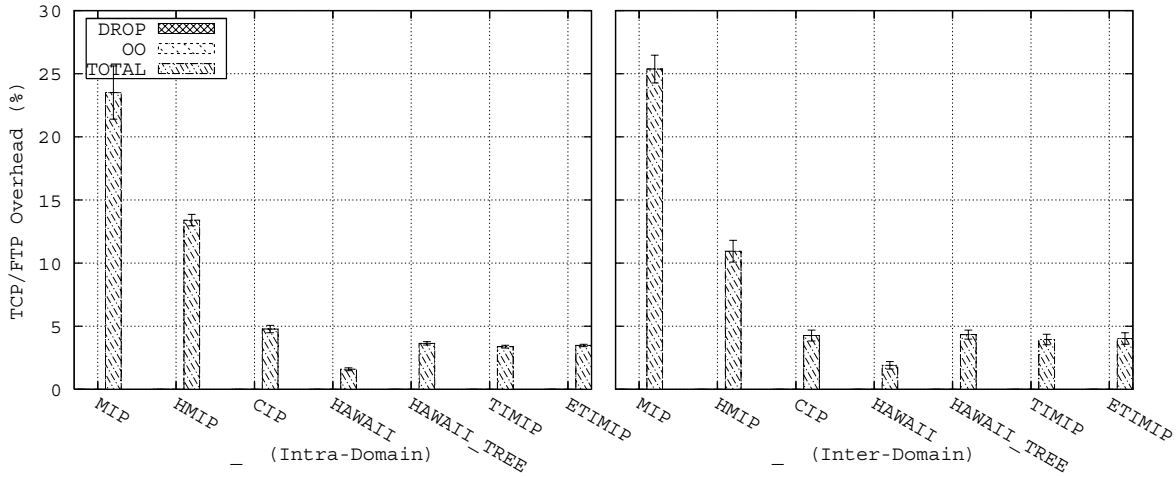


Figure 78: FTP/TCP Average TCP Overhead – reference case

The graphs of Figure 77 and Figure 78 show the average throughput and overhead for the case of the reference scenario. Here, all the previous conclusions are maintained, being clearer the differences of the protocols explained previously, namely the similarity of eTIMIP/TIMIP and HAWAII, the lower performance of CIP in intra-domain scenarios, the low performance of HMIP and MIP, and the poor performance of all protocols in inter-domain high-RTT scenarios.

Also, no significant difference exist for the case of mesh and tree topologies (HAWAII / HAWAII-TREE), having the latter a better throughput than the former (it should be stressed that the opposite behaviour was measured for UDP traffic, section 5.2.4). Such happens because even tough HAWAII in tree scenarios has higher handover latency and higher handover drops, it has a lower RTT by not having increasing routing paths (as per section 5.2.2) which enables a faster recuperation of the dropped packets. In addition, while HAWAII-TREE has a higher number of out-of-order packets per handover (Figure 53, section 5.2.4, in page 102), they do not further degrade the throughput, because the TCP receiver is able to sort the received packets in a buffer. The problem of these interesting features is an higher overhead metric, necessary for retransmitting the higher number of lost packets (comparing to HAWAII in a Mesh scenario).

5.3.4 Wired Load Effect - Continuous movement

This test recreates the previous test that modifies the reference scenario by introducing increasing amounts of load on all wired links, in order the study the effect of network load in the particular case of TCP traffic. Again, the update packets do not have any kind of QoS priority preference, being treated like the regular data packets, which are treated under the Best-Effort traffic class.

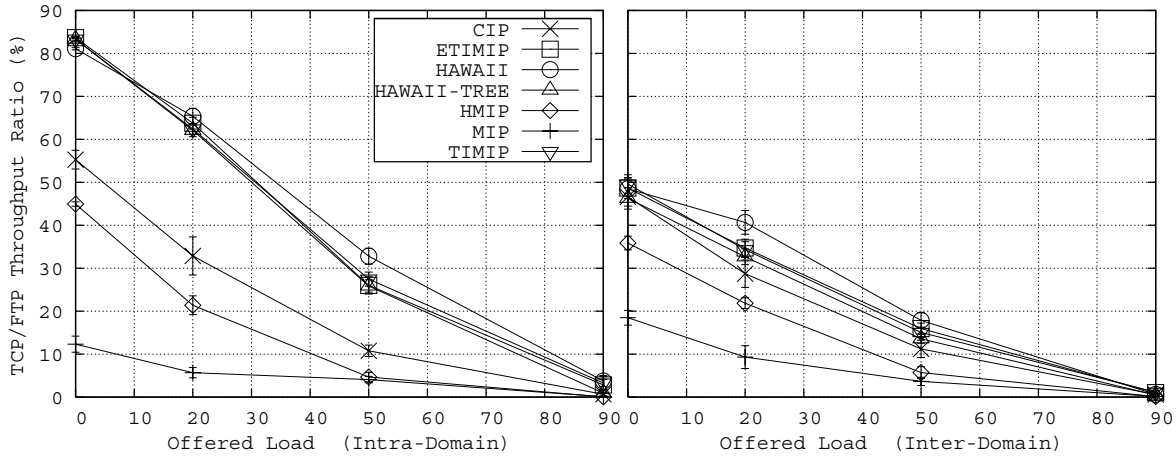


Figure 79: Wired load effect – TCP - total average throughput ratio

Figure 79 shows the corresponding throughput ratio for the same amounts of wired link loads considered previously. Here, it can be seen that all protocols are heavily affected by even small amount of load, particularly if they impose high RTTs (i.e., CIP, HMIP, MIP), for the same reasons described previously. Again, eTIMIP/TIMIP and HAWAII have clearly the best performance in all cases, especially for intra-domain traffic due to the lower RTTs achieved. Finally, the highest studied load ratio (90%) essentially neglects the service in the presence of fast handovers in all protocols (30 hand/min, as given from the reference scenario), as the excessive RTTs become higher then the TCP timeouts.

5.3.5 Transparency vs. efficiency - Number of agents

This test will repeat the previous test that evaluated the impact of different number of agent levels in eTIMIP basic protocol for TCP traffic. Again, the number of agent levels will vary from a minimum degenerate tree-less scenario, where the mobility agents are located in the single GW only and all the ARs, up to the full agent tree that was studied in the reference scenario.

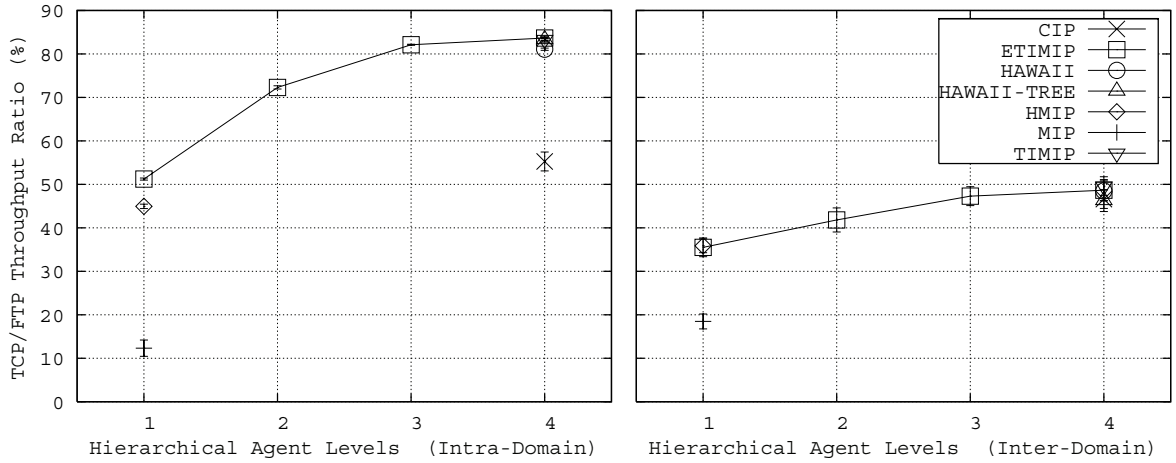


Figure 80: Number of agent levels (1 – GW + ARs only / 4 – Full agent tree) – Loss ratio

Figure 80 shows the average throughput for all protocols by varying the number of agent levels. As described previously, eTIMIP can support a transparency-to-efficiency trade-off, being similar to the classic mobility protocols when using a full agent tree mesh, and to hMIP when using a degenerate agent tree.

5.3.6 Degenerate tree test

The section repeats the previous degenerate tree test for basic eTIMIP with TCP traffic in greater detail. The following figures for TCP throughput and overhead confirms the previous conclusions.

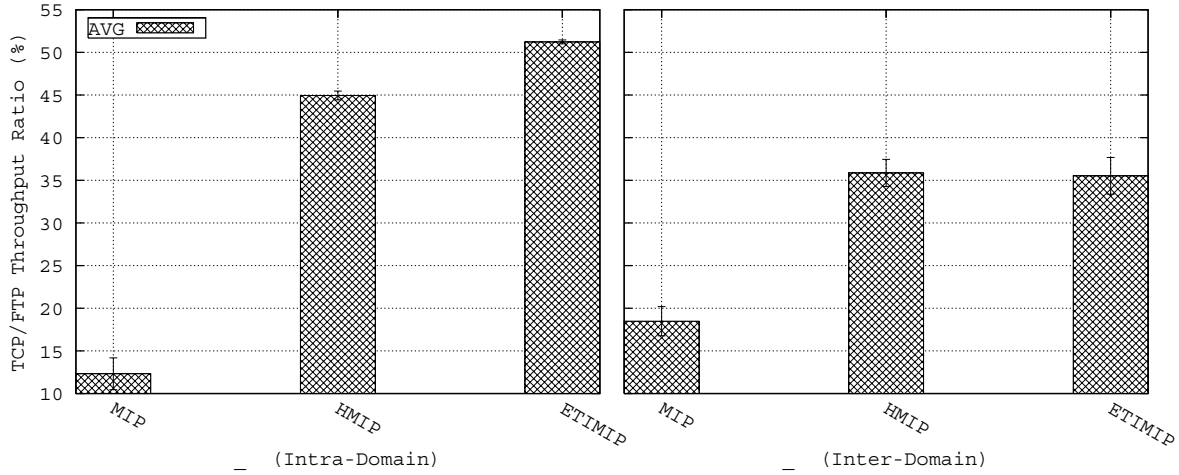


Figure 81: eTIMIP degenerate tree vs. other protocols – Throughput

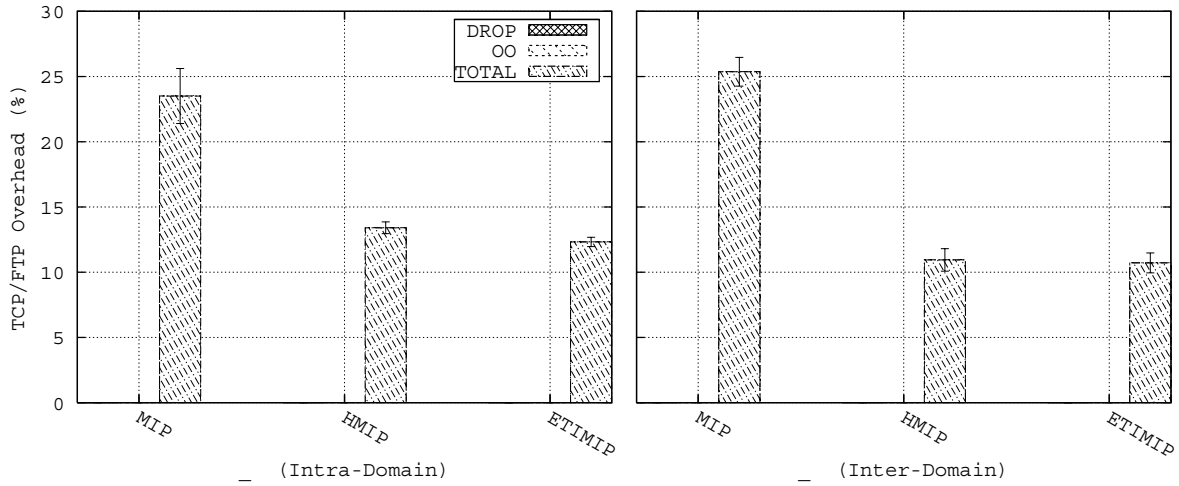


Figure 82: eTIMIP degenerate tree vs. other protocols – TCP overhead

5.3.7 Link Failures Effect - Stationary

This section repeats the previous test that evaluates the mobility service reliability, for the particular case of TCP traffic. As before, the results are presented in two different forms: Figure 83 shows the effect of a random link failure when the LMN is stationary at each possible location; Figure 84 shows the effect of a specific hierarchical level failure when the LMN is stationary at a random location.

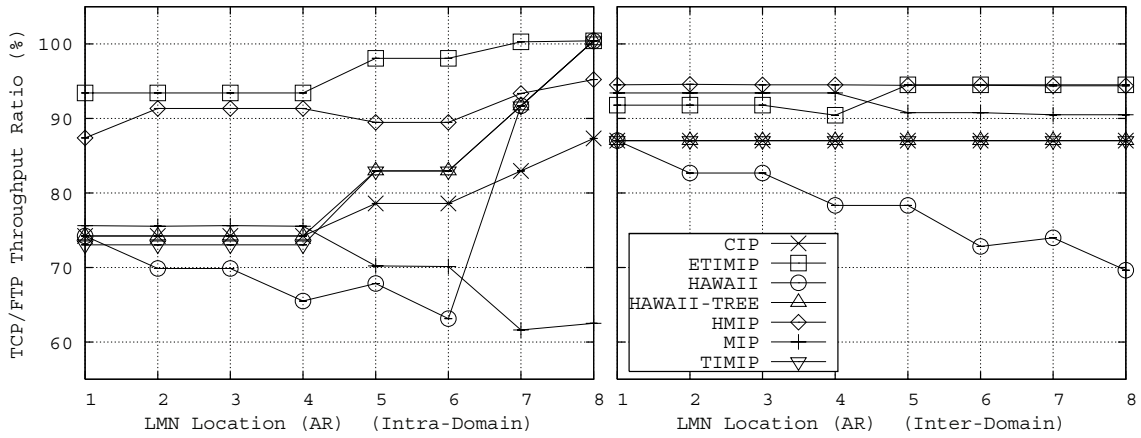


Figure 83: Throughput ratio with random link failures - TCP

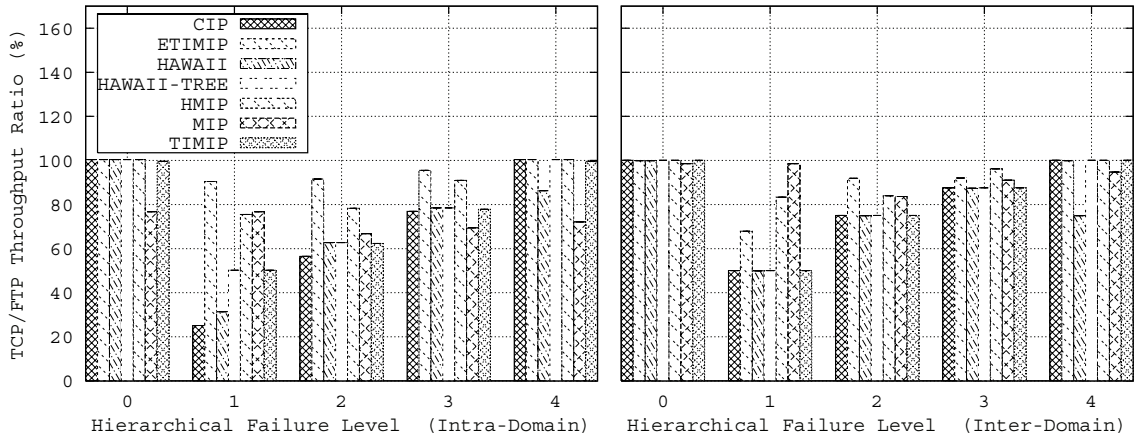


Figure 84: Throughput ratio with hierarchical link failures (0 – no failure / 4 – mesh links failure)

In all cases, the previous conclusions derived from both the previous UDP with link failures tests and the previous TCP tests are directly applicable: the protocols that feature tunnelling (eTIMIP / hMIP) are able to route packets through the alternative paths, and so they are reliable against link failures. Of those, better efficiency is achieved by eTIMIP for the lower RTT it imposes. The other classic protocols CIP and HAWAII suffer from the reliability problems discussed previously, being particularly penalized in the situations that involve high RTTs.

5.3.8 Link Failures Effect - Continuous movement

This section repeats the previous test that evaluates the mobility service reliability in the reference scenario (fast MN movements), for the particular case of TCP traffic.

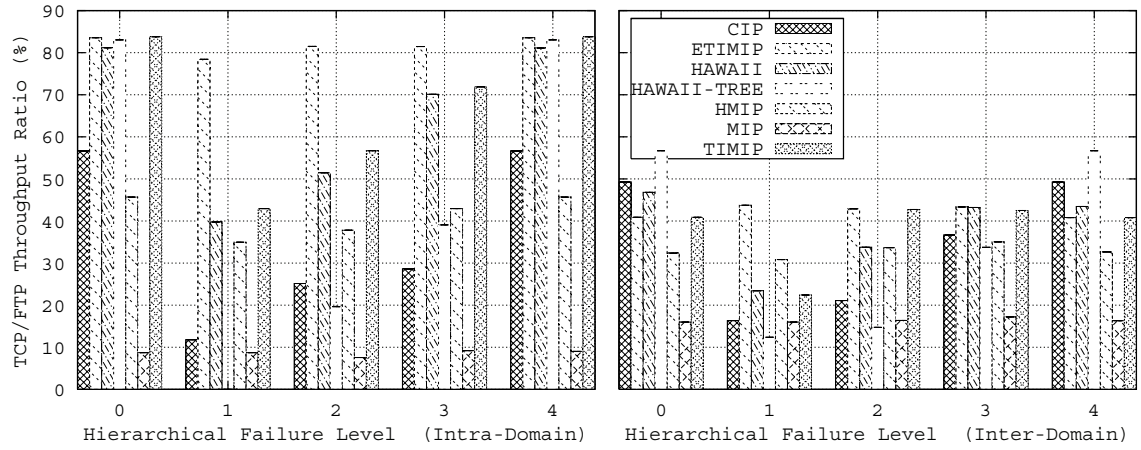


Figure 85: TCP throughput - Hierarchical link failures (0 – no failure / 4 – mesh links failure)

Here, eTIMIP has the clearest advantage of all protocols by supporting high reliability, low RTTs and low loss handover. All the other protocols fail to provide good performance, especially when the failures occur in the highest locations of the tree, either by not supporting reliability (HAWAII / CIP / original TIMIP), or by supporting it but not supporting lower RTTs and low loss handovers. As expected from the previous tests, in inter-domain scenarios the high RTT that all protocols present leads to major throughput degradation because none supports a zero-loss handover.

5.4 Basic eTIMIP Conclusion / Discussion

This section summarises the previous tests that compared the basic version of eTIMIP to other micro-mobility protocols and to the MIP macro-mobility protocol in high-speed scenarios. The basic routing mechanism of eTIMIP will be analysed regarding the major components of handover efficiency, routing efficiency, transparency, reliability and scalability support.

Handover Efficiency

On the pro side, the base eTIMIP specification presents a fairly simple distributed handover scheme that supports low latency in most of the cases and does not reorder the packet flows. This is achieved by quickly redirecting the in-flight packets to the new location, via a fast update of the close hierarchically distributed routing entries to the new MN location. As eTIMIP only uses a single update message, a low control load is achieved, which is also typically limited to the lower parts of the domain.

On the con side, the eTIMIP basic routing handover results in a small, but present, amount of lost in-flight packets that are incorrectly sent to the previous location during the handover. While having a small impact on UDP throughput, such small packet losses are the determining factor that leads to a large TCP throughput degradation, which is not higher only because of eTIMIP's low latency handover and low RTT support. Another problem concerning eTIMIP's basic handover is the worst-case behaviour in the middle of the tree, or when high transparency is desired by deploying fewer agent levels, as these situations can lead to a slower and lossier handover. Of the studied protocols, only HAWAII handles this problem efficiently.

Compared to other alternatives, eTIMIP's handover performance is as good as theirs in both UDP and TCP traffic types; however, both can be subject to a number of improvements, this necessity being more critical in the case of reliable TCP traffic.

Routing efficiency

On the pro side, the base eTIMIP specification also presents a fairly simple routing scheme that enables a tree-optimal forwarding service, where the packets always follow the most direct paths inside the tree. This directly enables a lower end-to-end delay, which is a key point for both UDP (benefiting real-time services) and TCP (enabling lower RTTs) traffic types. Additionally, this scheme also permits better resource utilization, especially in intra-domain traffic, which is kept in the lower parts of the domain and does not pass through the single GW.

On the con side, the eTIMIP basic routing handover is not able to achieve an even more efficient data routing service, due to the fact that it is not able to send the packets between the domain's agents directly. Such optimal routing scheme, which is only supported by the macro-mobility's MIP protocol, could be used to further lower the end-to-end delay, with the advantages previously pointed out. Another benefit would be to decouple the data and control paths, by enabling the bypass of data traffic through the single GW in inter-domain traffic scenarios, providing load balancing.

Compared to other alternatives, eTIMIP already provides the best routing performance for both UDP and TCP traffic types; however, it such routing can still be improved, which would greatly benefit both types of traffic.

Transparency support

On the pro side, the base eTIMIP specification presents a completely transparent service through the support of both Legacy Mobile Nodes and Legacy Routers. For this, the mobility service can be introduced in any pre-existing network domain while keeping the existing terminals, routers and topologies, just by deploying new eTIMIP agents adjacent to the terminals. However, eTIMIP can also introduce an efficiency-to-transparency trade-off possibility, as more agents may be added to the network to improve efficiency, through the deployment of an agent hierarchy that benefits fast handovers.

Regarding eTIMIP, no major downsides have been identified.

Compared to other alternatives, eTIMIP already provides the best round-up transparency of all protocols.

Reliability

On the pro side, the base eTIMIP specification presents simple reliability protection against wired link failures, using tunnelling mechanisms only when no direct paths between the agents exist. This simple mechanism, which is efficiently implemented in the IPv6 eTIMIP version, is sufficient to cater for the occurrence of wired link failures, by taking advantage of the fixed routing reconfiguration. While such events can be considered rare, the occurrence of a wired link failure can result in a major impact on the mobility service that can affect **all** terminals located in a certain AR, and thus is especially considered by eTIMIP.

On the con side, eTIMIP basic routing does not support a reliability mechanism for its own agent tree, as it is the case of the other efficient micro-mobility protocols. As such, the permanent failure or disconnection of a certain agent can have a high impact on the mobility service for the LMNs served by its sub-tree. Such problem can be minimized through the use

of a simpler or degenerate agent tree [34] [105] [25], which can impact on the handover efficiency of basic eTIMIP, or through the use of dynamic tree mechanisms like the ones proposed for the similar micro-mobility protocols [21] [55] or the ones featured in well-known networking equipment [172].

Compared to other alternatives, eTIMIP provides a reliability level in-between the other micro-mobility protocols - it is higher than the efficient protocols, but it is lower than the less efficient ones.

Scalability support

On the pro side, the base eTIMIP specification presents a scalable solution up to complete domains. As in the other micro-mobility solutions, this is done by hiding all frequent local handovers from the rest of the Internet, and by requiring only one routing entry per MN per tree level; as in other solutions, global scalable mobility is achieved using a complementary macro-mobility solution, being used the "surrogate MIP" (sMIP) extension for this task. Scalability is also aided by the limitation of internal traffic to the lower parts of the domain, where the data is forwarded by the GW only in the worst case for intra-domain traffic. Although it is not specifically measured in this simulation work, eTIMIP also presents an optimized state maintenance that requires no overhead for active terminals, and that incorporates a back-off in the explicit refresh mechanism.

On the con side, even though eTIMIP provides the mentioned state maintenance back-off mechanism, it does not provide a specific idle terminals support that removes state from the network, and it forces all inter-domain traffic to pass through the GW central point.

Compared to other alternatives, eTIMIP provides similar scalability to other alternatives, but which is capable of being improved by the support of idle terminals and load balancing.

6 Extended eTIMIP modules specification

This chapter presents a series of eTIMIP modules that complement the base protocol with mechanisms that complement it with additional efficiency, scalability, reliability and control gains. Using the proposed efficiency extensions, the resulting protocol is able to support seamless handovers and low mobility overhead, while maintaining the terminal and network independence already supported by the basic protocol.

Using the same architecture already defined for the base protocol, an extended secure mobility service is defined integrating all extensions (**full eTIMIP**) that, by using a route optimization and a seamless handover scheme on top of the overlay network, supports a mobility service with better transparency and efficiency than the best alternative solutions of the state of the art. Thus, full eTIMIP is able to improve efficiency without sacrificing full transparency, providing a flexible application of the mobility service in particular deployment scenarios.

In particular, eTIMIP efficiency features are an advance on traditional micro-mobility solutions: while these solutions seldom use the shortest paths to forward data packets, eTIMIP is able to do the same operation in a very efficient way, using a route optimization scheme. This feature enables the creation of direct paths inside the logical agent tree, which are used to directly route data packets to the final destination, in most situations. Used in conjunction with multiple mobility-aware points of attachment to the outside of the domain, this can offload the GW of data forwarding functions, providing load distribution.

On the other hand, even though the traditional micro-mobility solutions already feature mechanisms that can enable fast or smooth handovers, most of them are unable to **simultaneously** provide low latency and low loss handovers. Thus, eTIMIP advances the state of the art by introducing seamless handover procedures that support a zero drop / zero out-of-order handover, while minimizing the total handover latency. For the former, a low-loss handover is achieved by avoiding dropping packets at each L3 handover, and by avoiding disordering the flows when the routing entries are updated with the new LMN location. For the latter, a fast handover is maintained using a cross-layer detection support at the involved ARs and through a fast handover signalling to the physically adjacent AR neighbours.

In addition, the full protocol also improves scalability through specific idle terminals support, that extends the state maintenance operations of basic routing. Based on the LMN's current network usage, the nodes may be classified as active or idle. The former considers the LMNs that are actually sending and/or receiving data, in which the network will make all the effort to accurately track in order to provide a prompt service. The latter considers the LMNs that are not currently using the network, and thus can be subject to a minimal mobility service: for them, the network will perform a less exact location service, those being paged on demand by the network if and when there is new data for delivery.

Another original feature is the support of operator-centric scenarios, through the option of network-controlled handovers and a network-confirmed power-down. The first option enables the network to dynamically decide the best point of attachment of the LMNs to itself. This mechanism can be used to implement load-balancing capabilities, or being used with wireless technologies where the LMN can simultaneously access multiple APs belonging to different ARs. The second option increases the reliability of the terminal independent service for idle LMNs. This enables the confirmation, by part of the network, that the missing idle

LMNs have not passed unnoticed by either the generic detection algorithm or the cross-layer detection support, this being a necessity raised by the terminal independence feature of eTIMIP.

The rest of this chapter is organized as follows: the first section will contain the specification of the route optimization extension; the second section focuses on the seamless handovers extension; the third section will cover the idle terminals support; and the fourth section will focus on the operator-centric support.

6.1 Optimized eTIMIP routing operations

6.1.1 General Overview

This section presents the algorithms and operations needed for optimized eTIMIP's routing. This optional routing scheme is able to route data packets more efficiently inside the domain by being able to forward the data packets directly to their destinations instead of being restricted to following the tree paths as in basic routing. This enables the skipping of agents in the tree, which is a desirable feature in complex topologies, where extra links and routers are present between the agents.

The optimized routing is characterized by the distribution of additional soft-state routing entries per LMN, named optimized entries, which contain the last exact LMN location (e.g. its AR), in the regular update packets managed through basic routing. These entries will then be used to forward data packets directly to the LMN's AR, or, in transitory situations, to its vicinity. Besides of containing the exact LMN location, the optimized routing entries also contrast with the basic ones by being refreshed each time they are used, and not requiring explicit revoking of the outdated ones. This is made possible because eTIMIP consistence operations assure that the combination of basic and optimized entries will always point to the latest location of the terminals.

Besides the TRs covered in the basic update procedures (e.g. power-up and handovers), some TRs can also be dynamically chosen to be updated with the new information in order to improve efficiency. These dissemination methods tend to push the optimized entries to the network boundaries, which in turn tend to free the internal TR agents from data forwarding responsibilities, thus performing mobility management procedures only. Such data from control paths decoupling phenomena can increase scalability of the overall solution by balancing most of the load caused by data packet forwarding responsibilities off in the internal core TRs like the GW.

These route optimization mechanisms are totally optional and are not required for the correctness of the mobility service. The usage of these optimization mechanisms may be scheduled by the network's routers according to network policies. These policies should follow the IETF recommendations of route optimization triggering by the MNs, particularly the maximum signalling rates between a pair of nodes [2]. Alternatively, custom operator-defined metrics may be used for triggering optimization mechanisms, based on traffic utilization, MNs' movement patterns, and triangulation width, among others. The specification of such policies falls outside the scope of this thesis.

6.1.2 Registration phase

6.1.2.1 RO Distributed Update Procedure

General description

During the Registration phase, the previously described update packets are extended with an RO flag and an additional field containing the LMN's AR information. However, the resulting packet is processed using exactly the same operations as the previously described ones for basic routing, namely by being transferred agent-by-agent inside the tree and being subject to the same reliability and clock synchronization procedures.

When an agent receives an eTIMIP update message with RO information, the message is firstly processed according to the basic routing procedures. To have a coherent view of the network, the updating of basic information cancels any corresponding optimized entries that may be present in the TR. After that, the eTIMIP agent verifies the RO flag and, if it is available, it adds a corresponding optimized routing entry to the routing table with the LMN's AR information.

It should be noted that according to these rules, the previous RO information at the previous LMN locations is always either removed or updated with the new RO information. This mechanism ensures routing consistency as the chain of outdated RO entries coupled with updated basic or RO entries will always point to the correct LMN AR.

Application Example: Power-Up RO Operation

Figure 86 depicts the Power-Up operation with the RO option. This operation performs the same steps as the corresponding basic power-up (Figure 17), as the update messages continue to be propagated using the basic routing entries only. However, each step will additionally create or update the optimized entries at the agents it passes by with the AR information for this LMN (AR1). Thus, the initial power-up creates optimized entries at all agents up to the GW concerning the first LMN location (e.g. agent AR1).

In the figure, such optimized entries are identified at the routing tables with an adjacent arrow ("→"); the default fallback entries continue to be represented by a dash ("-"); and the AR's wireless interface is represented by "ITF"

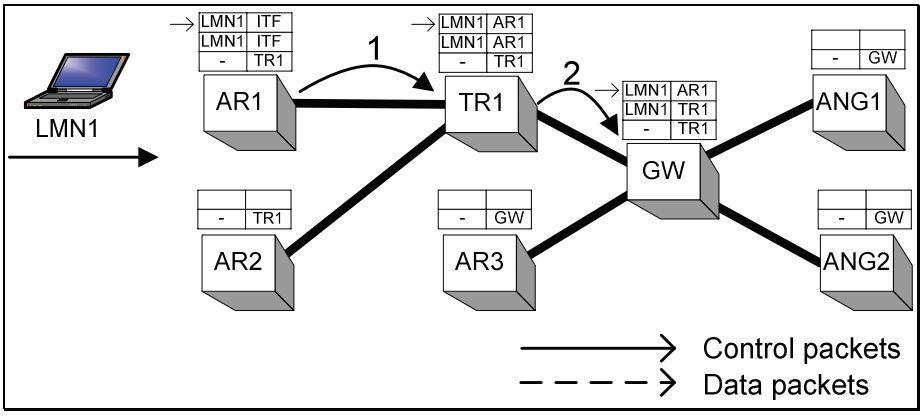


Figure 86: Optimized eTIMIP routing registration: Power-Up

Application Example: Handover RO Operation

Figure 87 depicts the handover operation with the RO option. As before (e.g., basic handover, Figure 18), this operation limits the signalling messages to the local sub-tree of the in-

volved AR agents, updating both basic and RO entries in it. In this case, an RO entry is created at AR1 pointing directly to AR2.

As this signalling is limited to the local sub-tree to provide low-latency handovers, there are agents that are not updated with the most current LMN location, like the GW which maintains its RO entry still pointing to AR1. However, registration consistency is always maintained, by the reasons explained previously.

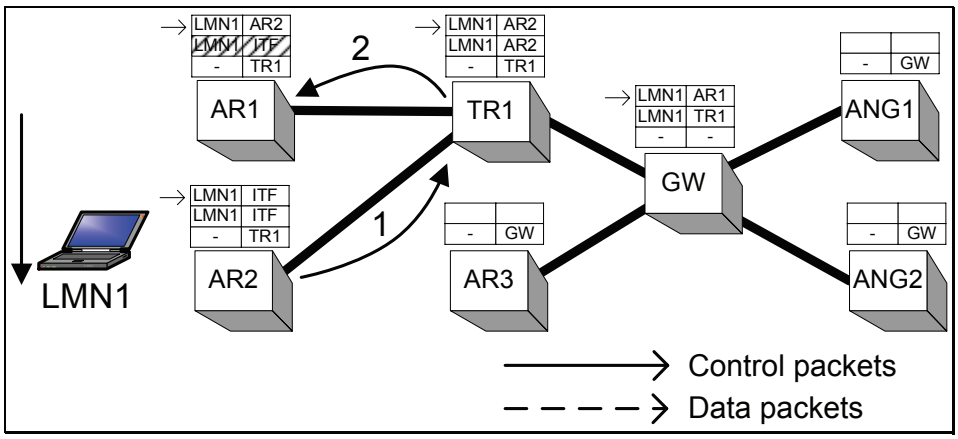


Figure 87: Optimized eTIMIP routing registration: Handover reconfiguration

Formal specification

The RO update procedures are formally described in the state machine of Figure 88, which all TRs execute in the registration phase. This formal definition expands the corresponding state machine of Figure 23 with the RO update-related mechanisms. All used functions are described in Appendix B.

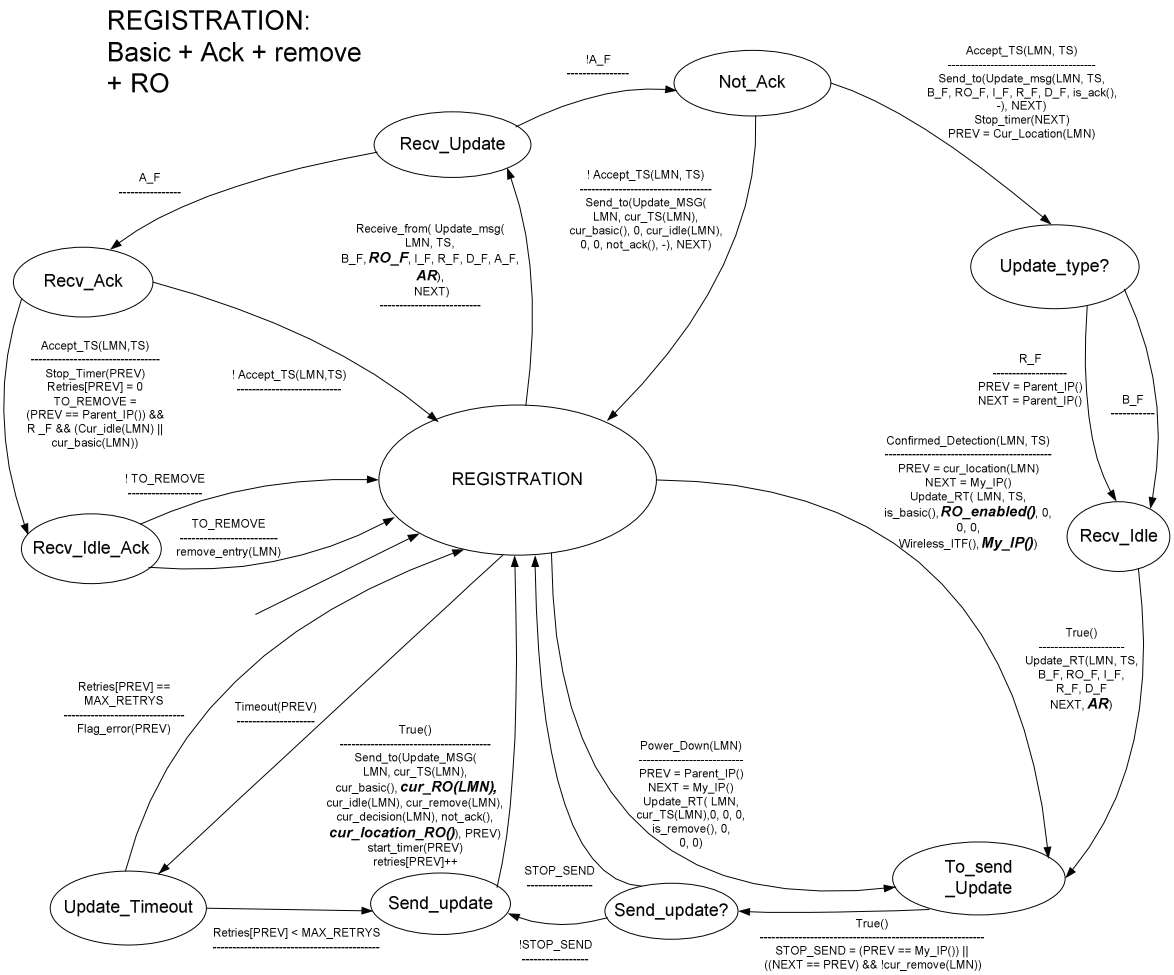


Figure 88: RO distributed update procedure formal specification

6.1.3 Execution phase

The optimized execution phase extends the basic execution phase with the ability to use the optimized routing entries, which take precedence over the existing basic entries in the forwarding process. Such entries can then be used to skip agents inside the tree by forwarding the data packets either directly to the final location of the terminal or, in transitory situations, to locations that are close to the real LMN's location, being then re-routed to the correct destination. Such is possible because the registration phase guarantees that the **combined** chain of routing entries, of either optimized or basic nature, always points to the most current LMN location.

6.1.3.1 RO data forwarding and RO entries refreshment

General description

When an eTIMIP agent receives a data packet, its destination is checked against the eTIMIP agent's routing table. If the node has an RO entry from the destination, the packet is forwarded to the next eTIMIP agent defined in that entry; when no RO entry is available, the packet is forwarded as previously described.

The RO entries are only refreshed if they are regularly used to forward data packets to a certain LMN. Therefore, inactive optimized entries are simply discarded, without any signal-

ling messages refreshment, reverting incoming packets to basic eTIMIP routing. This is possible because optimized routing entries are built on top of basic routing registration, which is itself a reliable process, and because the creation of a new basic entry in an agent invalidates any existing optimized entries for the LMN.

Application example: Intra-domain routing

Regarding the case of intra-domain traffic in the situation of Figure 89, the packets sent by the CN are first delivered to AR1 (step 1), which has an optimized entry point to the LMN's location on the domain (AR2). Thus, this routing entry can be used to route the packet directly to AR2 (step 2), instead of forwarding up the tree to TR1, as in the basic routing, which is able to use any direct links in the physical network between the two AR agents. Once in the AR2, the packet is simply delivered to the LMN, as usual.

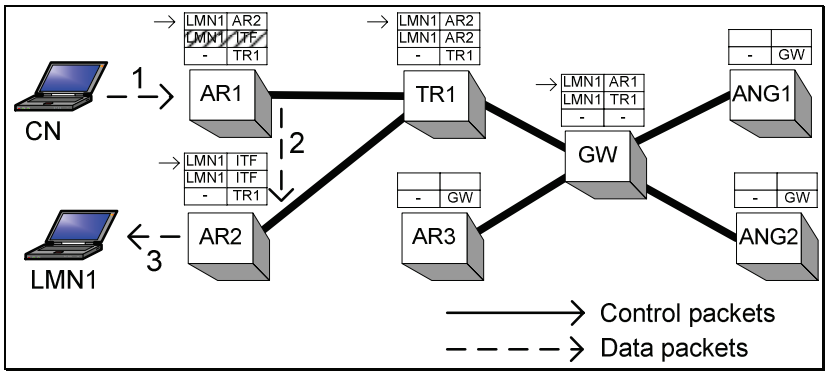


Figure 89: Optimized eTIMIP routing execution: Intra-Domain routing

Application example: Inter-domain routing

Regarding the case of inter-domain traffic, illustrated in Figure 90, packets arriving at the domain are forwarded by the initial ANG to the GW, as usual (steps 1 and 2). At this point, the outdated entry at the GW is used to route the packet directly to AR1 (step 3). At this node, the rest of the routing is done as in the previous case, using the most recent optimized routing information which points to AR2 (steps 4 and 5).

It should be noticed that this case illustrates the eTIMIP optimized routing consistency in **transitory** situations only, where the combined chain of optimized and basic entries guarantee routing consistency with a low-latency handover. Using the dissemination mechanisms that will be described in the next section, this local triangulation phenomenon will be removed through the notification of the necessary border nodes of the most current LMN location. This will enable direct forwarding of the data packets to the exact LMN location inside the network.

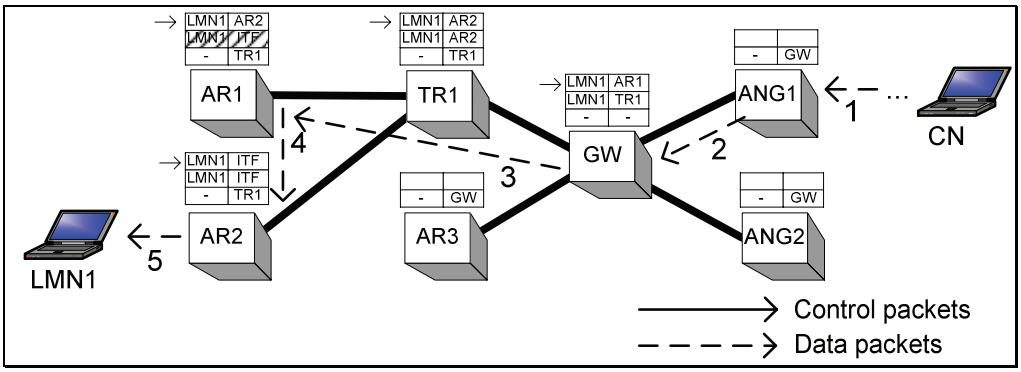


Figure 90: Optimized eTIMIP routing execution: Inter-Domain routing

Formal specification

The RO forwarding and refreshing procedures are formally described in the state machine of Figure 91, which all TRs execute in the execution phase. This formal definition expands the corresponding state machine of Figure 41, with the RO forwarding and refreshing-related mechanisms. All used functions are described in Appendix B, being the used variables described in Appendix C.

Of these, the most important added one is the LT_RO variable that holds the lifetime of the RO entry, and the usage of the cur_location_RO() to find the next agent to forward the data packet to, which uses the RO entry if available.

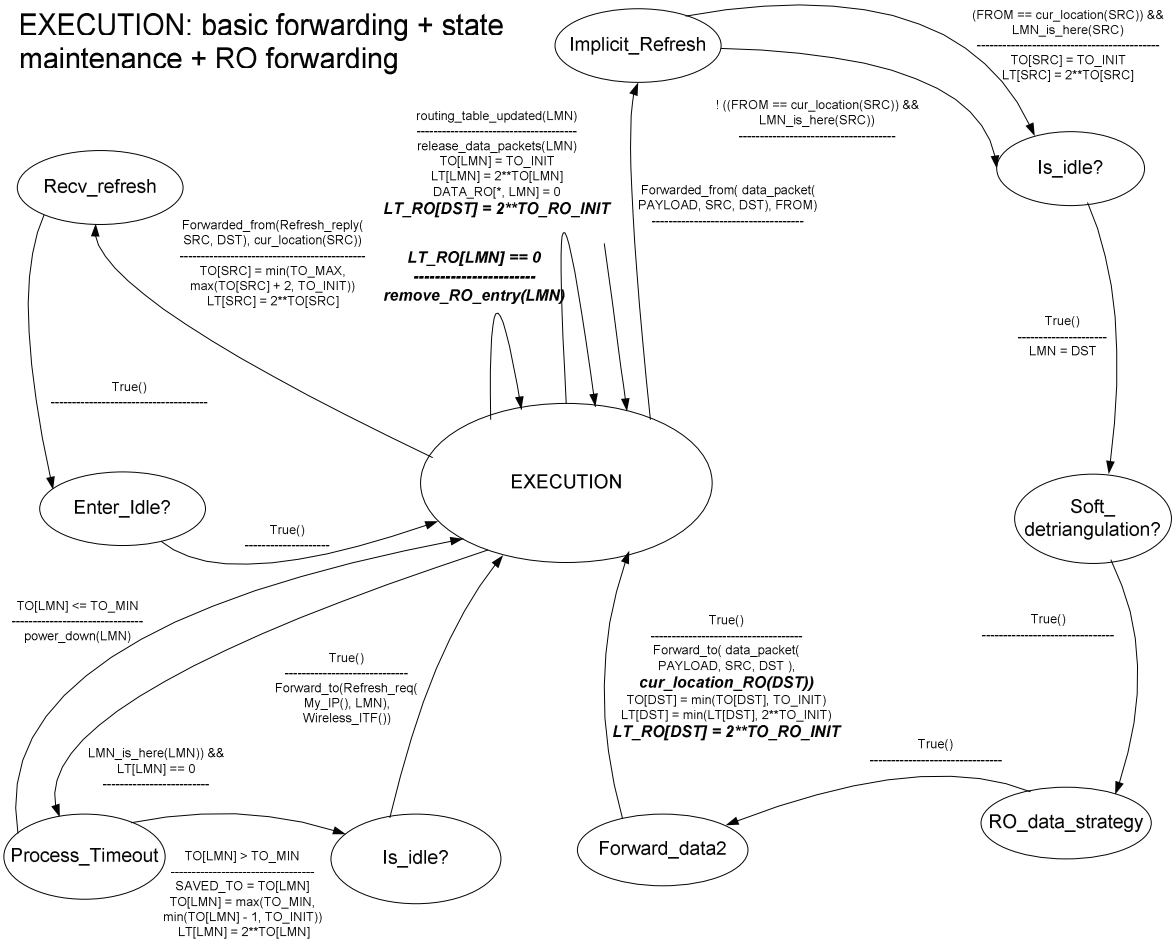


Figure 91: RO data forwarding and RO entries refreshment formal specification

6.1.3.2 Dissemination of optimized entries triggered by data packets

General description

The dissemination of optimized entries will incrementally disseminate the optimized entries on the network, by observing the data packets that are non-optimally routed inside the network.

When an agent forwards a data packet to a certain LMN using an optimized entry, it can deduce that the previous agent didn't have the latest optimized routing information – if it did, then it would have sent the data packet to the final TR directly. Using this information, the current agent can start an optimization update operation directed to the previous one, notifying it where it should send the LMN's packets directly. This triggering process can be sub-

ject to any number of operator-defined rules that define the specific conditions that start and end the dissemination triggering, in order to achieve a flexible trade-off between signalling overhead and data forwarding benefits. In particular, configured minimum and maximum values for either rate, number or type of misrouted packets can be used to trigger the dissemination operation.

This process produces an RO update packet, named “RO_inform”, which, unlike the previous basic or idle update messages, can be swapped between non-adjacent tree agents, and may be non-reliable. The message is marked with an RO flag and the AR information is associated with a LMN, enabling the receiver to deliver the next packets to the most current LMN location. As this update message lacks a “basic” flag, it is unable to modify existing basic entries, which enforces consistency managed in the registration phase. Also, the message contains the LMN’s detection timestamp, which is used to solve temporary inconsistencies caused by race conditions or lost control packets, where outdated entries are modified by a subsequent newer RO update.

When used simultaneously, the RO dissemination methods coupled with the RO entries state maintenance tend to restrict the **active** optimized entries to the border pairs that deal with the active data flows (ANG↔AR). This is a major step to remove additional state in internal eTIMIP agents of the network, namely the GW, TRs and the previous ARs, which also helps scalability when using the RO option. In particular, when using ANGs nodes, this feature can offload the single GW agent of data forwarding functions, which decouples the data from the control paths, as the GW tends to perform control functions only.

Application example: RO dissemination mechanism

The specific conditions that trigger the RO dissemination are illustrated in Figure 92. The initial state of the network is the same as the one in Figure 90, which suffers from the local triangulation problem previously pointed. When a data packet is received by AR1, this agent can start the optimized update packet process towards the source of the data packet (GW) (trigger “T”).

This update RO message will then modify the outdated RO entry at the GW with the latest AR information for this LMN (AR2). This causes future packets to be directly forwarded to the AR2. After some time, the intermediate RO entry at AR1 is dropped after an RO timeout. This is due to the fact that the RO entry stops being refreshed by not being used to forward LMN data traffic.

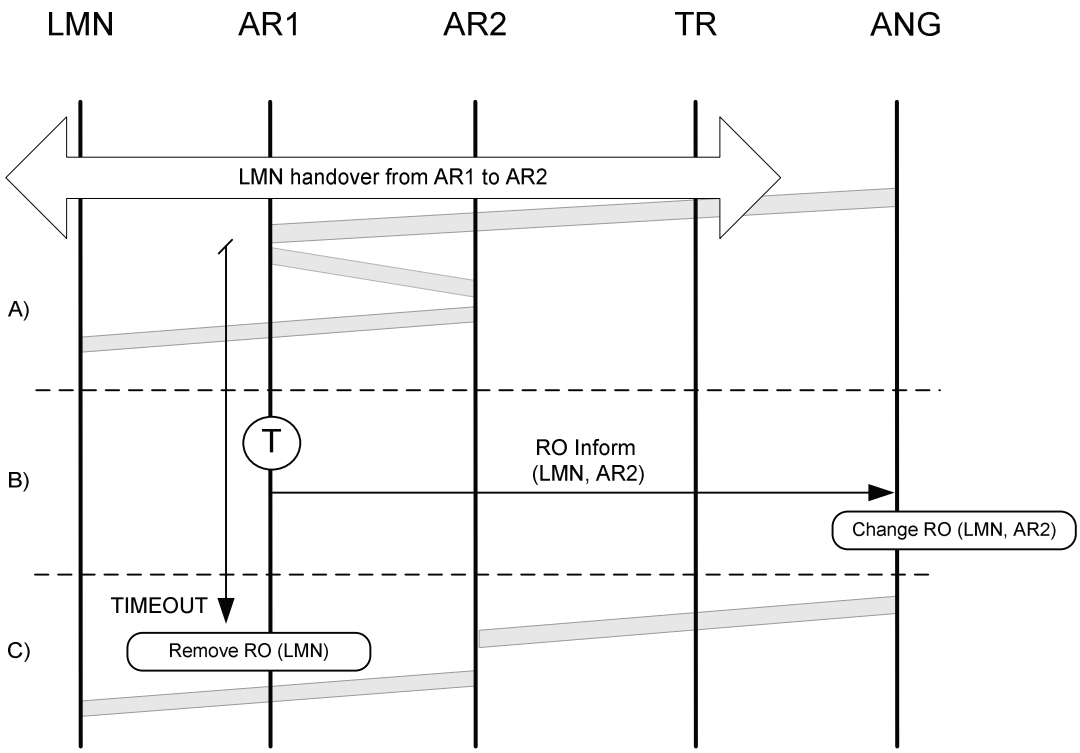


Figure 92: RO dissemination mechanism after handover

Application example: Control and Data Paths Decoupling

The combination of the RO dissemination and state maintenance operations can result in situations that are similar to the ones illustrated in Figure 93; while the dissemination mechanisms propagate optimized entries to the border of the network (node ANG1), state maintenance removes unused entries at core agents (TR1, GW) and previous LMN locations (AR1). Thus, the ANG1→AR2 direct logical link can avoid the router that contains the GW agent, freeing it from data forwarding functions.

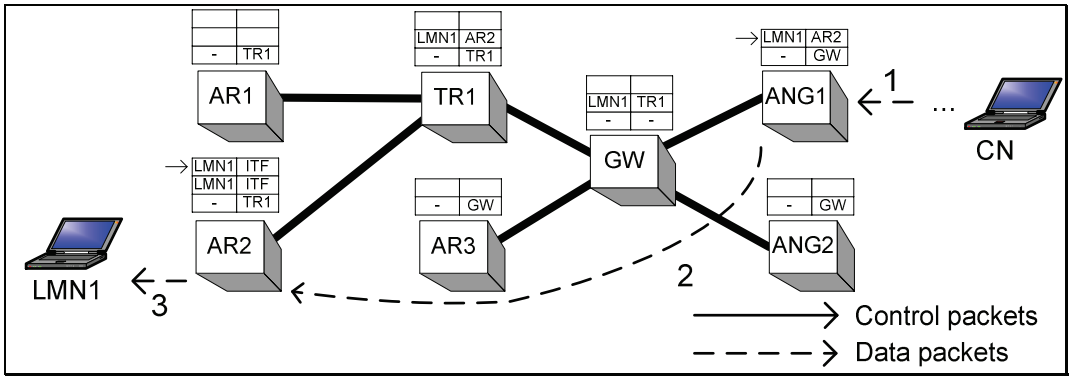


Figure 93: Combination of RO dissemination mechanism and state maintenance

Formal specification

The data dissemination procedures are formally described in the state machine of Figure 94, which all TRs execute in the execution phase. This formal definition expands the corresponding state machine of Figure 91 with the data triggering-related mechanisms. All used functions are described in Appendix B.

The triggered RO update mechanisms are formally described in the state machine of Figure 95, which all TRs execute in the execution phase, which details the RO information packets generated by these mechanisms. All used functions are described in Appendix B.

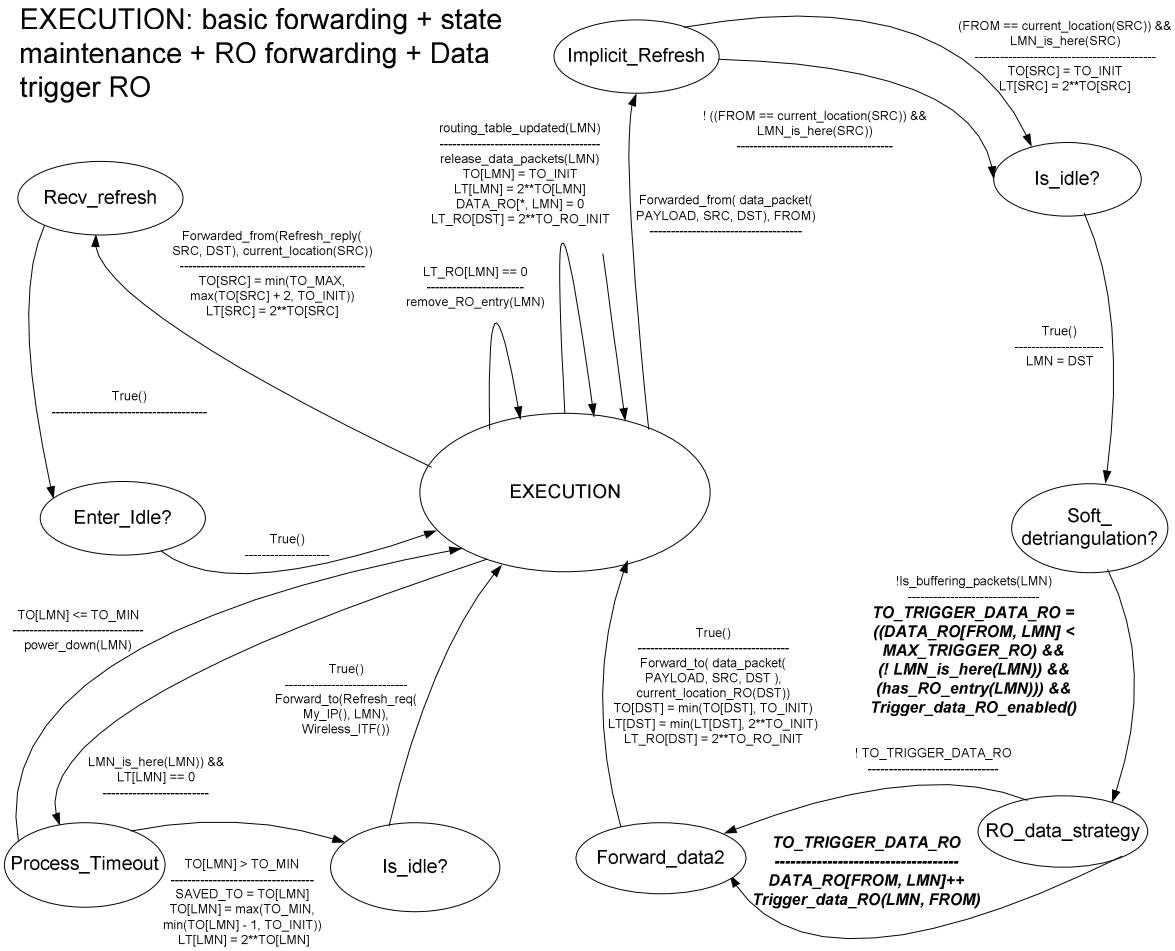


Figure 94: RO data dissemination triggering formal specification

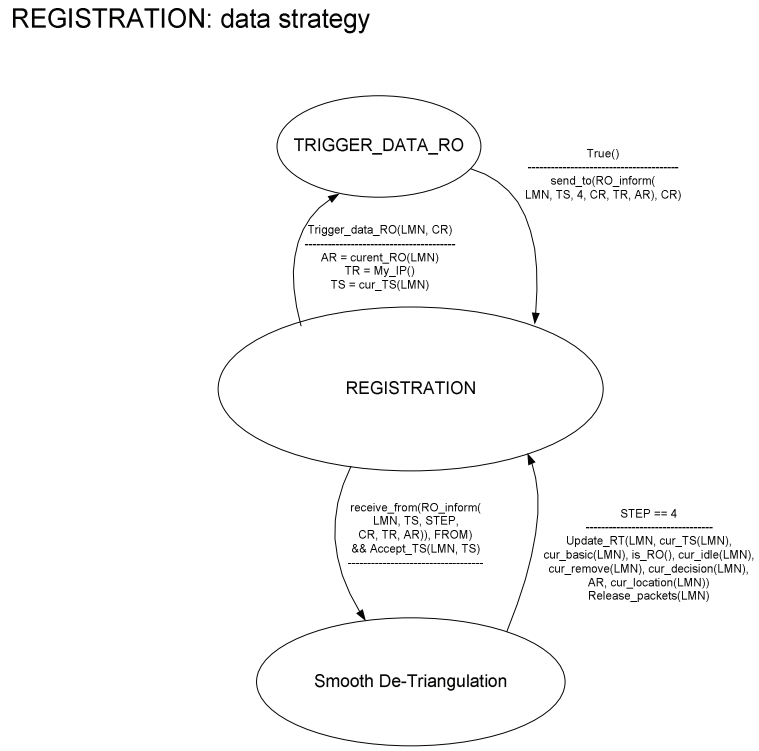


Figure 95: RO data dissemination update procedures formal specification

6.2 Seamless Handovers Support

6.2.1 General Overview

This section presents the proposed algorithms and operations that enable seamless handovers for the eTIMIP protocol, by minimizing both the handover latency and the handover packet losses that are incurred by the L3 handovers.

Concerning handover latency, even though this metric is already partially minimized by basic routing (as signalling is limited to the local sub-tree of the involved ARs, the authentication procedures are performed locally at the AR, and the LMN uses the same IP address through the domain), further latency improvements can be achieved with extensions to the detection phase, using L2 triggers that hint the LMN's arrival movements at the new AR, and by extending the registration phase with a fast handover signalling operation, where the new AR notifies its adjacent neighbours of the LMN movement, in parallel of the regular tree-based registration process.

Concerning handover packet loss, the previously described handover operations do not take any special care in the minimization of the losses incurred by the handovers. Such losses are divided into packets that are not received by the LMN (drops), and packets that are actually received but are out-of-order packets. The drops are avoided through the buffering of in-flight packets at the previous AR, until this node learns the new LMN location; this also requires an L2 trigger that hints the LMN's departure movements at the previous AR. The out-of-order packets are avoided by a smooth de-triangulation algorithm, which buffers the packets in the crossover node, to prevent them from arriving earlier than other in-transit packets.

Application example: Latency and Losses overview of Basic Routing' Handover

Figure 96 illustrates the above-mentioned problems related to latency and losses in the case of basic handovers. For this, the diagram simultaneously shows the control packets which instantiate the basic handover and their effect on the data flows destined to the LMN.

Firstly, when the LMN performs its L2 movement, all in-flight packets sent to the previous location (AR1) will be dropped. After the movement is detected using the heterogeneous GDA mechanism (step A), the new AR will update the parent node concerning the new LMN location, and the previous entry at AR1 will be removed (step B). As the crossover node is the parent node, this is sufficient to redirect the established flows to the new location, ending the effect of the handover.

This figure hints that the eTIMIP basic handover enables a low-latency handover, at the expense of dropped in-flight packets, until the crossover node is updated with the new LMN location.

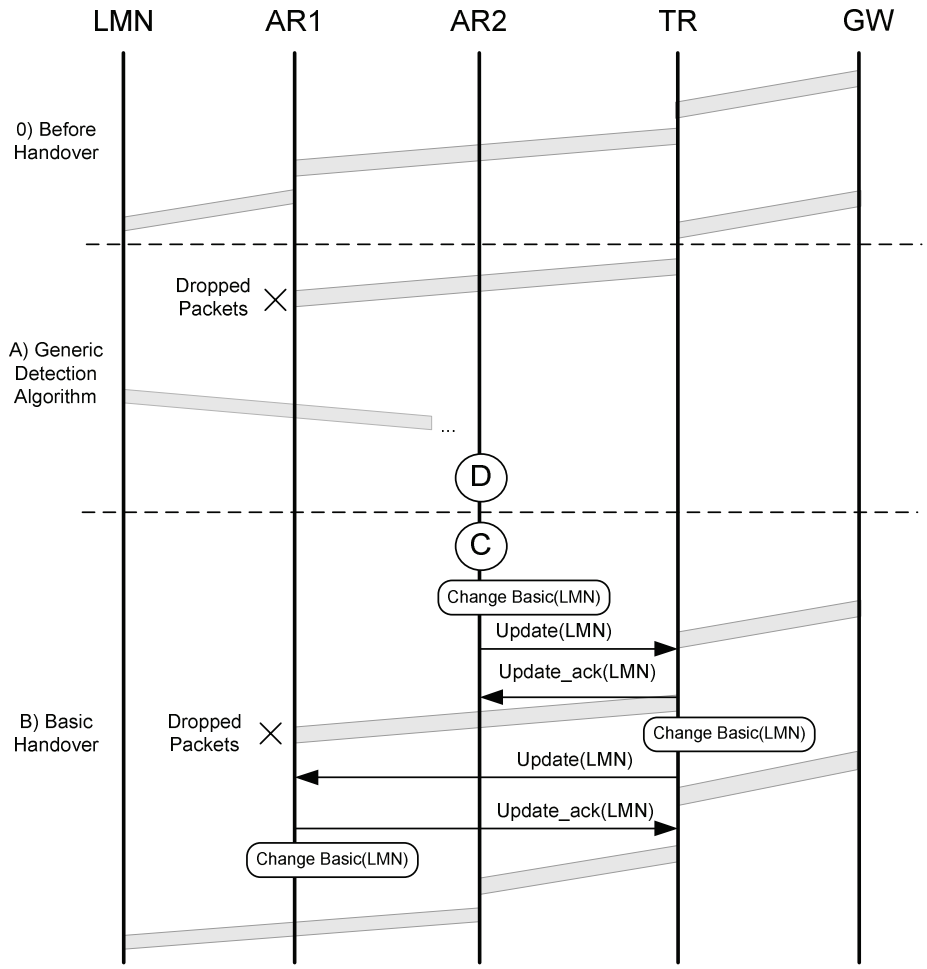


Figure 96: Basic eTIMIP handover analysis

Application example: Latency and Losses overview of RO Routing' Handover

Figure 97 presents the corresponding time diagram for the regular RO handover operation. Firstly, the same packet drop phenomena will occur, but will be more pronounced as the packets will be dropped until the previous AR is updated with the new LMN location. As this is done through the tree, this mechanism also incurs in higher handover latency (step A).

Then, the subsequent in-flight packets are tunnelled directly to the new AR, which avoids further packet drops. As previously described, this local triangulation permanently increases end-to-end delay and network resources; thus, it is made temporary using a direct de-triangulation operation at the ANG, which updates its RO entry to point to the new AR. However, this direct update can cause out-of-order packets due to the fact that the packets sent directly can be received before than the in-flight packets sent through the longer triangulated path. This is exemplified by the different delays imposed to data packet 1, which is sent through the long path (ANG→AR1→AR2), and data packet 2, which is sent through the direct path (ANG→AR2).

This figure hints that even tough the RO option introduces the benefits previously described (namely lower end-to-end delay, better resource utilization, load balancing and data paths decoupling), it has a lower handover efficiency than the corresponding basic handover. Besides the problems already present in basic handover, both handover latency and losses are further degraded - the former, by having the crossover node at the previous AR, and handling the notification through the tree; the latter, by introducing out-of-order packets in the de-triangulation operation.

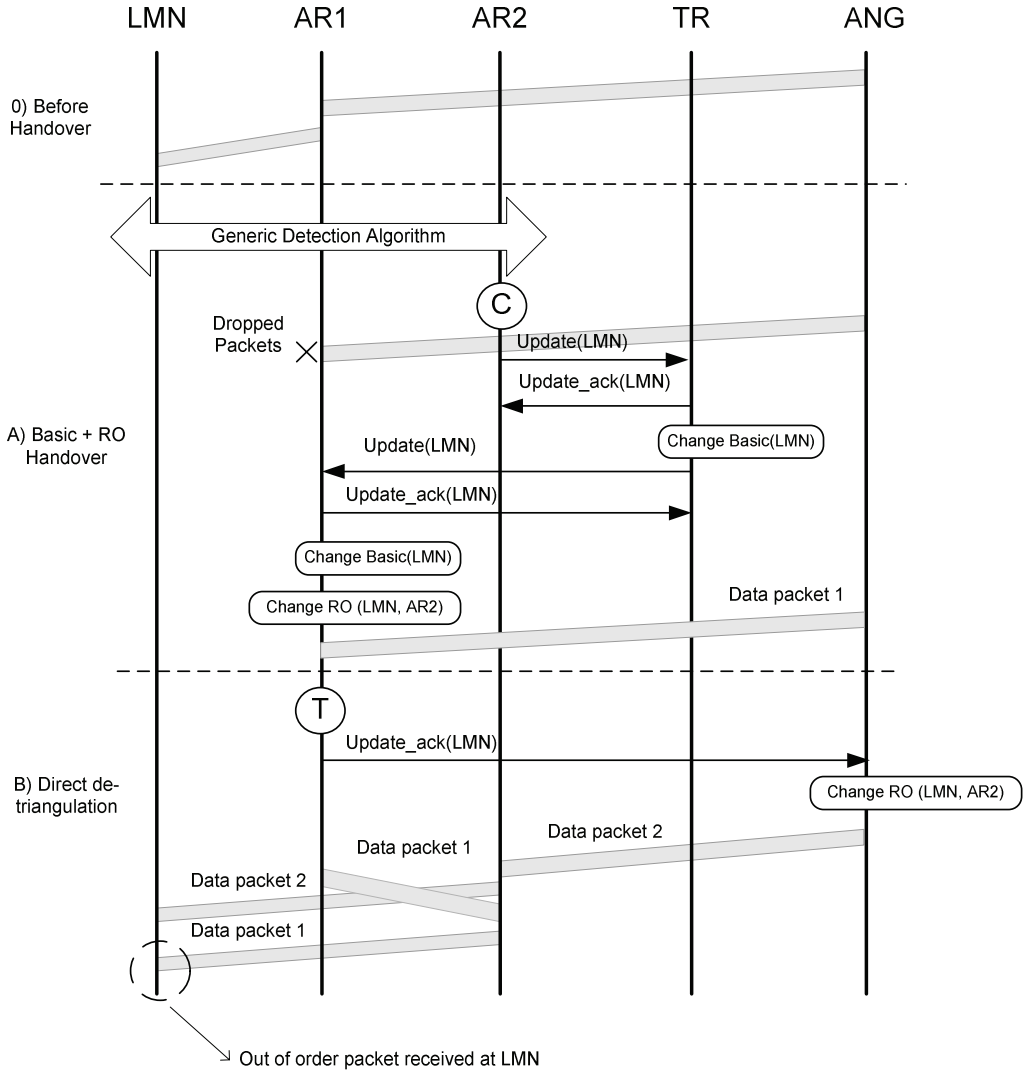


Figure 97: RO eTIMIP handover analysis, using direct de-triangulation

6.2.2 Detection phase

6.2.2.1 Cross Layer detection at the new AR for low-latency handovers

General description

To improve the detection efficiency, a technology-dependent mechanism is provided using cross-layer information that performs a faster localization of the LMN. For this, the L3 can receive L2 information from other layers or from the management plane in order for the new AR to be notified immediately after the LMN's attachment to it.

There are several ways of supporting such interaction, which are dependent on the existing network elements and their capabilities. The most efficient solution is provided when the LMN connects to an L3AP, which is able to collect and expose the L2 events that signal the LMN's arrival. In the case of the 802.11 infra-structured mode, this information can be derived from the L2 association message which is received at the AP when a LMN connects to it. In other technologies, other types of messages must be used, namely the 802.16e association message for WiMAX [126], or the packet data protocol (PDP) context activation process for UMTS [127]. This procedure can also be derived from the heterogeneous Layer-2 technology Media Independent Handovers (MIH) [74] [75]. If the terminals are identified by their L2 addresses, these are required to be mapped to their corresponding IP address, for instance by inverse neighbour discovery procedures [124] or by consulting distributed MAC-to-IP databases derived from DHCP procedures.

If this local L2 information is not available, which is the case of L2AP or AC, this interaction may be performed through standard remote management procedures operating in standard or proprietary Management Information Bases (MIB). The most efficient option is the use of asynchronous procedures based on alarms, or, alternatively, periodical queries to MIBs [38]. Currently, at least one vendor-provided MIB provides suitable information for querying via SNMP (table prAssociatedClientsTable of [117]).

Other options is to derive this information from existing mechanisms that implement such triggering paradigm, namely "Detecting Network Attachment" [151], Dynamic Host Configuration Protocol (DHCP) [155], fMIP [34], or MIP's low latency handovers Layer 2 triggers [23] [35]. In this latter case, the proposed "target trigger" is an L2 abstraction mechanism that notifies the AR's L3 about the arrival of the MN, which is already identified by its IP address instead of its L2 address.

It should be stressed that all these optimization handover hints might be unreliable and/or unsafe; the reliability problem is solved through the usage, possibly in a latter stage, of the generic detection algorithm; the security problem does not represent a security risk to the mobility service, as all handover hints are subject to the secure local authentication procedures previously described.

Application example: 802.11 association in a L3AP

Figure 98 shows the use of the 802.11 association message in an L3AP for triggering the detection process. When the LMN attaches to the AR, an 802.11 L2 Association Message is sent by the LMN; when the AR receives it, at the L2, it transfers it to the L3 and triggers the "D" primitive, signalling the LMN attachment to this AR.

Similar procedures may be envisaged for WiMAX or UTMS access technologies using their specific messages that signal the attachment to the network.

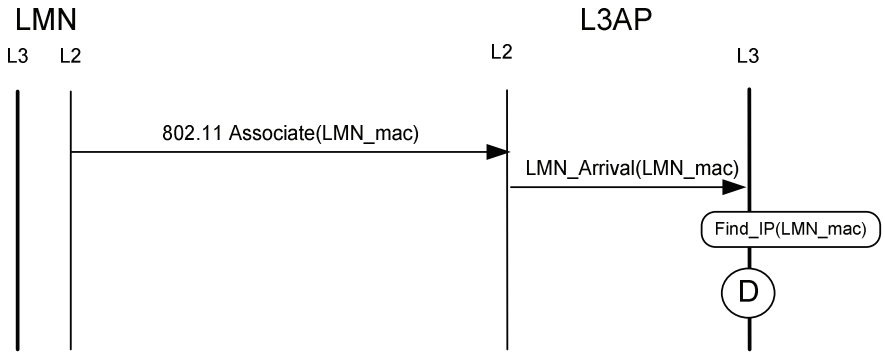


Figure 98: 802.11 Association in a L3AP

Application example: Asynchronous SNMP messages in L2AP

Figure 99 shows the use of asynchronous management procedures to accomplish the same results, when the LMN attaches to an AP (L2) of 802.11 technology that is managed by SNMP. In this case, when the 802.11 associate message is received, an SNMP Trap is issued by the SNMP agent, located in the AP, and sent to the manager, located in the AR. Upon reception of this trap, actions that are similar to the previous case are used to trigger the detection process.

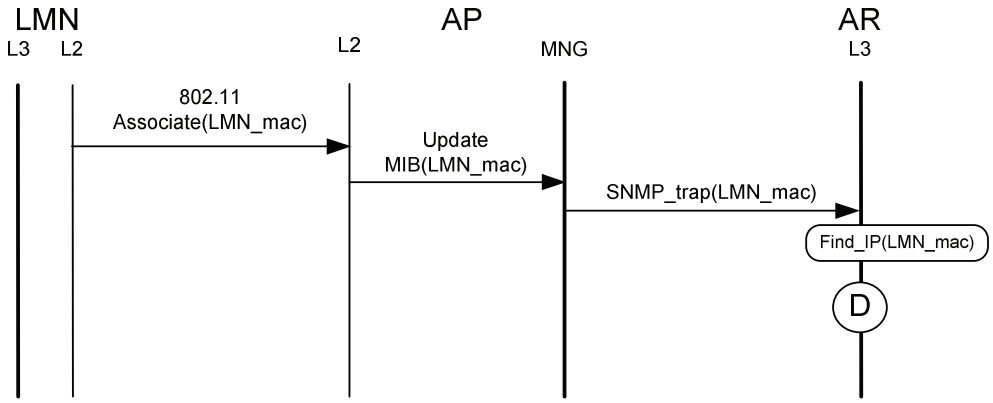


Figure 99: Asynchronous management procedures detection

Application example: Synchronous SNMP queries for L2AP

Figure 100 shows the complementary case, when no alarm is generated. The SNMP manager, located in the AR, must periodically poll the agents, located in the APs, to inform it about the LMNs that are associated to each one of them. Thus, the manager issues an SNMP-get message and the agent answers with an SNMP response message, containing the list of LMNs attached to it.

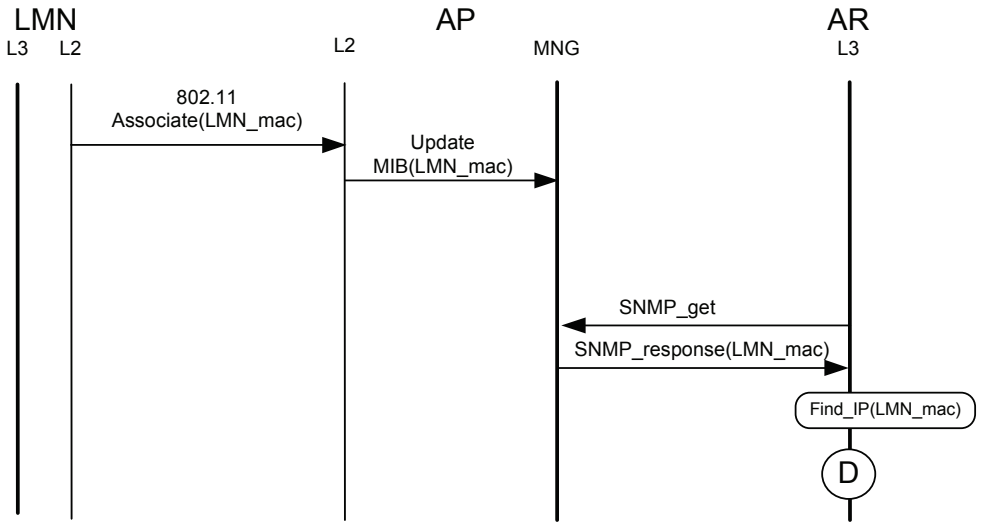


Figure 100: Synchronous management procedures detection

Application example: Cross layer detection using 802.21

Figure 102 shows the use of the Media Independent Handover framework of 802.21 to trigger the cross layer detection. When the LMN attaches to the AR, an 802.21 "link up" event message is triggered to the MIH component in the LMN, and it sends an Handover complete message to the MIH component in the AR; when the AR receives it, it transfers this trigger to the L3 and triggers the "D" primitive, signalling the LMN attachment to this AR.

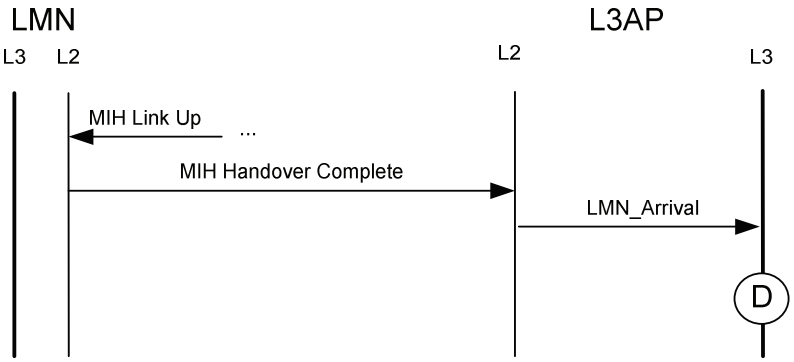


Figure 101: Cross layer detection using MIH 802.21

6.2.2.2 Cross Layer detection at the old AR for low-loss handovers

General description

To prevent packet drops, a technology dependent mechanism is provided using cross-layer information which notifies the previous AR that the LMN has stopped being attached to it. This operation is the complementary of the previous detection trigger, and thus it can essentially use the same methods.

Again, the most efficient solution is provided when the LMN disconnects from an L3AP which is able to locally expose the L2 information that signals the LMN's departure. In the case of the 802.11 infra-structured mode, this information can be derived from the L2 disassociation message, which is received at the AP when a LMN disconnects from it.

Application example: 802.11 dissociation in an L3AP

Figure 102 shows the use of 802.11 dissociation in an L3AP for triggering the detection process. When the LMN attaches to the AR, an 802.11 L2 dissociation Message is sent by the LMN; when the AR receives it, at the L2, it transfers it to the L3 and triggers a primitive that signals the LMN departure from this AR, named "U".

Similar procedures may be envisaged for WiMAX or UTMS access technologies using their specific messages that signal the attachment to the network, or the heterogeneous 802.21 MIH L2 functions.

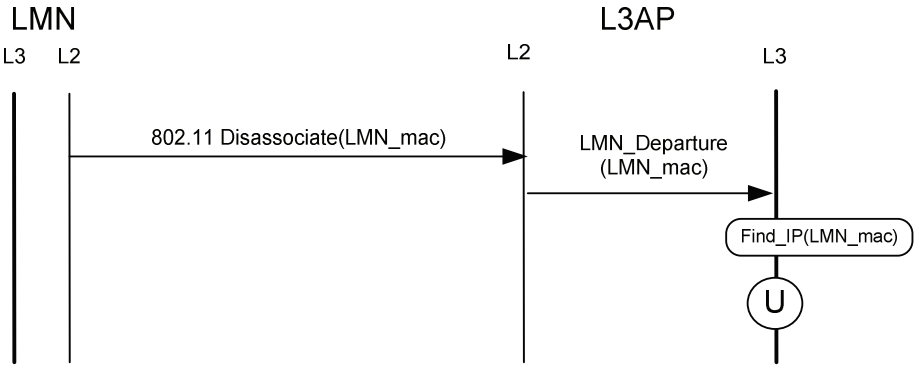


Figure 102: 802.11 Dissociation in an L3AP

6.2.3 Execution Phase

6.2.3.1 Packet Buffering

General description

This section specifies extensions to the basic routing's execution phase with methods that avoid the packet drop at the previous AR during a handover. When the previous AR node receives the departure trigger from the detection phase, it starts buffering all subsequently received packets of this LMN. This buffering operation is terminated when the AR learns the new LMN location, via the regular handover operations; in this situation, the buffered packets are released and routed normally to the LMN. Alternatively, selected packets are released if the buffer is full, to make room for more packets or after a timeout if no update message was received in the meantime. The decision of which packets to release can follow the QoS-aware recommendations defined in [52].

Application example: Low loss handover using buffering at the old AR

Figure 103 shows an example of how the packet buffering mechanisms are used at the old AR (AR1) to prevent packet losses during a basic handover operation. When the departure primitive (depicted as "U") is triggered at the old AR, the next received packets by the LMN are buffered instead of being sent to the wireless medium, where those would be dropped. When the handover operations update the routing table, the data packets are released and sent to the LMN directly using the established local tunnel (RO extension).

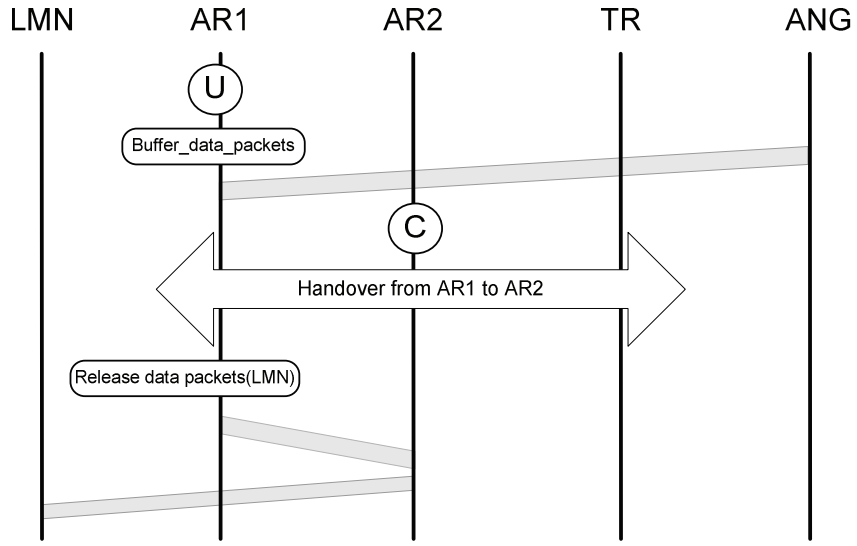


Figure 103: buffering of data packets at the previous AR

Formal specification

The packet buffering mechanisms are formally described in the state machine of Figure 104. This formal definition expands the corresponding state machine of Figure 94, with the packet buffering-related mechanisms. All used functions are described in Appendix B.

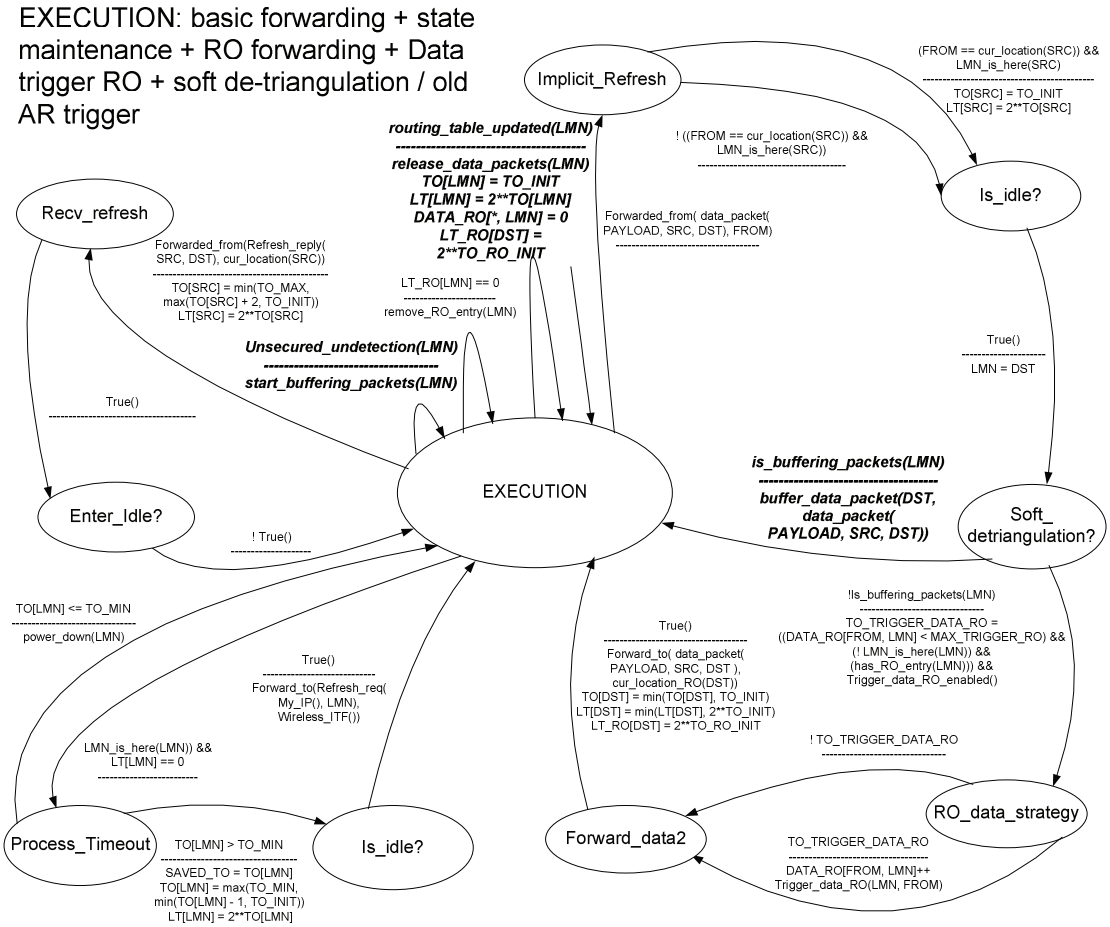


Figure 104: packet buffering formal specification

6.2.3.2 Fast Handover

General description

This section specifies extensions to the RO handover mechanism to reduce the handover latency during a handover, by creating the temporary local tunnel between the ARs at an earlier time than regular RO handover.

When the new AR node detects the arrival of a LMN, it starts the regular handover process as previously described, but can also directly notify the probable previous locations of the LMN with the latest RO information, using the same mechanisms used in the RO dissemination extension previously described. The choice of the neighbours to notify falls outside the scope of this specification, being suggested to notify the ARs that are geographically adjacent (on the physical network) to the new AR.

If, using this optimization, the previous AR is notified of the new LMN location, it can immediately create the local tunnel between the ARs. If the previous AR is not notified, then the same effect is reliably performed using the basic handover methods previously described.

Application example: Fast handover by direct notification of the adjacent ARs

Figure 105 shows an example of how the fast handover mechanisms are used to create the local tunnel earlier. When the LMN is detected at AR2 (trigger "C"), it starts the Fast handover process (trigger "F") and disseminates the latest RO information directly to its physically adjacent neighbours, which may or may not include the previous location of the terminal. When these set of ARs receive the latest RO information, they update their routing tables with the latest RO information, as previously described. In the given example, AR2 will notify both AR1 and AR3, which enables AR1 to immediately redirect the in-flight packets to the most current LMN location using such local tunnel (while the regular basic handover is being executed on the domain's tree).

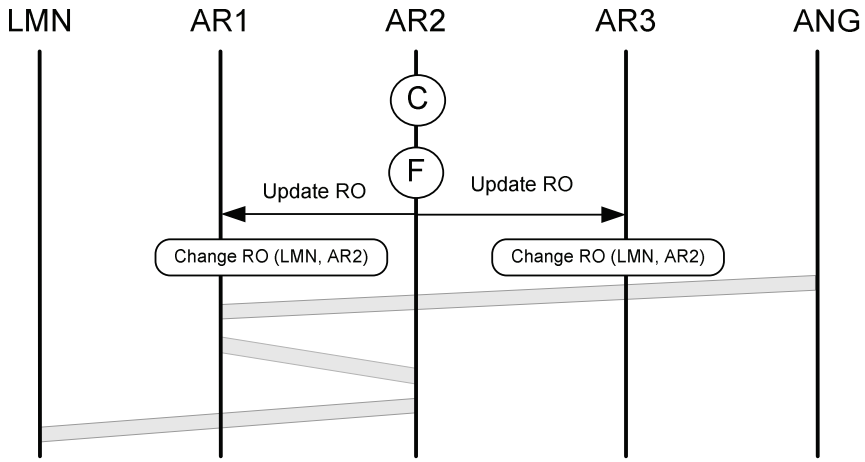


Figure 105: Fast handover by notifying directly the adjacent neighbours

Formal specification

The packet buffering mechanisms are formally described in the state machine of Figure 106. This formal definition expands the corresponding state machine of Figure 95, with the fast handover-related mechanisms. All used functions are described in Appendix B.

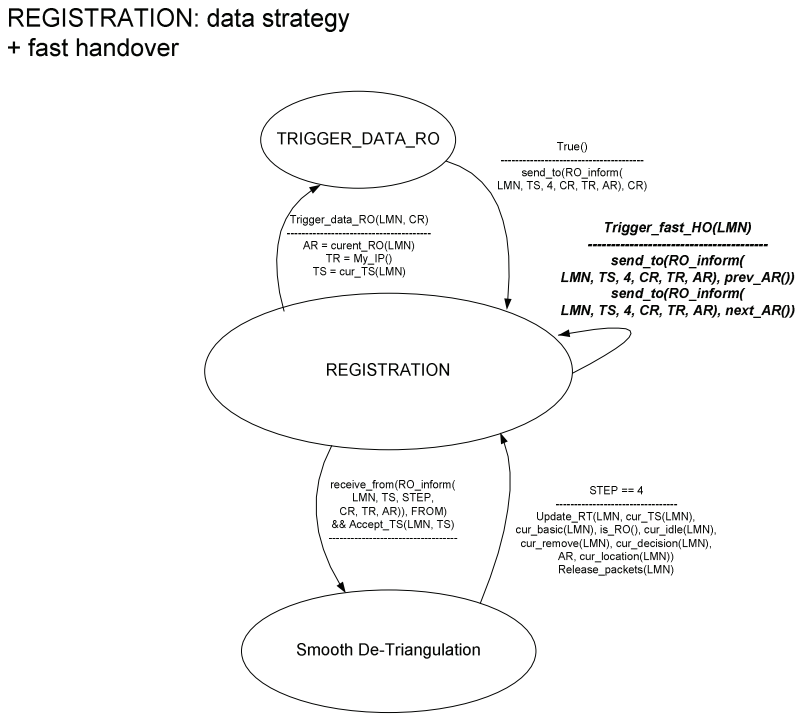


Figure 106: packet buffering formal specification

6.2.3.3 Symmetric Smooth de-triangulation Algorithm

General description

The smooth de-triangulation algorithms are a set of generic messages and procedures that improve the optimized routing's de-triangulation process without causing out-of-order packets **nor** increasing the packet delay of the in-flight packets. The proposed mechanism achieves its goals by delaying the packets, which will be sent via the direct path, for the exact minimum period of time that causes them to arrive at the destination immediately after the last in-flight packets sent via the triangulated path. Simpler versions of the mechanism were are defined that either remain optimal, but assume the presence of symmetric links, or that are non-optimal but guarantee packet ordering.

These mechanisms are necessary because the previously presented RO handover operation only considers the simplest form of removing the temporary local triangulation phenomena that occurs in each handover. Using the RO dissemination mechanisms, when the first packets arrive at the old AR, the old AR will notify the crossover node of the new LMN's AR (Figure 92). When the RO update message reaches the crossover node, the following data packets are sent directly to the correct AR, removing the triangulation effect.

Even though these operations do not typically incur in the drop of in-transit data packets, they can result in their reordering, as the direct path will typically have lower latency than the triangulated one; the exact latency difference will be the result of the actual link delays, the locations of the involved TRs and the amount of data packets queued in the routers. Depending on the type of receiver, these out-of-order packets can have a major impact on it, as additional actions may be needed to recover from such phenomena [164] [165]. These out-of-order packets can either be considered as lost packets by UDP receivers which expect ordered arrival [166], or can trigger the retransmit mechanism and needlessly reduce the contention window of the majority of the TCP senders, as no packets have actually been dropped [167].

The symmetric mechanism works by using a pair of messages called “buffer” and “flush” that are generated simultaneously at the new AR during a de-triangulation operation. The former (buffer) is sent directly from the new AR to the crossover node, and causes all the buffering of data packets destined to the given LMN. The latter (flush) is also sent to the crossover node, but is forced to first pass through the old AR (triangulating node); when is received, the message updates the routing table as before and releases the buffered packets using the new direct path. As the message pair measures the delay difference between the two paths, the packets will be buffered the optimal amount of time.

Regarding eTIMIP RO, the application of this option to the dissemination mechanism is straightforward, as it is only requires an extra step that transfers the process from the AR that detects the triangulation condition to the AR where the LMN is located. Then, the control messages, which feature a counter field used to distinguish between the “buffer” and “flush” messages, are sent as previously explained.

This algorithm has already been expanded to cover the case of asymmetric links, and was made generic in order to be possible to be applied to other mobility protocols that feature a de-triangulation component. Besides eTIMIP with route optimization, this can include at least the MIPv4-RO [10], MIPv6-RO [2], BCMP [25], and MEHROM [51] protocols. Such generic description is present in Appendix G, and is complemented with a analytic analysis of the efficiency of the algorithm.

Application example: symmetric smooth de-triangulation

Figure 107 shows an example of how the symmetric smooth de-triangulation algorithm is applied to eTIMIP RO. When the dissemination mechanism is triggered at the old AR (trigger “T”), the buffer message is first sent to the new AR. When this node receives this message, it generates a flush message on the reverse path, back to the old AR, and sends the buffer message to the crossover node (the GW), which uses it to update its routing table and starts buffering packets. When the old AR receives the flush message, it forwards the message to the GW, which uses it to release the buffered packets. At the final stage, the data packets are directly sent to the new AR, but are only received at the AR after the in-transit packets of the triangulated path, avoiding packet reordering.

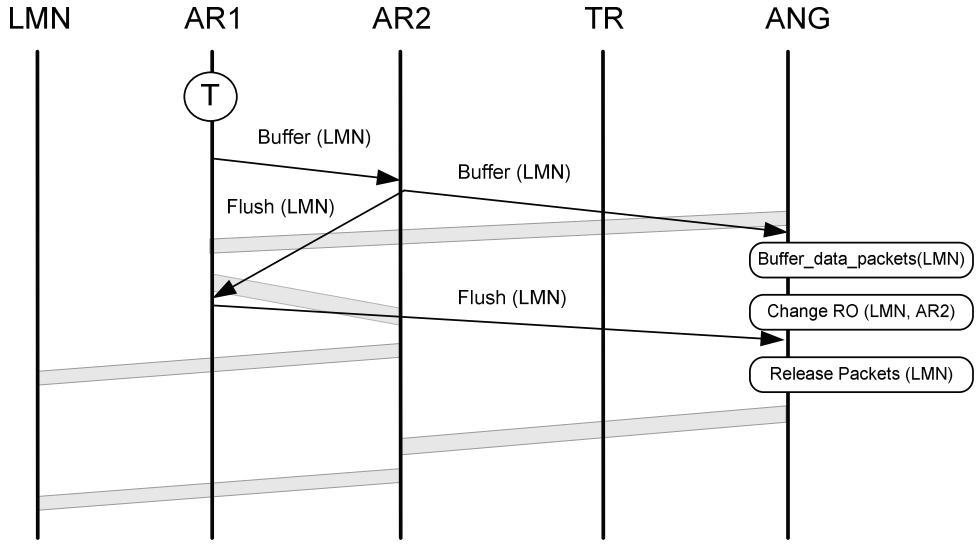


Figure 107: Symmetric Smooth de-triangulation

Formal specification

The described smooth symmetric de-triangulation algorithm applied to eTIMIP RO procedures are formally described in the state machine of Figure 108. This formal definition expands the corresponding state machine of Figure 95, with the symmetric smooth de-triangulation-related mechanisms. All used functions are described in Appendix B.

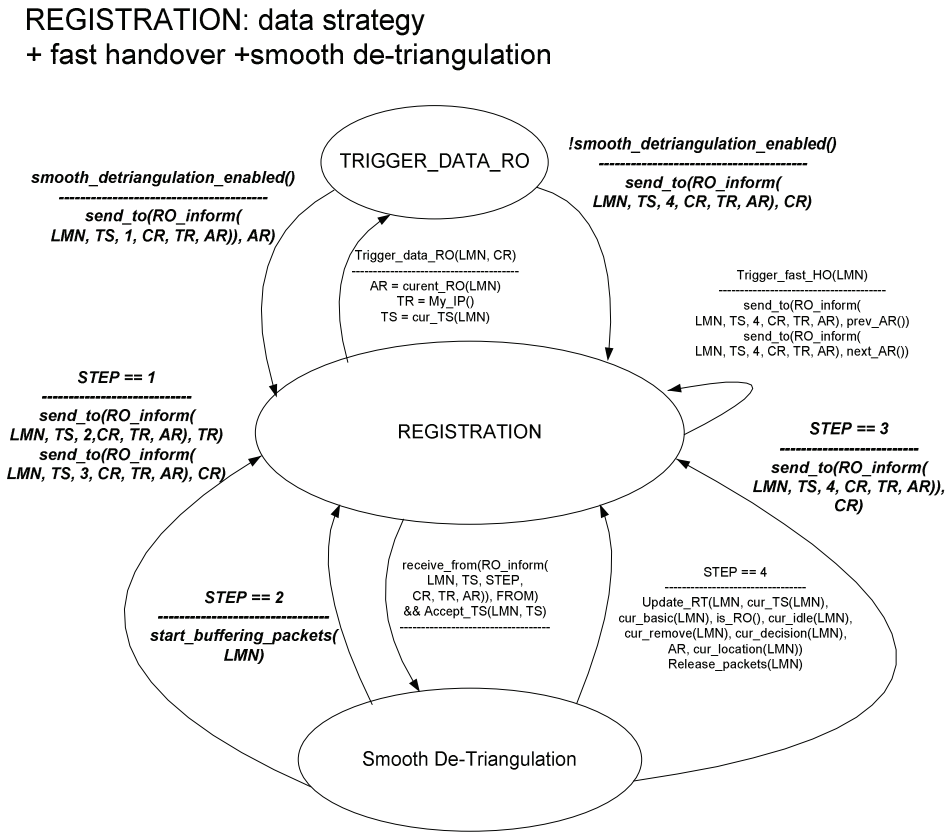


Figure 108: Smooth de-triangulation algorithm formal specification

6.2.3.4 Seamless Handovers combination

Application example: Seamless handovers

Figure 109 shows the combination of the described cross layer detection, fast handover, smooth handover, local triangulation and smooth de-triangulation mechanisms integrated to enable seamless handovers that simultaneously minimizes both the latency and the losses of each handover.

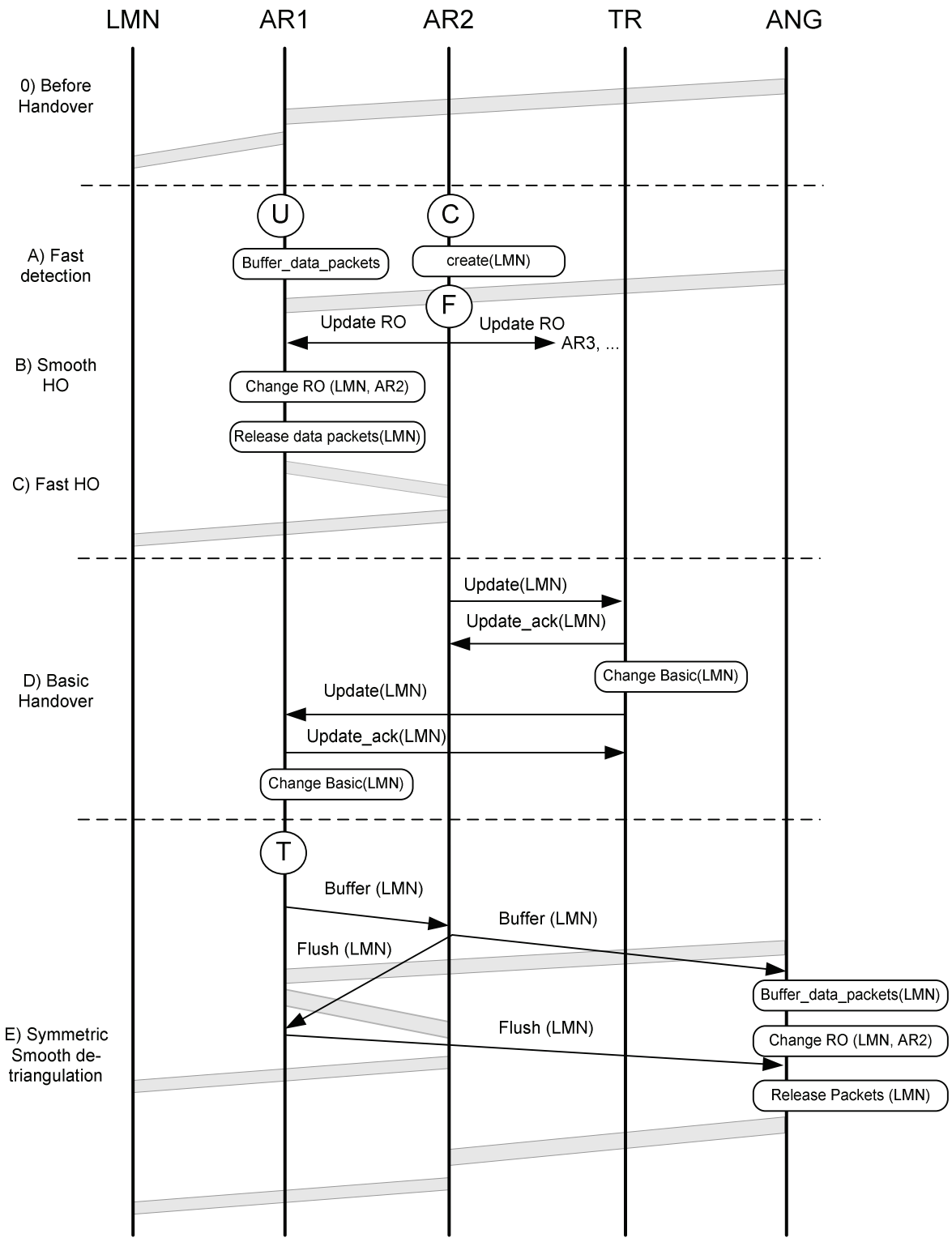


Figure 109: Seamless handovers combination

6.3 Idle Terminals Support

6.3.1 General Overview

This section presents a modular eTIMIP option that complements basic routing with support for idle terminals. This feature is able to both improve the scalability and efficiency of the protocol. Based on the LMN's current network usage, the eTIMIP agents may classify the LMNs into two different states: **active** or **idle**. The former considers the LMNs that are actually sending **and/or** receiving data, in which the network will make all the effort to accurately track, to provide a quick and prompt service. The latter considers the LMNs that are not currently using the network, and thus can be subject to a minimal mobility service. For them, the network will perform a lesser exact location service, being paged on demand by the network if and when there is new data for delivery. As it is widely studied, such support greatly helps increasing protocol scalability and reducing the state maintenance overhead [20].

This feature extends the previously described routing entries with a new class of entries, named idle entries, and adds a corresponding flag to the update control packets. In addition, the state maintenance operations are now able to trigger the idle operation, and are extended to permit the refresh of inactive terminals as well. In particular, this also requires a minor extension to the GDA to avoid LMN reactivations during the idle refresh state maintenance.

6.3.2 Detection phase

For idle support, the previously presented generic detection algorithm is slightly modified in order to support the temporary disabling of this process in the idle's entries refreshment process. This is necessary in order to prevent the regular detection processes described previously from unnecessarily reactivating the idle terminals during the refresh operation. Besides this, all the other detection functionality remains unaltered.

6.3.2.1 Deactivation of the Generic Detection Algorithm for idle terminals

General description

The previously proposed GDA is extended to support two primitives that signal the temporary disabling of the detection process for certain LMNs (GDA_ON / GDA_OFF). When GDA process is disabled, the first received data packet from a LMN will be ignored by the GDA if and only if it is a refresh response destined to an expected TR. This is support is useful to avoid unnecessary LMN reactivations during the idle refresh state maintenance.

Besides this special case, all the other situations are treated as before; in particular, other LMN data packets or unrelated refresh responses will always trigger the detection process as before.

Application example: Disabled Detection during Idle Refresh

Figure 110 shows a temporal diagram example which represents how an expected refresh reply packet, to a certain idle LMN, does not trigger the regular GDA procedures, and thus avoid LMN reactivation during idle refresh state maintenance.

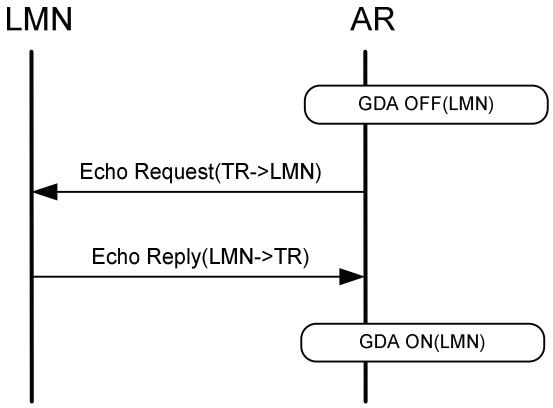


Figure 110: Generic algorithm detection

Application example: LMN re-activation during Idle Refresh

Figure 111 shows a temporal diagram example that clarifies that the above mentioned GDA disabling process is only related to expected refresh reply packets; if a regular data packet is received in this occasion, it triggers the detection process as usual, subject to the GDA timeout discussed earlier. This will eventually result in the idle LMN's re-activation, as discussed in section 6.3.4.1.

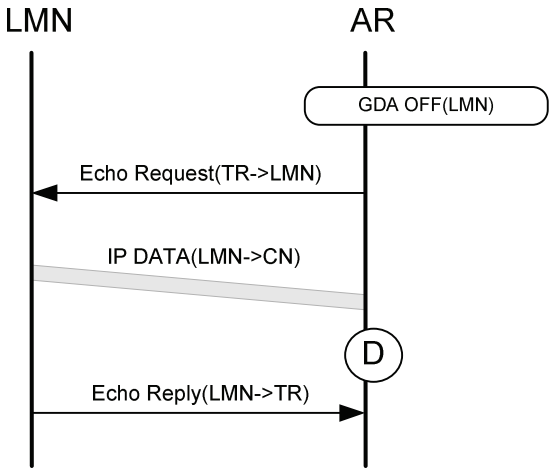


Figure 111: LMN re-activation during Idle Refresh

Formal Specification

The described generic detection algorithm is formally described in the state machine of Figure 112, which builds on the state machine of Figure 9 and adds the idle refreshment-related mechanisms to it, being these extensions identified by a bold typeface.

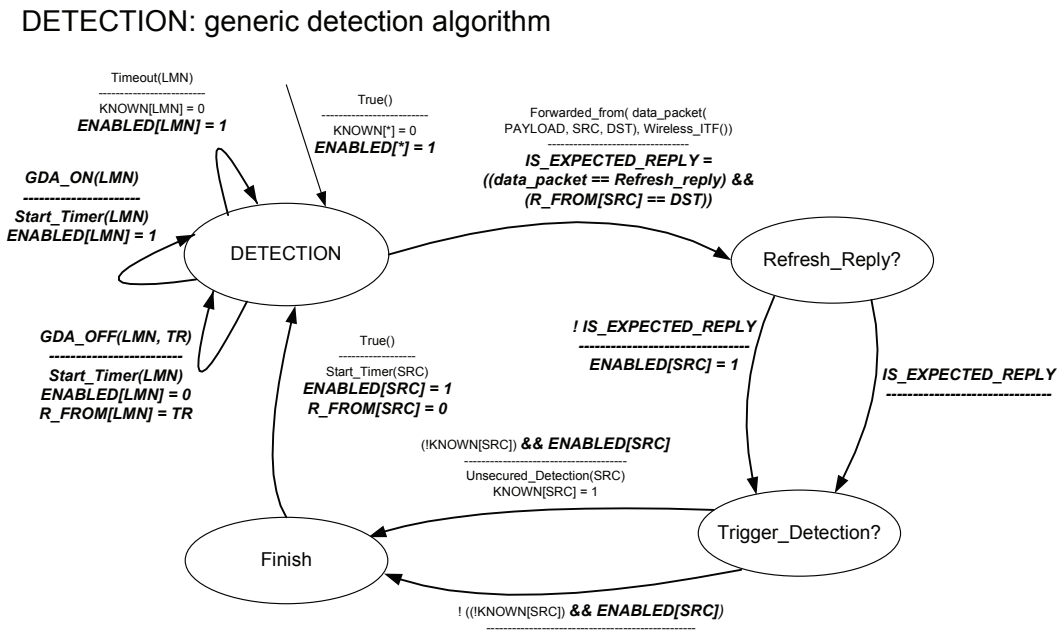


Figure 112: Generic Detection Algorithm formal specification

6.3.3 Registration Phase

For idle support, the previously presented basic update procedure messages and routing entries are expanded to support an extra “idle” flag. The other previously defined basic routing types of behaviour are maintained in this extension; in particular, the new idle update messages and entries use the same basic routing methods and messages, but feature an “idle” flag instead of a “basic” one.

6.3.3.1 Idle State Entry

General description

The idle procedures are triggered when the LMN’s AR detects that this terminal is idle for a long period of time, in the conditions of Figure 38 and Table 5, after multiple successful explicit refresh cycles were performed without LMN data being received. The idle procedures are triggered at the TR in the execution phase’s state maintenance operations, by a primitive named “init_idle()” (identified in the examples by “I”).

At the reception of this primitive, the agent responsible for the terminal starts by marking its LMN’s entry with the idle flag and sends an update packet with the idle flag to its parent node. By being a regular update packet, this control message is sent using the same mechanisms as the update messages described previously, namely using the reliability mechanisms via detection timestamps and retransmissions. Upon receiving this message, the parent agent modifies its LMN entry to “idle”, and replies with an acknowledgement packet. This ends the update process as the original node removes all LMN-related state from its memory.

Application example: State removal for Idle LMNs (idle entry)

Figure 113 extends the previous Figure 37 to illustrate how soft-state entries are removed for idle LMNs. After multiple successful explicit refresh cycles, where no LMN data is received (trigger “I”), the AR responsible for the LMN will transfer this responsibility to the TR agent above in the tree, which will modify its LMN entry to “idle”. Upon receiving this

message, the parent agent replies with an acknowledgement packet, ending the update process, where the original node removes all LMN-related state from its memory.

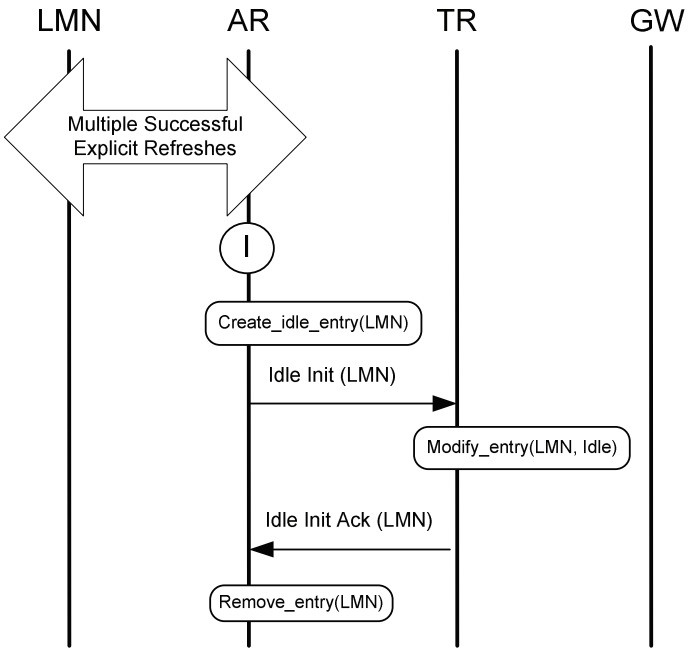


Figure 113: Idle state entry

Formal specification

The idle update methods are formally described in the state machine of Figure 114, which builds on the state machine of Figure 88 and adds the idle-related mechanisms to it, being these extensions identified by a bold typeface. All used functions are described in Appendix B.

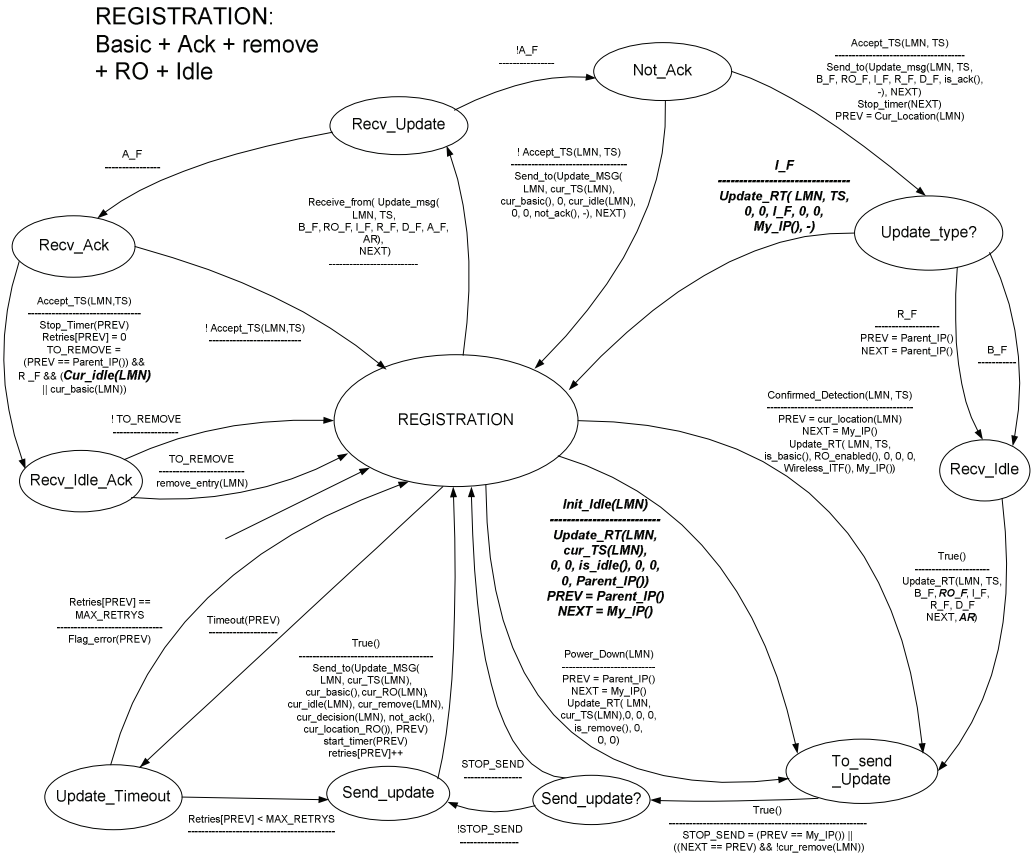


Figure 114: idle support formal specification

6.3.4 Execution Phase

For idle support, the previously presented state maintenance and forwarding procedures are extended to trigger the idle and paging operations, respectively, where the former concerns the detection of the idle LMNs, and the latter is their subsequent activation by part of the network when new data is available. All the other previously defined execution procedures maintain their functionality with this extension.

6.3.4.1 Idle triggering and Paging operation

General description

The idle triggering feature extends the previously described basic routing's state maintenance mechanisms by associating the upper bound of the refresh cycle procedure with the detection of idle terminals. When an explicit reply packet is received and the timeout value is already at its maximum, which is a sign that multiple explicit refresh cycles were tried **with success** for an idle terminal, then the idle support procedures are triggered for this LMN, in the registration phase. These idle support procedures will ultimately result in the transfer of LMN's state to the agent's tree parent, and the removal of the LMN entry in the current node.

When data is available on the network for an idle LMN, the data packet is routed in the usual way until it reaches the agent that has an idle entry for the LMN; at that moment, the packet is buffered and a paging operation is used in order to find the exact location of the LMN. This paging message is sent recursively through the sub-tree below the current node, containing a data_refresh flag that signals that the terminal is to be re-activated by the net-

work once it is found. When the paging messages reach the sub-tree's ARs, these nodes generate explicit refresh request packets to find the LMN.

When the terminal replies to one of these messages, the first packet transmitted by it is sufficient to recreate the missing routing entries in the agents between the AR and the original TR, with only the update procedures previously defined in the basic routing. This results in the release of all buffered packets and the end of the paging operation. Then, all subsequent LMN packets are routed directly using basic routing, without further overhead from paging operations.

Application example: Paging operation and LMN re-activation

Figure 115 illustrates how data is delivered for idle LMNs. Firstly, the data is routed in the usual way until it reaches the agent which has an idle entry for it (TR); at that moment (trigger "R"), the packet is buffered and the terminal is paged by the network, using a paging message that is sent to the sub-tree containing the terminal. All the sub-tree's ARs will send ICMP EchoRequest messages to the terminal, which forces it to reply with an EchoReply message destined to the AR (trigger "D").

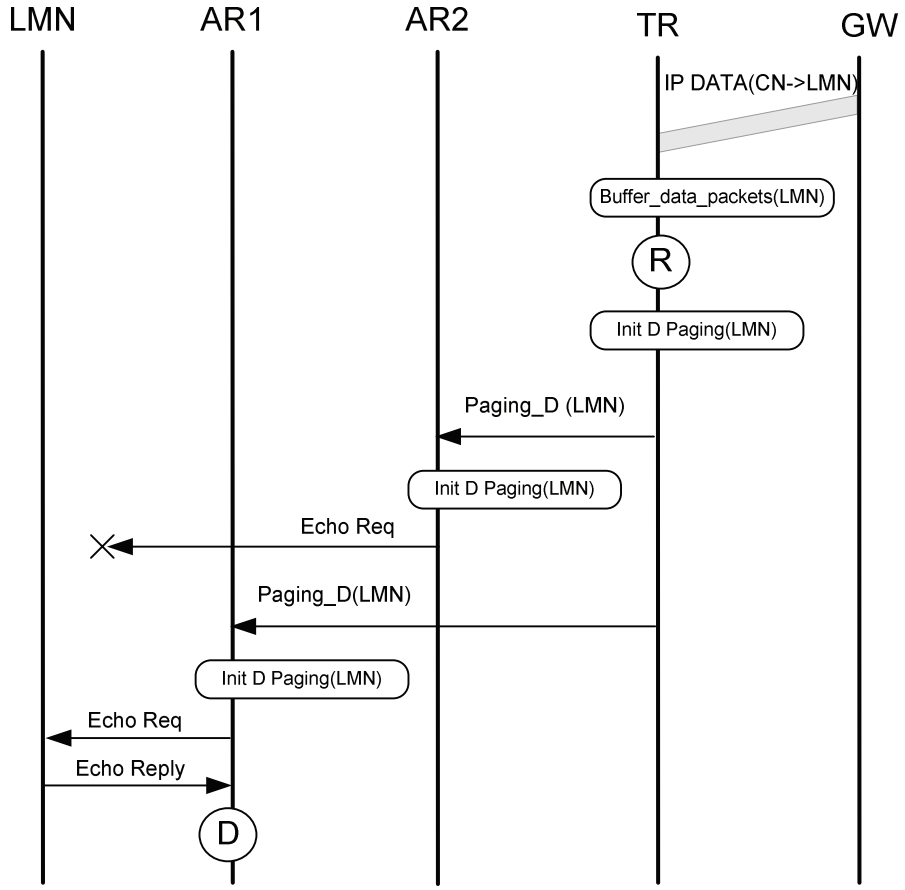


Figure 115: Paging operation

When the terminal responds, the first data packet transmitted by the terminal will re-create the routing entries in the necessary agents (trigger "C"). This effectively ends the paging operation, as buffered packets are flushed, and the following ones will be routed directly without any further paging operations (Figure 116).

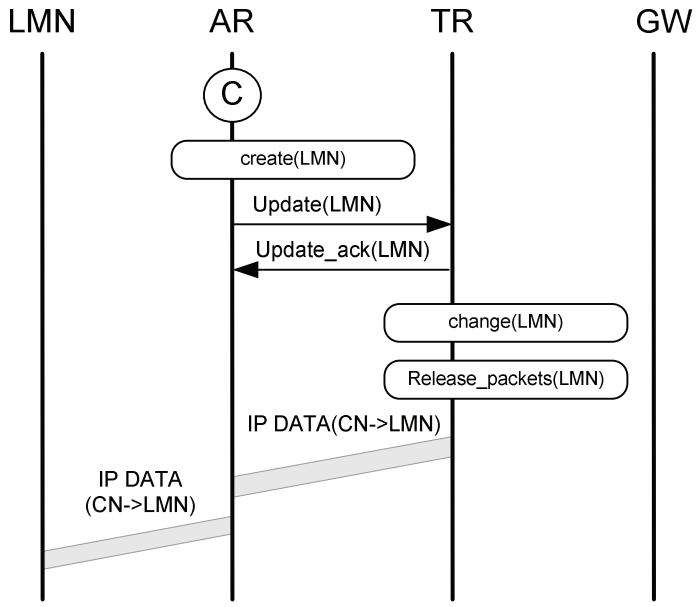


Figure 116: LMN re-activation

Formal specification

The idle triggering and paging triggering methods are formally described in the state machine of Figure 117, which builds on the state machine of Figure 41 and adds the idle-related mechanisms to it, being these extensions identified by a bold typeface. All used functions are described in Appendix B.

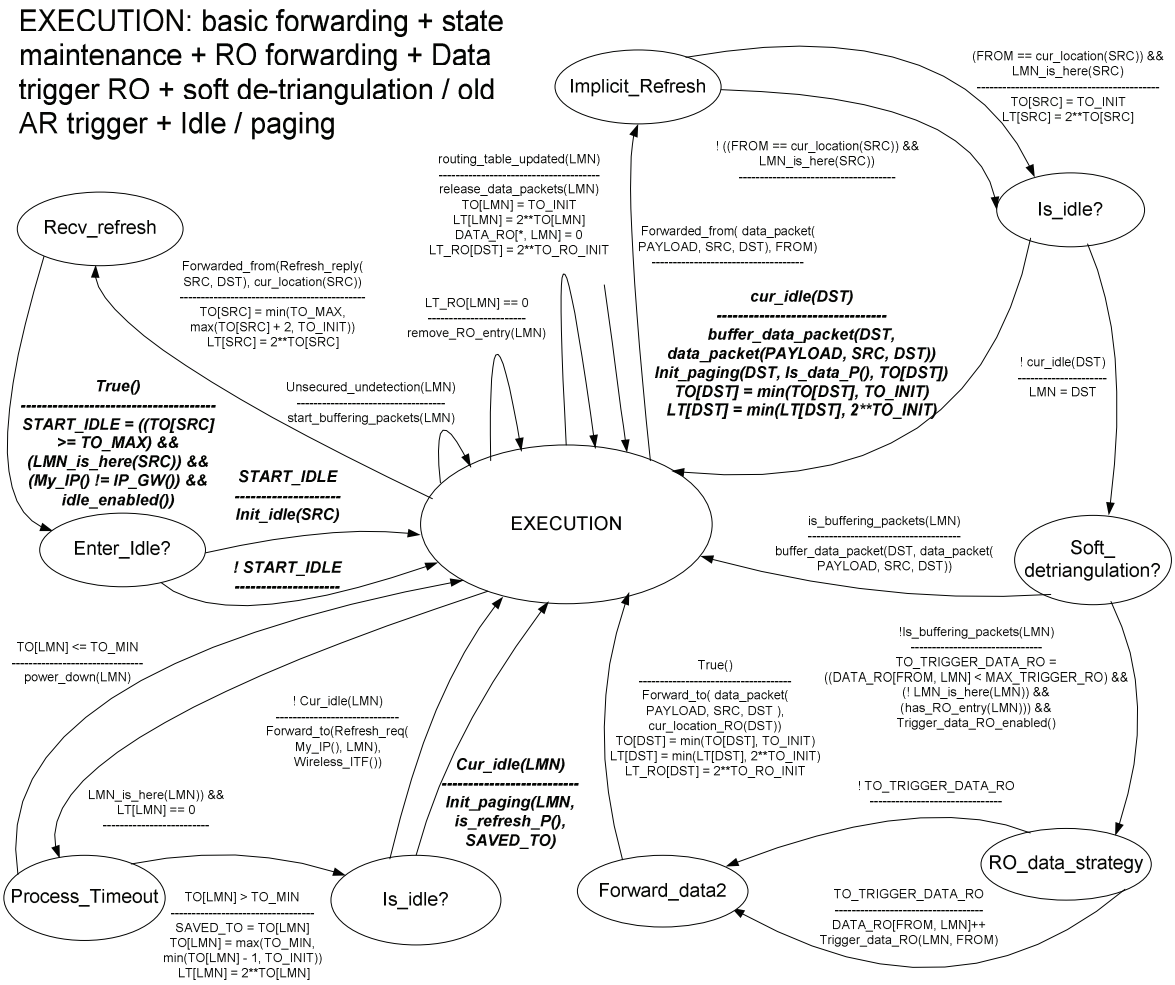


Figure 117: Idle state and paging triggering formal specification

The paging message methods are formally described in the state machine of Figure 118. All used functions are described in Appendix B.

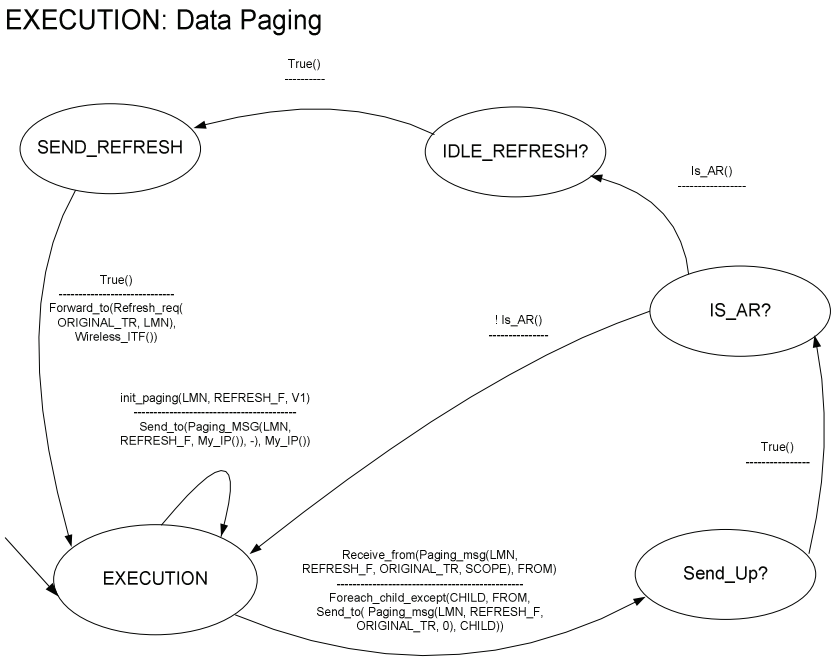


Figure 118: Paging mechanism formal specification

6.3.4.2 Idle entries refresh

General description

The idle entries refresh capability extends the previously described state maintenance mechanisms for support of idle entries, where the presence of the idle terminals somewhere inside the sub-tree where they were registered, is regularly confirmed. Similarly to the state maintenance of active hosts, the successful state maintenance of idle hosts is also used for triggering the idle operation to upper parts of the tree, up to the GW.

When a refresh is needed for an idle terminal, the TR node starts a paging operation for this terminal as before, but marks the generated packet with an idle refresh flag instead. This flag signals that the terminal is only to be found, and not made active again. When such paging messages reach the ARs, these nodes will temporally disable the generic detection algorithm and send an explicit refresh request packet to the LMN with the original TR which started the paging operation as the sender of the packet.

When the LMN replies to this refresh, it will not be reactivated, as the regular detection procedures are automatically skipped; on the other hand, the refresh reply message is also automatically sent to the TR that initiated the paging operation, which refreshes the idle LMN's state in this node. As before, after several successful refresh operations for an idle LMN, a further idle operation can be triggered to transfer the LMN's entry to the parent node, up to the GW.

Application example: Idle entries refresh

Figure 119 illustrates how idle entries are refreshed. Firstly, the TR which has a dirty idle entry (trigger "IR") sends a refresh paging message to all its sub-tree's TRs; all the sub-tree's ARs will send ICMP EchoRequest messages to the terminal, which forces it to reply with an EchoReply message destined to the original TR. In this process, the ARs temporarily disable their GDA capabilities for this LMN to prevent an undesired LMN's re-activation (thus, no trigger "D" is invoked). This process is subject to the minimum and maximum timeout values exemplified in Table 9.

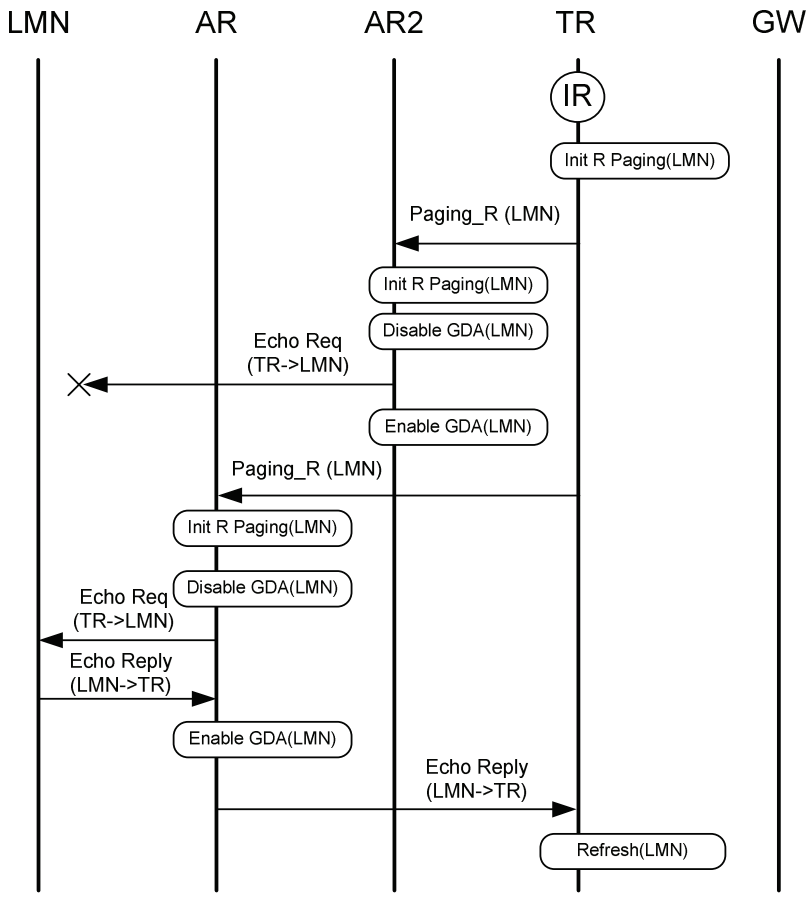


Figure 119: Idle entries refresh

Constants	MIN		INIT				MAX
TO	8	9	10	11	12	13	14
LF	256	512	1024	2048	4096	8192	2048
Time between refreshes	~4m	~8.5m	~17m	~34m	~1h	~2h	~4h
Refresh Involved levels	2	2	2	2	2	2	2

Table 9: Example configuration of state refresh values for idle terminals

Application example: Idle re-entry

Figure 120 illustrates how idle entries are transferred to the top portions of the tree. After multiple successful idle refresh cycles, where no LMN data is received, the TR responsible for the LMN will transfer this responsibility to the TR agent above it in the tree, which will modify its LMN entry to “idle”. Upon receiving this message, the parent agent replies with an acknowledgement packet, ending the update process, where the original node removes all LMN-related state from its memory. It should be noted that this process stops when this message reaches the GW, that has the longest configured timeouts.

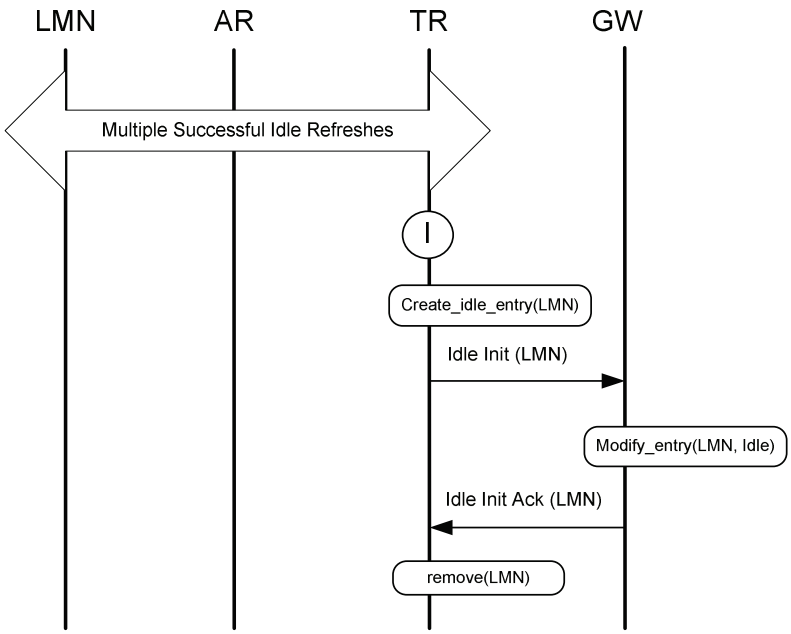


Figure 120: LMN idle entries re-entry

Formal specification

The idle entries refresh methods are formally described in the state machine of Figure 121, which builds on the state machine of Figure 118 and adds the idle-related mechanisms to it, being these extensions identified by a bold typeface. All used functions are described in Appendix B.

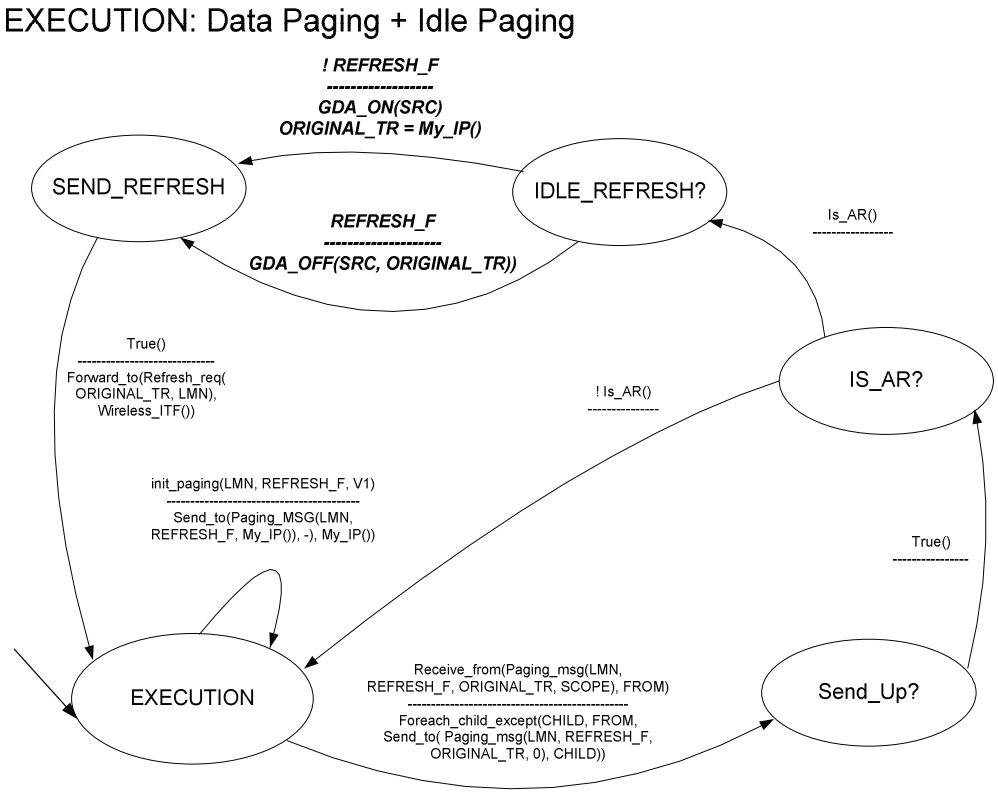


Figure 121: Idle entries refresh formal specification

6.4 Operator-Centric support

6.4.1 General Overview

This section presents an eTIMIP option that complements basic routing extensions useful for operator-centric scenarios, with the possibility of network-controlled handovers and network-confirmed departure detection.

In the first extension, it is the network that decides which is the best point of attachment of the LMNs to the network. In this model, the network's ARs will be able to report the perceived LMN movements to a common entity. The deciding node can then use such movement reports to dynamically assign the LMNs to specific ARs, possibly based on QoS metrics or other operator-defined policies. When a decision is made, the destination AR is notified, in a guaranteed way, to force a LMN handover to it. This mechanism is particularly useful for operator-centric scenarios, for implementing load-balancing capabilities, or for use with wireless technologies which support multi-access of a LMN by several APs simultaneously, where multiple ARs can detect the LMN at the same time.

In the second extension, the basic and idle routing's state maintenance operations are extended with the confirmation that the LMNs have not physically moved to other ARs and went undetected. When used, this option can increase the reliability of the protocol by ensuring that these LMNs, which are still connected to the network but went unnoticed, are not powered-down by the network. This need, which is not common in classic mobility protocols, arises from the terminal independency support, as it is the network that is responsible for tracking all LMN movements. As such, it is possible that an idle LMN makes a movement which passes unnoticed both by the generic detection algorithm and the cross-layer detection procedures, and has the risk of having its routing entries removed through the state maintenance operations[20].

6.4.2 Detection Phase

6.4.2.1 Network Handover Decision Engine support

General description

This section specifies extensions to the basic routing's detection phase with methods that enable the network to control the handover decision.

When the ARs securely detect the terminals using the previously described methods, instead of proceeding directly to the registration phase and performing a handover, it will first report the LMN's detection to the current AR of the terminal. This report message can optionally include additional L2 parameters, such as the received link quality, which quantify the prospective attachment quality of the LMN to this AR. As in the update messages, these

[20] It should be stressed that these situations are very rare, but nonetheless possible in the context of terminal independency. This can only happen to listener-only terminals that move away from its AR, or to idle terminals that move away from their paging area, and which, in both cases, are not detected either by the generic detection algorithm or the cross-layer detection support.

reports are sent to the current LMN's AR by following the domain's routing paths and old reports are discarded using the detection timestamps.

When the current AR of the LMN receives these reports, it decides which one is the next AR of the LMN. The definition of such decision mechanism falls outside the scope of this specification, being possible to use the L2 quality parameters, QoS metrics or any other operator-defined policy [52]. When a decision is made, the destination AR is notified of the decision in a guaranteed way, using an update packet with a "decision" flag, which triggers the regular registration operations previously described.

Other viable alternative that this procedure can use is a centralized decision engine entity, which processes the report messages, instead of the previous AR of the terminal [52]. In addition, this network-controlled support can be used in conjunction with 802.21 command services to provide enhanced handover services to 802.21-compliant terminals [69].

Application Example: Network Controlled Handovers

Figure 122 shows an example of how the network can control the handovers. Firstly, a LMN currently associated to AR1 is detected by both AR2 and AR3. These nodes will independently generate report messages that describe this event, optionally with specific L2 parameters. These messages are propagated in the tree until they reach the LMN's AR (AR1). After receiving the reports and reaching a decision, it sends a decision update packet to the chosen AR (AR2), which triggers the handover as normal.

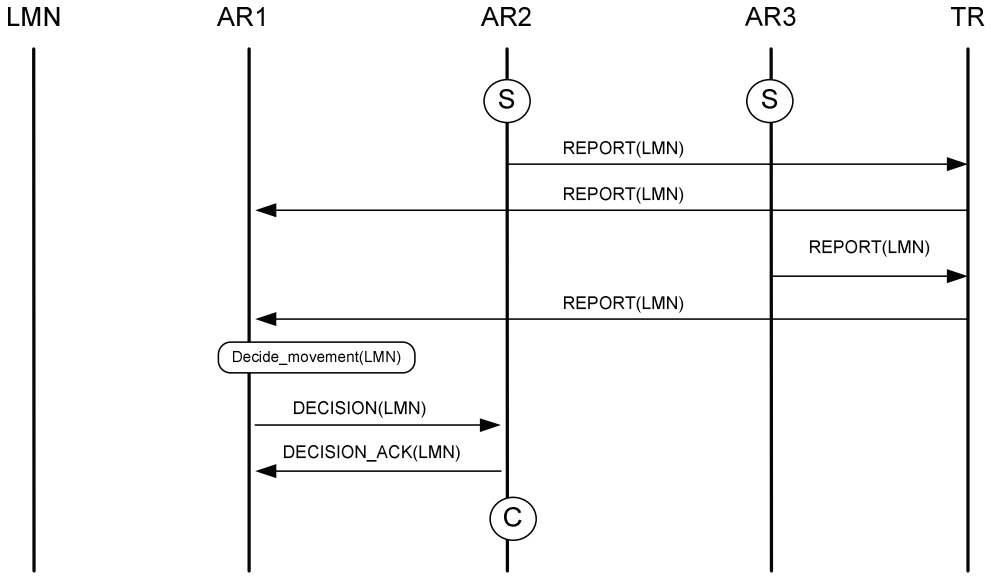


Figure 122: Network Controlled Handovers

Formal specification

The network controlled handover mechanisms are formally described in the state machine of Figure 123. This formal definition expands the corresponding state machine of Figure 16 with the network-controlled handover mechanisms. All used functions are described in Appendix B.

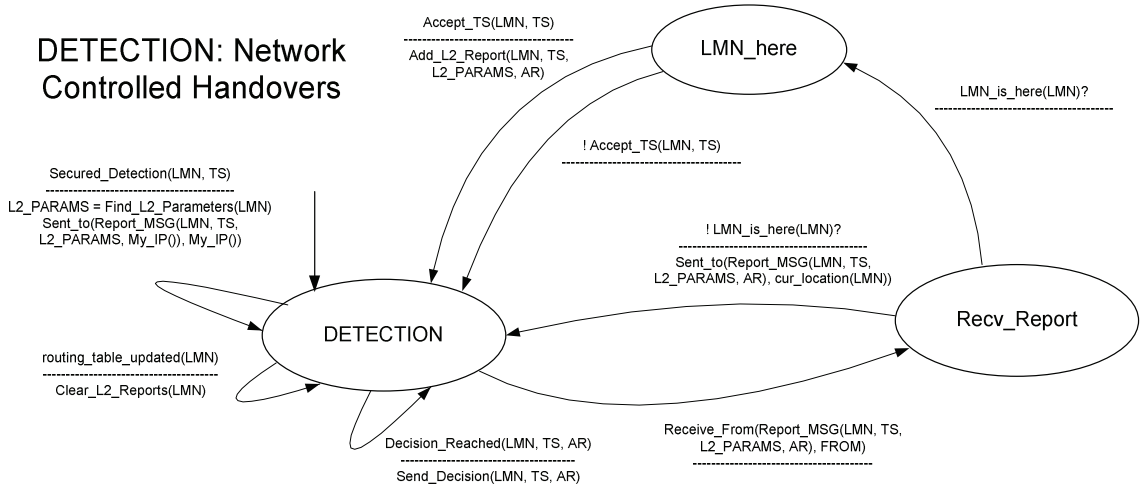


Figure 123: Network controlled handovers formal specification

6.4.3 Registration Phase

6.4.3.1 Network Handover Signalling support

General description

This section specifies the extensions to the basic routing's registration phase, by adding an additional flag to the update packets that signal the handover decision to the chosen AR. All the other update packets are functionally unchanged, namely the reliability via timestamp and timeout mechanisms.

Formal specification

The network controlled handover mechanisms are formally described in the state machine of Figure 124. This formal definition expands the corresponding state machine of Figure 114, with the decision update packet mechanisms. All used functions are described in Appendix B.

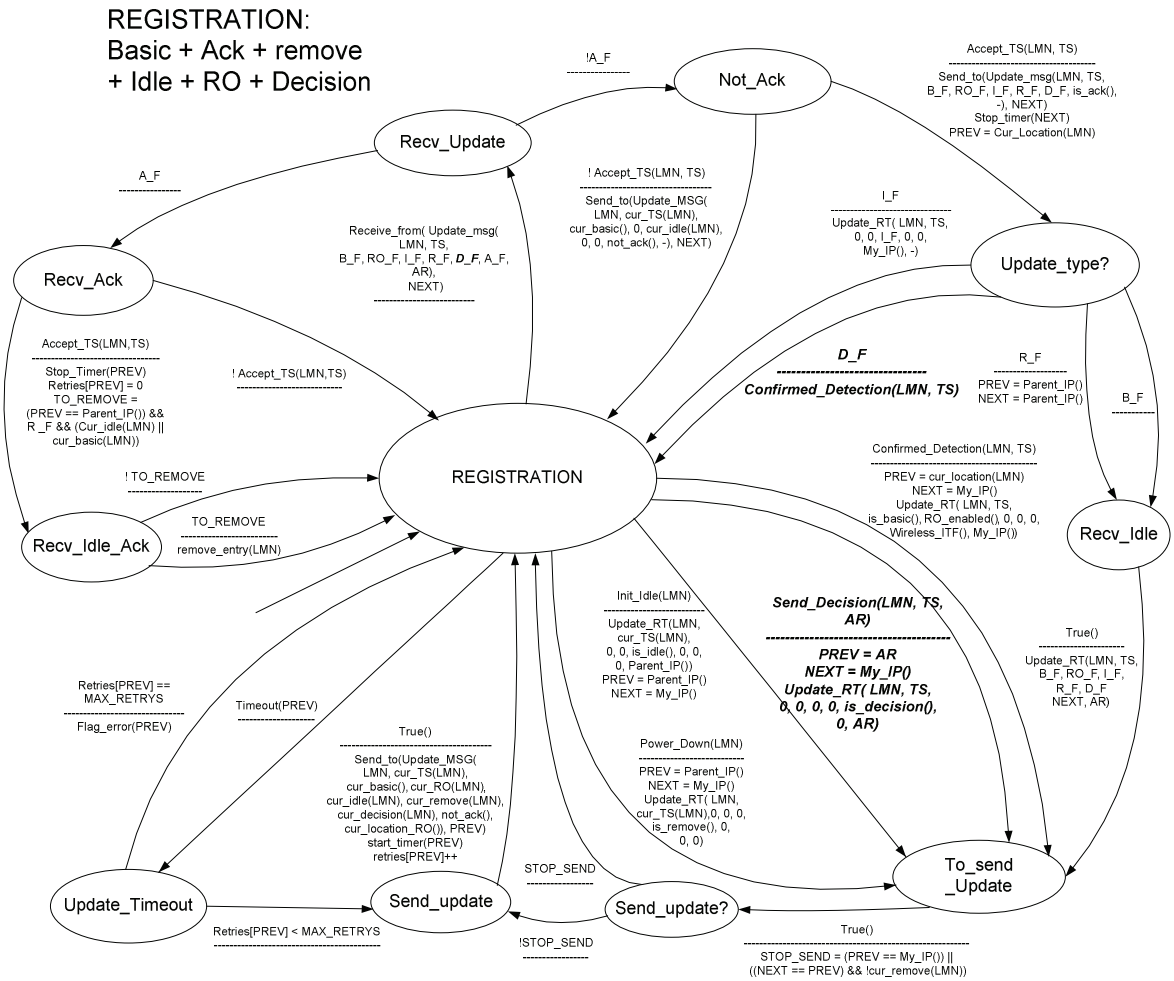


Figure 124: Decision message support

6.4.4 Execution Phase

6.4.4.1 Departure Confirmation Procedure

General description

The departure confirmation extension complements the basic and idle routing's state maintenance with the confirmation that the LMNs have not physically moved to other ARs and went undetected.

In this module, the state maintenance back-off operations are extended in order to additionally search for the terminal outside its normal location (the AR for active terminals, and the paging area for idle terminals) in the final stages of the back-off state maintenance procedures. For efficiency, these extra refresh operations are first limited to the hierarchical vicinity of the terminal and later extended to the whole network. It should also be noticed that, like the paging operations, such special measures are only needed until the terminal replies for the first time after being unnoticed, as it will triggers the regular update operations described previously.

The departure confirmation option is instantiated by an addition to the state maintenance paging messages through the inclusion of a "scope" field that signals the desired extent for the paging process. In the state maintenance process, the paging messages continue to be

forwarded for all children of a node, but can also be sent up the tree, as dictated by its scope value. The decision of the particular configurations to use such field is operator-defined.

When the parent node receives the paging message from its child, it decreases the scope value by one, sends the message to all the other children, and additionally sends the message to the parent node if the scope is still positive. If the ARs outside the LMN's sub-tree participate in this special paging operations, they will always perform the regular data refresh operations instead of idle refresh, to force the idle terminal's reactivation.

Application Example: Active state maintenance operations with departure confirmation

The following figures (Figure 125, Table 10 and Figure 126) present the state maintenance operations with departure confirmation for active terminals, being extended from the corresponding previous figures (Figure 37, Table 5 and Figure 39). Here, a network operator policy is chosen in which the LMNs are searched for in all the domain's tree levels up to 3 times before being removed, in the final back-off refresh operations (shaded cells of Table 10).

In Figure 125, a LMN moves from AR1 to AR2 but is not noticed either by the GDA or the cross-layer detection mechanisms. After multiple unsuccessful explicit refreshes limited to the local AR1, the next refresh operation (trigger "R") is additionally propagated up the tree (node TR), enabling the refresh operation at AR2. The refresh operation concludes when the LMN replies to the echo request packet, as AR2 detects the LMN using the GDA.

In Figure 126, a LMN moves away from the local domain. After multiple unsuccessful explicit refreshes limited to the local AR1, the next refresh operation (trigger "R") is additionally propagated up the tree (node TR), enabling the refresh operation at AR2, and at an later stage, also at AR3. At the end of this process, the AR will start the removal of the LMN's entries in the involved TRs, using a power-down procedure.

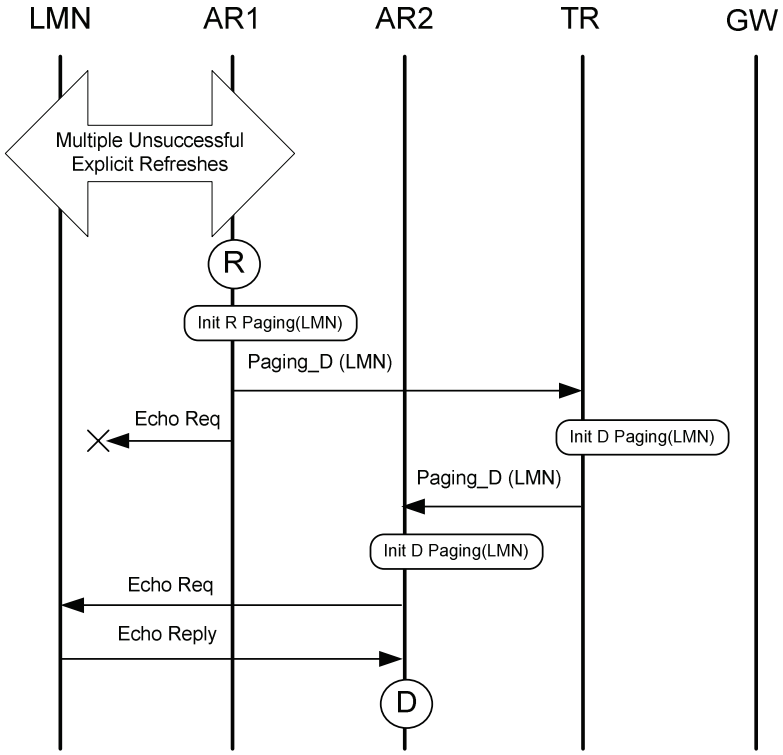


Figure 125: Active Terminal refreshment with confirmation

Constants	MIN					INIT					MAX
TO	0	1	2	3	4	5	6	7	8	9	10
LF	1	2	4	8	16	32	64	128	256	512	1024
Time between refreshes	1s	2s	4s	8s	16s	32s	~1m	~2m	~4m	~8.5m	~17m
Refresh Involved levels	3	3	3	2	1	1	1	1	1	1	1

Table 10: Example of state refresh values (with departure confirmation support - shaded cells)

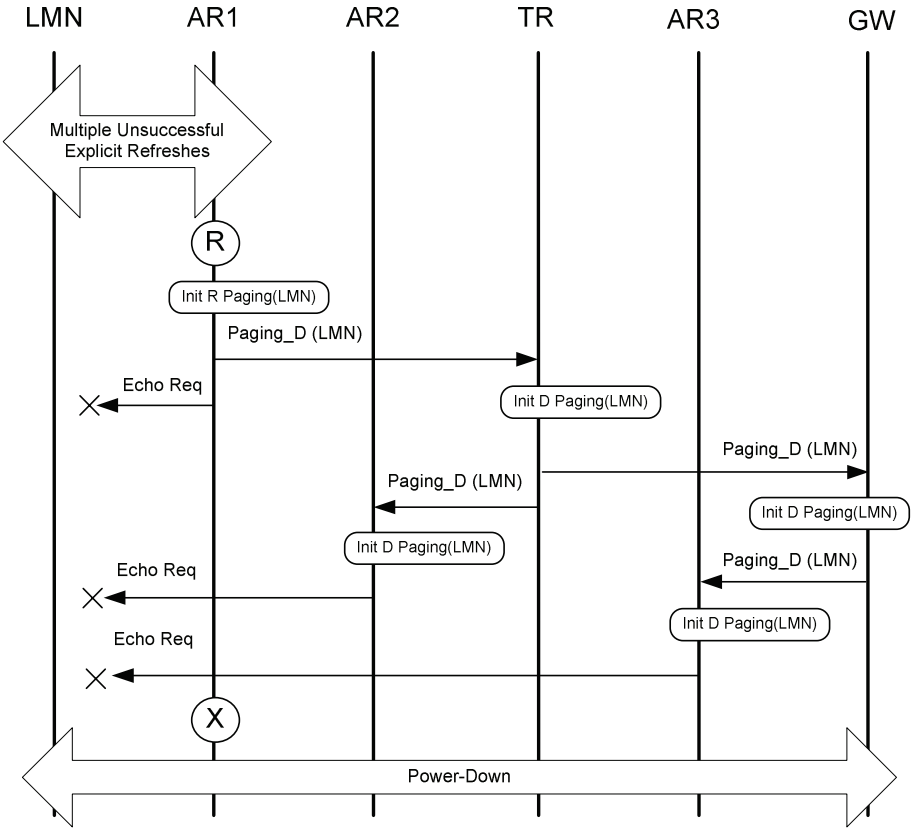


Figure 126: Power-down for Active LMN with confirmation

Formal specification

The departure confirmation mechanisms are formally described in the state machine of Figure 127. This formal definition expands the corresponding state machine of Figure 121 with the departure confirmation-related mechanisms. All used functions are described in Appendix B.

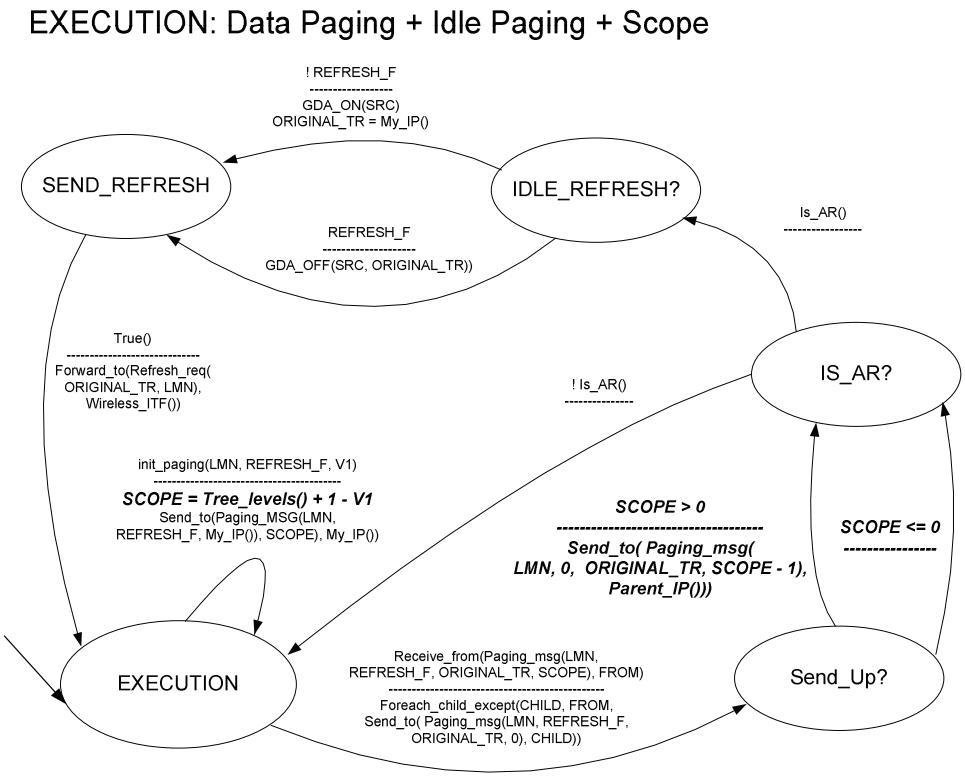


Figure 127: Departure Confirmation procedure formal specification

7 Extended eTIMIP module Tests

This chapter presents a complete performance evaluation of the eTIMIP extensions, concerning both its efficiency and its transparency aspects. For this, the base protocol will be studied, analysed and compared to various combinations of the extensions. The extensions were incrementally applied to the basic protocol, to better show their improvements and deteriorations as independently as possible. The combination of all extensions at the same time was also modelled, being designated as the "**full**" protocol. The presented tests follow the exact same topology, movement, traffic, metrics and test sets of the initial ones already presented in chapter 5.

The rest of this chapter is organized as follows: the first section will compare the basic version of eTIMIP to its extensions using UDP data traffic; the second section will repeat the same tests with TCP traffic. Finally, the final section will present a discussion part that summarizes the major pros and cons of the eTIMIP full protocol overall, on the same topics addressed previously in the first simulation study: handover efficiency, routing efficiency, transparency, reliability and scalability.

7.1 eTIMIP extensions tests using UDP data traffic

7.1.1 Discrete Isolated handover

This test will illustrate what happens during a single handover, from the point of view of the mobile node, when being served by the eTIMIP full protocol. For this, the MN will perform a single isolated handover in the middle of the domain, from node AR4 to node AR5, to show precisely what happens to the established data flows in each handover.

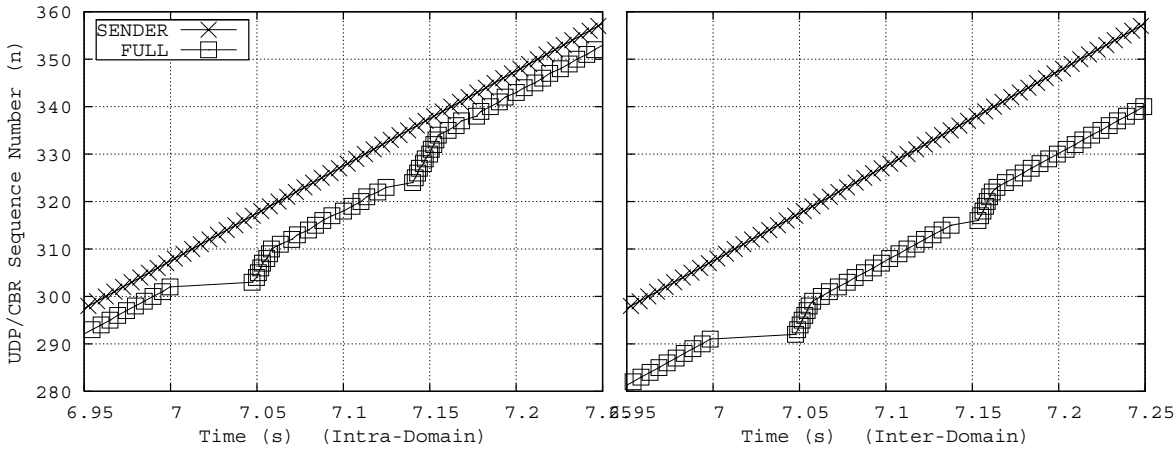


Figure 128: Isolated handover UDP time series - full eTIMIP protocol

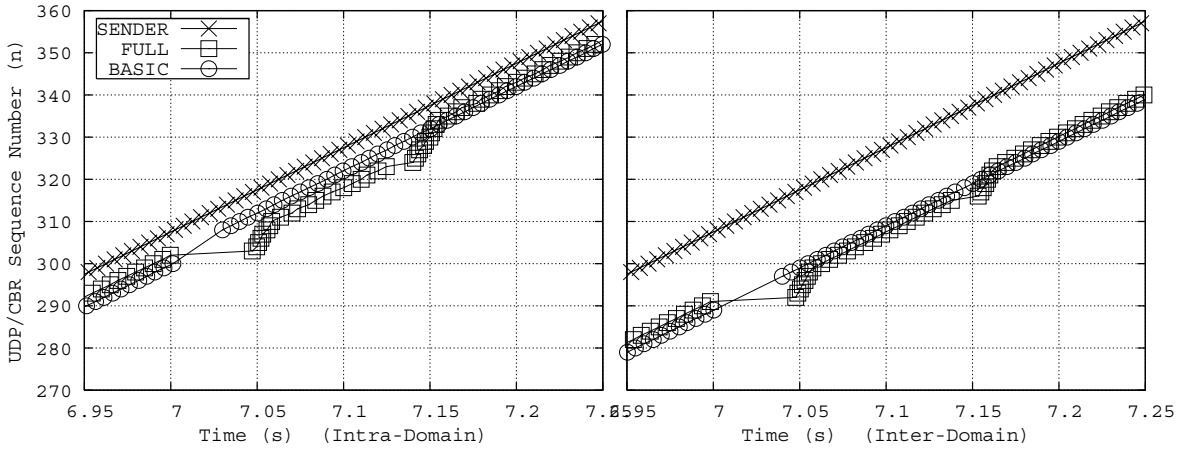


Figure 129: Isolated handover UDP time series - basic and full eTIMIP protocols

Figure 128 shows the time diagram of the full protocol in the same previously considered handover situation, featuring a single handover that occurrence at time 7.0s. For easier comparison purposes, the graph is superimposed with the previous graph of the basic protocol in Figure 129.

The comparison of both handovers shows that the full protocol imposes higher handover latency than the basic protocol, but in turn, it delivers all packets to the MN without any losses or reordering (starting at time 7.05s). This fast reestablishment, in the same magnitude order of the basic protocol, is possible without any losses through the combination of several eTIMIP extensions:

- The smooth handover extension enables the buffering of all in-flight packets that are received in the previous AR;
- The fast handover extension enables the fast notification of the adjacent ARs from the new AR, in parallel with the regular basic handover;
- The RO extension enables the creation of a temporary local tunnel between the involved ARs, to quickly deliver all packets in an ordered form.

At a later stage (time 7.15s), the local tunnel is released, in order to route the packets directly to the new AR, which ends the temporary tunnelling mechanism. However, this operation does not incur in packet drop or reordering, nor does it further delay the involved packets. This is possible using the smooth de-triangulation mechanism, which delays the packets, which will be sent via the direct path, for the exact minimum period of time that causes them to arrive at the destination immediately after the last in-flight packets sent via the triangulated path.

7.1.2 Stationary Measurements

This test will characterize the routing paths used by the eETIMIP extensions to stationary MNs. For this, the major extensions were applied to the basic protocol separately: the fast handover extension (HO_FAST), the smooth handover extension (HO_SMOOTH), and the route optimization (RO). Then, the separate RO extension was complemented with the three separate smooth de-triangulation algorithms defined in Appendix G (Asymmetric - RO_SOFT_A; Symmetric - RO_SOFT_S; Conservative - RO_SOFT_C). Only the “full” protocol combines all extensions at the same time, being used the symmetric de-triangulation algorithm option.

Again, for the stationary test, the MN will firstly move to each possible AR, starting from AR1, and remain there during the metrics measurement. The one-way delay results are summarized in Figure 130, where the X-axis refers to the MN's location in the domain (AR), and the Y-axis refers to the average one-way delay. This metric is related to the number of hops, being important for the delivery of UDP real-time services.

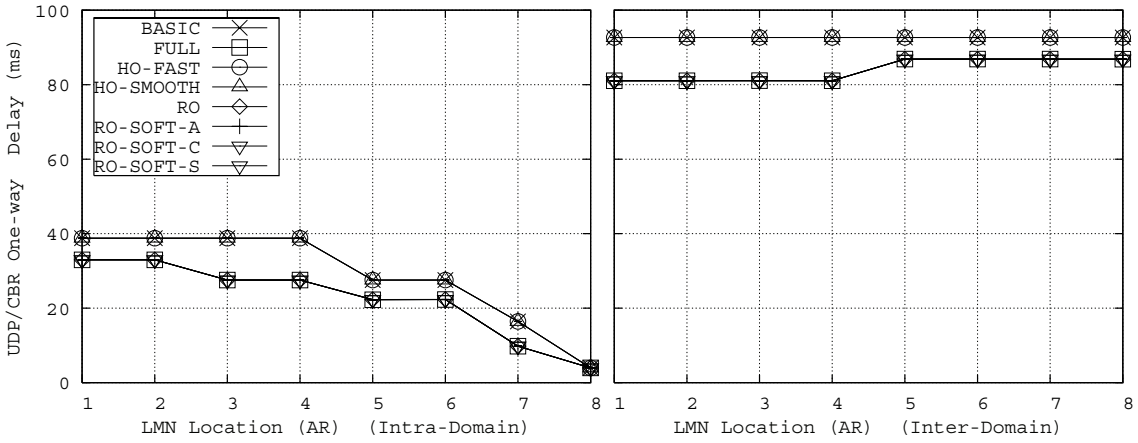


Figure 130: Stationary CBR/UDP one-way Delay per MN location

Regarding the intra-domain case, this experience shows that while basic eTIMIP already featured the most efficient tree-routing mechanism, the usage of the RO extension improves that result in stationary situations, by using optimal routing to **all** locations of the domain, bypassing the tree completely. Thus, the most direct paths between the borders of the domain - the involved ARs - are always used, resulting in the maximum possible routing efficiency: optimal routing. The other eTIMIP handover-related extensions have no impact on this metric.

Regarding the Inter-domain traffic, the previous constant one-way delay is also improved using the RO extensions, for the exact same reasons as the previous case. Here, the mobility-aware ANGs will be able to directly route the data packets to the correct AR, which also bypasses the agent's tree, and most importantly, the single GW. Thus, eTIMIP RO is able to support the same optimal routing paradigm that was only measured for MIP in inter-domain traffic scenarios (Figure 50 on page 99).

For both cases, it should be stressed that this extension already provides the maximum routing efficiency – optimal routing. Thus, the presented delay values are only limited by the existence of direct links of the simulated domain (Figure 46, page 94). As such, an even greater efficiency improvement could be achieved through the introduction of further additional direct links, which would be immediately used to directly route the data packets to their destinations.

7.1.3 Continuous Movement (multiple MN speeds)

This section presents simulations using continuous MN movements with different speeds, which are obtained by varying the time between the handovers. This will consider very slow movements of 60s inter-handover intervals, corresponding to 1 handover per minute, up to extremely fast 1s inter-handover intervals, corresponding to 60 handovers per minute.

The results of the various combinations of eTIMIP extensions are shown in Figure 131 and Figure 132, representing the average total losses and throughput for each MN speed.

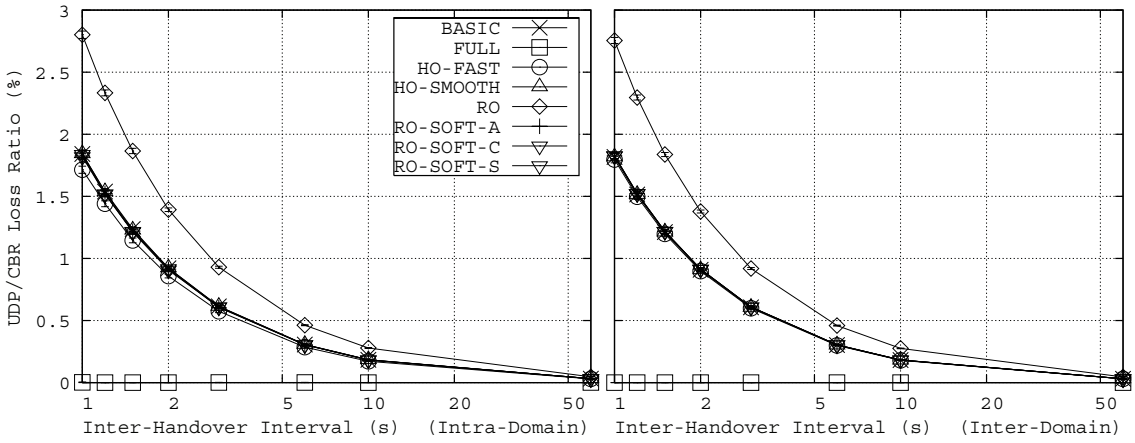


Figure 131: CBR/UDP Loss Ratio per inter-handover interval

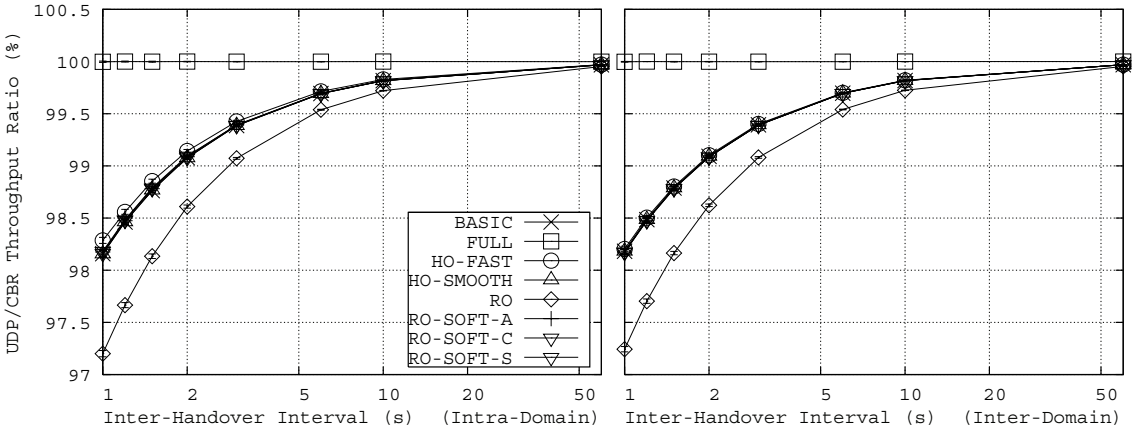


Figure 132: CBR/UDP Throughput per inter-handover interval

The graphs suggest that the isolated usage of the individual extensions do not seem to improve the previous results achieved by the basic handover. This happens because, although the isolated usage of the HO_smooth or the HO_fast extensions is actually able to reduce the amount of dropped packets, such is not visible in the graphs, as an approximately equal amount of out-of-order packets is introduced, maintaining the total loss ratio unaltered.

However, the real interest of the handover extensions is their combined usage (full protocol); specifically, the combination of the smooth handover, the RO, and a smooth de-triangulation mechanism present in the “Full” protocol is able to achieve a zero-loss handover. As described previously, such happens because the in-packets flight packets are buffered at the old AR until it is notified of the new MN location (HO_smooth extension), and are then sent in an ordered form to such location using a local tunnel (RO extension), which is additionally removed without causing out-of-order packets (smooth de-triangulation extension).

7.1.4 Reference Scenario (single average high MN speed, non-localized handovers)

This test recreates the reference scenario that will be used to analyse the extensions’ differences in greater depth. This is done by systematically analysing all the different mobility and UDP metrics defined in section 5.1.2, in a high-speed scenario with an average speed of 30 handovers per minute. In this test, the MNs will perform the previously described move-

ment that covers the entire domain, back-and-forth, resulting in an average inter-handover interval of 2 seconds.

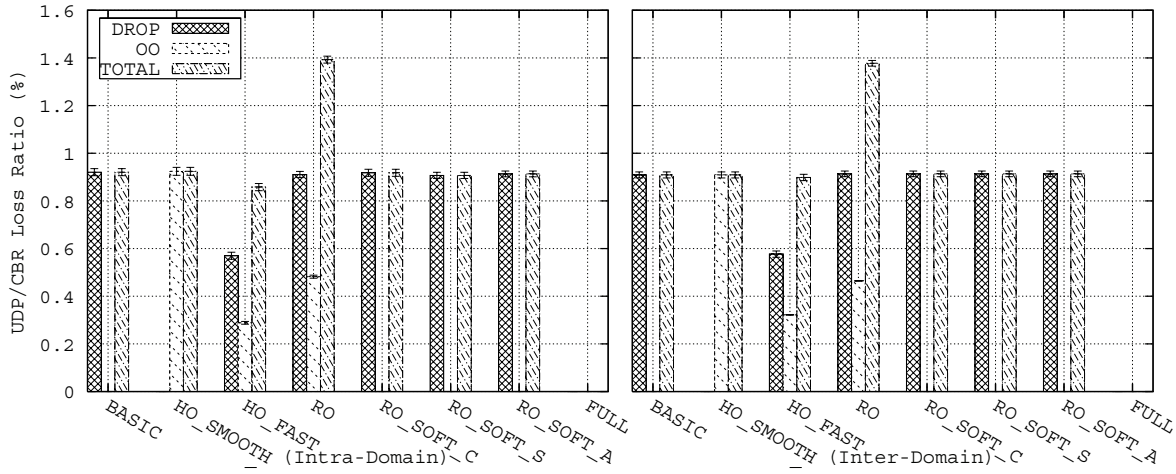


Figure 133: CBR/UDP Loss Ratio

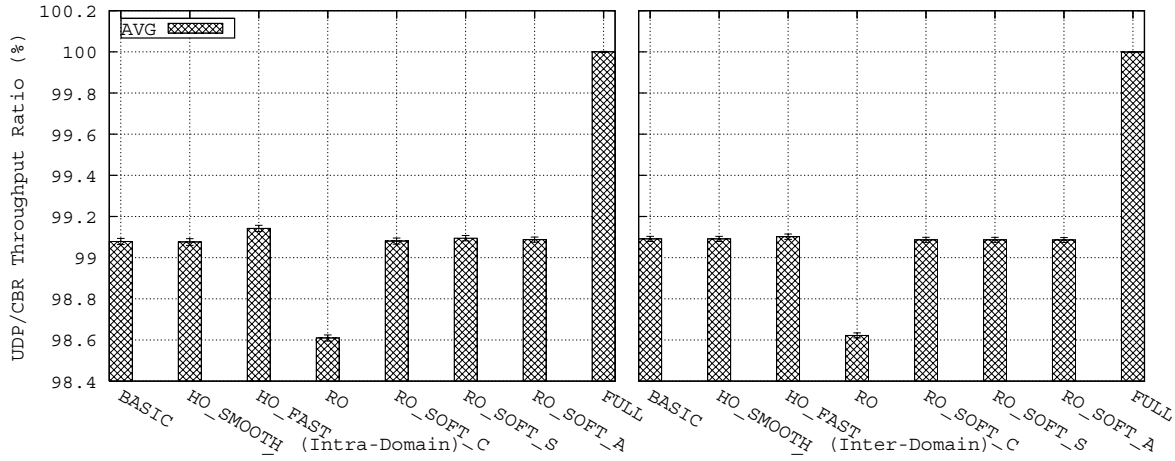


Figure 134: CBR/UDP Throughput

The graph of Figure 133 shows the total loss ratio separated by dropped packets and out-of-order packets, for the fixed average high-speed MN movement, and shows in detail the drop/out-of-order losses relation mentioned in the previous section.

Regarding the isolated use of the smooth handover extension (HO_smooth), this extension is able to buffer the in-flight packets that are being received at the previous AR while the basic handover proceeds normally. As explained previously, the regular basic handover operations will redirect the in-flight packets at the crossover node, located somewhere in the agent tree. After updating the crossover node, the update packet proceeds down the tree to update the agents up to the previous AR. When the previous AR is updated with the new basic location of the MN, it releases the buffered packets in the received order, and stops the buffering process. These packets are sent to the MN by the normal basic routing mechanisms – i.e. using the tree. However, such “salvaged” packets will be routed in the exact same paths as the in-flight packets to the new MN location – i.e. using the tree. Thus, both flows are blended with each other. While such mechanism does not cause any dropped packets, the salvaged packets will be delivered out-of-order, which causes the previously described problems. In order to solve this problem, it is necessary to deliver all buffered packets to the MN before the delivering to it the new packets.

The isolated use of the fast handover extension (HO_fast) is able to process the previous AR notification in a faster way, by informing all physically adjacent ARs of the new MN location. While previously received packets were already lost (as the test considered the isolated usage of HO_fast without buffering), the subsequent in-flight packets received at the previous AR can be redirected to the new location. However, most of these will actually be delivered to the MN out-of-order, for the same previously detailed reasons. Thus, the isolated usage of the fast handover results in a lower amount of dropped packets than the ones in the basic handover, and the reception of most (but not all) salvaged packets out-of-order, having been blended with the new packets sent directly to the new AR.

While providing major stationary benefits (section 7.1.2), the isolated use of the RO extension adds an additional number of out-of-order packets at each handover to the amount of dropped packets of the basic handover, a result of its direct de-triangulation scheme. This problem can be addressed by any of the three smooth de-triangulation mechanisms, which are able to avoid this out of order packets phenomenon.

Finally, the combined usage of the handover extensions, featured in the "full" protocol, is able to achieve a zero-loss handover without any throughput impact, by combining the smooth handover, the RO, and a smooth de-triangulation mechanism. As described previously, such happens because the in-flight packets are buffered at the old AR (HO_smooth) until it is quickly notified of the new MN location (HO_fast extension), and are then sent in an ordered form to such location using a local tunnel (RO extension), which is additionally removed without causing out-of-order packets (any of the smooth de-triangulation extensions).

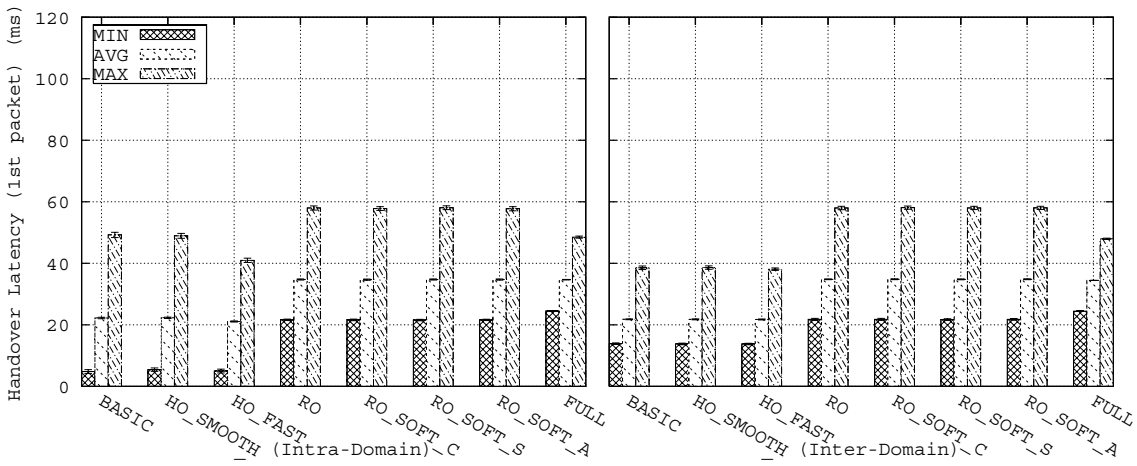


Figure 135: Handover latency - time of first packet received through the new AR

The graph of Figure 135 further reinforce the previous conclusions and characterize the eTIMIP extension's handovers, by showing the handover latency values for all considered extensions. The graph shows that the usage of RO results in higher and constant average handover latency in all cases. That happens because, in contrast with the basic handover or its non-RO extensions, the first packet is only redirected to the new AR after the previous AR is notified of the new MN location. In turn, such happens because the packets are sent directly via a tunnel from the ANG/AR to the previous AR, bypassing the tree completely, and thus being unable to be redirected earlier inside the tree as the regular basic routing does. It should be noted that such design behaviour is one of the key points that avoids packet reordering during a handover without having to support packet marking or separate buffers like the solution proposed in reference [52].

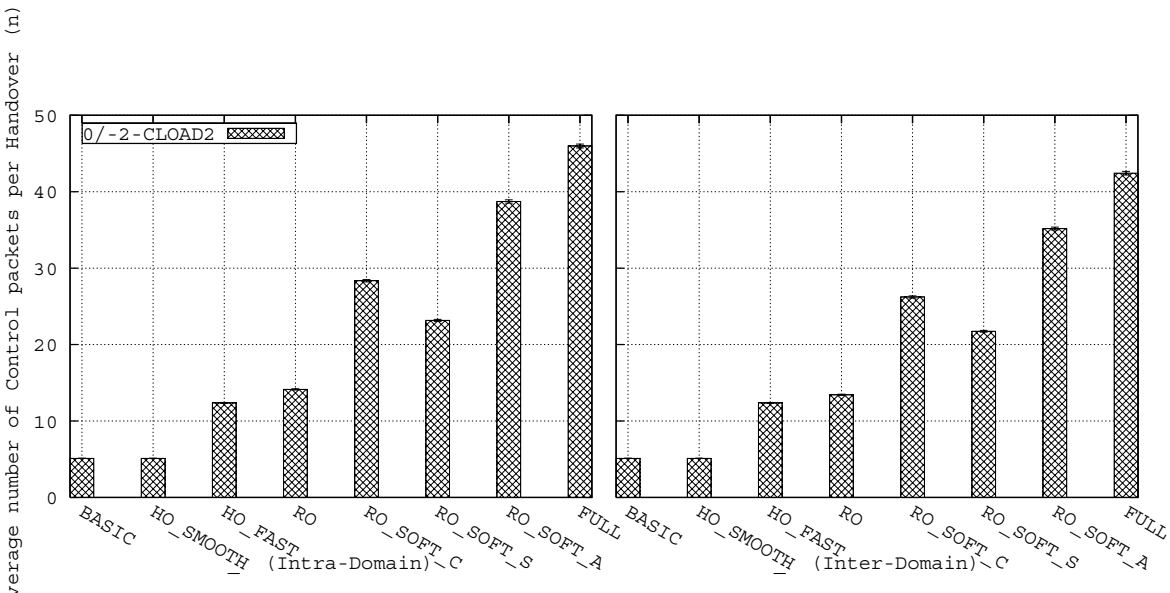


Figure 136: Control load - reference case - eTIMIP extensions

Figure 136 shows the average control load of the extensions that support the improved handovers and optimal routing, by measuring the average number of forwarded control messages. Here, it is clear that as each particular extension requires their own control messages, the final control load will be fairly higher when compared to the regular basic handover, especially for the "full" protocol.

This happens because the necessary messages cannot be multiplexed in shared control messages either by having to be sequentially generated, or for handover latency improvements. However, it should be stressed that all these control messages are short in nature (requiring an maximum of 44/88 bytes (IPv4/IPv6), as described in section 4.3), and are always limited to the infra-structured part of the domain, which is typically supported using high-bandwidth wired links, instead of being propagated to the wireless medium (e.g. fMIP [34], BCMP [25] and other similar protocols).

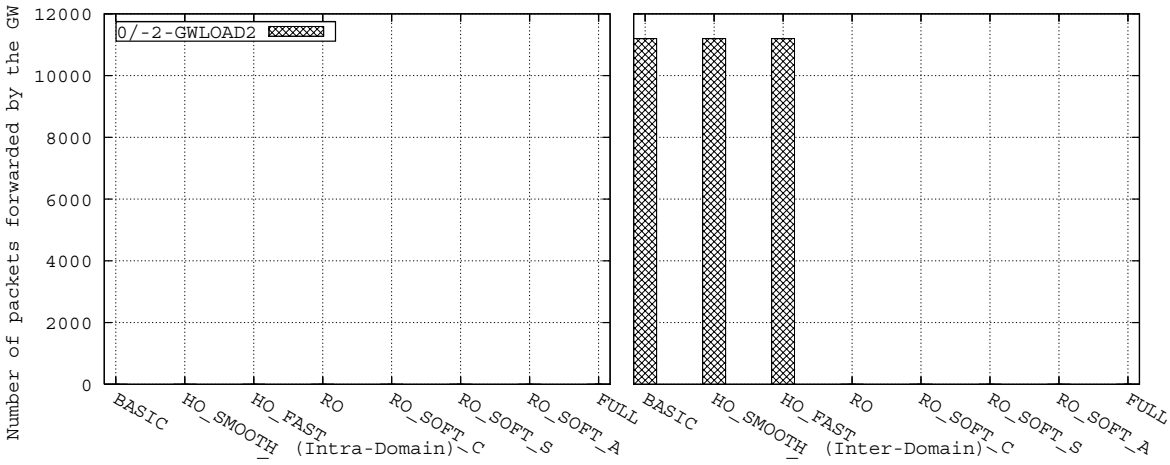


Figure 137: Data load on the GW - reference case - eTIMIP extensions

Figure 137 shows the average data load on the GW, by measuring the number of data packets forwarded by this central point. While eTIMIP's basic routing already managed to not involve the GW for intra-domain traffic forwarding, the inter-domain traffic was still handled as all the other protocols. However, the usage of the RO option, which enables direct optimal routing from the ANG to the LMN's AR, also has the important benefit of not forcing the inter-domain traffic's data packets to necessarily be forwarded by the single GW, off-

loading this central node. In conjunction, this feature is able to decouple the data from the control paths, as the GW tends to perform control functions only, with important scalability gains.

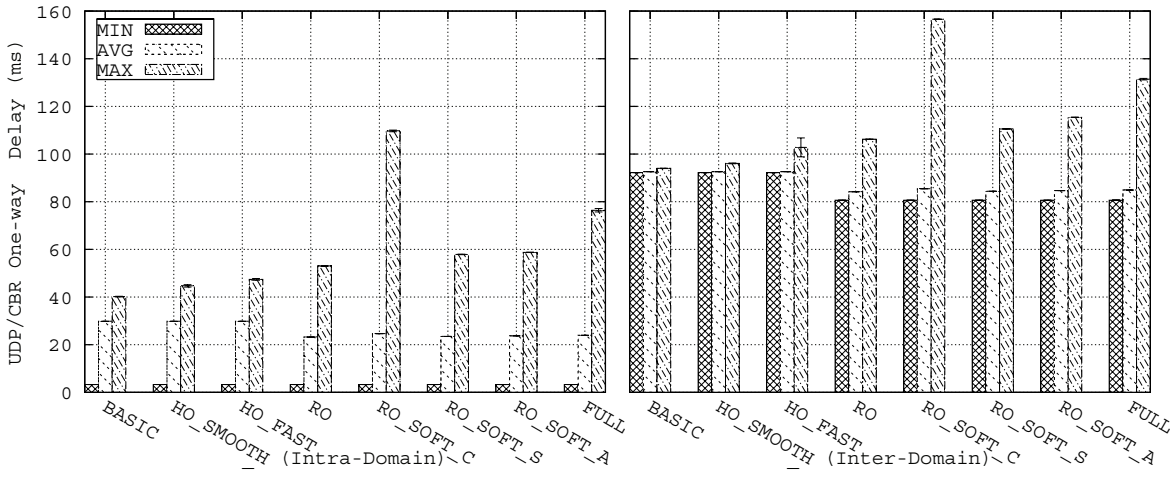


Figure 138: One way delay - reference case - eTIMIP extensions

Figure 138 shows the average one way delay imposed to the data packets for all handovers considered in the movement of this reference scenario. While the average delay values are essentially the ones already measured in the stationary conditions (Figure 58, page 105), the maximum value can show the differences between the various smooth de-triangulation options.

In fact, the graphs clearly show that the conservative smooth de-triangulation extension imposes a much higher maximum delay for the triangulated data packets, through the usage of the operations described previously that actively wait for the reception of the last packet through the triangulated path before sending the new packets through the direct path. However, both the symmetric and the asymmetric algorithms are able to perform the same zero-loss de-triangulation without imposing further delays either to the triangulated or to the packets sent directly, resulting in a much closer maximum delay value, which is similar to the one experienced by the RO extension.

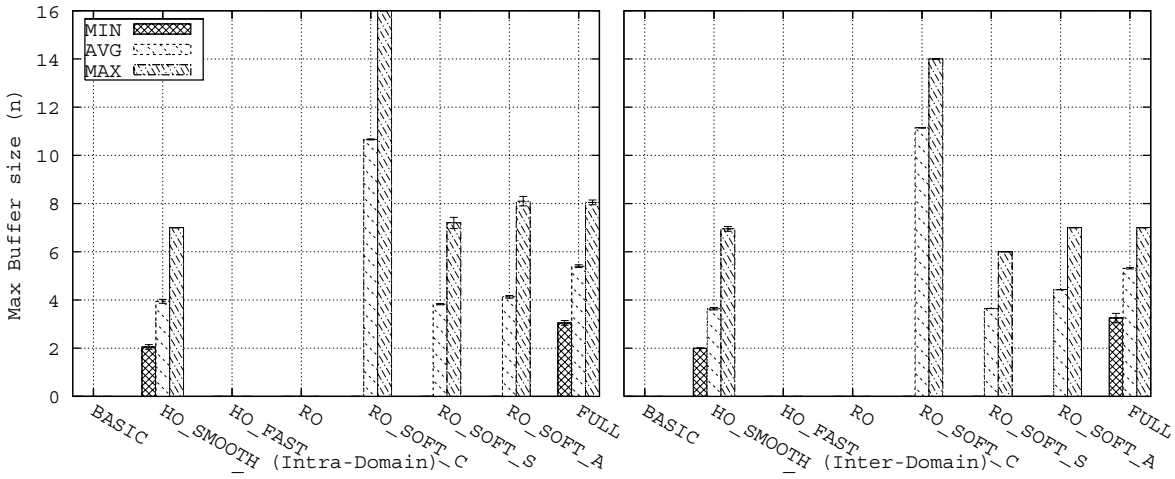


Figure 139: Maximum Buffer usage - reference case - eTIMIP extensions

Finally, Figure 139 considers the maximum buffer size of all eTIMIP agents per handover, and plots the minimum, average and maximum values for the movement considered in the

reference scenario. As described previously, the buffer mechanisms are used by the smooth handover mechanism and by the smooth de-triangulation mechanism. For smooth handover, the buffer is used until the previous AR is notified of the new LMN location; although not shown, the combination of the smooth with the fast handover extension is able to reduce this value, by notifying the previous AR at an earlier stage. For the smooth de-triangulation, the buffer is used for the time delay between the reception of the buffer and flush messages. Here, the same conclusion of the maximum delay graph can be verified, as the most advanced smooth de-triangulation algorithms impose a much lower buffer requirement than the base one.

In conclusion, by supporting zero-loss handovers with a low-latency handover support, the combination of all the eTIMIP extensions (protocol "full") is able to achieve a seamless handover support, which is defined as a low-loss, low-latency handover [23][21].The former is achieved through the support of a zero-loss handover, which is only limited by the available buffer space until the previous AR is notified of the new MN location; the latter is achieved through the support of a fast handover, where the handover occurs approximately after the time needed to directly notify the previous AR and to transfer the buffered data packets through the direct local tunnel. Although the resulting handover latency is higher than the one achieved by the basic routing (an increase that is only more pronounced in the best cases, and less pronounced in the worst cases), the resulting handover latency is still well within the accepted bound of fast handovers [23], and of the best alternative protocols.

7.1.5 Localized movement - best and worst cases

This test modifies the reference scenario to make the MNs perform their movements localized in a certain location of the domain, in order to recreate the previous test that studied both the best and the worst eTIMIP cases. In the former, the MN will perform its handovers between AR3 and AR4, which are located in the same sub-tree. In the latter case, the MN will perform its handovers between AR4 and AR5 which are located in the opposite parts of the main tree.

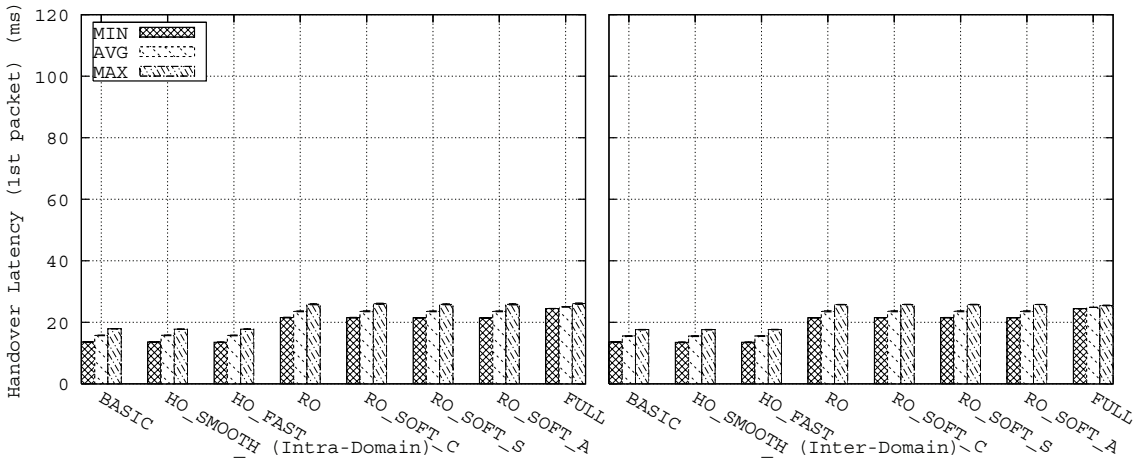


Figure 140: Localized handovers – handover latency - best case

21 Quoting from El Malki [23], page 4, "Seamless handoff - L3 handoff that is both low latency and low loss".

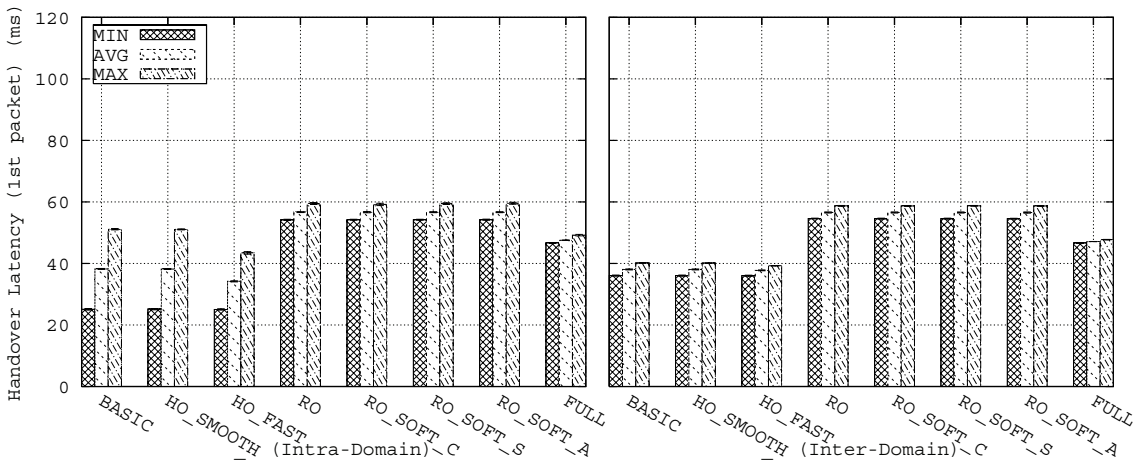


Figure 141: Localized handovers – handover latency – worst case

Figure 140 and Figure 141 show the handover latency results for the two opposite cases, which confirm the previous conclusions taken from the reference scenario adapted to localized handovers. The major point which this simpler experiment is able to show is how the HO_fast extension lowers the handover latency by creating the local tunnel at an earlier time. Thus, the major improvements occur in the worst case, when using the local tunnels from the RO extension (i.e. comparing the RO and the Full series in Figure 141). In that case, the achieved latency is only limited by the existing direct links, which account for a minimum of 3 hops for the handover message to be propagated, plus the hops for the forwarding of the data traffic between the ARs.

7.1.6 Wired Load Effect - Continuous movement

This test recreates the previous test that modifies the reference scenario by introducing increasing amounts of load on all wired links, in order to study the effect of varying link delays due to queuing for the various eTIMIP extensions. Again, the update packets do not have any kind of QoS priority preference, being treated like the regular data packets, which are treated under the Best-Effort traffic class.

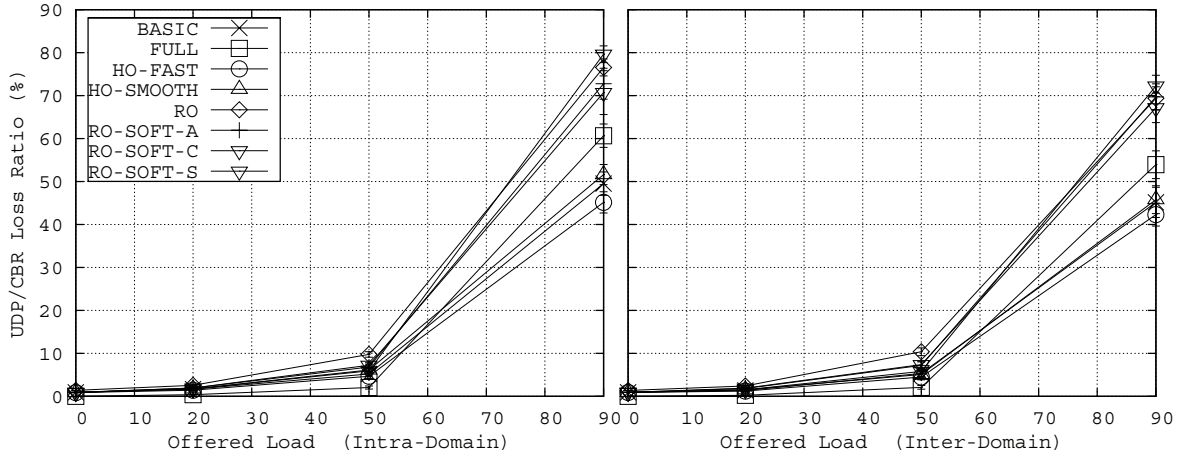


Figure 142: Wired load effect - total average loss ratio

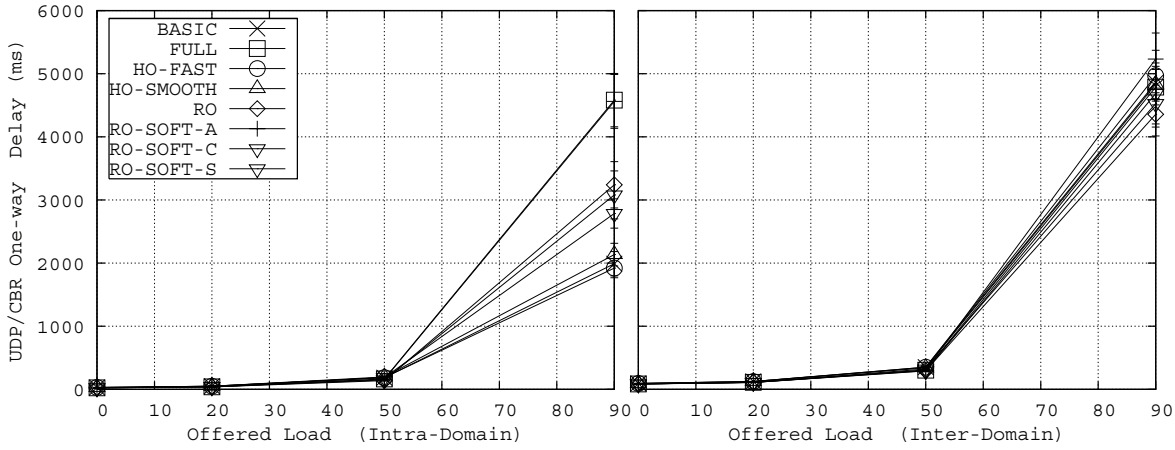


Figure 143: Wired load effect - total average one-way delay

The figures show the corresponding loss and delay values for the same amounts of wired link loads considered previously. Up to large values of link load (50%), the studied eTIMIP extensions show good results similar to the ones in the basic eTIMIP protocol. Only when massive loads are applied (90%), the efficiency drops for all combinations, for the same reasons already described. In this scenario, the simple HO_Fast extension has the best efficiency of all the studied combinations, as it features the simplest handover scheme with the least signalling update messages.

7.1.7 Transparency vs. efficiency - Number of Agents

This test will repeat the previous test that evaluated the impact of different numbers of agent levels on the eTIMIP extensions. Again, the number of agent levels will vary from a minimum degenerate tree-less scenario, where the mobility agents are only located in the single GW and all the ARs, up to the full agent tree that was studied in the reference scenario.

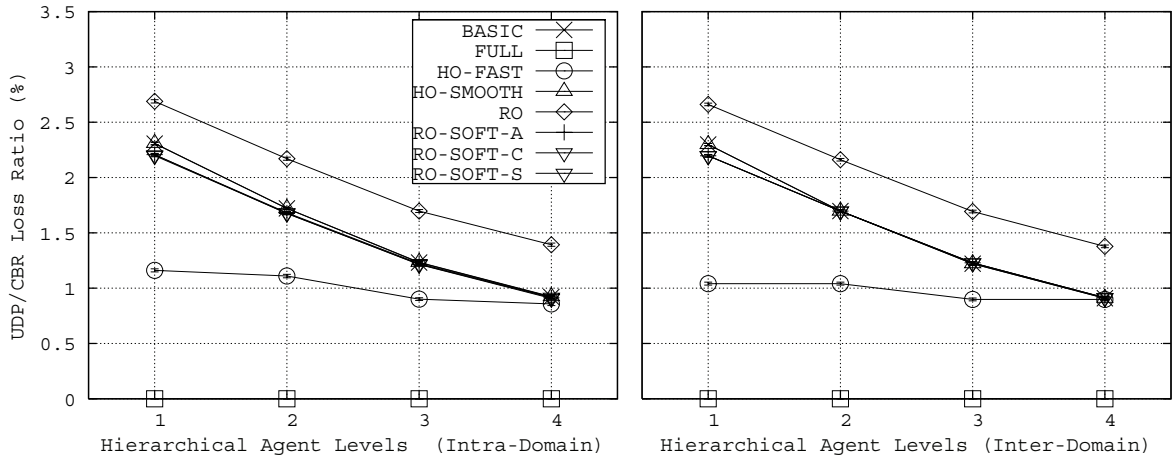


Figure 144: Number of agent levels (1 – GW + ARs only / 4 – Full agent tree) – Loss ratio

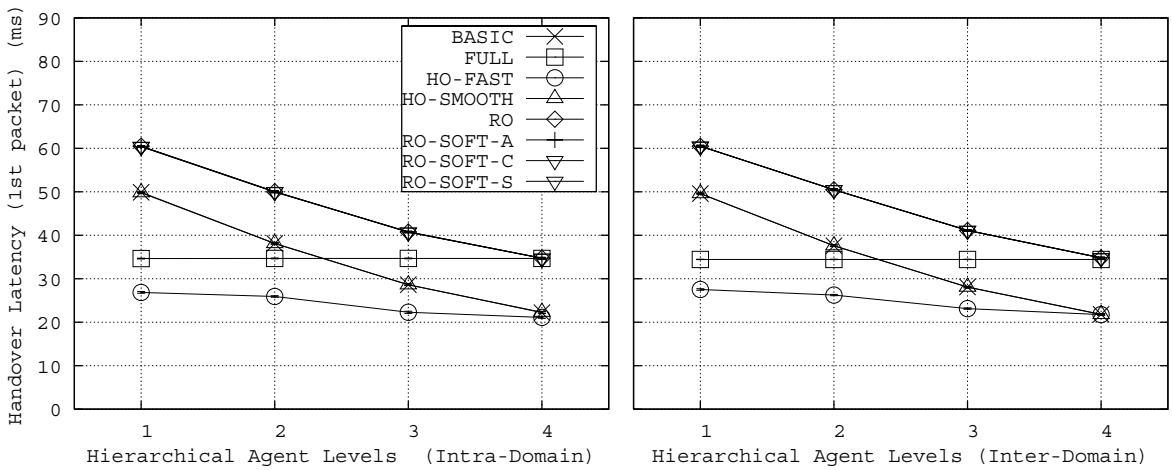


Figure 145: Number of agent levels (1 – GW + ARs only / 4 – Full agent tree) – Handover latency

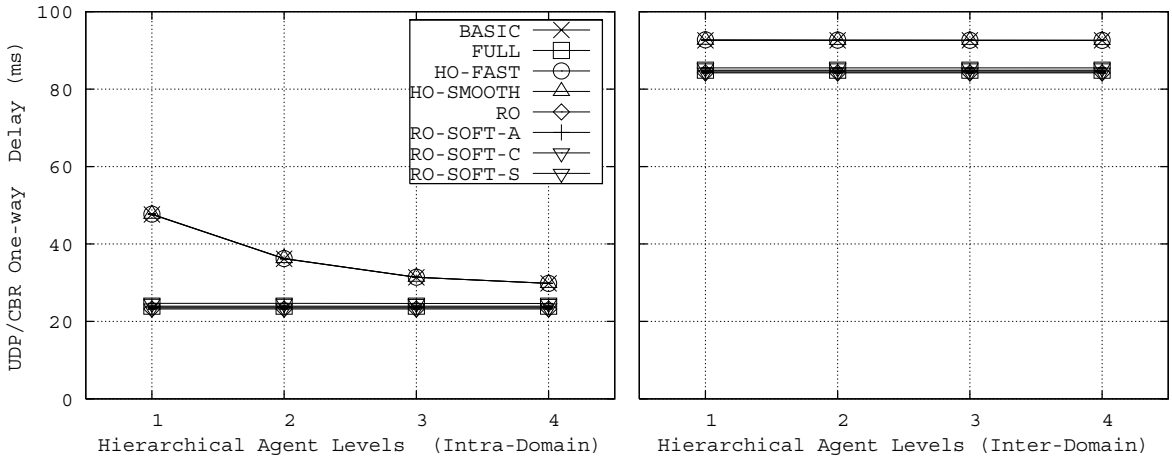


Figure 146: Number of agent levels (1 – GW + ARs only / 4 – full agent tree) – one way delay

The figures show that, while basic eTIMIP already provides the previously discussed efficiency to transparency trade-off, its extensions enable the support of both high transparency and high efficiency **simultaneously**. This happens because various optimized mechanisms of the “full” protocol are independent of the number of agent tree levels, resulting in a straight line in the graphs.

Specifically, while the loss ratio of the basic protocol only improved while the transparency level decreased, the full protocol supports a zero-loss handover for all situations (Figure 144). Regarding the handover latency, while the basic protocol only improved this metric as long as the transparency level decreased, the full protocol supports a constant low-latency handover in all situations (Figure 145). Finally, while the one-way delay metric of the basic protocol only improved while the transparency level decreased, the full protocol supports an optimal routing delay for all situations (Figure 146).

7.1.8 eTIMIP degenerate tree test

This test further details the important eTIMIP case where a degenerate tree is used instead of a partial or full agent tree mesh, which has important transparency and reliability gains. As already described, even in such scenario, the “full” protocol results in a seamless handover and optimal routing; this test will make clear that such results are possible using an higher control load, and the use of buffering at both the previous AR and the crossover node.

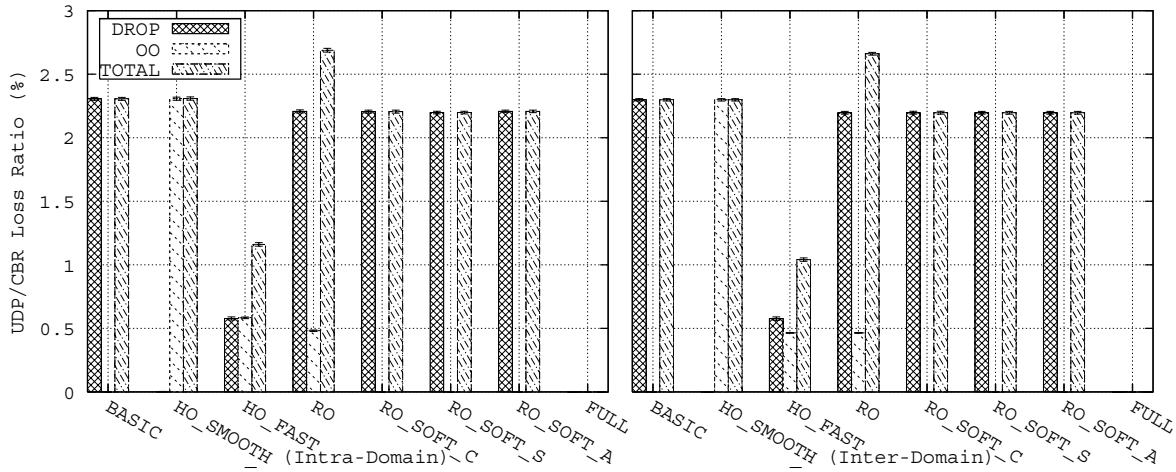


Figure 147: eTIMIP extensions in a degenerate tree – Loss Ratio

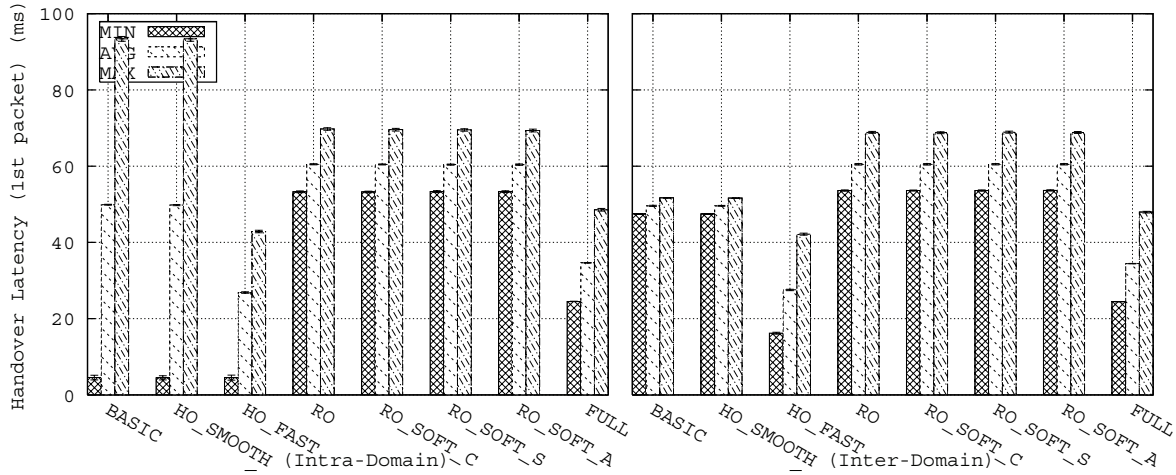


Figure 148: eTIMIP extensions in a degenerate tree – Handover Latency

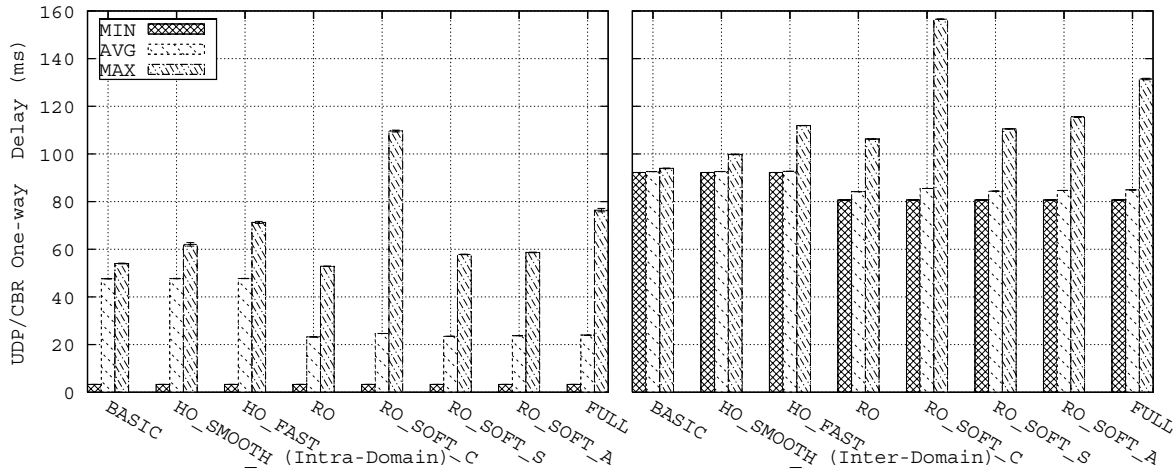


Figure 149: eTIMIP extensions in a degenerate tree – One-way delay

The results, presented in Figure 147 to Figure 149 , show that the loss, throughput and delay all confirm the previous conclusions in greater detail, being additionally possible to verify the minimum and maximum values of the handover latency and one-way delay in the eTIMIP degenerate case.

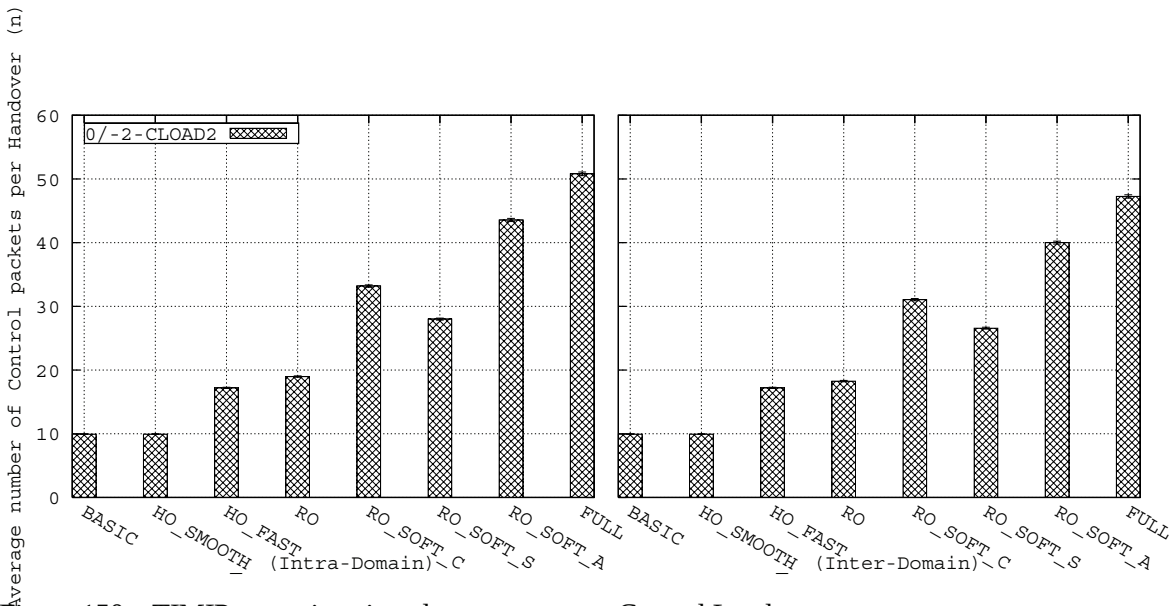


Figure 150: eTIMIP extensions in a degenerate tree – Control Load

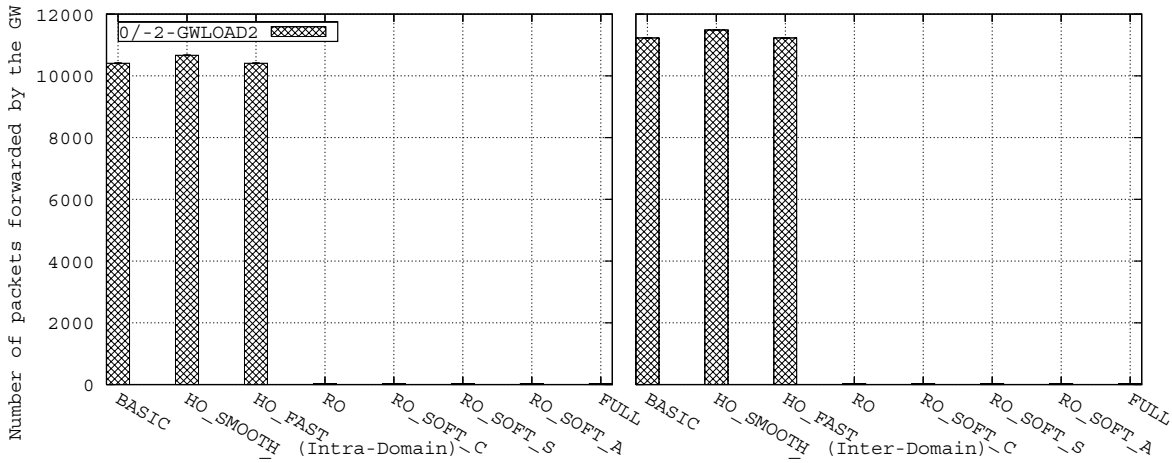


Figure 151: eTIMIP extensions in a degenerate tree – Data Load at the GW

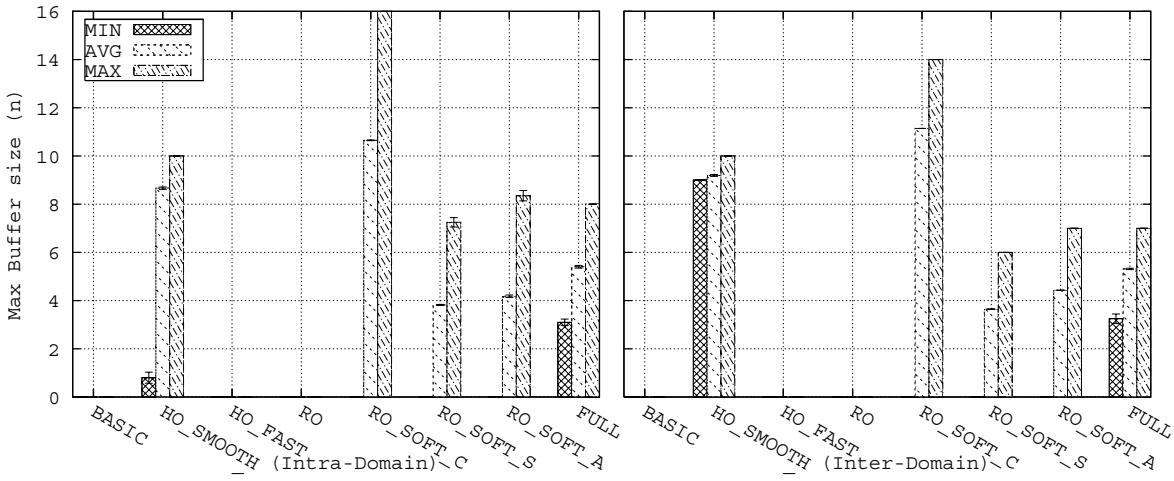


Figure 152: eTIMIP extensions in a degenerate tree – Maximum buffer size

The control load, data load and buffer size figures are also consistent with the previous conclusions. Again, the “full” protocol has the highest control load, which is nonetheless limited to the wired part of the domain (Figure 150). As the test uses a degenerate tree, non-RO intra-domain traffic passes through the GW again. This is solved for all cases using the

RO extension, as discussed previously (Figure 151). Finally, the buffer usage (Figure 152) is similar to the reference case in the degenerate tree (previously shown in Figure 139), as the mechanisms that make use of it (HO_smooth and smooth de-triangulation) are independent of the number of agent levels.

7.2 eTIMIP extensions tests using TCP data traffic

7.2.1 Discrete Isolated handover

This test will illustrate what happens during an illustrative handover of the full eTIMIP extensions, but using TCP traffic, mirroring the previous test of section 5.3.1.

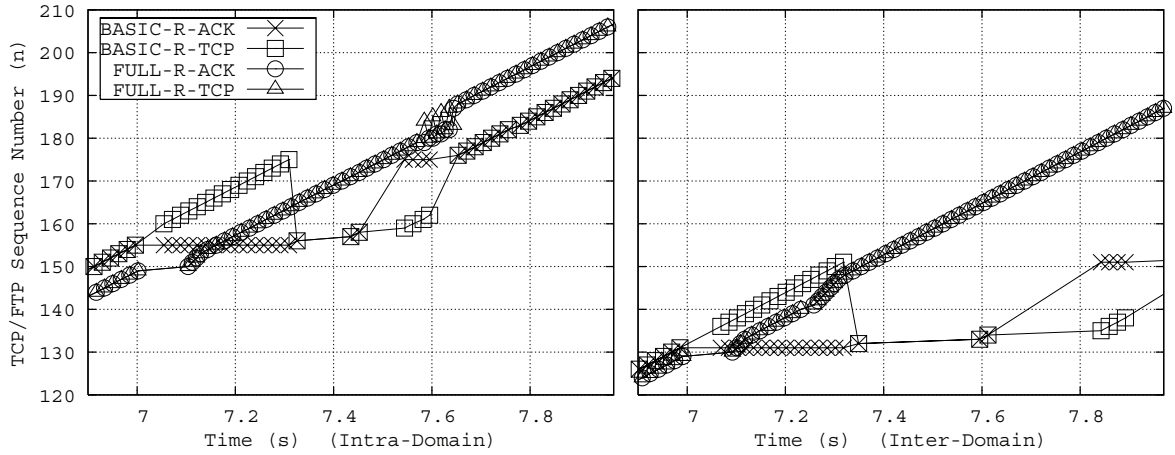


Figure 153: Isolated handover – TCP segment and acknowledgements at mobile receiver

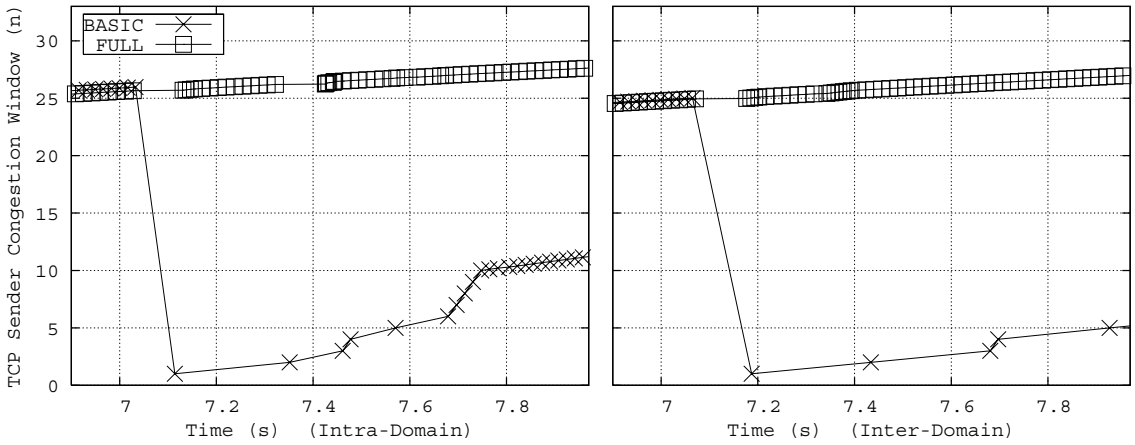


Figure 154: Isolated handover TCP time series – Congestion window at fixed sender

Figure 153 shows a time diagram that relates the time instants of the reception of TCP segments to generated TCP acknowledgements at mobile receiver, for both intra and inter-domain cases, while Figure 154 shows the corresponding congestion window value at the fixed sender, which regulates the generation of said TCP segments.

The comparison of both basic and full protocol handovers for TCP show that, while the former suffers from the problems previously pointed out, the latter is able to deliver the TCP segments without drops or reordering, in a way that prevents the TCP receiver from sending duplicate acknowledgements. Thus, the regular reception of the acknowledgements at the

sender results in an increasing congestion window without any timeouts, which avoids the throughput degradation discussed previously.

Specifically, when the handover occurs, the “full” protocol performs all the necessary operations for redirecting the buffered packets to the new AR using the local tunnel. Thus, after a short latency, the MN receives the burst of TCP segments without any drops or reordering, being this phenomena undistinguishable of any queuing that happens in any packet-switched network. As the TCP receiver gets all packets without drops or reordering, it continues to generate an increasing acknowledgement per received TCP segment (Figure 154). In turn, when the TCP sender receives it, it continues to increase the congestion window, avoiding the throughput degradation of basic routing. Later, the smooth de-triangulation mechanism of the full protocol is able to remove the local triangulation phenomenon without dropping or reordering the flow; as such, this operation is also unnoticed to the receiver, avoiding any throughput degradation.

7.2.2 Continuous Movement (multiple MN speeds)

This section presents simulations using continuous MN movements at varying speeds, by varying the time between the handovers, in the same conditions as in the previous tests of section 5.3.2. The respective results are shown in Figure 155, representing the average throughput for each MN speed, and Figure 156, representing the average TCP overhead required for instantiating a reliable service.

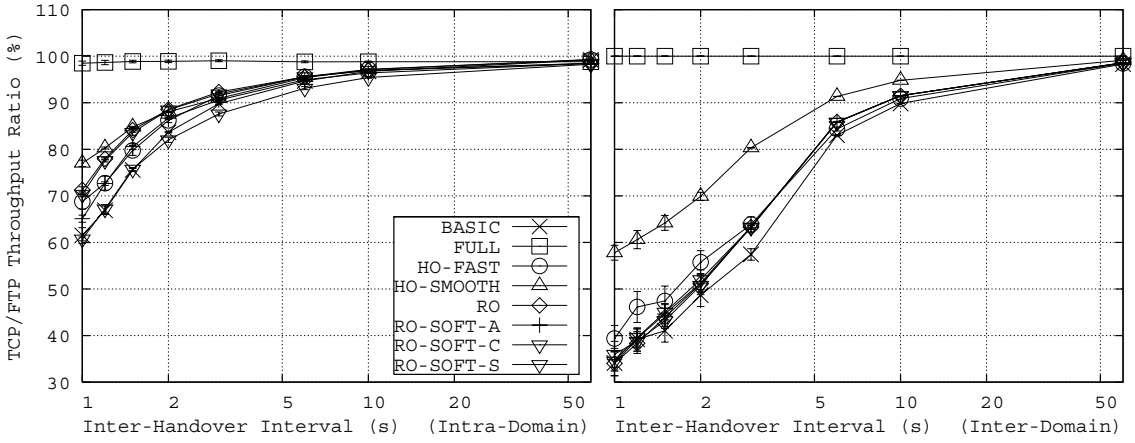


Figure 155: FTP/TCP Average Throughput per inter-handover interval

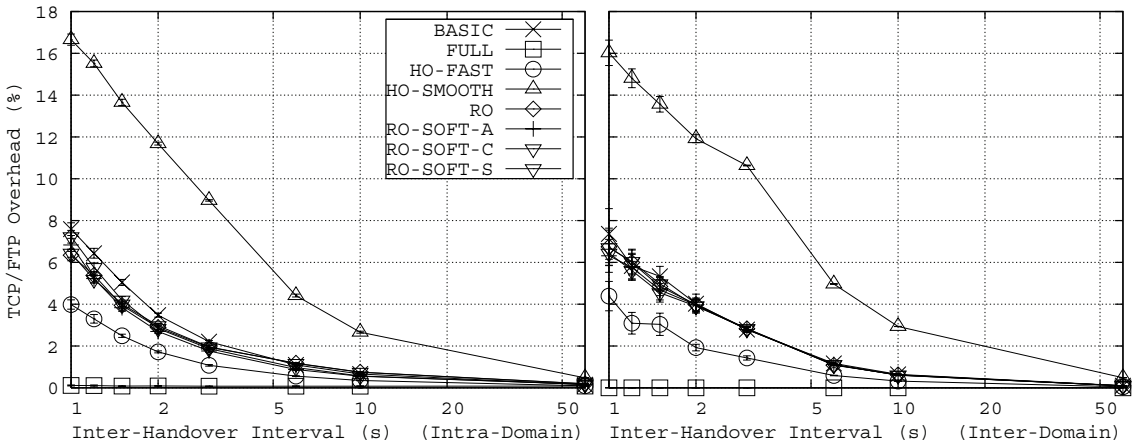


Figure 156: FTP/TCP Average Overhead per inter-handover interval

The graphs show that the individual extensions are unable to provide a seamless, unnoticed TCP handover, mainly due to introducing packet drops. By only requiring that the packets are reordered and not dropped, the HO_smooth extension achieves the lowest throughput degradation of the individual extensions, but it also achieves the worst overhead result due to its larger out-of-order ratio, which forces the unnecessary retransmission of all involved packets during the handover. Interestingly, the HO_fast induces the worst throughput degradation of the individual extensions, but requires the smallest amount of overhead. This happens because, even though it induces less dropped packets than the basic protocol, the salvaged out-of-order packets will confuse the sender and force it to generate duplicate acknowledgements.

Again, the combination of all extensions ("full") is able to provide a seamless handover service even at very high speeds, through its 100% throughput / 0% overhead in all speeds for the previously explained reasons.

7.2.3 Reference Scenario (single average high MN speed, non-localized handovers)

This test recreates the reference scenario with TCP traffic, in order to better study the impact of the handovers in this type of traffic in high speed scenarios (30 handovers/minute).

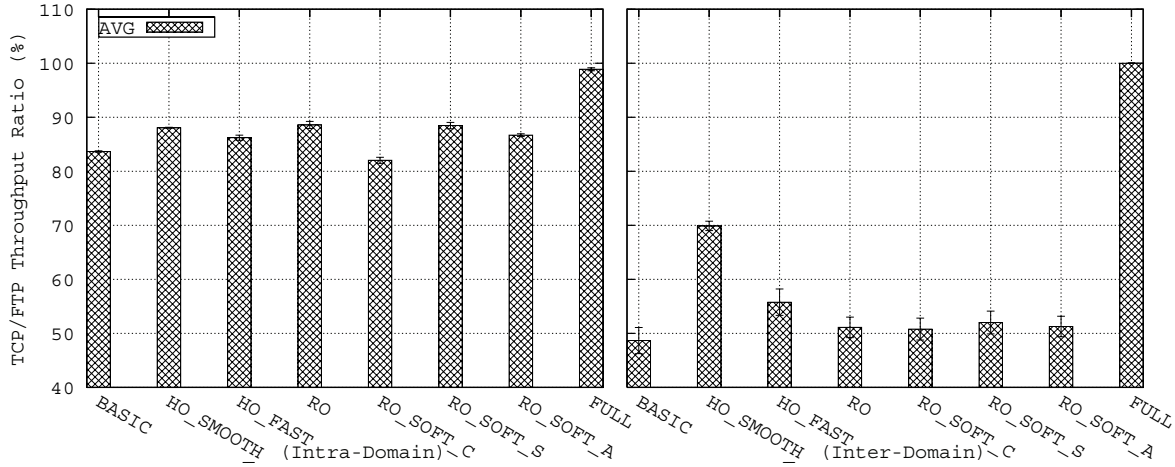


Figure 157: FTP/TCP Average Throughput – reference case

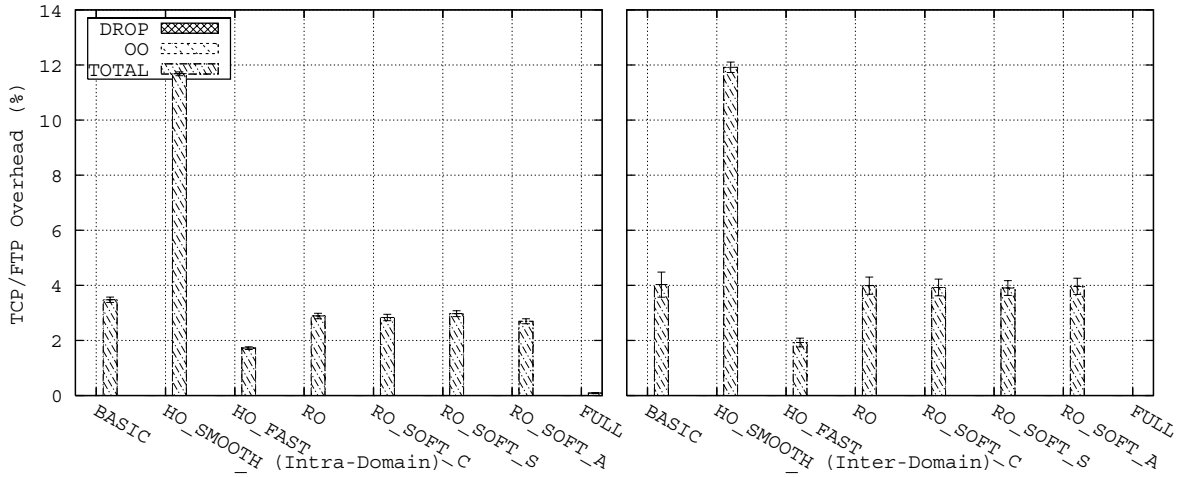


Figure 158: FTP/TCP Average TCP Overhead – reference case

The graphs of Figure 157 and Figure 158 show the average throughput and overhead for the case of the reference scenario. Here, all the previous conclusions are maintained, the differences of the extensions explained previously being clearer.

All the other metrics produce results that are similar to the ones in the test of the reference case with UDP traffic (section 7.1.3).

7.2.4 Wired Load Effect - Continuous movement

This test recreates the previous test that modifies the reference scenario by introducing increasing amounts of load on all wired links, in order to study the effect of varying link delays due to queuing in the particular case of TCP traffic. Again, the update packets do not have any kind of QoS priority preference, being treated like the regular data packets, which are treated under the Best-Effort traffic class.

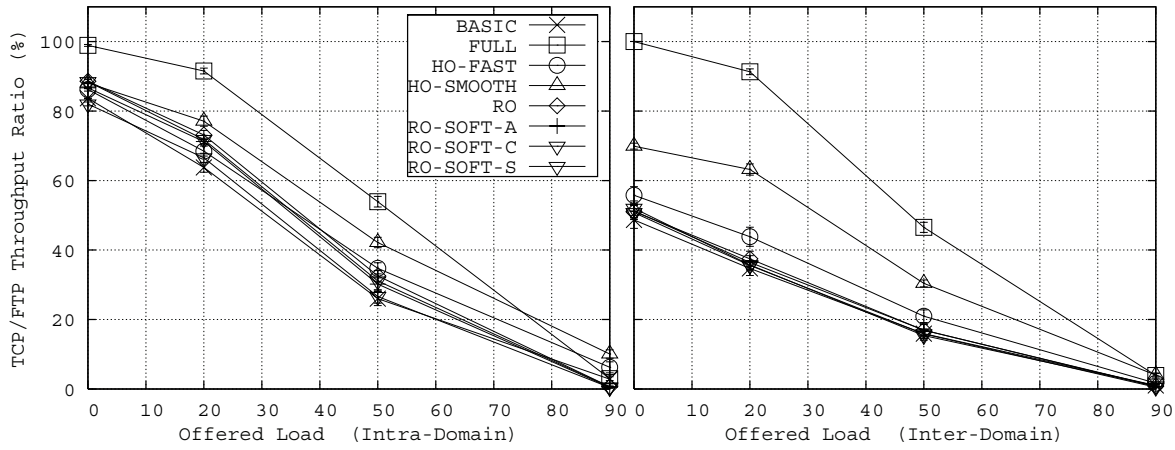


Figure 159: Wired load effect – TCP - total average throughput ratio

Figure 159 shows the corresponding throughput ratio for the same amounts of wired link loads considered previously. Under moderate loads, it can be seen that the full and, with an intermediate amount, the HO_smooth extensions are much more robust to support wired link load than the basic protocol or the other combinations. This happens because such extensions are able to avoid packet drops, which are very expensive to recover in such higher RTTs scenarios.

7.2.5 Transparency vs. efficiency - Number of agents

This test will repeat the previous test that evaluated the impact of different number of agent levels on the eTIMIP extensions, in the case of TCP traffic. Again, the number of agent levels will vary from a minimum degenerate tree-less scenario, where the mobility agents are located in the single GW only and all the ARs, up to the full agent tree that was studied in the reference scenario.

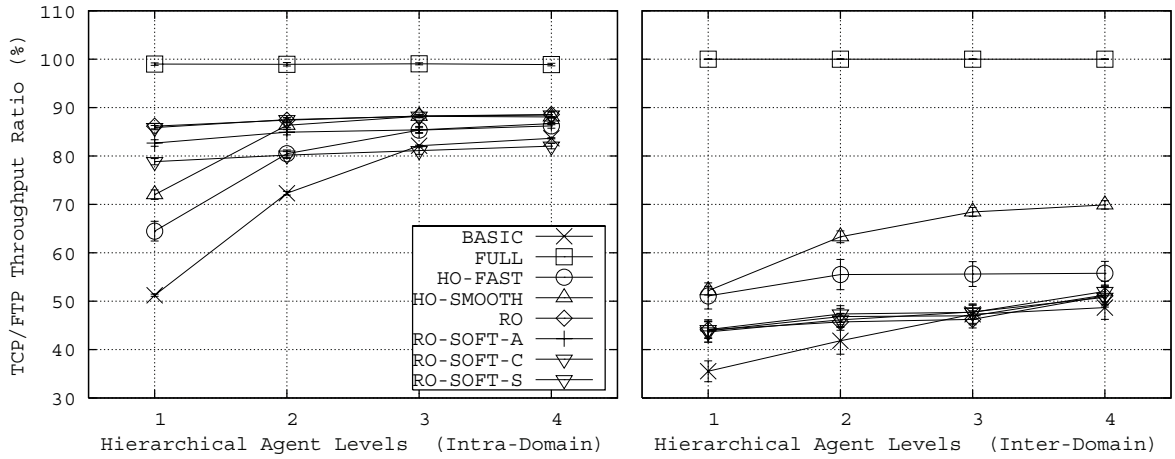


Figure 160: Number of agent levels (1 – GW + ARs only / 4 – full agent tree) – Throughput ratio

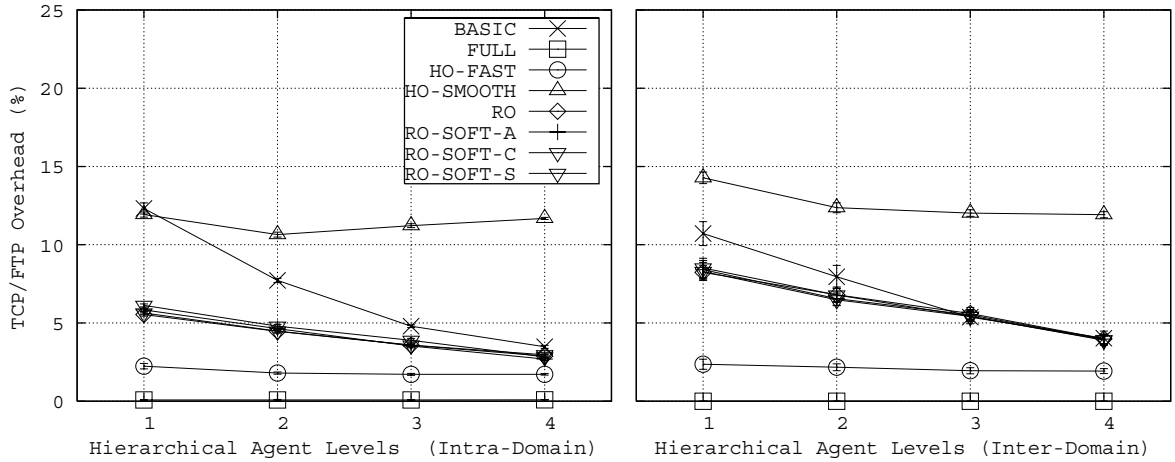


Figure 161: Number of agent levels (1 – GW + ARs only / 4 – full agent tree) – TCP Overhead

The TCP results are totally consistent with the previous transparency-to-efficiency trade-off discussion done in section 7.1.7. Specifically, by using mechanisms that are not based on the tree, the full protocol features an excellent performance even in the presence of a degenerate tree, by supporting seamless handovers for TCP traffic. Regarding the isolated extensions, the HO_smooth extension is the most important component for improving the basic protocol's throughput, while the HO_fast extension is the most important component for improving the TCP overhead of the basic protocol.

7.2.6 eTIMIP degenerate tree test

This test presents in greater detail the previous results for the case where a degenerate tree is used with TCP traffic, the details discussed previously being clearer.

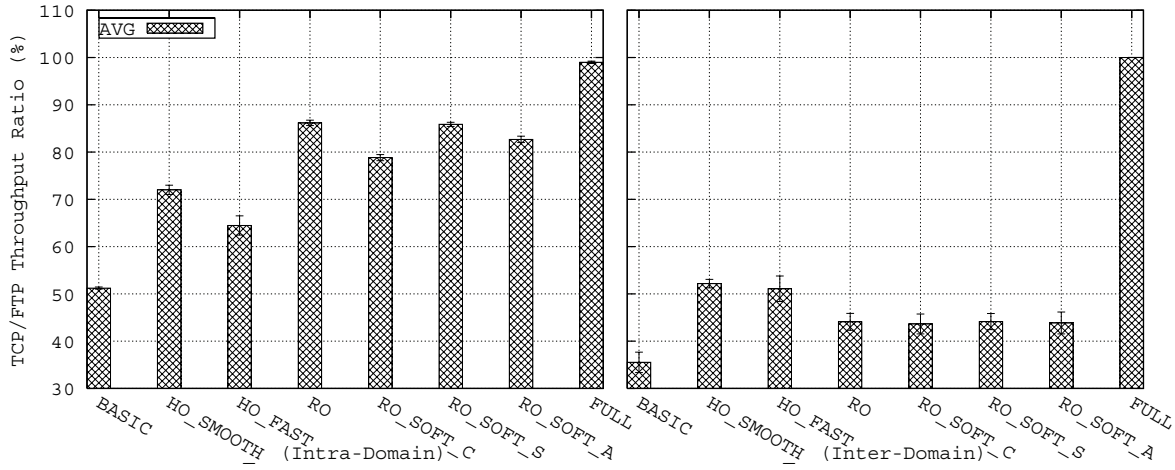


Figure 162: eTIMIP extensions in a degenerate tree – TCP Throughput

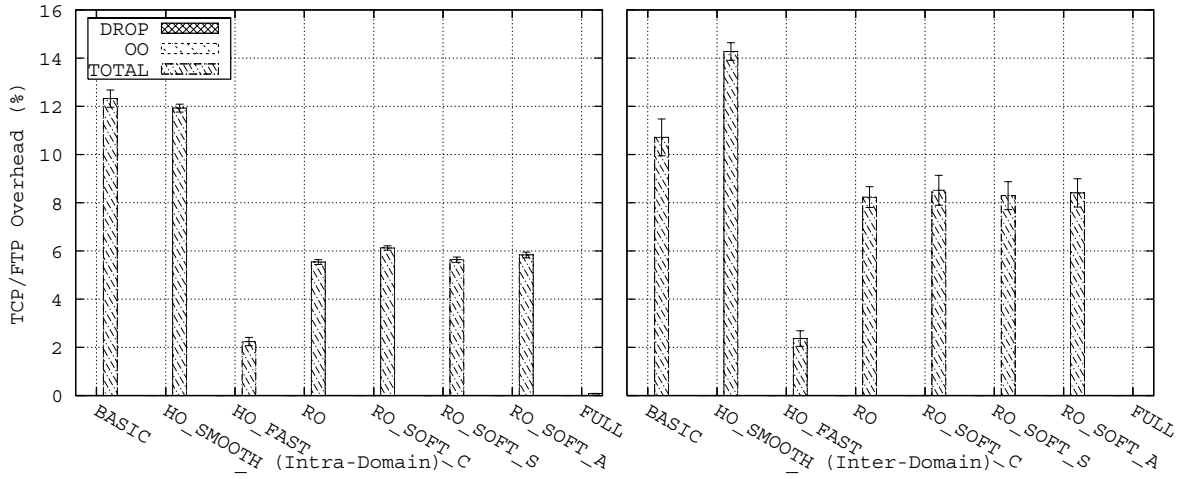


Figure 163: eTIMIP extensions in a degenerate tree – TCP Overhead

7.3 Full eTIMIP Conclusion / Discussion

This section summarizes the previous tests that compared the basic version of eTIMIP to its extensions, either in isolation or combined into a version of eTIMIP designated “full”. Again, such eTIMIP extensions will be analysed regarding the major components of handover efficiency, routing efficiency, transparency, reliability and scalability support.

Handover Efficiency

On the pro side, the full eTIMIP specification presents a seamless handover mechanism, which improves the original basic protocol’s handover with the support of a zero-loss handover mechanism and a low-latency handover support. This is achieved through the combination of multiple extensions, where the in-flight packets are buffered at the old AR, up to the time that this agent is quickly notified of the new MN location, and are then sent in an ordered form to the new AR using a local tunnel (which is additionally removed without causing out-of-order packets, nor increasing the packet delay). By leveraging the guaranteed basic handover with tree-independent mechanisms, such efficient support is made possible even when a degenerate agent tree is used, and in the case of handovers in the middle of the tree.

On the con side, the eTIMIP full routing handover is more complex, requires more control load and imposes a slightly higher handover latency when compared with the corresponding basic handover (being, nonetheless, still in the same magnitude order as the basic handover). However, such inherent characteristics of the handover scheme can be considered perfectly justified in the light of the achieved MN performance, where low-latency / zero-loss fast handovers are attained, and by the fact that all control packets are always limited to the wired part of the domain.

While basic eTIMIP provided one of the best handover performances in both UDP and TCP traffic types, at the expense of network transparency support through the use of a full agent tree, the full protocol provides the most efficient handover of all protocols, for all traffic types and without sacrificing transparency, by the use of tree-less mechanisms.

Routing Efficiency

On the pro side, the full eTIMIP specification presents an optimized routing scheme that manages to provide an optimal forwarding service by creating temporary direct tunnels between the involved agents. This feature enables the usage of the most direct paths inside the domain, benefiting both the packet delay and the resource usage, and offloads the internal tree agents, namely the single GW, from data forwarding services, by decoupling the data from the control paths and providing load-balancing support. Additionally, these RO routing paths are refreshed without any overhead, only through their simple usage.

On the con side, such efficient RO support for inter-domain traffic is only improved if mobility-aware agents (ANGs) are present on the borders of the domain, being the traffic forwarded to the GW by default if no such node exists.

While eTIMIP already provided the best round-up routing performance in both UDP and TCP traffic types, at the expense of network transparency support through the use of a full agent tree, the full protocol provides an even more efficient routing support for all types of traffic without sacrificing transparency, by the use of tree-less mechanisms.

Transparency support

By featuring exactly the same terminal and network transparency support as the base specification, full eTIMIP has the same transparency level as the basic protocol, which is greater than all the existing alternatives.

Reliability

On the pro side, the full eTIMIP specification leverages the existing basic eTIMIP reliability mechanisms, to provide a set of extensions that do not require complex reliability mechanisms, automatically reverting to basic mechanisms if control packets are dropped or race conditions occur. In addition, by supporting highly efficient handovers and routing using mechanisms that are not based on a tree as optimizations, eTIMIP can better support the case of a simpler or a degenerate tree, which improves its reliability against agent failures.

On the con side, the eTIMIP full specification continues to not support an agent reliability mechanism, which incurs in the problems discussed previously. However, such problem can be minimized using a simpler / degenerate tree that no longer impacts on the protocol's efficiency, and can be solved with the same mechanisms available for similar protocols discussed previously.

Compared to other alternatives, full eTIMIP provides an improved reliability level than the base protocol, being more reliable than alternative efficient micro-mobility protocols.

Scalability support

On the pro side, the full eTIMIP specification improves scalability by reducing the agent requirements while keeping efficient operation, and by proposing an idle terminals support, that results in fewer routing entries and higher refresh cycles. Also, the usage of the route optimization mechanism in combination with multiple access network gateways is able to provide load balancing capacities to the data traffic.

On the con side, the eTIMIP full specification requires additional signalling during its sequential handover operations. This can be a problem if a very high number of terminals are moving simultaneously, perhaps by having a conjugated physical movement, by resulting in increased overheads and handover latency. However, such signalling is always limited to the wired part of the domain only, and is composed of small control packets only.

Compared to other alternatives, full eTIMIP provides scalability similar to other alternatives.

8 Conclusions

This thesis has proposed a new heterogeneous IP micro-mobility protocol - **eTIMIP** - that advances the state of the art by simultaneously supporting increased efficiency and transparency than the best alternative state-of-the-art proposals. As such, eTIMIP is able to provide heterogeneous, efficient and transparent mobility support, suitable for a flexible application of the mobility service in particular deployment scenarios of both the existing and the future All-IP networks.

This is made possible as the eTIMIP protocol supports strong **efficiency** support, capable of enabling a low latency, low loss, low signalling and low data overhead mobility service through the support of seamless handovers and efficient resource utilisation, which do not conflict with the transparency support mechanisms.

In addition, this is also made possible as the eTIMIP protocol supports strong **transparency** support, capable of enabling immediate utilisation of the mobility service and smooth incremental upgrades of the existing infrastructure, through the support of mobility-unaware legacy terminals, legacy routers, and fixed legacy correspondent nodes, which do not conflict with the efficiency mechanisms either.

In addition, the protocol supports **modularity**, capable of enabling a flexible trade-off between the above-mentioned efficiency and transparency objectives, but also capable of enabling further scalability, reliability and control gains. For this, a generic architecture was defined, being defined a **basic protocol** that features better transparency with similar efficiency levels as the best alternative solutions. This generic architecture can also be used to support a number of eTIMIP extensions that form the **full protocol**, which can optionally and modularly be combined to provide better efficiency than the best alternative solutions, without sacrificing transparency, and to provide further scalability, reliability and control gains than the base protocol.

The eTIMIP basic and full protocol specifications are complementarily described through formal state machines, informal natural language and illustrative example descriptions. In addition, the specification is backed-up by the adaptation to IPv4 and IPv6 networks, by the description, evaluation and classification of the this field's state of the art through efficiency and transparency classification frameworks, and by the comparison of the base protocol with alternative proposals and its own extensions using simulation studies in the NS2 simulator.

8.1 Conclusions overview

This research work started by analysing the Next Generation Internet, where the major components that are required to be present in the future All-IP networks were identified - QoS, Security and Mobility. Regarding Mobility, it has been stated that the IP layer solutions emerge as the ones that are most suitable to provide mobility support to future heterogeneous mobile data networks, capable of featuring both scalability and efficiency in the mobility management mechanisms, coupled with transparency to the currently deployed Internet infrastructure. While the scalability problem has already been solved through the macro and micro mobility separation in the past, it has been identified that no state-of-the-art solution is able to support the efficiency and transparency aspects simultaneously and in a modular way, suitable for a flexible application of the mobility service in particular deployment scenarios

of heterogeneous all-IP networks. As such, a set of objectives were defined (**Efficiency** / **Transparency** / **Modularity**), that have been subsequently addressed through specific contributions.

Considering such objectives, this research work has firstly presented the current state of the art of the IP mobility field, for both macro and micro mobility. For this study, the existing standards have been described, alongside the selected proposals that are most related to this thesis' objectives. Some proposals, which are essentially compositions or minor extensions of the base protocols, were also briefly mentioned. In addition, the state-of-the-art study was also complemented with a description of representative above-IP mobility protocols that could also be used to support heterogeneous mobility support at the highest layers of the network stack.

In order to evaluate the possible contributions of previous research work on the solution to be proposed in this PhD thesis, an in-depth study was performed that analysed the state-of-the-art solutions' suitability concerning both the efficiency and the transparency topics. This was done using generic taxonomic frameworks, featuring multiple generic components and models, which classified and compared the selected state-of-the-art proposals for the fulfilment of this PhD thesis' objectives. In addition, this study was complemented by subjectively weighing the frameworks components to represent a possible view of the efficiency-to-transparency trade-off characteristics of each protocol in a single graph. From this study, it was concluded that the TIMIP protocol was the best-positioned state-of-the-art protocol to be extended to provide an efficient and transparent micro-mobility solution. Combined with the previously presented scalable and transparent sMIP macro-mobility protocol, the resulting eTIMIP/sMIP global mobility protocol is able to provide heterogeneous, efficient and transparent mobility support, suitable for a flexible application of the mobility service in particular deployment scenarios of both the existing and the future all-IP networks.

The proposed architecture was then described, providing modifications, optimisations and new support features that differentiated it from the earlier model, although backwards compatibility is maintained, if necessary. The eTIMIP architecture introduces a generic flexible mobility service based on an overlay network, which is able to separate the underlying physical network, with its existing legacy network elements and non-mobility-aware fixed routing, from the added or upgraded mobility-aware agents that create, among themselves, the mobile routing that supports an efficient and transparent mobility service.

Using this architecture, a base secure mobility service was formally defined (**basic eTIMIP**) that, by using an agent tree and a mobile subnet on the overlay network, supports a mobility service with better transparency and similar efficiency as the best alternative solutions of the state of the art. For the former, **transparency** is attained by supporting Legacy Mobile Nodes and Legacy Routers, where mobility-aware agents adjacent to the terminals enable the mobility support of unmodified LMNs; for the latter, **efficiency** is attained by supporting fast handovers, tree-optimal routing and an optimised state refreshment using the agent tree facilities. As such, basic eTIMIP is able to provide full transparency without sacrificing the already attained efficiency, enabling a flexible application of the mobility service in particular IPv4 and IPv6 deployment scenarios.

The next step of the research work focused on the evaluation of the basic eTIMIP proposal concerning both its efficiency and its transparency aspects. For this, the base protocol was studied, analysed and compared to selected state-of-the-art solutions through the use of simulation studies, modelled in the NS2 event-based simulator. Such studies considered both stationary and high-speed movement for multiple data traffic types, and compared basic

eTIMIP to other alternatives in scenarios that tested the seamless handover capabilities, the routing efficiency, the link failure reliability and the transparency-to-efficiency trade-off support. From these studies, it was concluded that basic eTIMIP presents an similar handover performance, a better routing capability and a better transparency capability than the alternative proposals. However, the same studies also identified that further handover and routing capabilities were required, and highlighted which components could be conditioning the observed eTIMIP efficiency.

Taking these identified shortcomings into consideration, an extended secure mobility service was formally defined (**full eTIMIP**), which supports a mobility service with better transparency and better efficiency than the best alternative solutions of the state of the art. This is made possible by a series of eTIMIP extensions that, using the same base architecture, extend the base protocol with mechanisms that complement it with the additional desired efficiency. In particular, a **route optimization** scheme enables the direct forwarding of the data packets between the domain edges, which, if used in conjunction with multiple mobility-aware points of attachment to the outside of the domain, can offload the internal agents of data forwarding functions, providing load distribution. In addition, a **seamless handover** scheme, which simultaneously supports a zero drop / zero out-of-order-handover while minimizing the total handover latency, is provided. As such, full eTIMIP is able to provide full efficiency without sacrificing the already attained transparency, enabling a flexible application of the mobility service in particular IPv4 and IPv6 deployment scenarios. In addition, the full protocol also improves scalability through a specific **idle terminals** support, and improves reliability and control through specific **operator-centric** scenarios.

The final step of the research work focused on the evaluation of the full eTIMIP proposal, again concerning both its efficiency and its transparency aspects, in the same scenarios as the previous simulation evaluation. For this, the base protocol was studied, analysed and compared to various incremental combinations of the eTIMIP extensions. From these studies, it was concluded that full eTIMIP presents a better handover performance and better routing capability than the base protocol, while maintaining the same transparency level; as the base protocol already featured a similar efficiency performance as the alternative proposals, it results in the conclusion that full eTIMIP features a better efficiency performance and a better transparency capability than the alternative proposals.

From what has been justified above, this PhD thesis states the following three major conclusions:

- The TIMIP/sMIP proposals were the best-positioned state-of-the-art protocols to be extended into an efficient and transparent micro-mobility solution;

- The basic eTIMIP protocol features an at least similar efficiency performance and a better transparency capability than the alternative proposals;

- The full eTIMIP protocol features a better efficiency performance and a better transparency capability than the alternative proposals;

8.2 Further Research Directions

The work described in this thesis can continue in five main directions, which may be seen as promising areas of research:

eTIMIP Policies Specification: Through this thesis, protocol support for multiple policies was specified. In some cases, however, there was no actual definition of a base policy, and a certain one was only suggested, or left to be implementation or deployment-dependent. This happened in the following cases: in the definition of the typical timeout values for the state maintenance operations; in the optimum cases for the triggering of the RO usage and its dissemination mechanisms; in the specification of which neighbours to notify in the fast handover extension; in the definition of the L2 parameters and algorithms to use in the network-controlled handovers; and in the "scope" parameter usage in the network-centric departure support.

Thus, additional research work is required to address these cases, either via the execution of further original specification and simulation research, or through the integration of normalization bodies' (e.g. IETF / IEEE / ITU) recommendations defined for similar cases into eTIMIP.

eTIMIP Scalability and Reliability Shortcomings: While this thesis was mostly focused on efficiency and transparency support, care was taken so it would not preclude other essential aspects of mobility management, namely scalability and reliability of the mobility service. Examples of these are: the basic protocol support of domain scalability, of optimized state refreshment and of link failure protection; and the full protocol support of idle terminals, of operator-centric scenarios, of RO load balancing between the network edges and of efficient handovers in a degenerate tree support.

However, further research may be necessary to extend full eTIMIP with the required amount of support of Scalability and Reliability for deployment in future All-IP networks. In particular, regarding scalability, the control load of the full protocol can be a problem in case a very large number of terminals are in movement simultaneously; a possible way to tackle this problem would be to integrate support for moving networks in eTIMIP (i.e. support of mobile routers using IETF's protocol NEMO). Regarding Reliability, even with the support of a degenerate tree provided in the full protocol, eTIMIP requires further reliability and redundant agent support; a possible way to solve this problem would be to integrate the robustness mechanisms of similar protocols that are also based on a degenerate tree into eTIMIP, by supporting multiple local anchor points [105] [25].

Seamless Support of Inter-System Handover and Host Multi-Homing: Undoubtedly, the success of next generation mobile networks will rely much on the degree of seamless mobility support among heterogeneous technologies. One key aspect of this trend is the support of inter-system handover and mobility management across heterogeneous networks and administrative domains. For this, a new class of multi-homing, multi-modal access devices, which can simultaneously connect to the network via different network interfaces, are required to be optimally supported in the next generation mobile networks, to better exploit the benefits of each technology [77].

While this PhD thesis already presented contributions to decouple the Identifier vs. Locator nature of IP addresses in a transparent way to LMNs, extensions to this paradigm may be required to better support the seamless handoff of the established flows between any of the available interfaces at any time by the network (based on the MN movements and network load). However, to support such facility in a transparent way, the LMNs would have to feature the same identifier-type IP address in all of its interfaces, having the network the exclusive responsibility of dynamically managing a set of locator IP addresses at the network-side only. This can be done if the terminal is configured by the network to have the same IP address in all interfaces, and if the network maintain multiple routing entries for the LMNs indexed by the access technology type.

Mobility Integration with remaining All-IP technologies: As it was described in this thesis' introduction, the future All-IP networks will undoubtedly feature integration of QoS, Security and Mobility technologies. While this PhD research was mostly focused on Mobility Management, care was taken to consider the existing and emerging All-IP key technologies, to ease their future integration using the proposed protocol.

In particular, regarding **QoS**, integration may be eased as the mobility protocol explicitly refrains from changing the IP address at each movement, which was proved to ease integration with QoS in the past [21]. However, further research focusing on context transfer technologies and network-controlled QoS handovers may be required. Regarding **Security**, the protocol already proposes a secure service supporting secure legacy IP stacks, either through custom security support or through the use of standard track protocols. Again, additional research work may be required to fully integrate the developed mobility technology with the newest trends in Security technologies, and with the adaptation of the already deployed security ones that the Legacy Mobile Nodes may support.

Transparent mobility support of IP-version agnostic networks: A recent IETF trend that emerged in the context of IPv6 migration is the idea that mobility protocols of a particular IP version could be extended to additionally manage the mobility of other IP versions; e.g. MIPv4 is in the process of being extended to additionally manage the mobility of IPv6 addresses as well, and vice-versa [106]. This new idea can ease the IPv6 transition process, as single or dual stack mobile nodes and mobile routers can roam in a more flexible mixture of IPv4 and/or IPv6 networks. However, such course of action still limits transparency by requiring mobility-aware terminals, which in the end could be another factor that would limit the IPv6 migration.

Thus, a very promising course of action would be the integration of the transparency technologies present in eTIMIP with the dual-stack MIP, enabling the strong transparency feature of eTIMIP to be additionally IP-version agnostic. For this objective, a new line of research work that analyses this problem and integrates the new IETF developments in this area needs to be performed.

References

IP Macro-Mobility Standards

[1] C. Perkins, Ed., "IP Mobility Support for IPv4", RFC-3320, IETF, January 2002.

[2] D. Johnson, C. Perkins, “Mobility Support in IPv6”, RFC-3775, June 2004

[3] J. Manner, ed, K. Mojo, ed, “Mobility Related Terminology”, RFC 3753, June 2004

[4] C. Perkins, P. Calhoun, “Authentication, Authorization, and Accounting (AAA) Registration Keys for Mobile IPv4”, RFC 3957, March 2005

[5] F. Johansson, T. Johansson, “Mobile IPv4 Extension for Carrying Network Access Identifiers”, RFC 3846, June 2004

[6] C. Perkins, P. Calhoun, “Mobile IP Challenge/Response Extensions”, RFC 3012 , November 2000

[7] C. Perkins, “IP Encapsulation within IP”, RFC-2003 , IETF, October 1996

IP Macro-mobility Extensions - IPv4

[8] M. Kulkarni, el al, “Mobile IPv4 Dynamic Home Agent Assignment”, draft-ietf-mip4-dynamic-assignment-03.txt, September 2004

[9] C. Perkins, “Foreign Agent Error Extension for Mobile IPv4”, draft-mip4-faerr-00.txt, October 2004

[10] C. Perkins and D. Johnson, “Route Optimization in Mobile IP,” draft-ietf-mobileip-optim-11.txt, IETF, September 2001.

[11] S. Vaarala, E. Klovning, “Mobile IPv4 Traversal Across IPsec-based VPN Gateways”, draft-ietf-mip4-vpn-problem-solution-01, January 2005

IP Macro-mobility Extensions - IPv6

[12] A. Patel, et al, “Authentication Protocol for Mobile IPv6”, draft-ietf-mip6-auth-protocol-04.txt, February 2005

[13] V. Devarapalli, “Mobile IPv6 Operation with IKEv2 and the revised IPsec Architecture”, draft-ietf-mip6-ikev2-ipsec-01.txt, February 2005

[14] P. Nikander, ed., “Mobile IP version 6 Route Optimization Security Design Background”, draft-ietf-mip6-ro-sec-02, October 2004

[15] B. Carpenter, J. Crowcroft, Y. Rekhter, “IPv4 Address Behaviour Today", IAB, RFC 2101, February 1997

[16] F. Dupont, J-M. Combes, “Using IPsec between Mobile and Correspondent IPv6 Nodes”, draft-ietf-mip6-cn-ipsec-00.txt, January 2005

[17] A. Patel, ed, "Problem Statement for bootstrapping Mobile IPv6", draft-ietf-mip6-bootstrap-ps-02.txt, March 2005

[18] S. Chakrabarti, E. Nordmark, "Extension to Sockets API for Mobile IPv6", draft-ietf-mip6-mipext-advapi-03.txt, September, 2004

[19] A. Patel, et al,February 2005, "Mobile Node Identifier Option for Mobile IPv6", draft-ietf-mip6-mn-ident-option-02.txt, February 2005

IPv4 Micro-Mobility Proposals Specification

[20] A. Campbell et al, “Design, Implementation and Evaluation of Cellular IP”, IEEE Personal Communications, Vol. 7 Nº4, August 2000.

[21] R. Ramjee, K. Varadhan, L. Salgarelli, S. Thuel, S-Y. Wang, T. La Porta, "HAWAII: a domain-based approach for supporting mobility in wide-area wireless networks", IEEE/ACM Transactions on Networking (TON), Volume 10 , Issue 3, Pages: 396 – 410 2002.

[22] E. Gustafsson et all, "Mobile IPv4 Regional Registration", draft-ietf-mip4-reg-tunnel-01, November 2005.

[23] K. El-Malki, ed., "Low Latency Handoffs in Mobile IPv4", draft-ietf-mobileip-lowlatency-handoffs-v4-11.txt, October 2005.

[24] R. Koodli, Ed., "Mobile IPv4 Fast Handovers", draft-ietf-mip4-fmipv4-00.txt, February 2006

[25] N. Georganopoulos, A. H. Aghvami, "Transport Protocol Performance over BCMP," Proc. IEEE Vehicular Technology Conference, November 2003.

[26] A. Grilo, P. Estrela, M. Nunes, Terminal Independent Mobility for IP (TIMIP)", IEEE Communications, Vol 39 Nº12, December 2001.

[27] P. Estrela, A. Grilo, T. Vazão and M. Nunes, "Terminal Independent Mobile IP (TIMIP)", draft-estrela-timip-01.txt, January 2003.

[28] R. Ramjee, T. La Porta et al.,"Paging support for IP mobility", draft-ietf-mobileip-paging-hawaii-01.txt, July 2000

[29] K. Leung, G. Dommety, P. Yegani, "Mobility Management using Proxy Mobile IPv4", draft-leung-mip4-proxy-mode-00.txt, February 26, 2006

[30] R. Koodli, C. E. Perkins, "Mobile IPv4 Fast Handovers", draft-ietf-mip4-fmipv4-00.txt, February 2006

[31] A. Misra et al., "IDMP-Based Fast Handoffs and Paging in IP-Based 4G Mobile Networks," IEEE Commun. Mag., Mar. 2002, pp. 138–45.

IPv6 Micro-Mobility Proposals Specification

[32] Z. D. Shelby, D. Gatzounas, A. T. Campbell, and C-Y. Wan, "Cellular IPv6," Internet Draft, draft-shelby-seamoby-cellularipv6-00, Work in Progress, November 2000.

[33] C. Castelluccia, "HMIPv6: A Hierarchical Mobile IPv6 Proposal", ACM Mobile Computing and Communication Review (MC2R), April 2000 issue

[34] H. Soliman, et al, "Hierarchical Mobile IPv6 mobility management (HMIPv6)", RFC 4140, August 2005

[35] R. Koodli, Ed., "Fast Handovers for Mobile IPv6", RFC 4068, July 2005

[36] P. McCann, "Mobile IPv6 Fast Handovers for 802.11 Networks", RFC 4260, November 2005

[37] H. Jang, ed., "Mobile IPv6 Fast Handovers over IEEE 802.16e Networks", draft-ietf-mipshop-fh80216e-00.txt, IETF, April 2006

[38] J. Matias, J. Saraiva, F. Silva, R. Rocha, "Mobilidade em IPv6: implementação e análise de técnicas de fast-handover", actas CRC 2004, FCCN, Outubro 2004

[39] F. Templin, S. Russert, K. Grace, "Network Localized Mobility Management using DHCP", draft-templin-autoconf-netlmm-dhcp-04.txt, October 2006.

[40] K. El Malki, H. Soliman, "Simultaneous Bindings for Mobile IPv6 Fast Handovers", draft-elmalki-mobileip-bicasting-v6-06 (work in progress), July 2005

IP Micro-Mobility Frameworks and Reviews

[41] P. Eardley, et al, "A Framework for the Evaluation of IP Mobility Protocols", proceedings of the 11th IEEE PIMRC, September 2000

[42] P. Eardley, N. Georganopoulos, M. West, "On the Scalability of IP micro mobility management protocols", IEEE Conference on Mobile and Wireless Communication Networks (MCWN2002), 2002

[43] P. ReinBold, O. Bonaventure, “IP Micro-Mobility Protocols”, IEEE Communications Surveys, Vol 5, Nº 1, 3º Quarter 2003

[44] A. Campbell, J. Gomez, “IP Micro-Mobility Protocols”, ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 4, pp. 45-53, October 2000

[45] A. Campbell, J. Gomez, S. Kim, Z. Turanyi, C-Y. Wan, and A. Valko, "Comparison of IP MicroMobility Protocols”, IEEE Wireless Communications, pp. 72-82, February 2002

[46] Lampropoulos, G. Passas, N. Merakos, L. Kaloxylos, A., “Handover management architectures in integrated WLAN/cellular networks”, Communications Surveys & Tutorials, IEEE, Volume: 7, Issue: 4, page(s): 30- 44, Fourth Quarter 2005

[47] N. Banerjee, W. Wei, S. Das, “Mobility Support in Wireless Internet”, IEEE Wireless Communications, Page(s): 54- 61, Oct 2003

[48] P. Estrela, T. Vazão, M. Nunes, “Performance Evaluation of a Terminal Independent Mobile Architecture”, accepted for publication on one of the planned special HET-NETs '06 journal issues

[49] G. Karagiannis, “Mobile IP - State of the Art Report”, Ericson research, technical report, Internet Next Generation project, July 1997

[50] K. Daniel Wong, A. Dutta, H. Schulzrinne, K. Young, “Simultaneous mobility: analytical framework, theorems and solutions”, Wireless Communications and Mobile Computing, Sep. 2006.

IP Micro-Mobility Comparisons

[51] L. Peters, I. Moerman, B. Dhoedt, P. Demeester, "MEHROM: Micromobility support with Efficient Handoff and Route Optimization Mechanism", published in 16th ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems, Antwerp, Belgium, August 31 - September 2, 2004, pp. 269-278

[52] R. Hsieh, A. Seneviratne, “A comparison of mechanisms for improving mobile IP handoff latency for end-to-end TCP”, MOBICOM 2003: 29-41, September 2003

[53] X. Pérez-Costa, M. Torrent-Moreno, H. Hartenstein, “A performance comparison of Mobile IPv6, Hierarchical Mobile IPv6, fast handovers for Mobile IPv6 and their combination”, ACM SIGMOBILE Mobile Computing and Communications Review, Volume 7, Issue 4 , October 2003

[54] X. Perez Costa, et. al, "A MIPv6, fMIPv6 and HMIPv6 handover latency study: analytical approach", IST Mobile & Wireless Telecommunications Summit 2002, Thessaloniki, Greece, June 17-19, 2002.

[55] S. Imre, M. Szalay, “Modeling the Reliability of Ring Topology IP Micro Mobility Networks”, 12th International World Wide Web Conference – Emerging Applications for Mobile and Wireless Access, Hungary, May 2003

[56] C. Blondia, O. Casals, L. Cerdà, N. Wijngaert, G. Willems, P. Cleyn, “Performance Comparison of Low Latency Mobile IP schemes”, Proceedings of WiOpt’03, 3-5 March 2003, Sophia Antipolis, France, pp. 115-124.

[57] C. Keszei, N. Georganopoulos, Z. Tyranyi, A., Valko, "Evaluation of the BRAIN candidate mobility management protocol. In Proceedings of the IST Mobile Summit (September 2001).

[58] P. Estrela, T. Vazão, M. Nunes, “De-Triangulation Optimal solutions for mobility scenarios with asymmetric links”, International Conference on Information Networking (ICOIN 2007), January 2007

[59] P. Estrela, T. Vazão, M. Nunes, “Design and Evaluation of eTIMIP - Overlay Micro-Mobility Architecture based on TIMIP”, International Conference on Wireless and Mobile Communications (ICWMC 2006), July 2006.

[60] P. Estrela, T. Vazão, M. Nunes “Performance Evaluation of Micro-Mobility Protocols in Fail-Tolerant Mesh Networks”, PIMRC 2006, September 2006.

[61] G. Bhaskara, A. Helmy, "TCP over Micro Mobility Protocols: A Systematic Ripple Effect Analysis", IEEE Vehicular Technology Conference (VTC), September 2004

[62] P. Estrela, T. Vazão, M. Nunes, “Micro Mobility Performance Evaluation of a Terminal Independent Mobile Architecture”, Second International Working Conference on Performance Evaluation of Heterogeneous Networks, July 2004

[63] L. Peters, I. Moerman, B. Dhoedt, P. Demeester, “Influence of the topology on the performance of micromobility protocols”, Proceedings of WiOpt’03, 3-5 March 2003, Sophia Antipolis, France, pp. 287-292.

[64] L. Peters, I. Moerman, B. Dhoedt, P. Demeester, “Performance of micro-mobility protocols in an access network with a tree, mesh, random and ring topology”, Proceedings of the IST Summit 2003, 15-18 June 2003, Aveiro, Portugal, pp. 63-67.

[65] F. Vena, L. Cerdà, O. Casals, “Study of the TCP dynamics over wireless networks with micro-mobility support using the ns simulator”, ACM Wireless Networks, Volume 10 , Issue 1, Pages: 17 - 27 , 2004

Non IP Layer Mobility Specification

[66] SIP Network Working Group, “SIP: Session Initiation Protocol”, RFC-3261, June 2002

[67] 3GPP TS 23.060, “General Packet Radio Service (GPRS), Service Description, Stage 2,” December 2001.

[68] A. C. Snoeren and H. Balakrishnan, “An End-to-End Approach to Host Mobility,” Proc. 6th Int’l. Conf. Mobile Comp. and Net., Boston, MA, August 2000.

[69] IEEE P802.21/D00.01, “Draft IEEE Standard for Local and Metro-politan Area Networks: Media Independent Handover Services”, July 2005.

[70] R. Moskowitz and P. Nikander: “Host Identity Protocol (HIP) Architecture”, RFC: 4423, May 2006.

[71] G. Patel and S. Dennett, “The 3GPP and 3GPP2 Movements Toward an All-IP Mobile Network,” IEEE Pers. Commun., Aug. 2002, pp. 62–64.

[72] Gonzalo Camarillo et al., “The 3G IP Multimedia Subsystem”, England: Wiley, 2006.

[73] N. Banerjee, S. Das and A. Acharya, “SIP-based Mobility Architecture for Next Generation Wireless Networks.”.3rd IEEE International Conference on Pervasive Computing and Communications (Percom), 2005.

[74] G. Gupta, D. Johnston, “IEEE 802.21 A Generalized Model for Link Layer Triggers”, IEEE 802.21, March 2004

[75] K. Mitani, "Unified L2 Abstractions for L3-Driven Fast Handover", draft-koki-mobopts-l2-abstractions-03 (work in progress), October 2005.

All-IP networks / 4G Evolution / Mobility Layer

[76] D. Saha, A. Mukherjee, I. S. Misra, and M. Chakraborty, "Mobility Support in IP: A Survey of Related Protocols", IEEE Network, vol. 18, no. 6, pp. 34-40, Nov/Dec 2004.

[77] Suk Yu Hui; Kai Hau Yeung, “Challenges in the migration to 4G mobile systems” Communications Magazine, IEEE , Volume: 41 , Issue: 12 , Dec. 2003, Pages:54 - 59

[78] U. Varshney, R. Jain, “Issues in emerging 4G Wireless Networks”, IEEE Computer 34(6) pages:34-40, 2001.

[79] W. Kellerer, H.-J. Vögel, K.-E. Steinberg, “A Communication Gateway for Infrastructure Independent 4G Wireless Access”, IEEE Communications Magazine, Vol.9 No.3, March 2002

[80] G. Carneiro, J. Ruela, M. Ricardo, "Cross-layer design in 4G Wireless Terminals," IEEE Wireless Communications, April 2004, pp. 7-13.

[81] Y. Kim, et al, “Beyond 3G: Vision, “Requirements, and Enabling Technologies”, IEEE Communications Magazine, pp. 120-124, March 2003

[82] K. Tachikawa, “A perspective on the Evolution of Mobile Communications”, IEEE Communications Magazine, pp. 66-73, October 2003

[83] Steven J. Vaughan-Nichols, "The Challenge of Wi-Fi Roaming", IEEE computer magazine (Vol. 36, No. 7), pp. 17-19, July 2003

[84] S. Keshav, “Why Cell Phones Will Dominate the Future Internet”, ACM SIGCOMM Computer Communication Remix, vol 35, issue 2, pages 83-86, April 2005

[85] Uskela and Sami, “All IP Architectures for Cellular Networks,” Proceedings of 3G Mobile Communications Technologies Conference, pp.180-185, March 2001.

[86] J.L. Chen, W.H. Chen and S.Y. Kuo, “All-IPv6 Service Interworking Gateway,” International Journal of Network Management, March 2005.

[87] F. M. Chiussi, D. A. Khotimsky, and S. Krishnan, “Mobility Management in Third-Generation All-IP Networks,” IEEE Commun. Mag., Sept. 2002, pp. 124–35.

[88] A. Lewis, “Let's ditch the switch”, IEE Review Volume 50, Issue 8, Aug. 2004 Page(s):18 - 21

[89] T.B. Zahariadis, K. G. Vaxevanakis, C.P. Tsantilas, N.A. Zervos, N.A. Nikolaou, “Global roaming in next-generation networks”, IEEE Communications Magazine, Volume 40, Issue 2, Feb. 2002 Page(s):145 - 151

[90] T. Henderson, “Host mobility for IP networks: a. comparison”, IEEE Network, Nov. 2003, pp. 18-26

[91] W. Eddy, “At What Layer Does Mobility Belong?", IEEE Communications Magazine, October 2004.

[92] D. Le, X. Fu and D. Hogrefe, “A Review of Mobility Support Paradigms for the Internet”, IEEE Communications Surveys and Tutorials, Jan 2006.

[93] E. Wedlund and H. Schulzrinne, "Mobility support using SIP," in Second ACM/IEEE International ConJerence on Wireless and Mobile Multimedia (WoWMoM'99), Aug. 1999.

[94] I. Akyildiz, J. Xie, S. Mohanty, "A Survey of Mobility Management in Next-Generation All-IP-Based Wireless Systems," IEEE Wireless Communications, special issue on Mobility and Resource Management, vol. 11, no. 4, pp. 16-28, August 2004.

[95] P.J. McCann and T. Hiller, “An Internet Infrastructure for Cellular CDMA Networks using Mobile IP,” IEEE Personal Comm., vol. 7, no. 4, pp. 26-32, Aug. 2000.

Transparency References

[96] R. Braden, Ed., “Requirements for Internet Hosts - Communication Layers”, RFC 1122, October 1989

[97] E. Gustafsson, ed., “Requirements on Mobile IP from a Cellular Perspective”, Internet draft, draft-ietf-mobileip-cellular-requirements-02, work in progress, June 1999

[98] V. Navda, A. Kashyap and S. Das, “Design and Evaluation of iMesh: an Infrastructure-mode Wireless Mesh Network”, IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM), Italy, June 2005.

[99] M. Lee, Y. Yoon, “MTI: Integration of UNA and MTAII toward the Next Generation Internet”, proceedings of the 1st EuroNGI Conference on Next Generation Internet Networks, Italy, April 2005

[100] G. Albertengo, C. Pastrone, G. Tolu, “MOON: a New Overlay Network Architecture for Mobility and QoS Support”, proceedings of the 1st EuroNGI Conference on Next Generation Internet Networks, Italy, April 2005

[101] A. Giovanardi, G. Mazzini, “Transparent mobile IP: an approach and implementation” Global Telecommunications Conference (GLOBECOM '97), IEEE Volume 3, Page(s):1861 – 1865, November 1997

[102] D. Clark, J. Wroclawski, K.R. Sollins, R. Braden , "Tussle in cyberspace: defining tomorrow's internet", IEEE/ACM Transactions on Networking, Volume 13 , Issue 3, Pages: 462 - 475 , June 2005.

[103] J. Kempf, Ed., "Goals for Network-Based Localized Mobility Management (NETLMM)", RFC 4831, April 2007.

[104] J. Kempf, Ed., "Problem Statement for Network-Based Localized Mobility Management (NETLMM)", RFC 4830, April 2007.

[105] J. Laganier, S. Narayanan, F. Templin, “Network-based Localized Mobility Management Interface between Mobile Node and Access Router”, draft-ietf-netlmm-mn-ar-if-01.txt, June 2006

[106] Tsirtsis, G., et.al, "Problem Statement: Dual Stack Mobility", draft-ietf-mip6-dsmip-problem-03.txt, January 2007.

Other Network layer references

[107] Jangeun Jun et al, "Theoretical Maximum Throughput of IEEE 802.11 and its Applications," in Proc. of the 2nd IEEE International Symposium on Network Computing and Applications (NCA'03), Cambridge, April 2003

[108] Yang, ed., et al, “CAPWAP Architecture Taxonomy”, Internet draft, draft-ietf-capwap-arch-06.txt, work in progress, November 2004

[109] G. Malkin, “RIP Version 2”, IETF RFC 2453, November 1998

[110] J. Moy, “OSPF Version 2”, IETF RFC 2328, April 1998

[111] Y. Rekhter, T. Li, eds., “A Border Gateway Protocol 4 (BGP-4)”, IETF, RFC-1771, March 1995

[112] Y. Rekhter, T. Li, eds, “An Architecture for IP Address Allocation with CIDR”, RFC-1518, September 1993

[113] R. Rivest, “RFC 1321: The MD5 Message-Digest Algorithm, IETF, April 1992.

[114] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

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

[116] D. Mills, “Network Time Protocol (Version 3) - Specification, Implementation and Analysis”, RFC 1305, March 1992.

[117] AvantCom, “AvantCom Private 802.11 MIB”, available at http://www.avantcom.com/snmpAVC80211.aspx, September 2003

[118] ISO, "Intermediate system to Intermediate system routing information exchange protocol for use in conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)," ISO/IEC 10589:2002, Second Edition, 2002.

[119] I.F. Akyildiz, W. Xudong, “A survey on wireless mesh networks”, IEEE Communications Magazine, IEEE, Volume: 43, Issue: 9, Sept. 2005.

[120] M. Patrick, “DHCP Relay Agent Information Option”, RFC 3046, January 2001.

[121] G. McGregor, “The PPP Internet Protocol Control Protocol (IPCP), RFC 1332, May 1992.

[122] David C. Plummer, “An Ethernet Address Resolution Protocol”, RFC 826, November 1982.

[123] A. Conta, S. Deering, M. Gupta, Ed., “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification”, RFC 4443, March 2006.

[124] A. Conta, “Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification”, RFC 3122, June 2001

Below-IP Access Technologies references

[125] IEEE, "IEEE 802.11-99 IEEE WLAN MAC and PHY Layer Specifications", August 1999.

[126] IEEE 802.16E-2005, “IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands”, 802.16e, December 2005

[127] 3GPP, “Combined GSM and Mobile IP mobility handling in UMTS IP CN”, 3GPP TR 23.923, June 2006

[128] 3GPP, “reserved for: All-IP network (AIPN)”, work-in-progress, 3GPP TS 23.258

[129] A. Mishra et al, “An empirical analysis of the IEEE 802.11 MAC layer handoff process”, ACM SIGCOMM Computer Communication Review, Vol 33, issue 2, April 2003.

QoS references

[130] X. Xipeng, M. Lionel, “Internet QoS: A Big Picture”, IEEE Network Magazine March/April 1999.

[131] DiffServ Working Group, http://www.ietf.org/html.charters/diffserv-charter.html

[132] IntServ Working Group, http://www.ietf.org/html.charters/intserv-charter.html

[133] Y. Bernet, “The Complementary Roles of RSVP and Differentiated Services in Full-Service QoS Network”. IEEE Communications Magazine, February 2000.

[134] E. Mannie, Ed, “Generalized Multi-Protocol Label Switching (GMPLS) Architecture”, RFC 3945, October 2004.

[135] F. Faucheur, et al, “Multi-Protocol Label Switching (MPLS) Support of Differentiated Services”, IETF RFC 3270, May 2002.

[136] B. Teitelbaum, S. Hares, L. Dunn, V. Narayan, R. Neison and F. Reichmeyer, “Internet2 QBone: Building a Testbed for Differentiated Services”. IEEE Network Magazine, Special Issue on Integrated and Differentiated Services for Internet, September 1999.

[137] S. Mangold, C. Sunghyun, G.R. Hiertz, O. Klein, B. Walke, “Analysis of IEEE 802.11e for QoS support in wireless LANs”, IEEE Wireless Communications, Volume: 10 , Issue: 6, page(s): 40 - 50, December 2003

[138] R. Braden, Ed., “Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification”, RFC-2205, IETF, September 1997

Security References

[139] Jon Allen, Jeff Wilson, “Securing a wireless network”, Proceedings of the 30th annual ACM SIGUCCS conference on User services, Pages: 213 - 215, 2002

[140] S. Kent and R. Atkinson, “IP Authentication Header”, RFC 2402, November 1998.

[141] IEEE, “IEEE Standard for Local and metropolitan area networks - Port-Based Network Access Control”, IEEE Std 802.1X-2001, 2001.

[142] C. Rigney, A. Rubens, W. Simpson, and S. Willens, “Remote authentication dial in user service (RADIUS)”, RFC 2138, April 1997.

[143] IEEE, “IEEE Std 802.11i”, July 2004.

[144] A.Yegin, Ed., “Protocol for Carrying Authentication for Network Access (PANA) Requirements”, RFC 4058, May 2005

[145] B. Aboba and D. Simon. “PPP EAP TLS Authentication Protocol”. RFC 2716, October 1999.

[146] PEAP - http://www.ietf.org/ietf/IPR/MICROSOFT-PEAP.txt

[147] IPSEC web reference, freeswan.org, http://www.freeswan.org/freeswan_trees/freeswan-1.5/doc/links.ipsec.html

[148] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko, “Diameter Base Protocol”, RFC-3588, September 2003

[149] J. Arkko, J. Kempf, B. Zill, P. Nikander, "Secure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[150] T. Aura, "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[151] J. Kempf, S. Narayanan, et al, "Detecting Network Attachment in IPv6 Networks (DNAv6)", "draft-ietf-dna-protocol-01.txt", June 2006

[152] C. Vogt, J. Kempf, "Security Threats to Network-Based Localized Mobility Management (NETLMM)", RFC 4832, April 2007

[153] C. Kaufman, Ed., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[154] B. Aboba, M. Beadles, "The Network Access Identifier", RFC 2486, January 1999.

[155] R. Droms, ed, W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001

QoS / Security / Mobility Services integration

[156] A. Prasad, P. Schoo, "IP Security for Beyond 3G towards 4G", in proceedings of WWFR 7, Eindohoven, Netherlands, December 2002

[157] X. Fu, D. Hogrefe, R. Soltwisch, and S. Narayanan, "QoS and Security in 4G Networks", Proceedings of the 1st CIC/IEEE Global Mobile Congress (GMC 2004), Shanghai, China, pp. 117-122, October 2004.

[158] J. Garcia-Macias, F. Rousseau, G. Berger-Sabbatel, L. Toumi, A. Duda, "Quality of service and mobility for the wireless internet", ACM Wireless Networks, Volume 9 , Issue 4, Pages: 341 - 352 , July 2003

[159] J. Manner et al., "Evaluation of Mobility and Quality of Service Interaction," Comp. Networks, vol. 38, 2002, pp. 137–63.

[160] A lopez et al, "A Study on QoS Provision for IP-based Radio Access Networks", Lecture Notes In Computer Science; Vol. 2170, Proceedings of the Thyrrhenian International Workshop on Digital Communications: Evolutionary Trends of the Internet, Pages: 139 - 157, 2001

[161] Romdhani , M. Kellil , H-Y. Lach , A. Bouabdallah, H. Bettahar, "IP Mobile Multicast: Challenges and Solutions" IEEE Communications Surveys & Tutorials, Volume 6, No. 1, Pages 18-41, First Quarter 2004.

Simulation / Packet reordering

[162] C. Cicconetti, E. Mingozzi, G. Stea, "An Integrated Framework for Enabling Effective Data Collection and Statistical Analysis with ns-2", Workshop on the NS-2: The IP Network Simulator, in conjunction VALUETOOLS 2006 Conference, Pisa, October 2006

[163] S. Floyd, "Maintaining a Critical Attitude towards Simulation Results", Keynote Presentation, Workshop on the NS-2: The IP Network Simulator, in conjunction VALUETOOLS 2006 Conference, Pisa, October 2006

[164] A.Morton, L.Ciavattone, G.Ramachandran, S.Shalunov, J.Perser, "Packet Reordering Metric for IPPM", draft-ietf-ippm-reordering-13.txt, May 2006

[165] J. Bennet, C. Partridge, N. Shectman, "Packet reordering is not pathological behaviour", IEEE/ACM Transactions on Networking, vol 7, issue 6, December 1999

[166] M. Laor and L. Gendel. "The effect of packet reordering in a backbone link on application throughput", IEEE Network, 7(6), September 2002.

[167] X. Zhou and P. Mieghem. "Reordering of IP packets in Internet." In Proc. Passive and Active Measurement, April 2004.

[168] S. Floyd, ed, "Metrics for the Evaluation of Congestion Control Mechanisms", draft-irtf-tmrg-metrics-09.txt, IETF, March 2007

Internet Principles and Protocols specification

[169] R. Perlman, “Protocol Design Folklore”, Addison Wesley Professional, Jan 2001, available http://www.awprofessional.com/articles/article.asp?p=20482, retrieved March 2007

[170] B. Carpenter, Architectural Principles of the Internet, Internet informational RFC 1958, June 1996.

[171] R. Bush, D. Meyer, “Some Internet Architectural Guidelines and Philosophy”, RFC 3439, December 2002

[172] J. Kurose and K. Ross. “Computer Networking -- A Top Down Approach Featuring the Internet, 3rd edition”; Addison-Wesley, 2005.

[173] D. Thaler, “Multilink Subnet Issues”, draft-iab-multilink-subnet-issues-03.txt, IAB, January 2007.

[174] S. Deering, R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification” , RFC 2460 December 1998

[175] D. Harel and Chaim-arie Kahana, On statecharts with overlapping, ACM Transactions on Software Engineering and Methodology, 1(4):399-421, October 1992.

[176] M. Blumenthal and D. Clark. “Rethinking the design of the internet: the end-to-end arguments vs. the brave new world”. ACM Transactions on Internet Technology, 1(1), August 2001.

[177] B. Carpenter, "Internet Transparency," Internet RFC 2775, February 2000.

Web Resources

[178] J. Kristoff, “Mobile IP”, http://condor.depaul.edu/~jkristof/mobileip.html, DePaul University, Chicago, October 1999, acessed March 2007

[179] “NS-2 Home Page”, http://www.isi.edu/nsnam/ns, acessed March 2007

[180] “Columbia IP Micro-Mobility Suite (CIMS)”, http://www.comet.columbia.edu/micromobility, acessed March 2007

[181] “TIMIP Homepage”, http://tagus.inesc-id.pt/~pestrela/timip, acessed March 2007

[182] J. Xie, "Mobility Management in 4G Wireless Systems ", http://users.ece.gatech.edu/~jxie/4G/, acessed March 2007

[183] VECTEC, “Wireless: General Trends”, http://www.vectec.org/researchcenter/stats.html?category=12, acessed March 2007

[184] “Mobile IP Implementations”, http://www.mip4.org/2004/implementations/, acessed July 2006

[185] “Gnuplot homepage”, http://www.gnuplot.info, acessed March 2007

Appendix A Smooth Upgrade process example

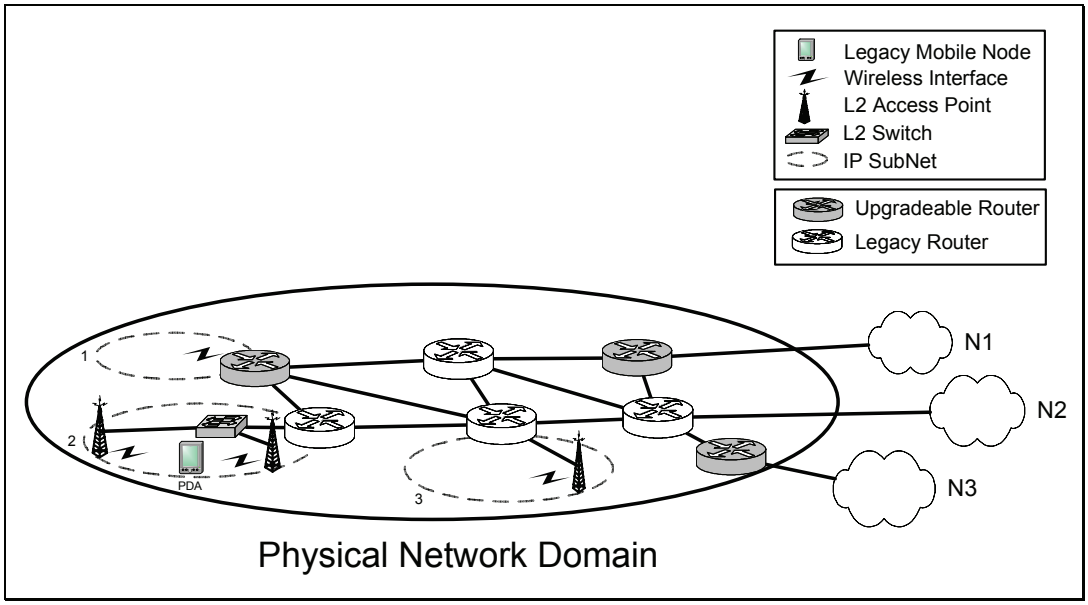


Figure 164: Example network before mobility services introduction

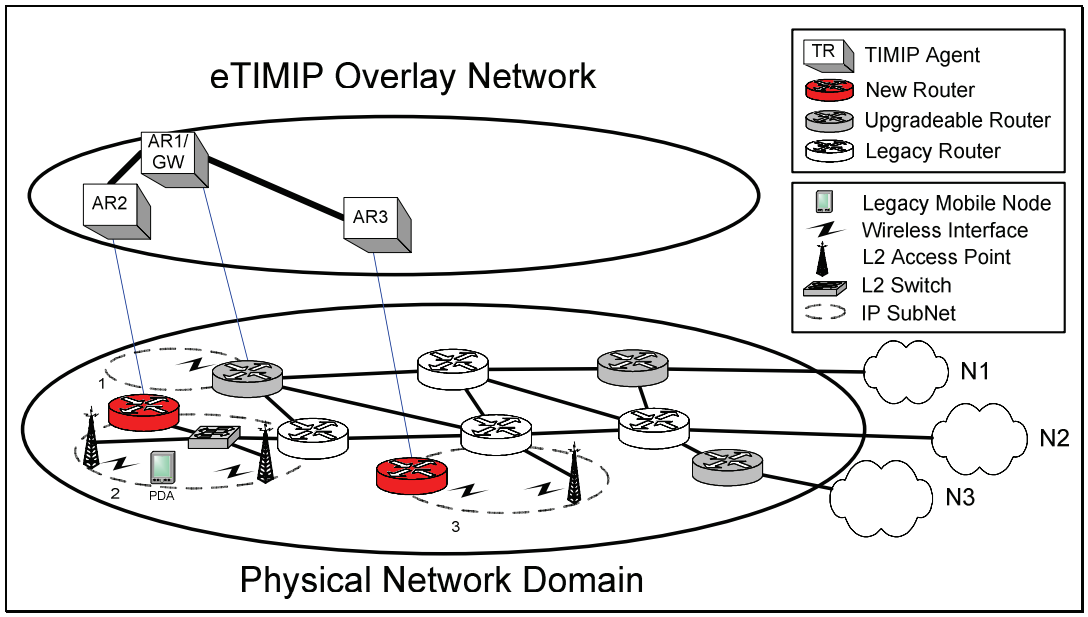


Figure 165: Example network after minimum mobility services introduction, using only ARs

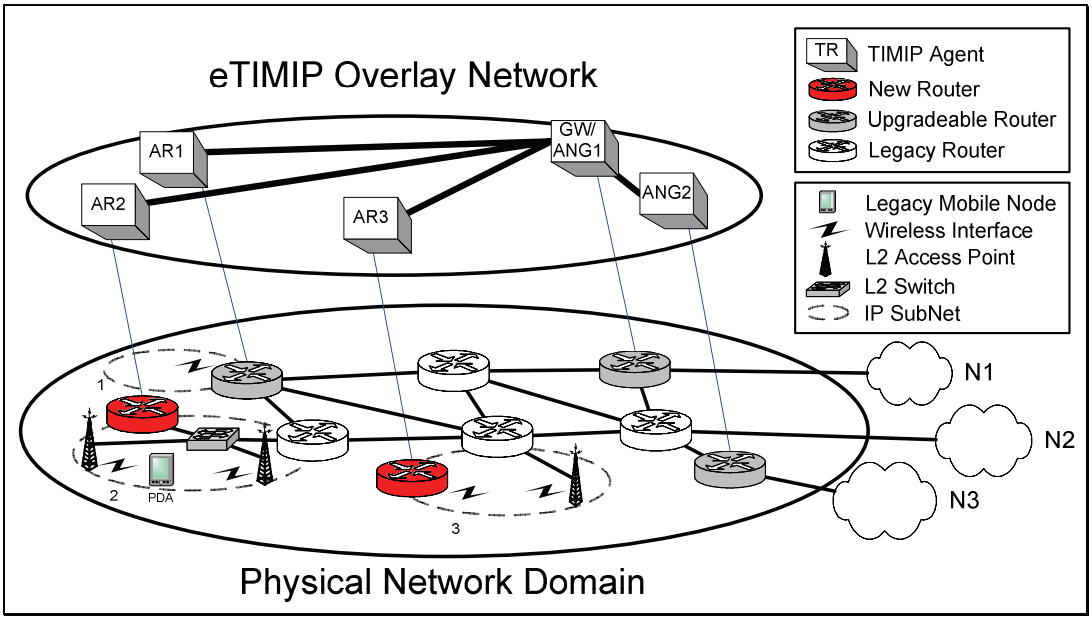


Figure 166: Example network after medium mobility services introduction, using multiple ARs, a GW and a second ANG.

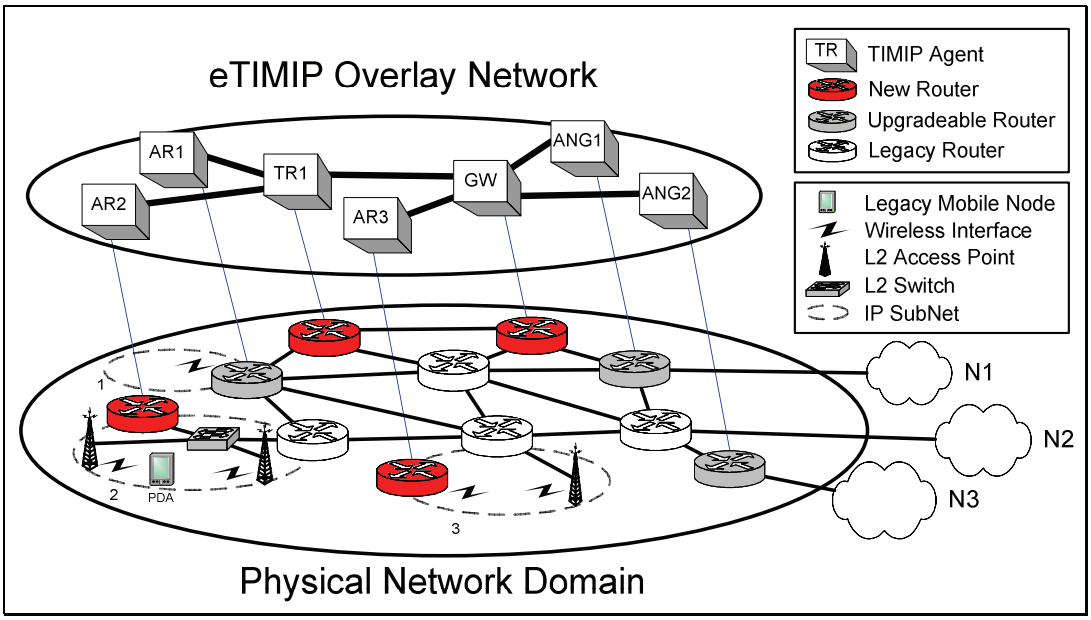


Figure 167: Example network after full distributed mobility services introduction, using multiple ARs, TRs and ANGs, but still keeping the existing legacy terminals, legacy routers and topologies unaltered.

Appendix B Formal Specification helper functions and constants

By Type

Generic and Timer Functions

True()	Logical function that always succeeds
False()	Logical function that always Fails
Min(A, B)	Returns the minimum of two values
Max(A, B)	Returns the maximum of two values
Rand()	Returns a random number
Now()	Returns the timestamp value of the node's local clock
Flag_error(ADDR)	Flags a serious error in the system's MIB for entity with address ADDR
Start_timer(TR)	Cancels the timer associated with TR in the current node, if any, and installs a timer associated with TR in the current node with the default duration
Stop_timer(TR)	Cancels timer associated with TR in the current node, if any
Timeout(TR)	Signals the elapsement of time period associated with node TR

Tree and Wireless interfaces support

My_IP()	Returns the current node's IP address
GW_IP()	Returns the GW's IP address
Parent_IP()	Returns the current node parent's IP address, or My_IP() if its the GW
Next_AR()	Returns the next adjacent AR's IP address, or My_IP() if none
Prev_AR()	Returns the previous adjacent AR's IP address, or My_IP() if none

Foreach_child_except(CHILD, except_TR, action):
Iterates a certain action for all the node's child nodes except a certain one

Is_AR()	Returns if the current node has an wireless interface
Wireless_ITF()	Returns the wireless interface of this AR

Control and Data Forwarding

Send_to(MSG, TR)	Sends message MSG from the current node to node TR
Receive_from(MSG, TR)	Receives a message MSG from source TR destinated to the current node
Forward_to(MSG, NEXT_HOP)	Forwards the unmodified message MSG to node NEXT_HOP if adjacent node or Wireless_ITF(), or encapsulated with a soft or full tunnel to node NEXT_HOP in all other cases
Forwarded_from(MSG, PREV_HOP)	Receives the unmodified message MSG from adjacent node TR or from Wireless_ITF(), or desencapsulatates message MSG from node TR

Routing Table Updating and Timestamps

Remove_entries(LMN)	Removes all the LMN basic and RO entries on the system's routing table, clears the generic detection algorithm's LMN memory (if at an AR); This function Invokes Routing_table_updated(LMN)
Remove_RO_entry(LMN)	Removes the LMN RO entry on the system's routing table
Routing_table_updated(LMN)	Signals the modification of a routing table entry for a given LMN

Update_RT(LMN, TS, BASIC_FLAG, RO_FLAG, IDLE_FLAG, REMOVE_FLAG, DECISION_FLAG, NEXTHOP, AR)
Updates the routing table with the next-node and final-node information, a given timestamp for LMN and the set of specified flags. If BASIC_FLAG is present, any existing RO entry is removed first for this LMN.
This can either result on the creation, modification or removal of a corresponding routing entries at the system's routing table.
This function invokes Routing_table_updated(LMN)

Cur_TS(LMN)	Consults the routing table of the current node, and returns the timestamp associated with this LMN
Accept_TS(LMN, TS)	Verifies that the given timestamp TS is equal or greater than Cur_TS(LMN)

Basic entries

Is_basic()	Flag that signals a basic routing entry
Not_basic()	Flag that signals the inexistence of a basic routing entry
Cur_basic(LMN)	Returns true if a basic entry flag is present on the routing table for this LMN
Cur_location(LMN)	Consults the routing table of the current node, and returns the next-node associated with this LMN, or the parent node if no entry is found, or the current node if no entry is found and node is the GW
LMN_is_here(LMN)	Calls Cur_location(LMN) and returns true if the LMN is located at this node

RO entries

Is_RO()	Flag that signals a RO routing entry
Not_RO()	Flag that signals the inexistence of a RO routing entry
Cur_RO(LMN)	Returns true if a RO entry flag is present on the routing table for this LMN
Cur_location_RO(LMN)	If a RO entry exists on the routing table, returns it; else return Cur_location(LMN)

Idle, decision, remove and ack entries

Is_Idle()	Flag that signals an idle routing entry
Not_Idle()	Flag that signals the inexistence of an idle routing entry
Cur_Idle(LMN)	Returns true if a idle entry flag is present on the routing table for this LMN
Is_decision()	Flag that signals an decision routing entry
Not_ decision()	Flag that signals the inexistence of an decision routing entry
Cur_ decision(LMN)	Returns true if a decision entry flag is present on the routing table for this LMN
Is_remove()	Flag that signals an remove routing entry
Not_ remove()	Flag that signals the inexistence of an remove routing entry
Cur_remove(LMN)	Returns true if a remove entry flag is present on the routing table for this LMN
Is_ack()	Flag that signals an ACK
Not_ ack()	Flag that signals the inexistence of an ACK

Control packets

Update_msg(LMN, TS, BASIC_FLAG, RO_FLAG, IDLE_FLAG, REMOVE_FLAG, DECISION_FLAG, ACK_FLAG, AR)
: Generates an update message for LMN with timestamp TS, optional RO AR information, and the set of specified flags

Report_msg(LMN, TS, L2_PARAMS, AR)
: Generates an Report message for LMN with timestamp TS, AR information, and optional L2 parameters information

Paging_msg(LMN, REFRESH_FLAG, ORIGINAL_TR, SCOPE)
: Generates an Paging message for LMN with TR information, Scope, and a given refresh flag

RO_inform(LMN, TS, STEP, CR, TR, AR))
: Generates an RO_inform message for this LMN, with a certain Timestamp TS, de-triangulation step STEP, for informing the crossover node CR to send LMN's packets directly to node AR instead of triangulating via node TR

Data packets, Buffering and Refreshes

Data_packet(PAYLOAD, SRC, DST)
: Pseudo-function that represents a LMN data packet from source SRC to destination DST, with a given payload

Buffer_data_packet(LMN, PACKET)
: Stores a given data packet in a buffer associated to this LMN, being released when release_data_packets(LMN) is called, buffer is full, or after an timeout with the default duration

Release_data_packets(LMN)	Releases LMN's buffered data packets in First-in, First Out order
Start_buffering_packets(LMN)	Starts buffering subsequent received data packets for LMN until release_data_packets() is called, or an internal timeout occurs.
Is_buffering_packets(LMN)	Returns true when data packets are being buffered for LMN
Is_idle_P()	Flag that signals an idle refresh paging operation
Is_data_P()	Flag that signals an regular data paging operation; equal to !is_idle_P()

Refresh_req(SRC, DST)	Generates a standard refresh request with given source and destination pair
Refresh_reply(SRC, DST)	Represents a standard refresh reply with given source and destination pair
Send_surrogate_ND(LMN)	Sends a surrogate neighbor discovery for given LMN to associate the default router IP address with this node's MAC address

eTIMIP Internal Primitives, Optional modules and Address Resolution

RO_enabled()	Returns true when the route optimization option is enabled
Idle_enabled()	Returns true when the idle support option is enabled
Trigger_Data_RO_enabled()	Returns true when the data RO dissemination mechanism is enabled
Trigger_Fast_RO_enabled()	Returns true when the data Fast HO dissemination mechanism is enabled
Smooth_detriangulation_enabled()	Returns true when the smotth de-triangulation option is enabled
Unsecured_Detection(LMN)	Invoked after a detection at an AR, for a given LMN without authentication guarantees
Secured_Detection(LMN, TS)	Invoked after a successful detection at an AR, for a given LMN with an associated timestamp TS, with authentication guarantees
Confirmed_Detection(LMN, TS)	Invoked after a successful confirmed detection at an AR, for a given LMN with an associated timestamp TS, with authentication guarantees
Init_paging(LMN, REFRESH_FLAG, SCOPE)	Starts paging procedures for this LMN
Init_idle(LMN)	Starts idle support procedures for this LMN
Power_Down(LMN)	Starts power-down removal procedures for this LMN
Trigger_data_RO(LMN, CR)	Starts data RO dissemination mechanism for this crosshover node CR
Trigger_Fast_RO(LMN, TS)	Starts data Fast HO dissemination mechanism at this AR. Equal to Confirmed_Detection(LMN, TS) && Trigger_Fast_RO_enabled()
GDA_ON(LMN)	Disables the generic detection algorithm feature
GDA_OFF(LMN)	Enables the generic detection algorithm feature
Find_L2_Parameters(LMN)	Returns the Layer 2 parameters from the current AP of the LMN, for supporting the network-controlled handover decision
Decision_Reached(LMN, TS, AR)	Event invoked when the Deciding Node decides the new location of the terminal, based on the L2 reports generated by the Ars that detect the LMN
Send_Decision(LMN, TS, AR)	Invoked to notify the chosen AR of the network-controlled handover
Add_L2_Report(LMN, TS, L2_PARAMS, AR)	Collects a new AR L2 report for a given LMN, at the Deciding node
Clear_L2_Reports(LMN)	Clears all previous L2 reports for a given LMN
L2_discovery_req(X_IP, LMN)	IP address resolution message request
L2 discovery resp(LMN, X_IP, AR_MAC)	IP address resolution message response

Generic Security functions

Security_validate_req(LMN, GW, RAND, AR_TS)	Generates a security validate message for a single LMN, with a random number RAND, timestamp AR_TS and gateway information GW
Security_validate_ack(LMN, GW, RAND, AR_TS, MNID, LMN_TS)	Receives a security validate ack message for a single LMN, with a random number RAND, timestamps AR_TS and LMN_TS, encrypted LMN MNID and gateway information GW
Security_init_req (LMN)	Generates an Security Update Request to update the new PID at the LMN
Security_init_ack (LMN)	Generates an security update acknowledge to trigger revalidation for this LMN
Compute_PID(LMN, NK)	Computes the unique session key for this LMN and this domain using the secret network key NK
Sign(MSG, PID)	Digitally sign message MSG with key PID, by appending a encrypted digest of the message to it.
Verity_PID(message, PID)	Verifies the digital signature of a message with key PID
Get_public_key(LMN)	Starts the LMN authorization check via an AAA infrastructure via the GW, and securely returns the public key of LMN
Encript(MSG, K)	Encrypts the whole contents of message MSG with key K and returns the resulting message
Decript(MSG, K)	Decrypts The Whole Contents Of Message MSG With Key K and returns the resulting message
My_PrivK()	Returns the LMN's Private Key

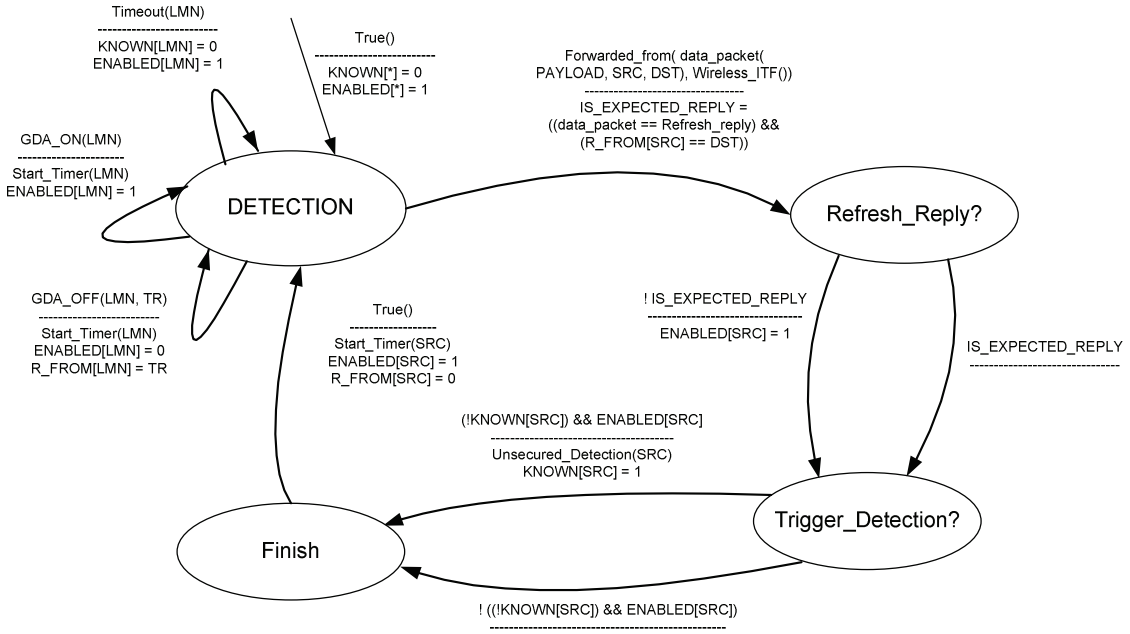
Alphabetically Sorted

Accept_TS(LMN, TS) Verifies that the given timestamp TS is equal or greater than Cur_TS(LMN)
Add_L2_Report(LMN, TS, L2_PARAMS, AR) Collects a new AR L2 report for a given LMN, at the Deciding node
Buffer_data_packet(LMN, PACKET) Stores a given data packet in a buffer associated to this LMN, being released when release_data_packets(LMN) is called, buffer is full, or after an timeout with the default duration
Clear_L2_Reports(LMN) Clears all previous L2 reports for a given LMN
Compute_PID(LMN, NK) Computes the unique session key for this LMN and this domain using the secret network key NK
Confirmed_Detection(LMN, TS) Invoked after a successful confirmed detection at an AR, for a given LMN with an associated timestamp TS, with authentication guarantees
Cur_ decision(LMN) Returns true if a decision entry flag is present on the routing table for this LMN
Cur_basic(LMN) Returns true if a basic entry flag is present on the routing table for this LMN
Cur_Idle(LMN) Returns true if a idle entry flag is present on the routing table for this LMN
Cur_location(LMN) Consults the routing table of the current node, and returns the next-node associated with this LMN, or the parent node if no entry is found, or the current node if no entry is found and node is the GW
Cur_remove(LMN) Returns true if a remove entry flag is present on the routing table for this LMN
Cur_RO(LMN) Returns true if a RO entry flag is present on the routing table for this LMN
Cur_TS(LMN) Consults the routing table of the current node, and returns the timestamp associated with this LMN
Cur_location_RO(LMN) If a RO entry exists on the routing table, returns it; else return Cur_location(LMN)
Data_packet(PAYLOAD, SRC, DST) Pseudo-function that represents a LMN data packet from source SRC to destination DST, with a given payload
Decision_Reached(LMN, TS, AR) Event invoked when the Deciding Node decides the new location of the terminal, based on the L2 reports generated by the Ars that detect the LMN
Decript(MSG, K) Decrypts The Whole Contents Of Message MSG With Key K and returns the resulting message
Encript(MSG, K) Encrypts the whole contents of message MSG with key K and returns the resulting message
False() Logical function that always Fails
Find_L2_Parameters(LMN) Returns the Layer 2 parameters from the current AP of the LMN, for supporting the network-controlled handover decision
Flag_error(ADDR) Flags a serious error in the system's MIB for entity with address ADDR
Foreach_child_except(CHILD, except_TR, action) :
Iterates a certain action for all the node's child nodes except a certain one.
Forward_to(MSG, NEXT_HOP) Forwards the unmodified message MSG to node NEXT_HOP if adjacent node or Wireless_ITF(), or encapsulated with a soft or full tunnel to node NEXT_HOP in all other cases
Forwarded_from(MSG, PREV_HOP) Receives the unmodified message MSG from adjacent node TR or from Wireless_ITF(), or desencapsulatates message MSG from node TR
GDA_OFF(LMN) Enables the generic detection algorithm feature
GDA_ON(LMN) Disables the generic detection algorithm feature
Get_public_key(LMN) Starts the LMN authorization check via an AAA infrastructure via the GW, and securely returns the public key of LMN
GW_IP() Returns the GW's IP address
Idle_enabled() Returns true when the idle support option is enabled
Init_idle(LMN) Starts idle support procedures for this LMN
Init_paging(LMN, REFRESH_FLAG, SCOPE) Starts paging procedures for this LMN
Is_ack() Flag that signals an ACK
Is_AR() Returns if the current node has an wireless interface
Is_basic() Flag that signals a basic routing entry
Is_buffering_packets(LMN) Returns true when data packets are being buffered for LMN
Is_data_P() Flag that signals an regular data paging operation; equal to !is_idle_P()
Is_decision() Flag that signals an decision routing entry
Is_Idle() Flag that signals an idle routing entry
Is_idle_P() Flag that signals an idle refresh paging operation
Is_remove() Flag that signals an remove routing entry
Is_RO() Flag that signals a RO routing entry
L2 discovery resp(LMN, X_IP, AR_MAC) IP address resolution message response
L2_discovery_req(X_IP, LMN) IP address resolution message request
LMN_is_here(LMN) Calls Cur_location(LMN) and returns true if the LMN is located at this node
Max(A, B) Returns the maximum of two values
Min(A, B) Returns the minimum of two values
My_IP() Returns the current node's IP address
My_PrivK() Returns the LMN's Private Key
Next _AR() Returns the next adjacent AR's IP address, or My_IP() if none.
Not_ ack() Flag that signals the inexistence of an ACK
Not_ decision() Flag that signals the inexistence of an decision routing entry
Not_ remove() Flag that signals the inexistence of an remove routing entry

Not_basic()	Flag that signals the inexistence of a basic routing entry
Not_Idle()	Flag that signals the inexistence of an idle routing entry
Not_RO()	Flag that signals the inexistence of a RO routing entry
Now()	Returns the timestamp value of the node's local clock
Paging_msg(LMN, REFRESH_FLAG, ORIGINAL_TR, SCOPE)	Generates an Paging message for LMN with TR information, Scope, and a given refresh flag
Parent_IP()	Returns the current node parent's IP address, or My_IP() if its the GW
Power_Down(LMN)	Starts power-down removal procedures for this LMN
Prev _AR()	Returns the previous adjacent AR's IP address, or My_IP() if none
Rand()	Returns a random number
Receive_from(MSG, TR)	Receives a message MSG from source TR destinated to the current node
Refresh_reply(SRC, DST)	Represents a standard refresh reply with given source and destination pair
Refresh_req(SRC, DST)	Generates a standard refresh request with given source and destination pair
Release_data_packets(LMN)	Releases LMN's buffered data packets in First-in, First Out order
Remove_entries(LMN)	Removes all the LMN basic and RO entries on the system's routing table, clears the generic detection algorithm's LMN memory (if at an AR); This function Invokes Routing_table_updated(LMN)
Remove_RO_entry(LMN)	Removes the LMN RO entry on the system's routing table
Report_msg(LMN, TS, L2_PARAMS, AR)	Generates an Report message for LMN with timestamp TS, AR information, and optional L2 parameters information
RO_enabled()	Returns true when the route optimization option is enabled
RO_inform(LMN, TS, STEP, CR, TR, AR))	Generates an RO_inform message for this LMN, with a certain Timestamp TS, de-triangulation step STEP, for informing the crossover node CR to send LMN's packets directly to node AR instead of triangulating via node TR
Routing_table_updated(LMN)	Signals the modification of a routing table entry for a given LMN
Secured_Detection(LMN, TS)	Invoked after a successful detection at an AR, for a given LMN with an associated timestamp TS, with authentication guarantees
Security_init_ack (LMN)	Generates an security update acknowledge to trigger revalidation for this LMN
Security_init_req (LMN)	Generates an Security Update Request to update the new PID at the LMN
Security_validate_ack(LMN, GW, RAND, AR_TS, MNID, LMN_TS)	Receives a security validate ack message for a single LMN, with a random number RAND, timestamps AR_TS and LMN_TS, encrypted LMN MNID and gateway information GW
Security_validate_req(LMN, GW, RAND, AR_TS)	Generates a security validate message for a single LMN, with a random number RAND, timestamp AR_TS and gateway information GW
Send_Decision(LMN, TS, AR)	Invoked to notify the chosen AR of the network-controlled handover
Send_surrogate_ND(LMN)	Sends a surrogate neighbor discovery for given LMN to associate the default router IP address with this node's MAC address
Send_to(MSG, TR)	Sends message MSG from the current node to node TR
Sign(MSG, PID)	Digitally sign message MSG with key PID, by appending a encrypted digest of the message to it
Smooth_detriangulation_enabled()	Returns true when the smotth de-triangulation option is enabled
Start_buffering_packets(LMN)	Starts buffering subsequent received data packets for LMN until release_data_packets() is called, or an internal timeout occurs
Start_timer(TR)	Cancels the timer associated with TR in the current node, if any, and installs a timer associated with TR in the current node with the default duration
Stop_timer(TR)	Cancels timer associated with TR in the current node, if any.
Timeout(TR)	Signals the elapsement of time period associated with node TR.
Trigger_data_RO(LMN, CR)	Starts data RO dissemination mechanism for this crosshover node CR
Trigger_Data_RO_enabled()	Returns true when the data RO dissemination mechanism is enabled
Trigger_Fast_RO(LMN, TS)	Starts data Fast HO dissemination mechanism at this AR. Equal to Confirmed_Detection(LMN, TS) && Trigger_Fast_RO_enabled()
Trigger_Fast_RO_enabled()	Returns true when the data Fast HO dissemination mechanism is enabled
True()	Logical function that always succeeds
Unsecured_Detection(LMN)	Invoked after a detection at an AR, for a given LMN without authentication guarantees
Update_msg(LMN, TS, BASIC_FLAG, RO_FLAG, IDLE_FLAG, REMOVE_FLAG, DECISION_FLAG, ACK_FLAG, AR)	Generates an update message for LMN with timestamp TS, optional RO AR information, and the set of specified flags
Update_RT(LMN, TS, BASIC_FLAG, RO_FLAG, IDLE_FLAG, REMOVE_FLAG, DECISION_FLAG, NEXTHOP, AR)	Updates the routing table with the next-node and final-node information, a given timestamp for LMN and the set of specified flags. If BASIC_FLAG is present, any existing RO entry is removed first for this LMN. This can either result on the creation, modification or removal of a corresponding routing entry at the system's routing table. This function invokes Routing_table_updated(LMN)
Verity_PID(message, PID)	Verifies the digital signature of a message with key PID
Wireless_ITF()	Returns the wireless interface of this AR

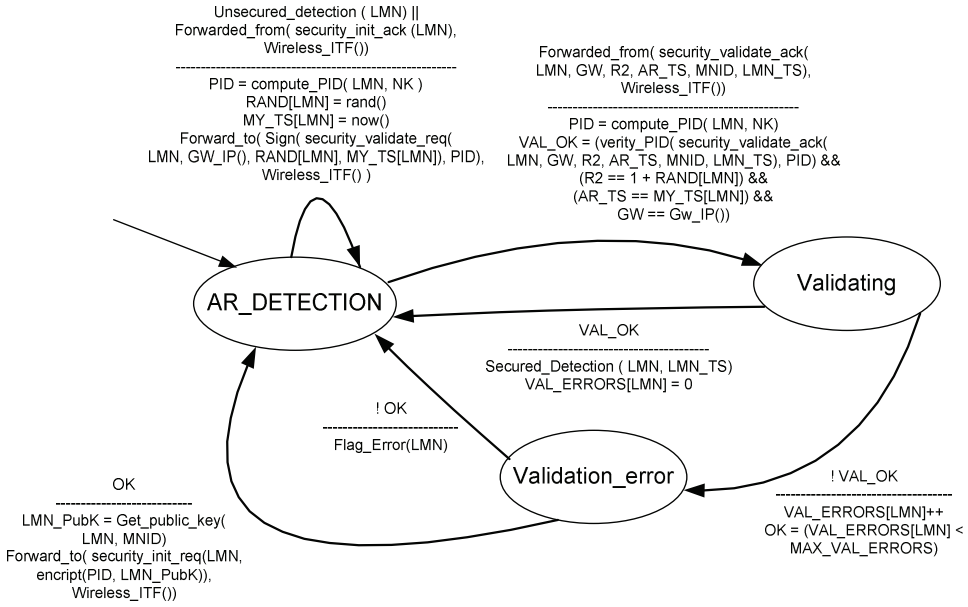
Appendix C Formal Specification State Machines and Variables

DETECTION: generic detection algorithm



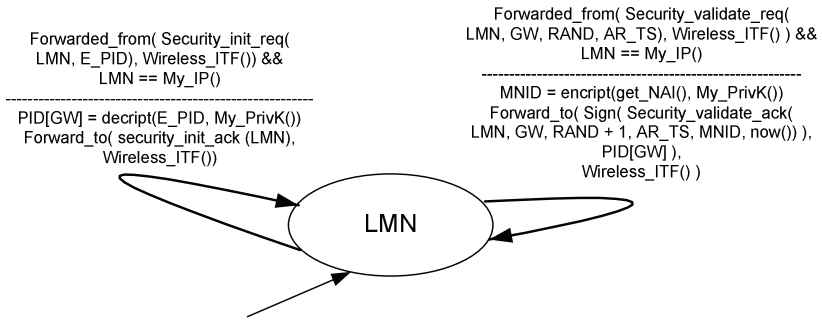
KNOWN[]	Array that holds the information that this LMN is already been detected in the current GDA timeout period
PAYLOAD	Payload of the received data packet
SRC	IP Source address of the received data packet
DST	IP Destination address of the received data packet
IS_EXPECTED_REPLY	Decision variable that holds if the received request reply packet was the correct reply to the last refresh request generated for this LMN by the current AR.
R_FROM[]	Array that holds the source address of the last refresh request for this LMN
ENABLED[]	Selectively disables the GDA detection for a certain LMN, used in the Idle refresh mechanism.
LMN	Legacy Mobile Node's IP address
TR	Original agent that requested the Refresh operation

DETECTION SECURITY: AR NODES

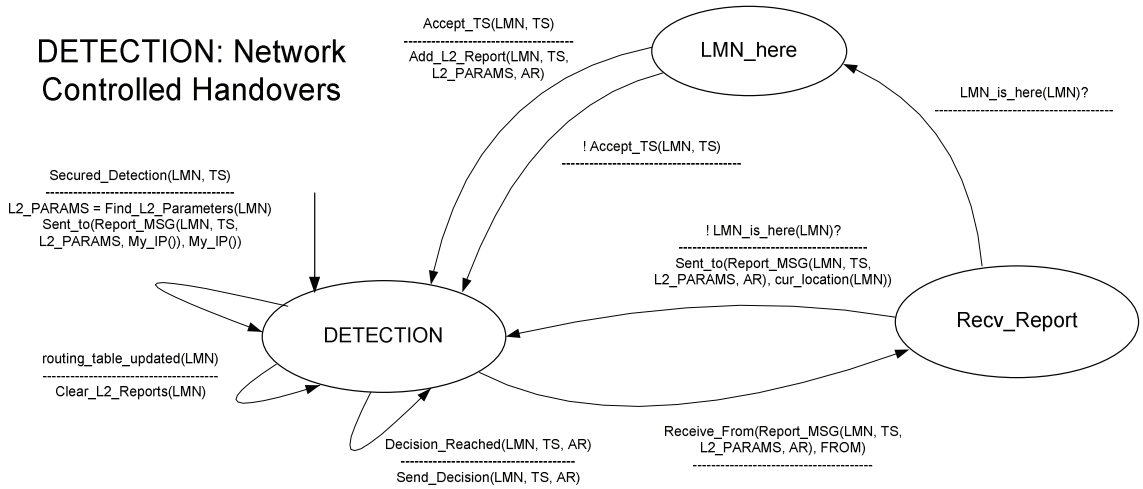


LMN	Legacy Mobile Node's IP address
RAND[]	Array that holds the last AR random number for this terminal
MY_TS[]	Array that holds the last AR timestamp for this terminal
LMN	Legacy Mobile Node's IP address
GW	Gateway IP address given by the LMN
AR_TS	Timestamp nonce of the AR given by the LMN
LMN_TS	Timestamp nonce of the LMN given by the LMN
R2	Random number given by the LMN
MNID	opaque LMN identifier encrypted with its private key
VAL_OK	Decision variable that holds if the received validation reply packet passes all the security requirements.
NK	Secret network key shared between the network's agents
PID	Session key shared between the LMN and this network
VAL_ERRORS[]	Counter of sequential failed validations, per LMN
OK	Decision variable that holds if the last failed validation is below the maximum administrative limit
LMN_PubK	Public key of this LMN

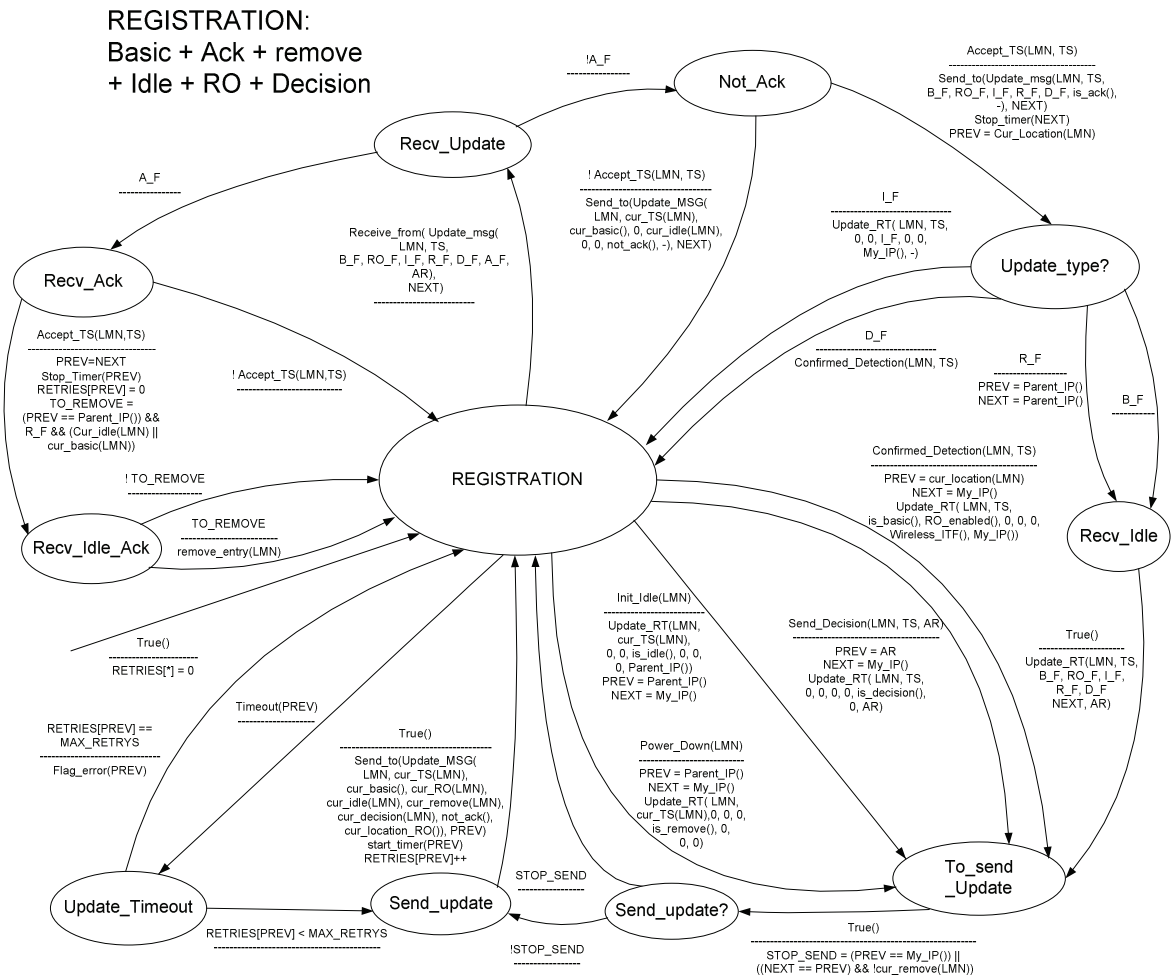
DETECTION SECURITY: LMN NODES



LMN	Legacy Mobile Node's IP address
GW	Gateway IP address
RAND	Random number given by the AR
AR_TS	AR timestamp given by the AR
PID[]	Session key shared between the LMN and this network, per network
E_PID	Session key given by the network in the initialization procedure
MNID	opaque LMN identifier encrypted with its private key

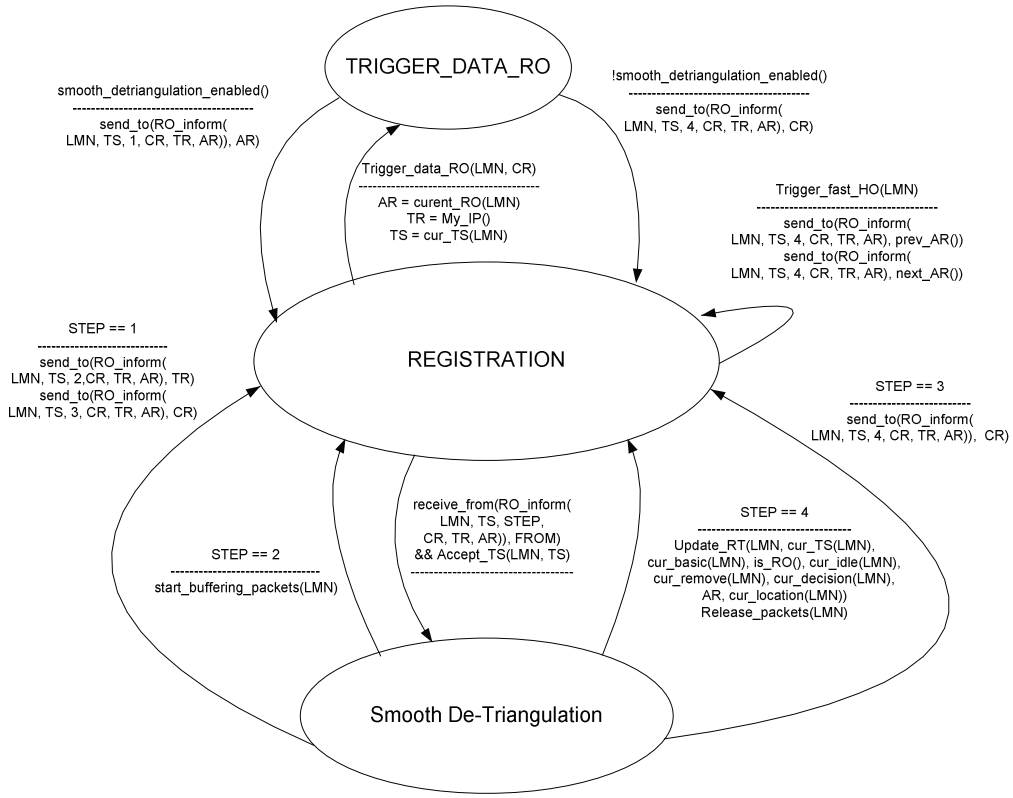


LMN	Legacy Mobile Node's IP address
TS	Timestamp
L2_PARAMS	Specific Layer 2 parameters acquired from the AP
AR	LMN's location
FROM	Previous agent that sent this message

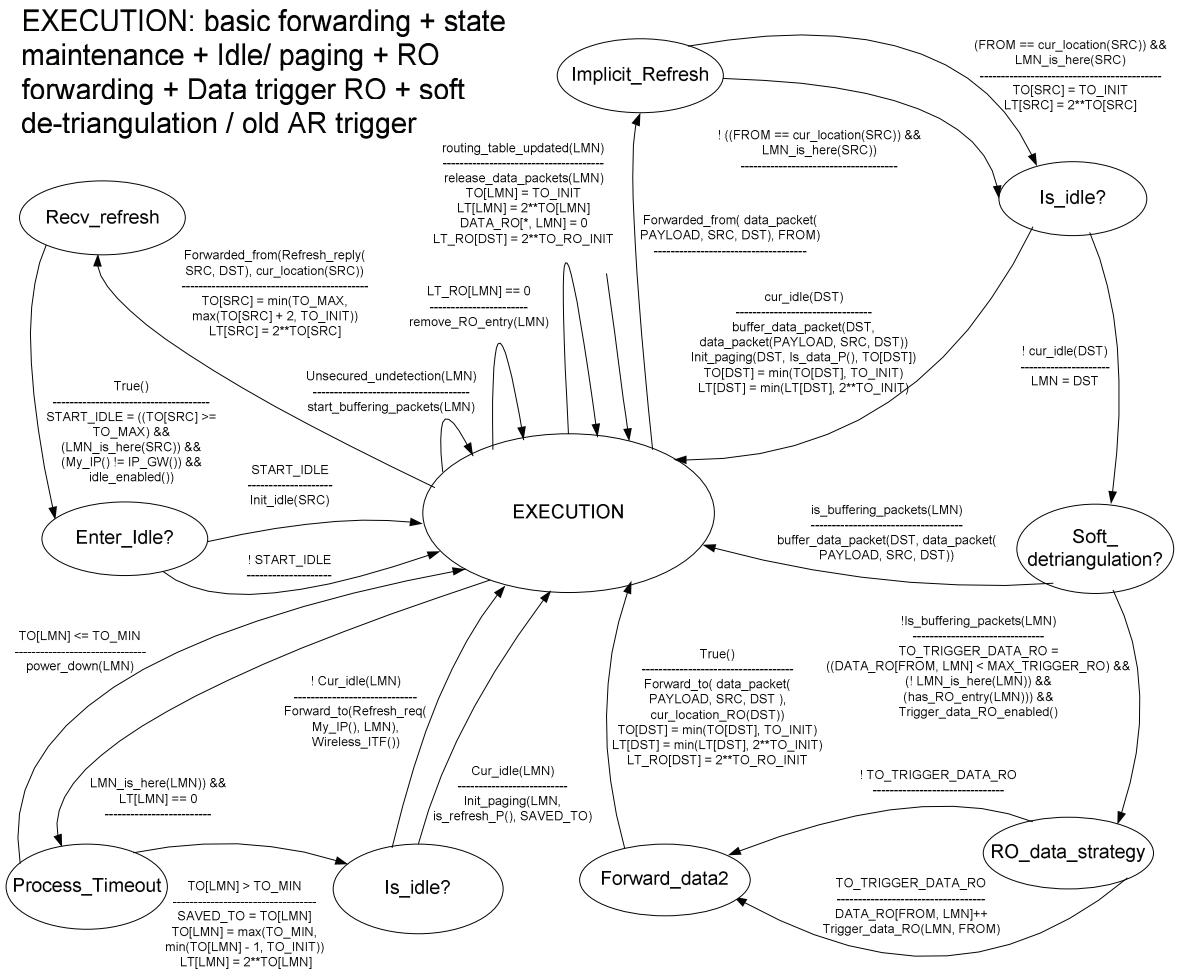


LMN	Legacy Mobile Node's IP address
TS	Message TimeStamp
B_F	Basic Update Flag
RO_F	Route Optimized update Flag
I_F	Idle Update Flag
R_F	Power-down Update Flag
D_F	Network-controlled Decision Update Flag
A_F	Ackowledgement Flag
NEXT	agent that has sent this update message for this LMN to the current node, being updated in the routing table as the new next-hop agent for this LMN
PREV	previous agent where the LMN was located (by consulting the routing table before updating it with the value of the NEXT variable), and where the update message will be sent to
AR	LMN's real AR location, (if known)
TO_REMOVE	Decision variable that holds if the LMN's basic or idle entries should be removed during an Power-down or successful reception of acknowledgement
TO_SEND	Decision variable that holds if the update message should be propagated to the previous location of the terminal (agent PREV)
RETRIES[]	Counter of sequential unackowledged sent messages, per agent

REGISTRATION: data strategy
+ fast handover +smooth de-triangulation

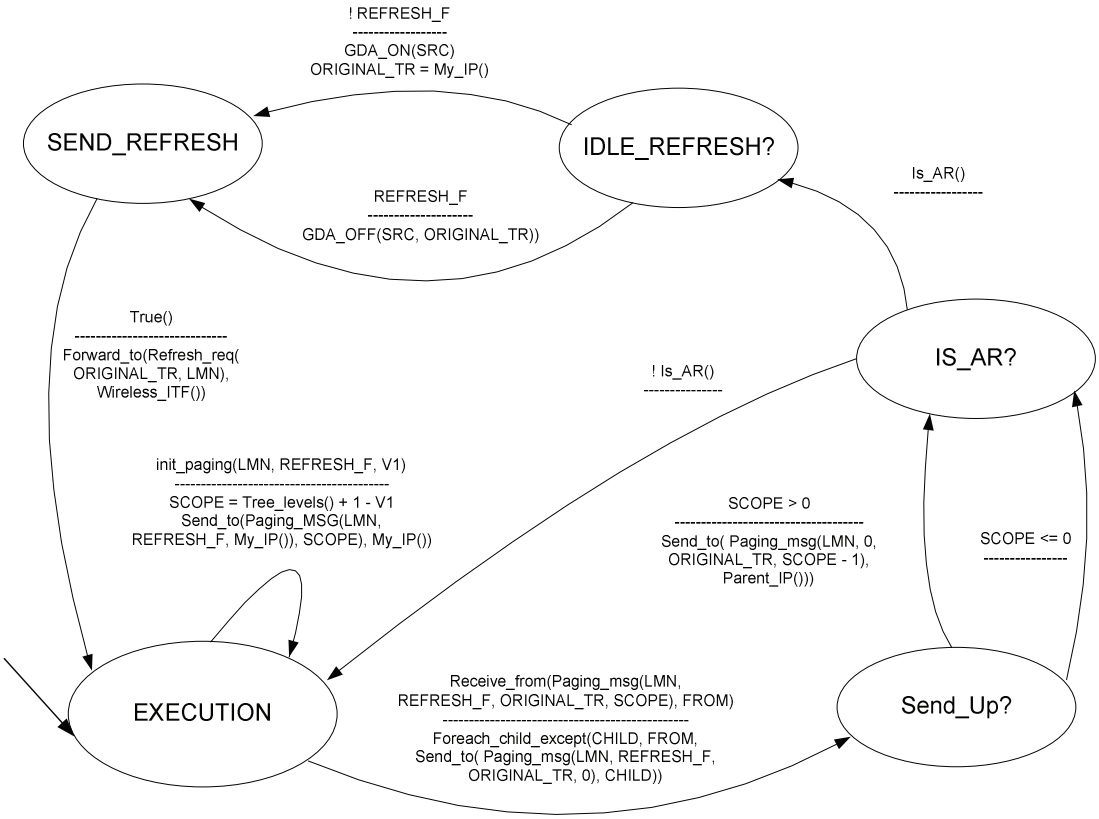


LMN	Legacy Mobile Node's IP address
CR	Crossover Node Agent
AR	LMN's Location
TR	Triangulating Node
TS	LMNs's Timestamp
FROM	Previous agent that has sent this control message to the current agent
STEP	Current Step of the De-triangulation process



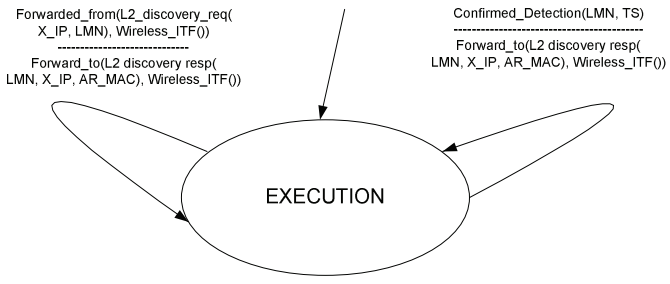
PAYLOAD	Payload of the received data packet
SRC	IP Source address of the received data packet
DST	IP Source address of the received data packet
FROM	Previous agent that has sent this data packet
LT[]	Current basic soft-state entries Lifetime until the next refresh operation, per LMN
LT_RO[]	Current RO soft-state entries lifetime
TO[]	Current basic soft-state refresh cycle length, per LMN
LMN	Legacy Mobile Node's IP address
TO_TRIGGER_DATA_RO	Decision variable that holds if the RO dissemination should be triggered
START_IDLE	Decision variable that holds if the Idle operations should be triggered for this LMN
DATA_RO[]	Number of triggered Data RO Disseminations, per LMN and crossover node.
SAVED_TO	Previous refresh cycle length, determines which tree levels are paged on-demand

EXECUTION: Data Paging + Idle Paging



LMN	Legacy Mobile Node's IP address
V1	Current Refresh Cycle Length, determines which tree levels are paged on-demand
REFRESH_F	Flag, distinguishes between Paging and Idle Refresh Operations
SCOPE	Number of parent tree levels involved in this Paging Operation (if positive)
ORIGINAL_TR	Original agent that requested this paging operation
FROM	Previous agent that has sent this data packet
CHILD	Iterator for each Node's children agents

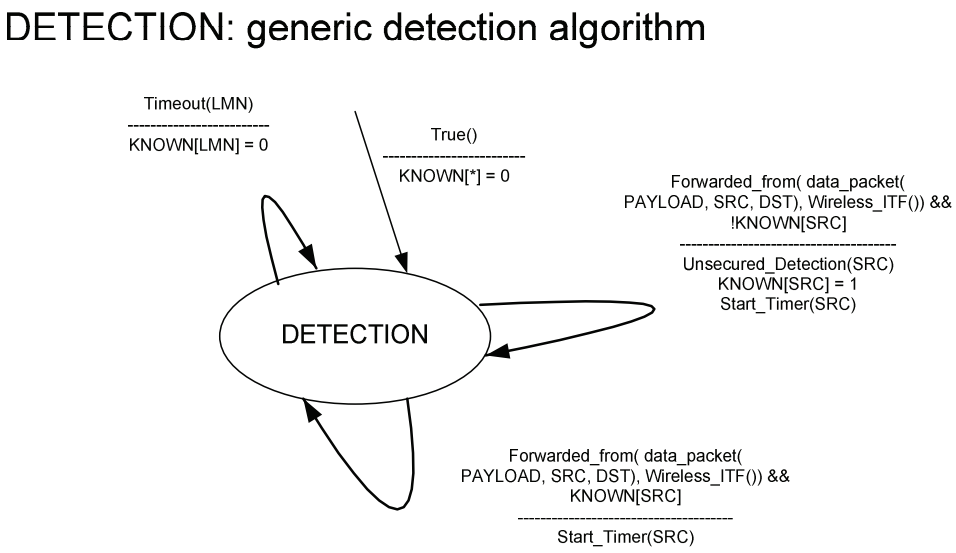
EXECUTION: Neighbor Discovery LMN support



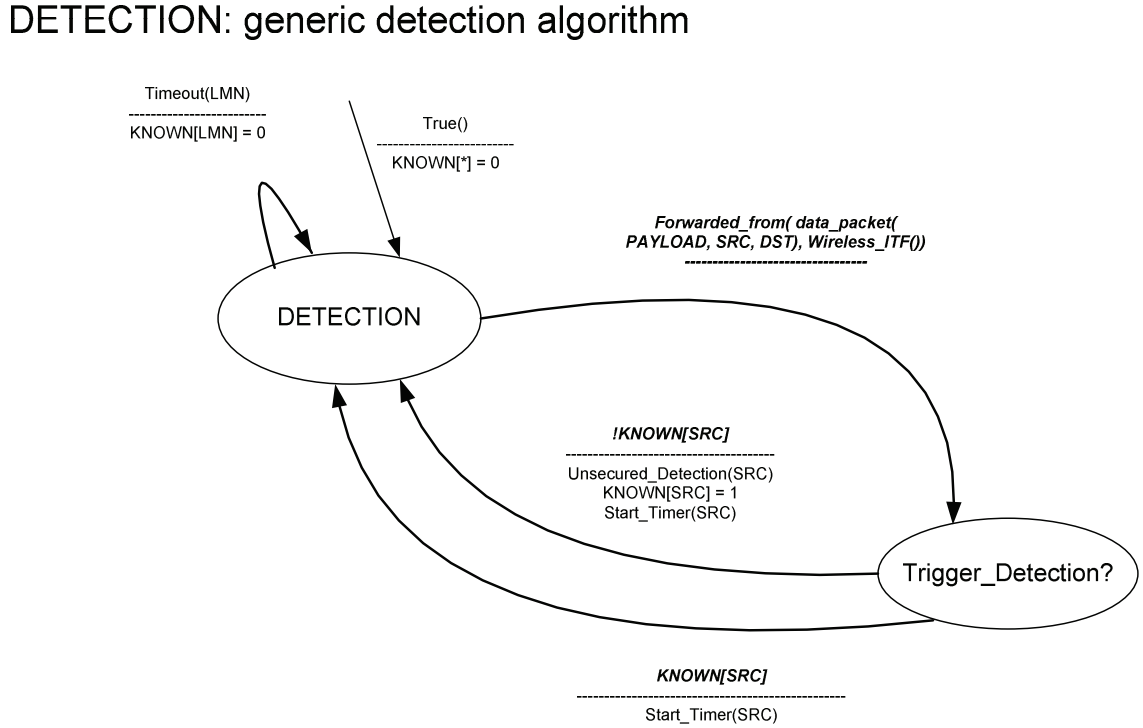
X_IP	IP address to resolve in address resolution
LMN	Legacy Mobile Node's IP address
AR_MAC	L2 address of the AR

Appendix D State Machine Manipulation Example

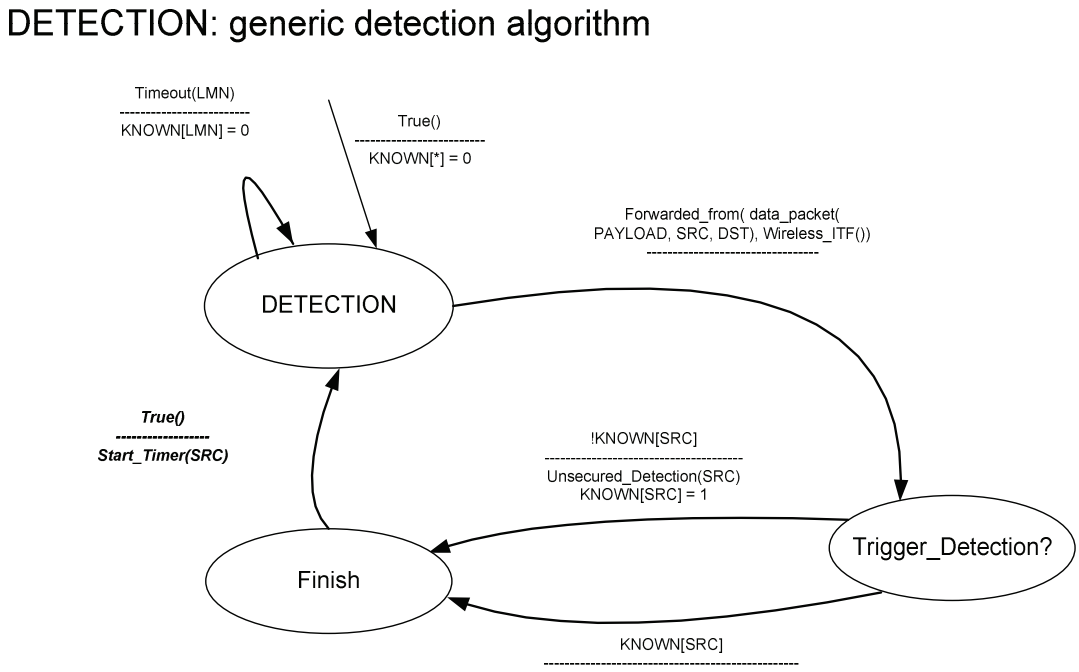
Step 1 - Original Optimized State Machine



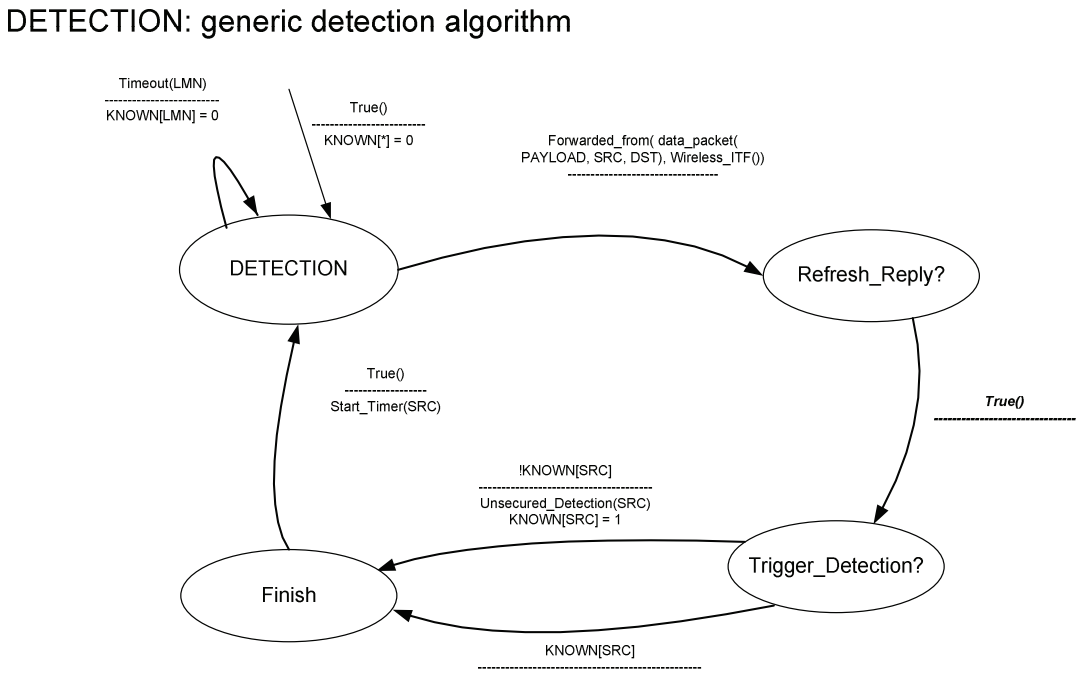
Step 2 - Separation of Common Event



Step 3 - Separation of Common Action



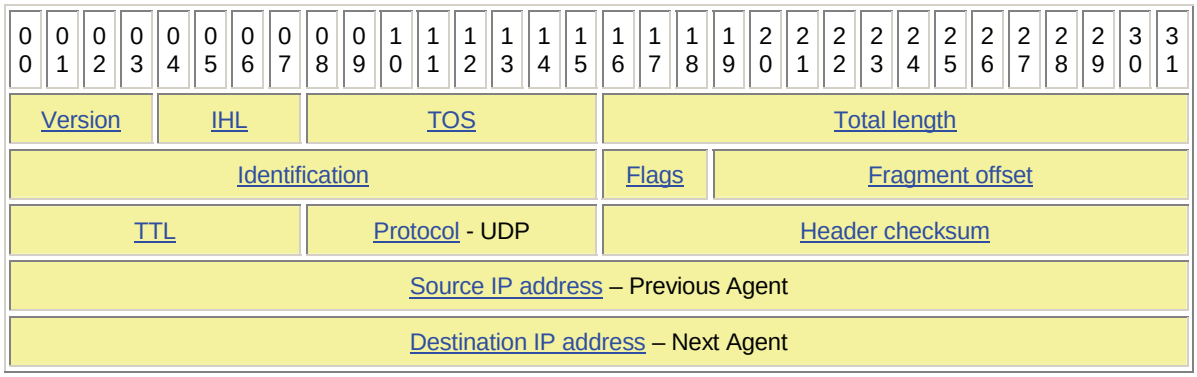
Step 4 - Dummy State introduction to enable extra modular functionality



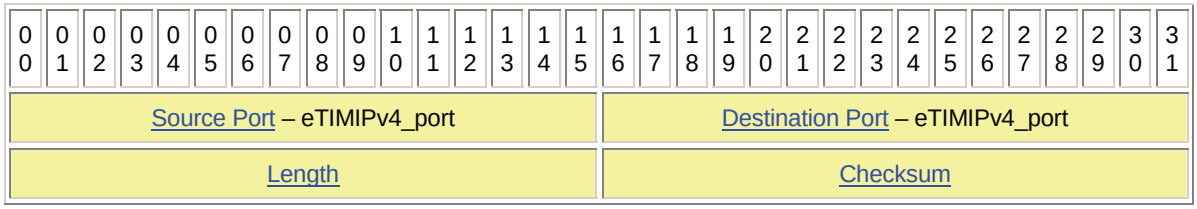
Appendix E eTIMIP packet formats

eTIMIPv4 Control Packets

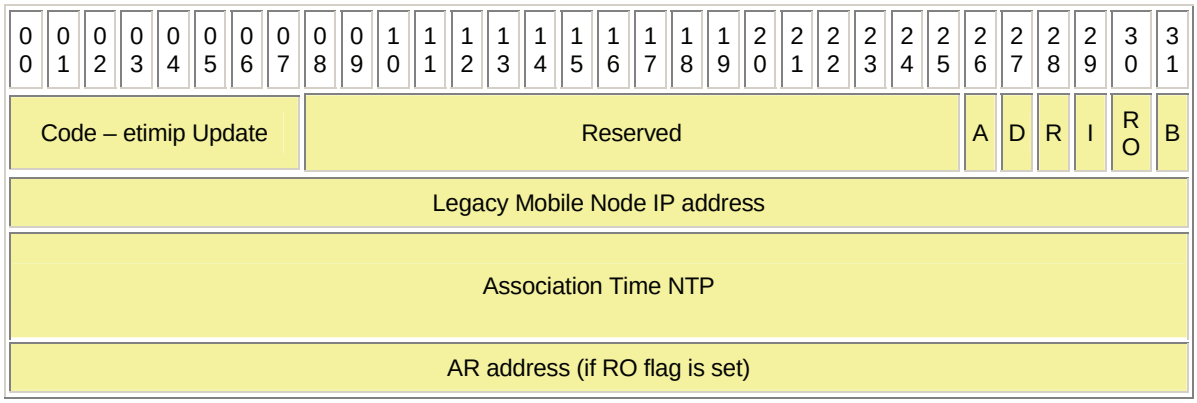
IP header:



UDP header:

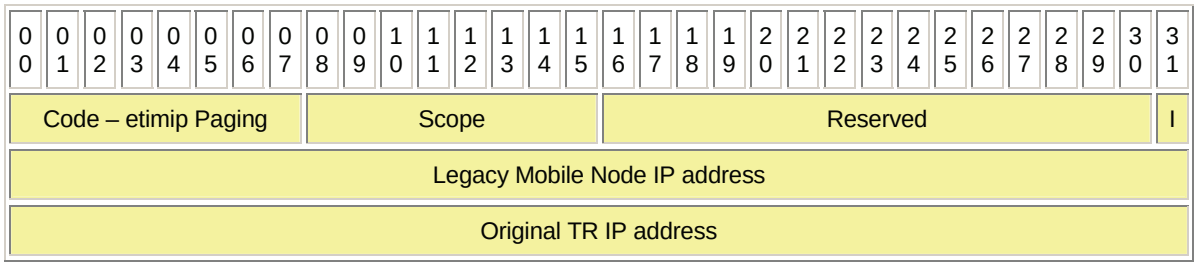


eTIMIP Update header:

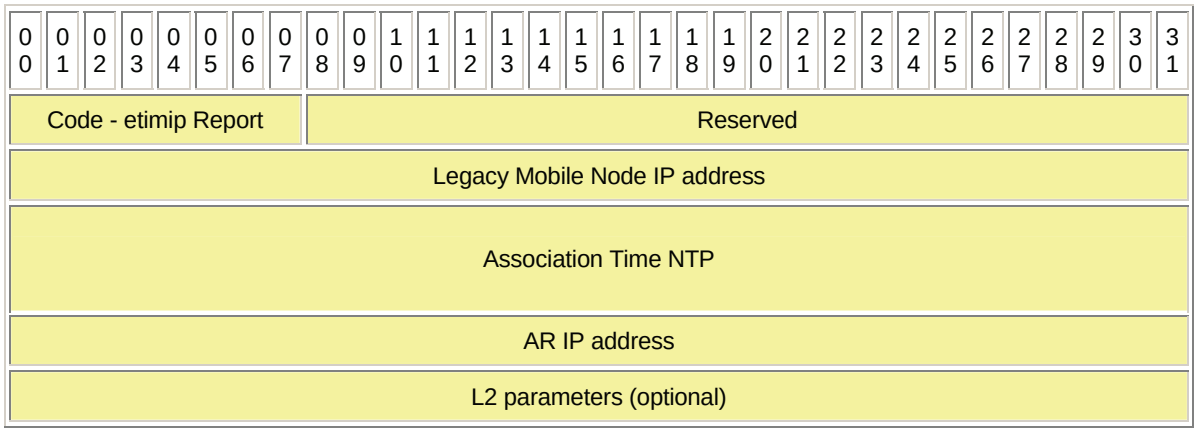


Meaning of flags: **A**cknowledge / **D**ecision / **R**emoval / **I**dle entry / **R**oute **O**ptimization / **B**asic Routing

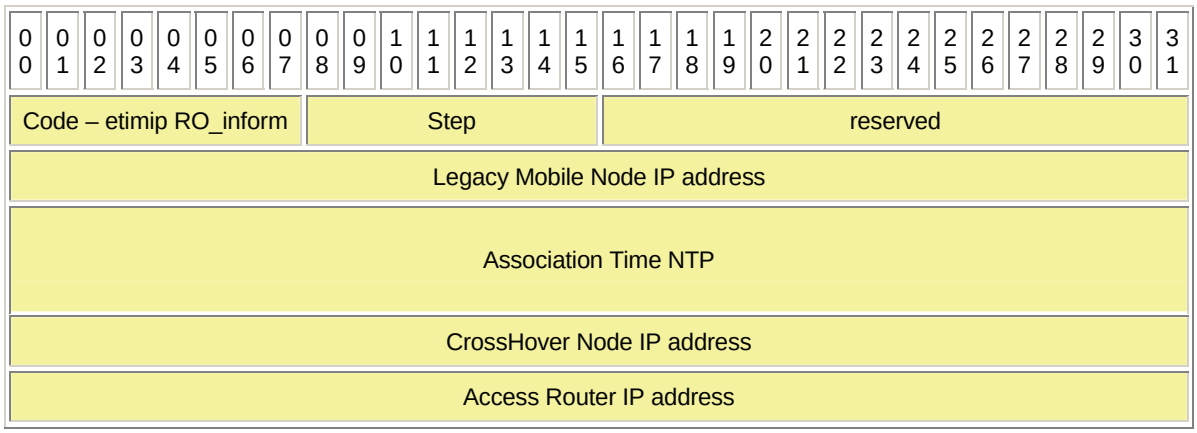
eTIMIP Paging header:



eTIMIP Report header:

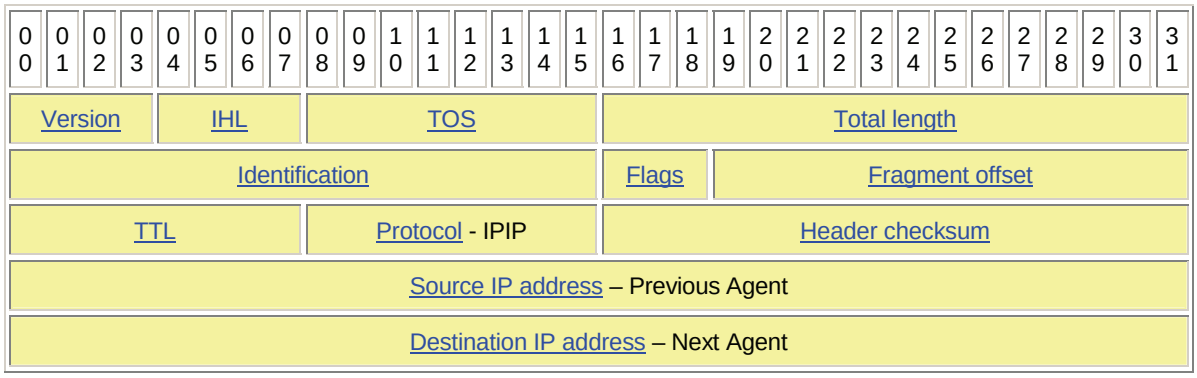


eTIMIP RO_Inform header:

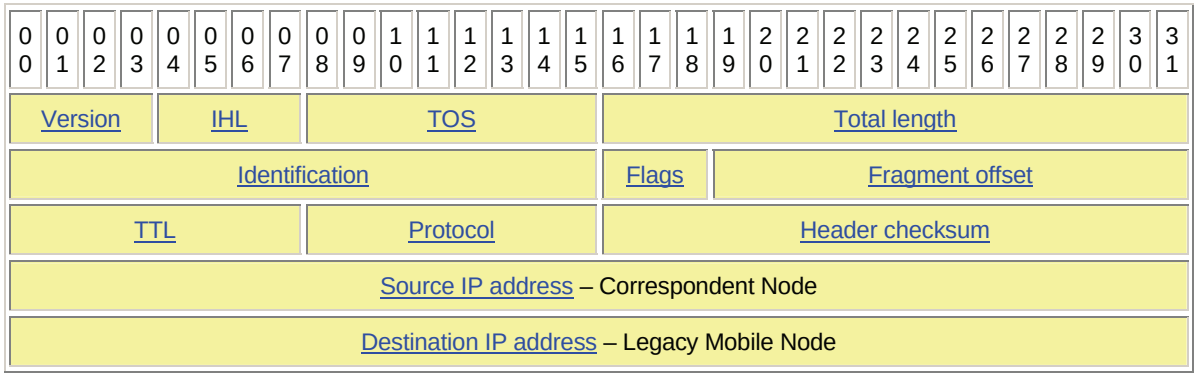


eTIMIPv4 data packets

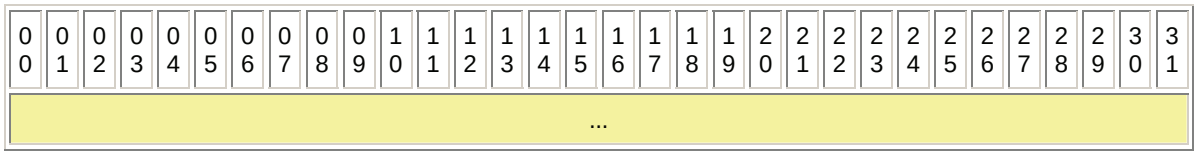
Outer IP header:



Inner IP header:

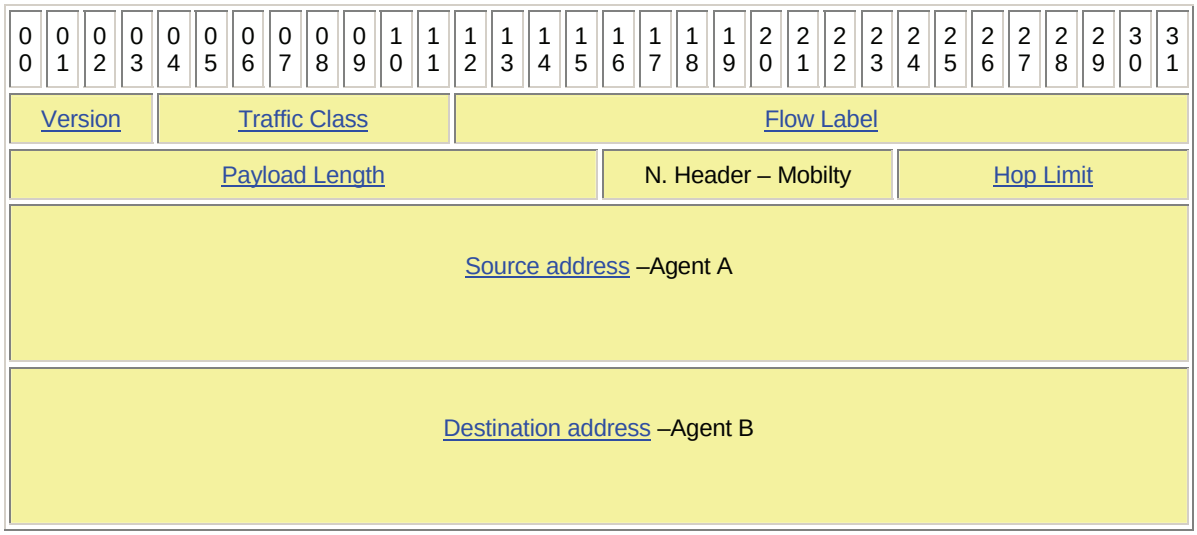


Higher levels:

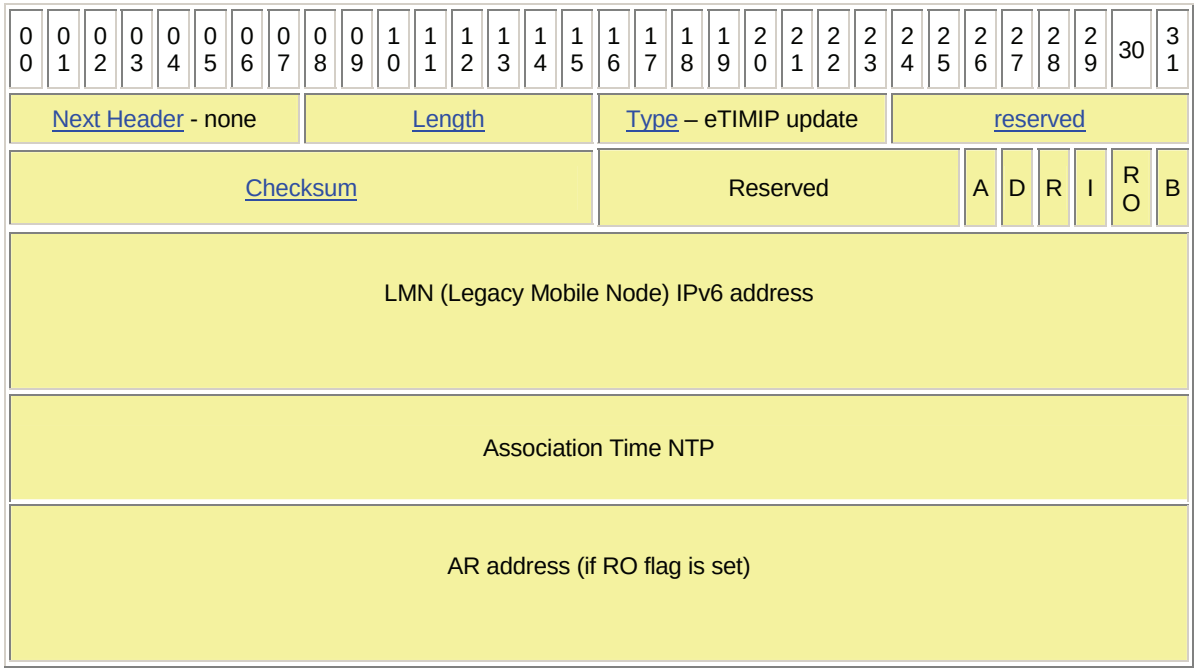


eTIMIPv6 Control Packets

IP header:

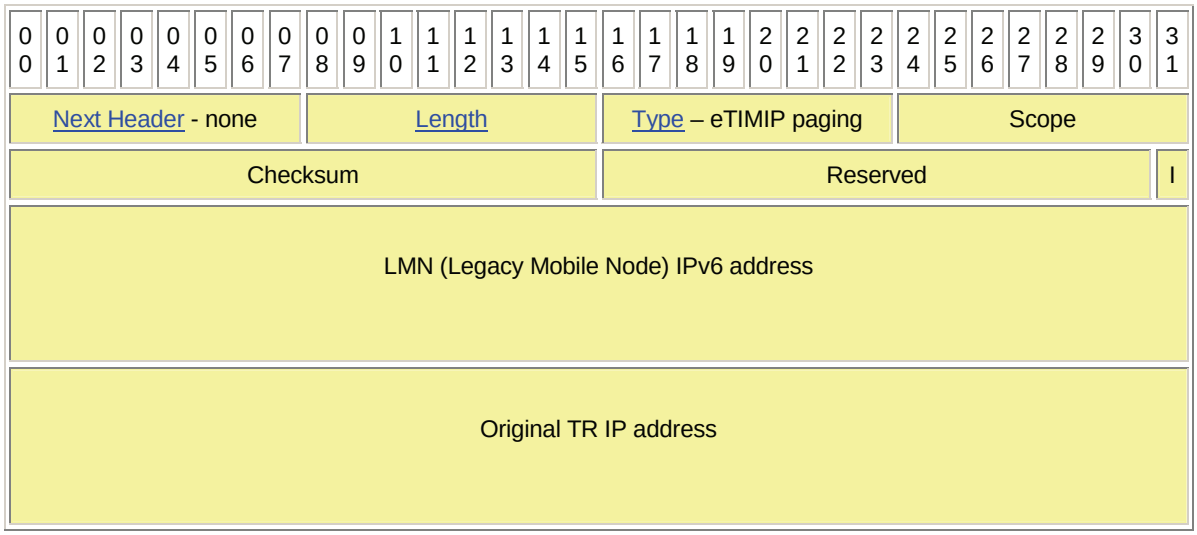


Update Mobility Header:

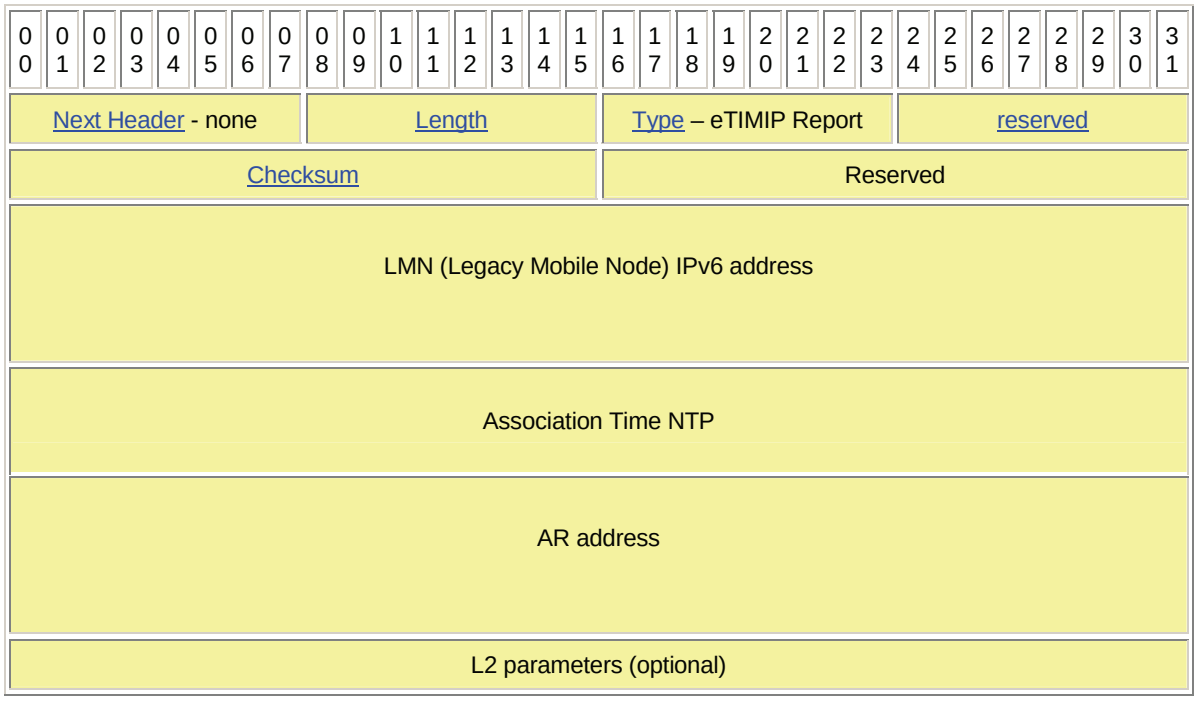


Meaning of flags: **A**cknowledge / **D**ecision / **R**emoval / **I**dle entry / **R**oute **O**ptimization / **B**asic Routing

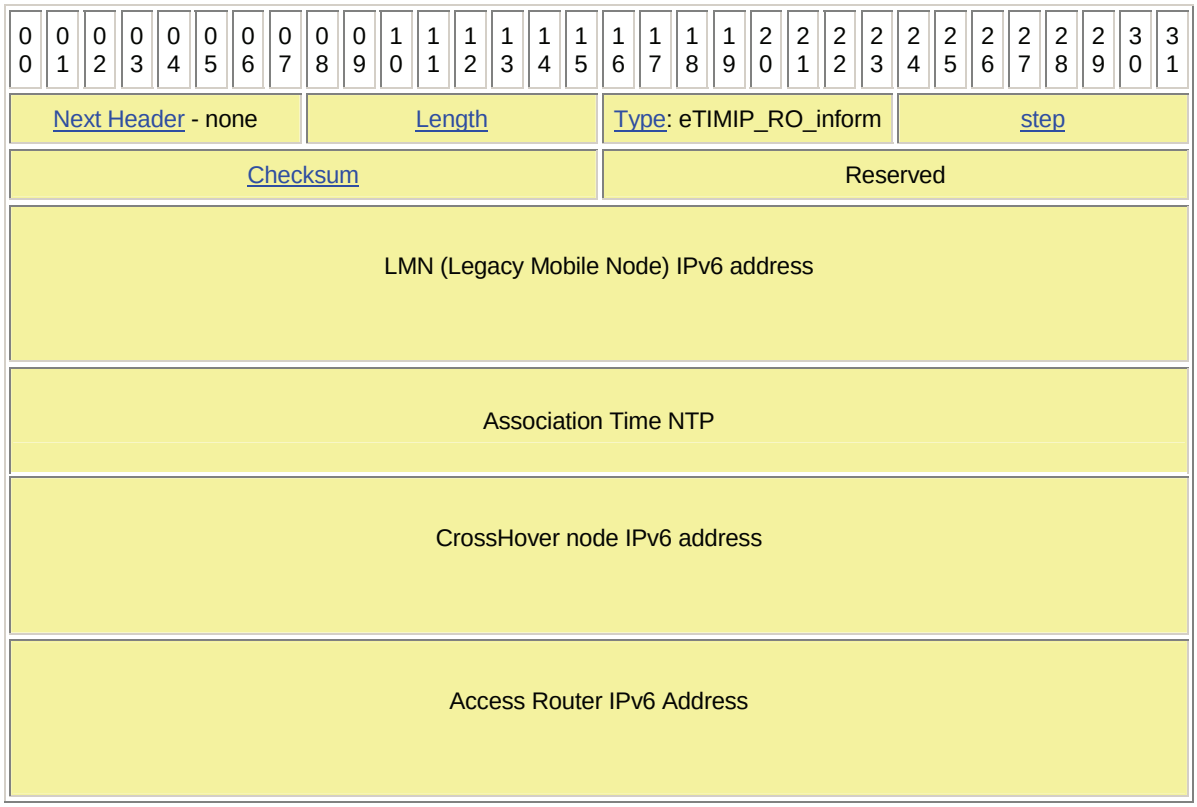
Paging Mobility Header:



Report Mobility Header:

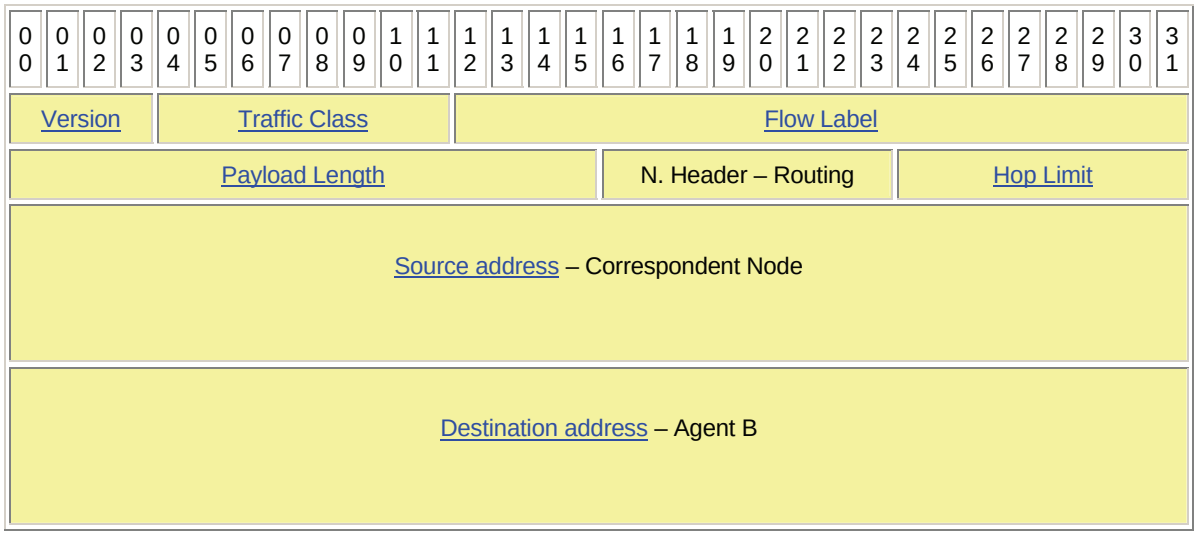


eTIMIP RO_Inform header:

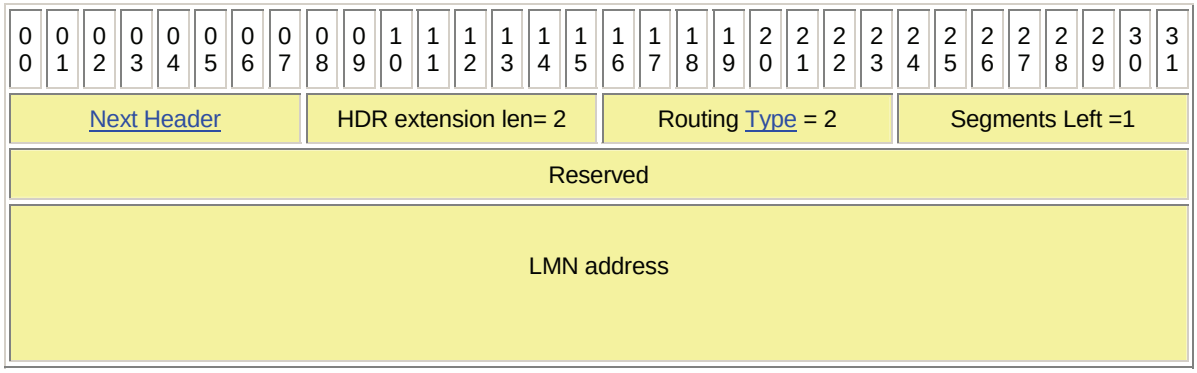


IPv6 Data packets

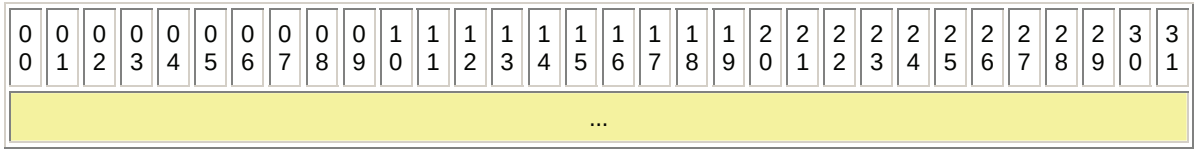
IPv6 Header:



Routing Header:



Higher levels:



Appendix F Simulation Software description

Simulation Pipeline description

Pipeline overview

Figure 168 illustrates a general overview of the simulation pipeline that was implemented to automatically generate the graphs results presented in the simulation studies. This pipeline is composed of multiple bash shell scripts that feed the simulator with series of parameters, in order to produce sequences of individual simulation scenarios, which in turn will be aggregated to produce the final graphs.

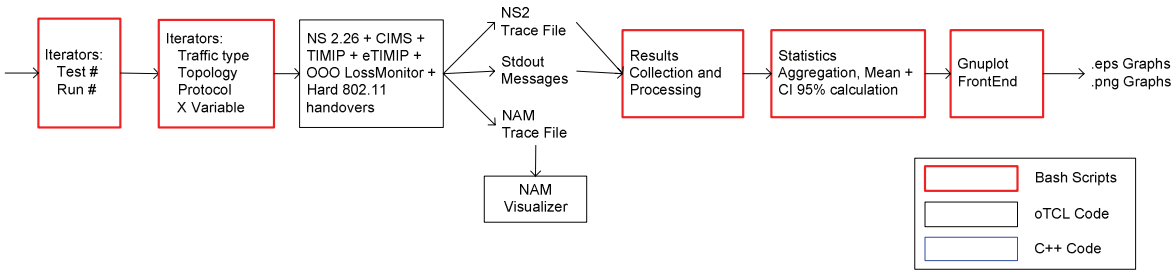


Figure 168: Simulation Method Pipeline

Pre-simulation Iterators

The first module will iterate the available test sets, which are represented in Figure 169. Each test set will focus on a particular test scenario, in which the effect of a single parameter (like mobile speed, mobile position, link failure existence, etc) will be evaluated in all the available metrics for all protocols, traffic types and topologies. Once the test set is chosen, it will be iterated a sufficient number of times with random components, in order to find the average values for the available metrics, and their associated 95% confidence interval.

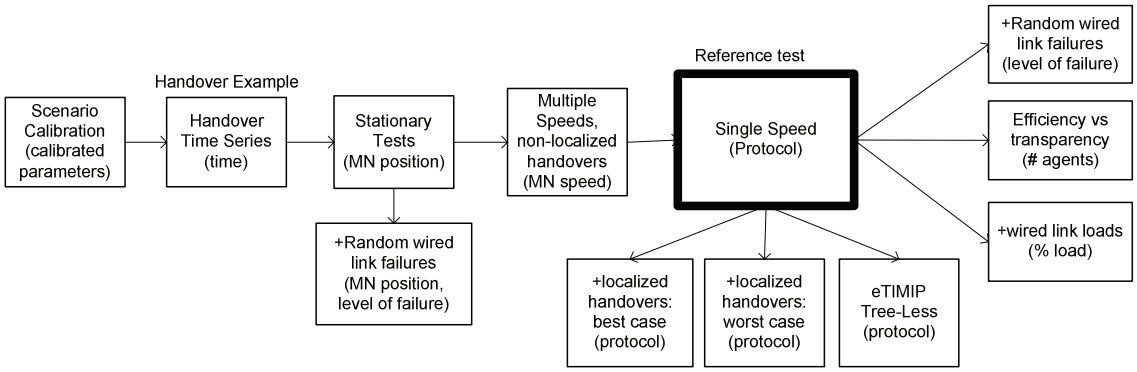


Figure 169: Test set description and iterated parameters

The second module will be specific for each individual test set, and will fix all parameters except the one that is currently under test, which will be sequentially iterated (“X variable”). In addition, the test set will repeat the same scenario for all considered traffic types (UDP / TCP), all topologies (Tree / Mesh) and all considered protocols. Each instrumented individual test will then invoke the simulator, by passing the chosen parameters via the command line.

NS2 simulation

The next component of the pipeline is the NS2 simulator extended with the previously mentioned modifications. The simulator, based on the command line parameters (detailed in Table 11 and Table 12), will instantiate the specific simulation scenario derived from the base one, and model the processes and mechanisms of the chosen protocol.

Each individual simulation run will produce up to three different output results files: The first file will contain the measurements performed by the agents and oTcl components, which are calculated in run-time during the simulation. The NS2 trace file will be generated by the simulator, in the mixed wired+wireless format, being used for post-processing the results after the simulation process. Finally, the simulator can optionally output a NAM trace file that is used for visualization of specific test conditions, using the external NAM animator program.

Option	Description
ang	Enables support for access network gateways
end	Time of simulation stop
h	Simple help command
hh	Verbose help command
d	Debug level
ho_fast	Use etimip fast handover
ho_smooth	Use etimip smooth handover
icoin	Perform specific icoin 2007 paper simulations
l2_hard	Perform l2 802.11 hard handovers
la	Use asymmetric links
lf	Id of link failure
lh	Use hierarchical delays links
ll	Base link delay
ls	Base link speed
lt	Topology type
lth	Number of hierarchical agent levels
ml	Perform localized handovers
mp	Initial mn position
mr	Number of mn round movements
ms	Mn speed, in handovers/minute
nam	Generate nam trace and run animator
p	Mobility protocol
ro	Use etimip route optimization
ro_soft	Use smooth de-triangulation option

Table 11: Mobility simulator command line options

Option (cont)	Description (cont)
soft_alg	Smooth de-tringulation algorithm
run	Random seed
tb	Maximum size of traffic buffer
tbe	Amount of load traffic
ti	Generate inter/intra domain traffic
tr	Cbr packet rate/min
trace	Disable trace files
treeless	Use degenerate etimip tree
ts	Data packet size
tt	Generator type (TCP/UDP)

Table 12: Mobility simulator command line options (cont)

Post-simulation Iterators, Statistics calculation

Taking the two generated files, the subsequent pipeline component will collect the already processed results in the stdout file, and will process the trace file in order to generate the final metrics for this individual simulation run.

The next pipeline component will be executed after all independent runs for a given individual protocol and set of parameters are finished. For this, each measured metric will be aggregated to the average of the measured independent runs, this average value being used to calculate the 95% confidence interval value that expresses measured error range induced by the multiple independent runs [162].

Finally, the measured metrics of each individual test are presented in graphs, through the usage of a front-end to the Gnuplot program [185] that generates the appropriate Gnuplot scripts and that produces the .eps and .png versions of the graphs. Essentially, the pipeline produces two types of graphs, linked to the type of iteration. In the former type of graphic, line graphs show the variation of each metric (y axis) in relation to a varying parameter (X Axis); each line will then link the Y values of the same protocol. In the latter type of graphic, bar graphs show the variation of each metric (y axis) in relation to the protocols, represented in the X Axis.

NS2 Simulation modelling description

NS2 Mobility Support

The developed NS2 mobility code was built on top of the Network Simulator (NS), version 2.31, a discrete event simulator targeted at networking research, which provides substantial support for simulation of TCP, routing, and multicast protocols over wired, wireless (local and satellite) and hybrid networks. This simulator was developed by the VINT project and for the last few years has been maintained by the Information Sciences Institute (ISI) group of the University of Southern California.

Figure 170 illustrates the existing options for the simulation of IP mobility protocols in NS2. The figure shows that only the base MIP protocol is included in the core simulator; all the other protocols and support modules are maintained as external contributions, which are typically tied to the particular version of NS2 that the code has been branched-off from. Almost all such modules are publicly available for research with open licenses; however, at the time of writing, some modules were not available by the authors yet.

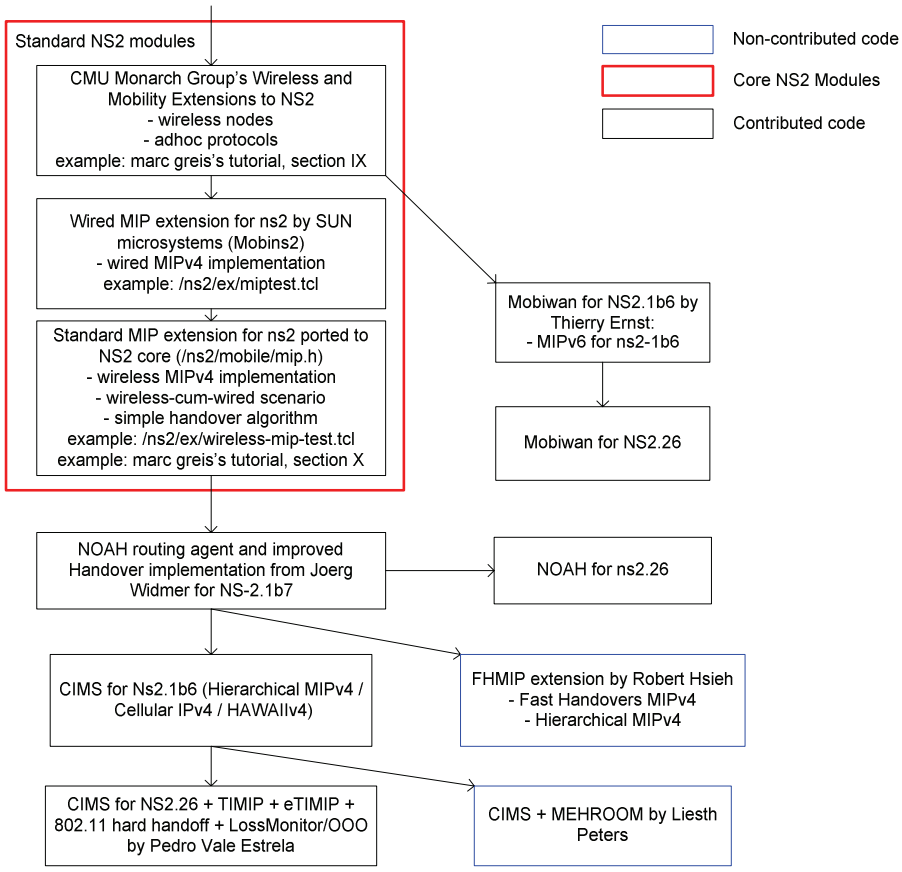


Figure 170: NS2 IP mobility modules

Regarding the module contributed by this PhD thesis research, it extends the version 2.31 of the base simulator with the eTIMIP model, featuring mobility of Legacy mobile nodes, support of Legacy Routers, basic routing, optimized routing, multiple access network gateways, and modelling of the optimal smooth de-triangulation algorithms. For comparison purposes, the code includes an original implementation of the TIMIP protocol, and implementations of the CIP, HAWAII and hMIP protocols. These implementations were derived from the Columbia IP Micro-Mobility Suite (CIMS) v1.0 [180], originally developed for NS2.1b7, being upgraded to this recent version of the simulator. To integrate all the different protocols in common scenarios, an object-oriented oTcl framework was developed to ease the systematic test of the mobility-protocols in the same scenarios.

All the protocols use the hybrid wired-cum-wireless NS2 scenario that supports hybrid base stations, composed of both wired and wireless interfaces. As all wireless nodes in this scenario must **always** have an ad-hoc mobility agent, these protocols use the "no-ad-hoc" (NOAH) [181] module that emulates a simple infra-structured 802.11 interface supporting soft-handovers. This generic module, which was also upgraded to NS2-26, was also customized to simulate 802.11 infra-structured behaviour with multiple channels. This forces L2 hard handover operations, where a station can only receive packets via its current associated AR, this behaviour being closer to real 802.11 networks [129] than the soft-handovers paradigm that were considered in previous simulations studies, as in reference [45].

Finally, the code also includes a generic model of a UDP receiver, extended and compatible with the standard one, that besides the detection of dropped packets, also detects out-of-order and late packets.

Internal NS2 software components

Figure 171 illustrates the internal components that constitute the micro-mobility simulator, which is instrumented using the scripts previously described.

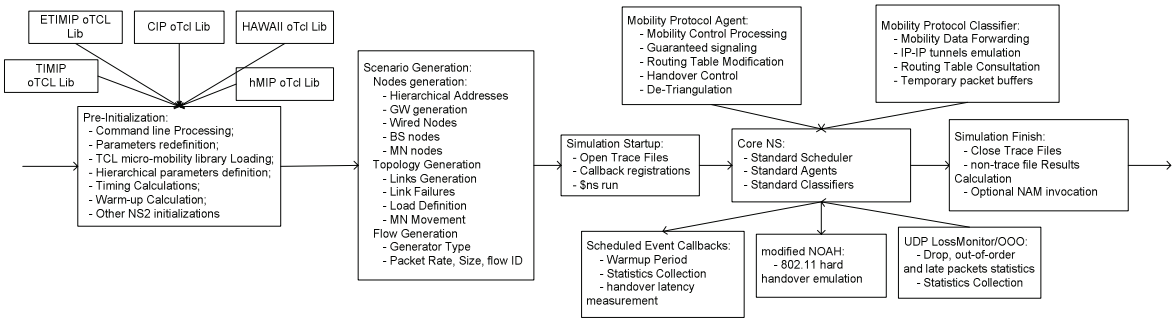


Figure 171: NS2 micro-mobility simulator pipeline

Pre-initialization

When the simulator is invoked, a pre-initialization mechanism coded in oTCL is invoked that starts by processing the command line parameters and that redefines the default options. Then, based on the indicated protocol, a specific micro-mobility library is loaded into the simulator, which defines a set of object-oriented functions that model each protocol. Lastly, a series of parameters and values are calculated automatically, namely the NS2 Hierarchical definitions, timing values and the length of the warm-up cycle of the traffic generators.

Scenario generation

Then, a series of oTCL functions will start-up the specific scenario chosen for the simulation. This process starts by generating the nodes; these are divided into: the purely wired nodes, that compose the micro-mobility's GW, routers and external nodes; the purely wireless nodes, which are the MNs; and the hybrid NS2 base stations (ARs). All these nodes are automatically initialized using standard NS2 methods, and additionally initialized using the specific micro-mobility functions loaded from the mentioned micro-mobility library.

The process continues by generating the topology of the reference scenario, which was previously described in Figure 46 (located in section 5.1.1, page 94). This process considers the instantiation of the specific base tree and the additional mesh links and links that connect to the outside of the domain. The scenario initialization continues by scheduling the link failures occurrence, if any, and the load ratio of the backbone links, which only force extra queuing delay (as sufficient buffer space is allocated to avoid drops). The load generation is emulated as follows: each link features an Exponential On/Off traffic source, with the parameters calibrated in such a way that produces bursts of queued packets that results in a variable long-term average link usage. For this, the generator "data_rate" variable was set to the double of the link capacity, while the "burst_time" variable was set to a fraction of the total generator period, where 100% of load equals half the total generator period.

Finally, the specific MN location and movements are instantiated, which will depend if stationary or continuous movement is desired: the former only considers the MN's desired AR location; the latter additionally considers the desired average speed, number of handovers, and the first and finishing ARs.

The final part of the scenario generation deals with the test flow generation. As mentioned previously, in all tests the moving MN will be the receiver of a UDP or TCP test flow, that is

generated either inside (MN2) or outside the domain (CN). This is instantiated by the appropriate NS2 generator and agent objects.

The functions that implement the scenario generation will schedule the series of ordered events, depicted in Figure 172. The simulator always starts in time zero (tag “t0”). Immediately after that, the MNs are scheduled to move to their initial positions.

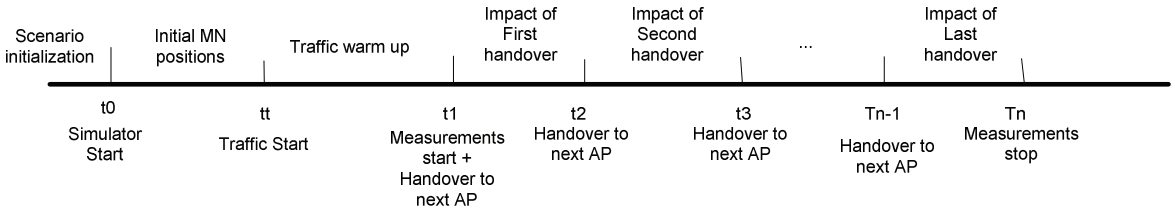


Figure 172: Calculated time instants for moving MN

After the traffic has been started and the warm-up period has passed, the start-up functions will schedule the first handover and the start of the measurements at time t1. Then, the following handover is scheduled at time t2, at a particular time instant determined by the desired MN movement speed, and so on. Finally, after the last handover at time Tn, the measurements will be stopped at the time instant of the following handover, as determined by the MN speed, which enables the measurement of the impact of the last handover on the test flow. Regarding the stationary tests, the simulation runs for 20-seconds after the warm-up period, being this a constant time period sufficient for the precise measurement of the available metrics.

All these time instants are randomized depending on the random seed, which is the same for all protocols per simulation run. For this, the handover time instants are added by a uniform random variable of interval [-100..+100] ms, which by being centred on 0, maintains the average simulated mobility speed constant in each simulation run. Besides the handover time instants, the random seed also randomizes other NS2 components, namely the traffic start, load start and multiple TCP and wireless back-off random variables.

Simulation Start-up

The last stage of the initialization of the simulator occurs by setting up the desired trace files and by a series of otcl call-back registrations, and the simulation is started by calling “$ns run”.

Simulation Execution and Event handling

When this happens, the control is given to the core C++ NS2 scheduler, which starts simulating the specified scenario by means of internal messages passed between the components. At certain points of the code, the mobility protocol model is invoked, being the control and the data paths treated separately.

The control path is handled by the eTIMIP Protocol Agent component, coded in C++, which generates and processes the mobility messages featuring the same fields defined in the eTIMIP specification (section 4.3), performs the optimized surrogate detection based on the 802.11 association, and implements the various handover schemes, including the optimized routing, smooth handover, fast handover and smooth de-triangulation schemes. The data path is handled by the eTIMIP Protocol classifier component, also coded in C++, which performs the data forwarding functions to LMNs. This operation is done by consulting the shared routing table maintained by the eTIMIP agent, to find the next agent where this data packet should be sent to. This module also emulates the IP-in-IP tunnelling which is used in the

cases where encapsulation is required, and triggers the RO dissemination when it detects that a sending node has an outdated RO routing entry. Finally, also driven by the agent component using internal triggers, the classifier will also implement the internal buffering used by the smooth handovers and the smooth de-triangulation schemes.

The lower part of Figure 171 also depicts some of the modified modules and scheduled support call-backs. The NOAH agent, coded in C++, was modified to emulate a 802.11 infra-structured behaviour which is closer to existing deployed networks, by emulating 802.11 hard handovers between different frequencies. In this modification, the AP will drop the packets destined to stations that are currently associated to other APs. Also coded in C++, the extended LossMonitor will be invoked for every received UDP packet at the MN. While the original agent only supported the counting of dropped packets, by observing the discontinuities of the sequence number of the received packets[22], the extended LossMonitor receiver uses the sequence number and a timestamp field to detect drop, out of order and late packets, and to measure the one-way delay. Finally, the Scheduled Event call-backs contain oTCL functions that are invoked at run-time and that implement the mentioned warm-up period, the traffic/load start instants, and the statistics calculation, either global or per handover.

Simulation Finish

The last stage of Figure 171 is executed at the end of the simulation, at time Tn, in which the trace files are closed and the final results are calculated and sent to the standard output file.

Simulation Parameters Calibration

This section presents the preliminary simulations that were performed to calibrate selected simulation parameters, using the scenario described previously. These parameters include the amount of generated UDP packets, the number of sequential handovers and the value of the internal link delays. Such parameters were sequentially calibrated, in a way that does not benefit or harm any of the studied mobility protocols. In addition, the load generation mechanism was also calibrated and measured the amount of load that it introduces in the network.

Number of UDP probe packets

The objective of the first calibration test is determine in which conditions the measured dropped packets at the receiver are due to the mobility protocols' handovers effect only. For this, a stationary situation will be considered, with increasing amounts of UDP/CBR traffic being generated, for both inter and intra-domain traffic types.

[22] In NS2, this is possible as the UDP header includes a sequence number field for this task.

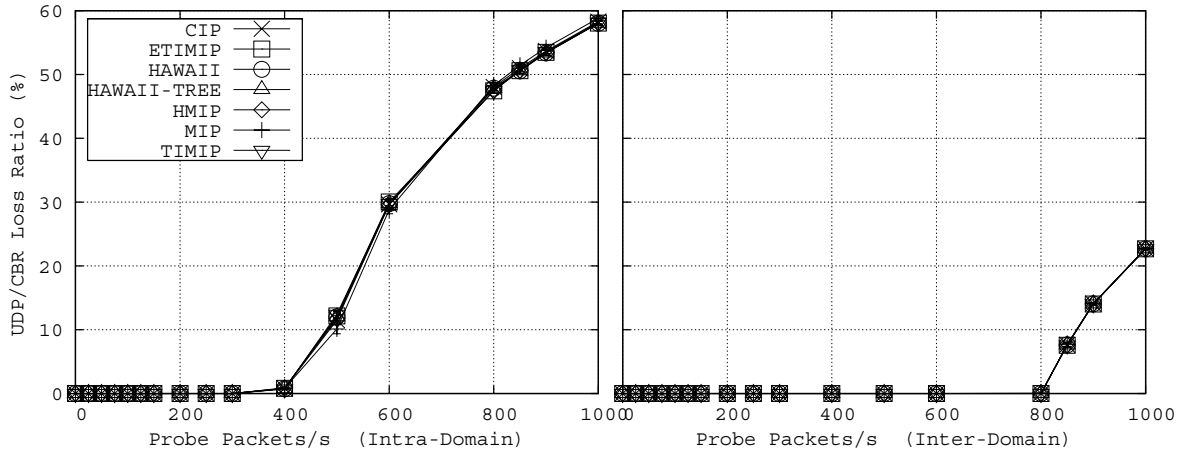


Figure 173: Number of UDP probes calibration - stationary - handover loss

Figure 173 shows that for a low amount of traffic no losses are experienced in all protocols, confirming that in such range the amount of dropped packets is directly related to the mobility protocols. However, at a certain stage, losses are measured, as the amount of traffic imposed is higher than the one supported by the 802.11 channel. The maximum amount of traffic is lower in the intra-domain traffic scenario, as the packets pass through the wireless channel twice.

Taking this test into consideration, the number of UDP probe packets is calibrated to be of 200 packets/s.

Number of UDP probe packets - Continuous handovers

The objective of the second calibration test is to determine the minimum amount of UDP/CBR traffic that must be generated in order to precisely determine the handover losses and handover latency metrics. This calibration is necessary, as a too low UDP rate may not be able to precisely measure the proposed metrics of handover latency and loss; however, an lower value also results in faster simulation times. For this test, a high-speed movement is introduced that covers the whole domain, once from AR1 to AR8 and back to AR1, at a speed of 30 handovers/minute.

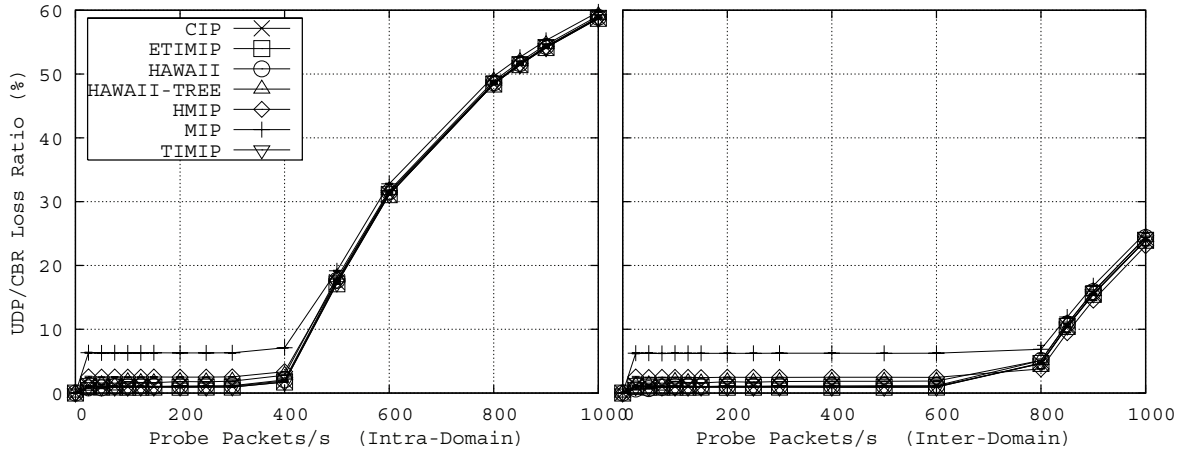


Figure 174: Number of UDP probes calibration - continuous handovers - handover loss

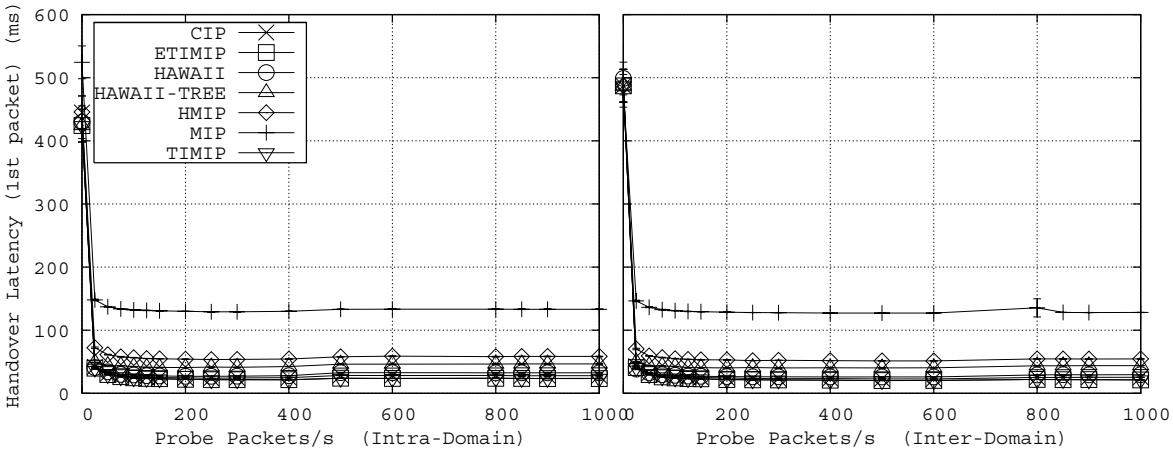


Figure 175: Number of UDP probes calibration - continuous handovers - handover latency

Figure 174 show the handover losses. While a too-low traffic value is unable to detect the handover losses properly, a too-large value incurs in the same phenomena as the previous stationary calibration test. Figure 175 shows that the handover latency also requires a minimum value of generated packets per second in order to properly measure the handover latency metric, by the same reasons for all protocols.

Taking this test into consideration, the number of UDP probe packets is confirmed to be of 200 packets/s.

Number of MN Handovers

The objective of the third calibration test is to determine the minimum number of handovers that must be performed by the MN to differentiate between the protocols. For this, the MN will perform increasing amount of complete back-and-forth movements in the domain, (AR1→AR8→AR1) with the UDP/CBR data traffic rate found previously.

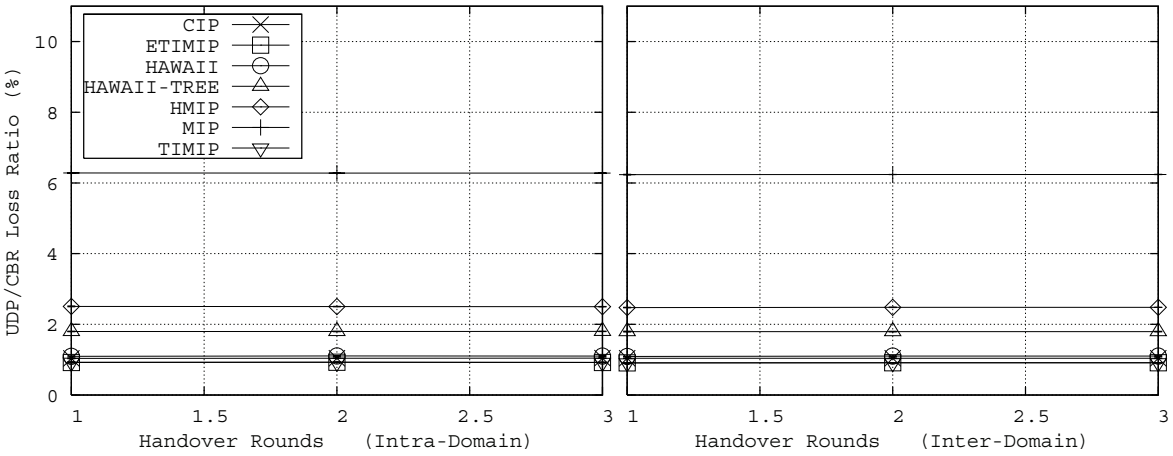


Figure 176: Number of handovers calibration - Handover Loss

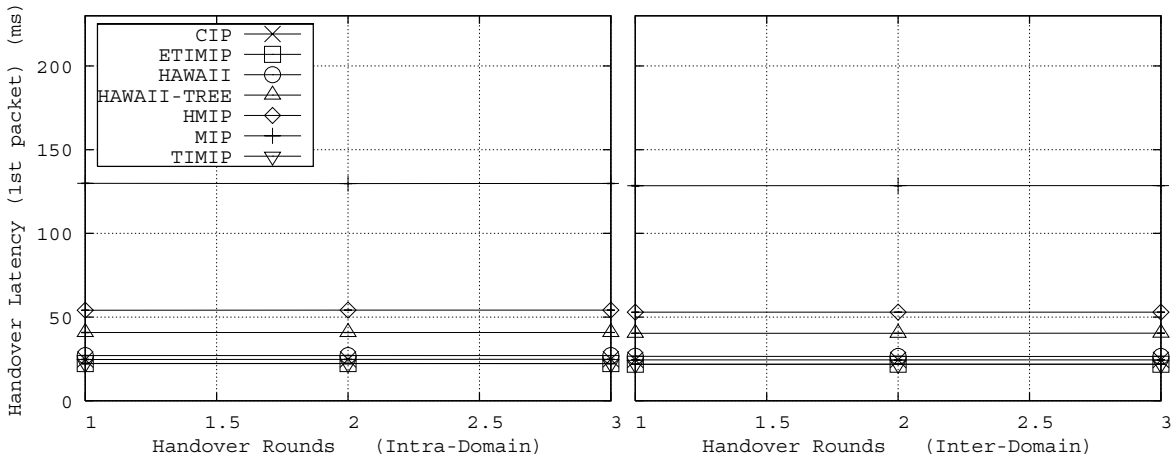


Figure 177: Number of handovers calibration - Handover Latency

Figure 176 shows the total handover losses, while Figure 177 shows the measured handover latency. The figures show that all metrics are unaffected by performing a larger number of handovers, being the base movement sufficient to characterize the handover.

Taking this test into consideration, the number of MN movements is confirmed to be one complete back-and-forth movement round from AR1→AR8→AR1.

Link delays calibration

The objective of the forth calibration test is to study the impact of the link delays in the measured handover metrics. For this, the reference scenario will be used, but the link delays will be tested for a variety of values.

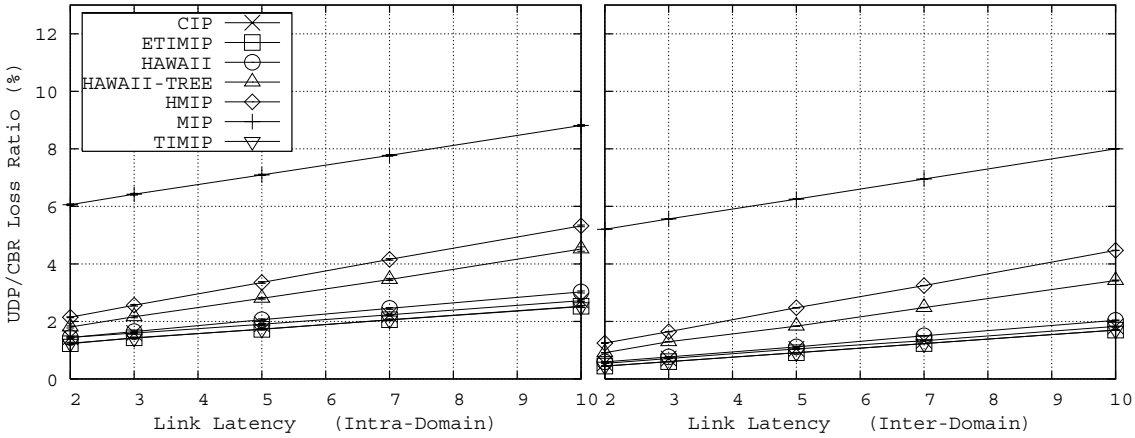


Figure 178: Link delay calibration - influence on handover loss

Figure 178 shows the corresponding loss ratio for the chosen link delays. As expected, higher link delays result in higher losses, but the relative differences among the protocols are maintained. As such, the intermediate value of 5ms is used for all internal links of the domain.

Link load calibration

The objective of the final calibration test is to measure the actual load that is injected in the network by using the method outlined previously using Exponential traffic sources with variable “rate” and “on_period” variables. For this, increasing amounts of load will be generated on the network, without mobility nor data test flows, and the average link usage will be measured.

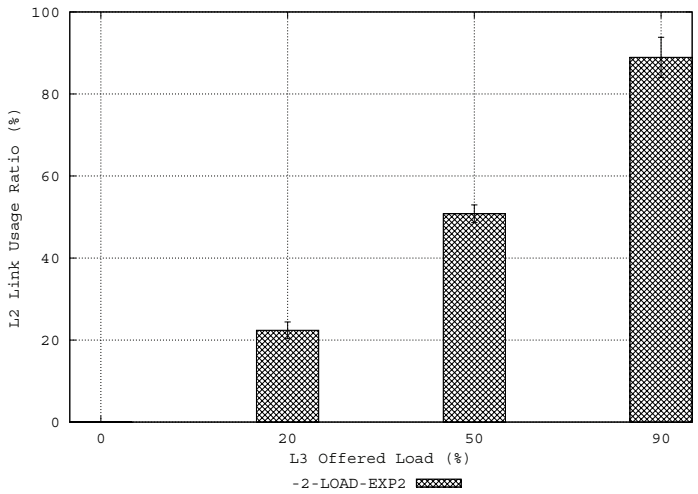


Figure 179: Link load calibration

Figure 179 shows the measured average link utilization for no load, and for 20%, 50% and 90% of L3 load. In all cases the resulting L2 link utilization was within the desired load range, showing that the chosen method is appropriate for generating abritary amounts of link load in NS2.

Appendix G Formal analysis of Soft de-triangulation algorithms

General description and Related work

The Soft de-Triangulation algorithms are a set of generic messages and procedures that improve the optimized routing's de-triangulation process without causing out-of-order packets **nor** increasing the packet delay of the in-flight packets. The proposed mechanism achieves its goals by delaying the packets, which will be sent via the direct path, for the exact minimum period of time that causes them to arrive at the destination immediately after the last in-flight packets sent via the triangulated path. Simpler versions of the mechanism that either remain optimal, but assume the presence of symmetric links, or that are non-optimal but continue to guarantee packet ordering, are also presented.

These mechanisms are necessary because previous the presented RO handover operation only considers the simplest form of removing the temporary local triangulation phenomena that occur in each handover. Using the RO dissemination mechanisms, when the first packets arrive at the old AR, the old AR will notify the crossover node of the new LMN's AR (Figure 92). When the RO update message reaches the crossover node, the following data packets are sent directly to the correct AR, removing the triangulation effect. Even though these operations do not typically incur in the drop of in-transit data packets, they can result in their reordering, as the direct path will typically have lower latency than the triangulated one; the exact latency difference will be the result of the actual link delays, the locations of the involved TRs and the amount of data packets queued in the routers. Depending on the type of receiver, these out-of-order packets due to such de-triangulation operations can have a major impact on it, as additional actions may be needed to recover from such phenomena [164] [165]. These out-of-order packets can either be considered as lost packets by UDP receivers which expect ordered arrival [166], or can trigger the retransmit mechanism and needlessly reduce the contention window of the majority of the TCP senders, as no packets have actually been dropped [167].

This problem was addressed by previous research work. The Celular IP semi-soft handover [20] solves it by delaying the packets sent through the direct path at a specific node, known as crossover node, for a constant amount of time, in order to allow an earlier reception of the triangulated packets. In spite of avoiding out-of-order packets, this mechanism forces a constant delay, which is independent of actual de-triangulation phenomena, and results in excessive majorated delays to ensure an ordered reception. Alternatively, seamless MIP [52] marks the packets and reorders them at the destination. This is an inefficient and of limited applicability process, as it requires packet marking in the IP header of all data packets

Problem illustration

To illustrate the problem, let us consider the triangulation situation depicted in Figure 180a. The figure comprises the crossover node, Node A, which is the node where the triangulated and the direct paths diverge; the old destination node, Node B, which is the one that was previously used to transfer information with the MN and will be used as a triangulated node; and the new destination node, Node C, which is the one that should be used to access the MN after it roams from Node B to Node C. For the sake of simplicity, the MN is omitted in the figure.

At first, when Node A receives data to the MN, it uses Node B the reach the MN, although it is now accessible through Node C; then, Node C will send an update message to Node A, so that data may be routed directly to it, removing the triangulation effect. Even though such operations do not typically incur in the drop of in-transit data packets, they can result in their reordering, as the direct path will typically have a lower latency than the triangulated one (otherwise, the routing would benefit from being triangulated in the first place, and no de-triangulation mechanism should be performed). The exact latency difference will be the result of the actual link delays and the amount of data packets queued in the routers, among other factors.

This problem is illustrated in b, where packets sent through the direct path (Data packet 2, A→C) can be received earlier than packets sent through the triangulated path (Data packet 1, A→B→C).

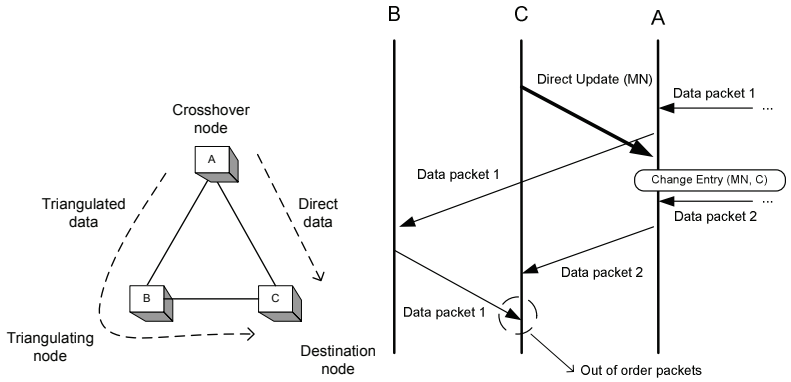


Figure 180: De-triangulation problem illustration. a) Overall problem b) Messages Exchanged

De-triangulation analytic framework

In this section, we introduce a new framework for analysing the de-triangulation problem.

Let us considered the situation represented in Figure 181a where the data packet transmission and reception instants are depicted and the time instants represent:

- $t_{0:}$ the time when the de-triangulation process will start at the destination node (Node C);
- $t_x > t_{0:}$ the time when the **last** data packet is sent by the crossover node (Node A), through the **triangulated** path;
- $t_{x'} > t_x$ the time when this packet is received by the destination node;
- $t_y > t_{x:}$ the time when the **first** data packet is sent by the crossover node (Node A) through the **direct** path;
- $t_{y'} > t_y$ the time when this packet is received by the destination node.

Considering also the mean delays that occur between the involved nodes for the transmission of packets, depicted in Figure 181b, which represent, separately, the downstream ($d_{(A,B)}$, $d_{(B,C)}$ and $d_{(A,C)}$) and the upstream ($d_{(B,A)}$, $d_{(C,B)}$ and $d_{(C,A)}$) paths. Each mentioned delay will contain the time needed to support all the operations involved to transmit a packet and to receive it, of which the propagation and queuing delays will be the main drivers.

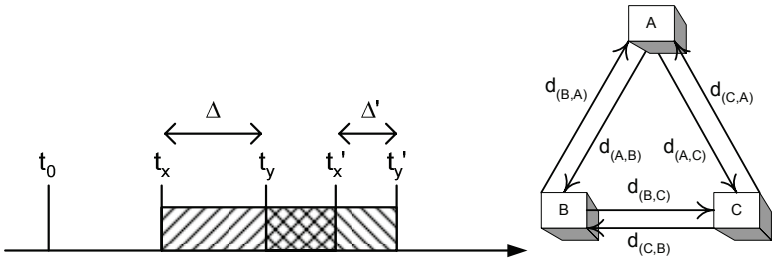


Figure 181: a) Time instants of data packet transmission and reception b) asymmetric link delays

Using this notation, the following relations occur:

$$t_{x'}=t_x+d_{(A,B)}+d_{(B,C)} \quad \text{(Eq. 5)}$$

$$t_{y'}=t_y+d_{(A,C)} \quad \text{(Eq. 6)}$$

Considering the time Δ since the transmission of the last triangulated packet until the transmission of the first direct packet at the crossover node; and the time Δ' since the arrival of the first direct packet until the arrival of the last triangulated packet a the destination node, which are given by:

$$\Delta = t_y - t_x \quad \text{(Eq. 7)}$$

$$\Delta' = t_{y'} - t_{x'} \quad \text{(Eq. 8)}$$

A positive value of Δ results in buffering being applied to the packets at the crossover node, as incoming packets need to wait before they are forwarded. The value is directly proportional to the maximum buffering capacity required. A value of zero results in no buffering at all. Finally, a negative value of Δ is not used in the context of de-triangulation.

A negative value of Δ' results in the occurrence of out-of-order packets, but no additional delay, as triangulated packets are received later than the direct ones. A positive value results in ordered delivery, but with an additional delay being experienced by the involved packets, which is equal to Δ'. Finally, a value of zero will result in an optimal de-triangulation mechanism – no out-of-order packets and no delay increase.

Thus, the objective of an ordered de-triangulation mechanism is defined as the operation where the packets will be delayed at the crossover node for a certain period of time Δ, so that Δ' will be positive; additionally, the objective of an optimal ordered de-triangulation mechanism is to ensure that Δ' will be zero.

Direct de-triangulation analysis

To illustrate the generic solution, let us consider the regular de-triangulation situation depicted in Figure 182.

Firstly, the destination Node C sends an Update Message directly to the crossover node (step 1). Then, the crossover node will update its own Routing Table, being able to start using the direct path (step 2). From now on, incoming packets will be sent directly to Node C, while already in-transit packets continue to use Node B (step 3). Every in-flight packet received by Node B is sent to Node C (triangulated path) (step 4), being received later than the direct paths, unless an ordered de-triangulation mechanism is used.

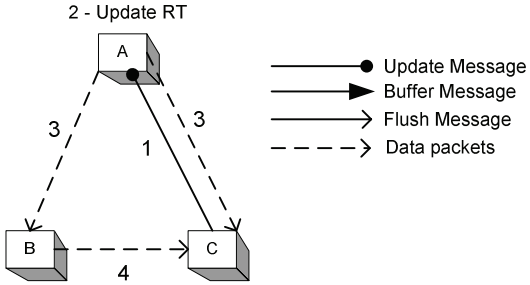


Figure 182: Direct de-triangulation algorithm

Using the same notation as before:

$t_x = t_0 + d_{(C,A)}$ (Eq. 9)

$t_y = t_x$ (Eq. 10)

$\Delta' = (t_y + d_{(A,C)}) - (t_x + d_{(A,B)} + d_{(B,C)})$ (Eq. 11)

$\Delta' = d_{(A,C)} - d_{(A,B)} - d_{(B,C)}$ (Eq. 12)

In order to avoid out-of-order packets, Δ' must be positive or equal to zero:

$\Delta' >= 0$ (Eq. 13)

$d_{(A,C)} >= d_{(A,B)} + d_{(B,C)}$ (Eq. 14)

As this last relation is not verified in the typical situations, since it would negate the major benefit of removing the de-triangulation, this results in the out-of-order packets phenomena illustrated in section 2.

Proposed generic solutions

Conservative Algorithm

The first presented algorithm, named **conservative algorithm** (Figure 183, part a), will ensure that no out-of-order packets will occur, by buffering the received packets at the cross-over node, while waiting for the reception of the last packet via the triangulated path. Only when this happens, the buffered packets will be released by the use of a new message called **Flush**.

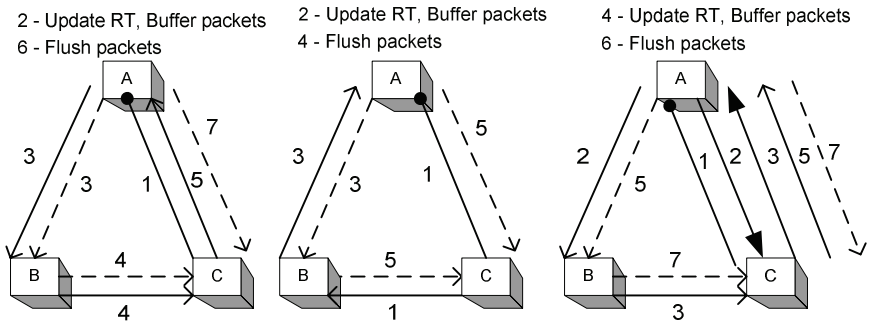


Figure 183: Proposed solutions: a) Conservative b) Symmetric c) Asymmetric

Firstly, the destination Node C sends an Update Message directly to the crossover node, as in the previous case (step 1), but now it starts buffering all incoming packets to a certain MN (step 2). Then, node A will send a Flush message that will pass through the triangulated and the destination node before returning to itself (step 3, 4 and 5). This forces the control packet to pass through the same paths as the in-transit data packets in the triangulated path (also

steps 3 and 4, dashed lines). When node A receives the Flush Message, it transmits the buffered packets via the direct path, and stops the buffering of additional packets (steps 6 and 7).

Using the same notation as above, the last packet forwarded using the triangulated path will guaranteedly be received at node C **earlier** than the first packet received via the direct path. However, this operation increases the previous handover latency for an additional amount of time, as shown by the following analysis:

$$t_x = t_0 + d_{(C,A)} \quad \text{(Eq. 15)}$$

$$t_y = t_x + d_{(A,B)} + d_{(B,C)} + d_{(C,A)} \quad \text{(Eq. 16)}$$

$$\Delta' = (t_y + d_{(A,C)}) - (t_x + d_{(A,B)} + d_{(B,C)}) \quad \text{(Eq. 17)}$$

$$\Delta' = (t_x + d_{(A,B)} + d_{(B,C)} + d_{(C,A)} + d_{(A,C)}) - (t_x + d_{(A,B)} + d_{(B,C)}) \quad \text{(Eq. 18)}$$

$$\Delta' = d_{(C,A)} + d_{(A,C)} \quad \text{(Eq. 19)}$$

Thus, out-of-order packets are avoided ($\Delta' > 0$), but the data packets are always unnecessarily delayed for ($d_{(C,A)} + d_{(A,C)}$) time units.

Optimal algorithm for Symmetric links

If the link delays can be assumed to be symmetric, then the previous algorithm can be refined to not incur in any extra delay, besides the imposed by triangulation itself. This has the advantages of solving the latency increase of the previous mechanism, and of reducing the buffering requirements of the **symmetric algorithm** (Figure 183, part b).

Again, the destination node sends an Update Message directly to the crossover node (step 1) to start the buffering of all data packets (step 2). However, at the same time, the destination node also sends the Flush Message in parallel to the crossover node, which now passes through the triangulated node in the opposite direction of the data packets (step 1 and 3). As before, when the crossover node receives this message, it stops buffering additional packets (step 4) and transmits the buffered packets via the direct path (step 5).

Using the same notation as before:

$$t_x = t_0 + d_{(C,A)} \quad \text{(Eq. 20)}$$

$$t_y = t_0 + d_{(C,B)} + d_{(B,A)} \quad \text{(Eq. 21)}$$

$$\Delta' = (t_y + d_{(A,C)}) - (t_x + d_{(A,B)} + d_{(B,C)}) \quad \text{(Eq. 22)}$$

$$\Delta' = (t_0 + d_{(C,B)} + d_{(B,A)} + d_{(A,C)}) - (t_0 + d_{(C,A)} + d_{(A,B)} + d_{(B,C)}) \quad \text{(Eq. 23)}$$

$$\Delta' = (d_{(C,B)} + d_{(B,A)} + d_{(A,C)}) - (d_{(B,C)} + d_{(A,B)} + d_{(C,A)}) \quad \text{(Eq. 24)}$$

As symmetric link delays are assumed, $d_{(C,B)}=d_{(B,C)}$; $d_{(B,A)}=d_{(A,B)}$; and $d_{(A,C)}=d_{(C,A)}$. Thus:

$$\Delta' = 0 \quad \text{(Eq. 25)}$$

Thus, the algorithm is optimal: out-of-order packets are avoided, and no extra delay is incurred in the data packets.

Optimal algorithm for Asymmetric links

If the link delays are asymmetric, e.g. due to different propagation delays or queuing, then the conservative algorithm can again result in out-of-order packets, in the situations where Δ' is negative. Such situation might happen when the control messages (Update and Flush messages) travel faster than the data packets and the delay imposed to data packets at the cross-

over node is smaller than needed. Although this could be solved by the addition of an extra small delay after the reception of the Flush Message, this would not result in an optimal solution. In contrast, the final **asymmetric algorithm** presented in this section is able to maintain optimality through slightly higher control and data loads (Figure 183, part c).

Firstly, the destination node sends an Update Message directly to the crossover node (step 1). When it receives the message, it sends a Flush Message as in the conservative algorithm via nodes B (step 2) and C (step 3 and 5), but also a new message to itself via node C (steps 2 and 3), called **Buffer Message**. These pair of messages are used to measure the actual delays experienced by data packets, taking account the asymmetric nature of the links. When the crossover node receives the Buffer Message (step 3), it updates the Routing Table and starts buffering data packets (step 4). The reception of the Flush Message is dealt with as before (step 6 and 7).

Using the same notation as above:

$$t_x = t_0 + d_{(C,A)} + d_{(A,C)} + d_{(C,A)} \quad \text{(Eq. 26)}$$

$$t_y = t_0 + d_{(C,A)} + d_{(A,B)} + d_{(B,C)} + d_{(C,A)} \quad \text{(Eq. 27)}$$

$$\Delta' = (t_y + d_{(A,C)}) - (t_x + d_{(A,B)} + d_{(B,C)}) \quad \text{(Eq. 28)}$$

$$\Delta' = (t_0 + d_{(C,A)} + d_{(A,B)} + d_{(B,C)} + d_{(C,A)} + d_{(A,C)}) - (t_0 + d_{(C,A)} + d_{(A,C)} + d_{(C,A)} + d_{(A,B)} + d_{(B,C)}) \quad \text{(Eq. 29)}$$

$$\Delta' = (d_{(A,B)} + d_{(B,C)} + d_{(C,A)} + d_{(A,C)}) - (d_{(A,B)} + d_{(B,C)} + d_{(C,A)} + d_{(A,C)}) \quad \text{(Eq. 30)}$$

Without the need to assume symmetric link delays, equation 30 simplifies to:

$$\Delta' = 0 \quad \text{(Eq. 31)}$$

Thus, the algorithm is always optimal regardless of the combination of asymmetric links: out-of-order packets are avoided, and no extra delay is incurred.