

Mobility Management in IP-Based Networks

**Analysis, Design, Programming and Computer-
Based Learning Modules**

**Dissertation
Zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)**

**vorgelegt der Fakultät für Informatik und Automatisierung
der Technischen Universität Ilmenau**

**von Dipl.-Ing. Ali Diab
geboren am 10.10.1976 in Damaskus/Syrien**

vorgelegt am

Gutachter:

- 1. Prof. Dr.-Ing. habil. Andreas Mitschele-Thiel**
- 2. Prof. Dr. rer. nat. habil. Jochen Seitz**
- 3. Prof. Dr.-Ing. Jochen Schiller**

urn:nbn:de:gbv:ilm1-2010000098

**Mobility Management in IP-Based
Networks**
**Analysis, Design, Programming and Computer-
Based Learning Modules**

Copyright © 2010

by

Ali Diab

Acknowledgement

First, I would like to thank my advisor Prof. Mitschele-Thiel for his support and consistent academic guidance, which significantly helped me in accomplishing this thesis. I would like also to thank Prof. Seitz and Prof. Schiller for being co-supervisor of the thesis.

Second, I would like to thank all members of the Integrated Communication Systems group for their involvements in discussions and verifications of the thesis.

Finally, I would like to thank my family, especially my father and mother, for their encouragement and support.

Abstract

Mobile communication networks experience a tremendous development clearly evident from the wide variety of new applications way beyond classical phone services. The tremendous success of the Internet along with the demand for always-on connectivity has triggered the development of All-IP mobile communication networks. Deploying these networks requires, however, overcoming many challenges. One of the main challenges is how to manage the mobility between cells connecting through an IP core in a way that satisfies real-time requirements. This challenge is the focus of this dissertation.

This dissertation delivers an in-depth analysis of the mobility management issue in IP-based mobile communication networks. The advantages and disadvantages of various concepts for mobility management in different layers of the TCP/IP protocol stack are investigated. In addition, a classification and brief description of well-known mobility approaches for each layer are provided. The analysis concludes that network layer mobility management solutions seem to be best suited to satisfy the requirements of future All-IP networks. The dissertation, therefore, provides a comprehensive review of network layer mobility management protocols along with a discussion of their pros and cons. Analyses of previous work in this area show that the proposed techniques attempt to improve the performance by making constraints either on access networks (e.g. requiring a hierarchical topology, introducing of intermediate nodes, etc.) or mobile terminals (e.g. undertaking many measurements, location tracking, etc.). Therefore, a new technique is required that completes handoffs quickly without affecting the end-to-end performance of ongoing applications. In addition, it should place restrictions neither on access networks nor on mobiles. To meet these requirements, a new solution named Mobile IP Fast Authentication protocol (MIFA) is proposed. MIFA provides seamless mobility and advances the state of the art. It utilizes the fact that mobiles movements are limited to a small set of neighboring subnets. Thus, contacting these neighbors and providing them in advance with sufficient data related to the mobiles enable them to fast re-authenticate the mobiles after the handoff. The dissertation specifies the proposal for both IPv4 and IPv6. The specification of MIFA considers including many error recovery mechanisms to cover the most likely failures. Security considerations are studied carefully as well. MIFA does not make any restrictions on the network topology. It makes use of layer 2 information to optimize the performance and works well even if such information is not available.

In order to analyze our new proposal in comparison to a wide range of well-known mobility management protocols, this dissertation proposes a generic mathematical model that supports the evaluation of figures such as average handoff latency, average number of dropped packets, location update cost and packet delivery cost. The generic model considers dropped control messages and takes different network topologies and mobility scenarios into account. This dissertation also validates the generic mathematical model by comparing its results to simulation results as well as results of real testbeds under the same assumptions. The validation proves that the generic model delivers an accurate evaluation of the performance in low-loaded networks. The accuracy of the model remains acceptable even under high loads. The validation also shows that simulation results lie in a range of $\pm 23\%$, while results of real testbeds lie in a range of $\pm 30\%$ of the generic model's results. To simplify the analysis using the generic mathematical model, 4 new tools are developed in the scope of this work. They automate the parameterization of mobility protocols, network topologies and mobility scenarios. This dissertation also evaluates the new proposal in comparison to well-known approaches (e.g. Mobile IP, Handoff-Aware Wireless Access Internet Infrastructure (HAWAII), etc.) by means of the generic mathematical model as well as simulation studies modeled in the Network Simulator 2. The evaluation shows that MIFA is a very fast protocol. It outperforms all studied protocols with respect to the handoff latency and number of dropped packets per handoff. MIFA is suitable for low as well as high speeds. Moreover, there is no significant impact of the network topology on its performance. A main advantage of MIFA is its robustness against the dropping of control messages. It remains able to achieve seamless handoffs even if a dropping occurs. The performance improvement is achieved, however, at the cost of introducing new control messages mainly to distribute data concerning mobile terminals to neighbor subnets. This results in more location update cost than that resulting from the other mobility management protocols studied. Due to excluding any constraints on the network topology, MIFA generates the same packet delivery cost as Mobile IP and less than other protocols.

An additional focus of this dissertation is the development of an adaptive eLearning environment that personalizes eLearning contents conveying the topics of this dissertation depending on users' characteristics. The goal is to allow researchers to quickly become involved in research on mobility management, while learners such as students are able to gain information on the topics without excess detail. Analyses of existing eLearning

environments show a lack of adaptivity support. Existing environments focus mainly on adapting either the navigation or the presentation of contents depending on one or more selected users' characteristics. There is no environment that supports both simultaneously. In addition, many user characteristics are disregarded during the adaptivity process. Thus, there is a need to develop a new adaptive eLearning environment able to eliminate these drawbacks. This dissertation, therefore, designs a new Metadata-driven Adaptive eLearning Environment (MAeLE). MAeLE generates personalized eLearning courses along with building an adequate navigation at run-time. Adaptivity depends mainly on providing contents with their describing metadata, which are stored in a separate database, thus enabling reusing of eLearning contents. The relation between the metadata that describe contents and those describing learners are defined accurately, which enables a dynamic building of personalized courses at run-time. A prototype for MAeLE is provided in this dissertation as well.

Zusammenfassung

Mobilkommunikationsnetze erleben eine enorme Entwicklung, die durch die über Telephonie hinaus zahlreichen neuen Applikationen und Dienste zu sehen ist. Der große Erfolg des Internets sowie die Anforderung von „always-on“ Konnektivität hat die Entwicklung von All-IP Mobilkommunikationsnetzen angetrieben. Eine der wesentlichen Herausforderungen in diesen Netzen ist die Entwicklung von Mobilitätsmanagementansätzen, die die Echtzeitanforderungen erfüllen können. Dieser Herausforderung stellt sich die Dissertation.

Die Dissertation liefert eine detaillierte Analyse der Mobilitätsmanagementthematik in IP-basierten Mobilkommunikationsnetzen. Die Vor- und Nachteile verschiedener Konzepte zur Mobilitätsmanagementimplementierung in verschiedenen Schichten des TCP/IP Protokollstapels werden untersucht. Die Dissertation beschreibt die bekannten Ansätze jeder Schicht des TCP/IP Protokollstapels und stellt eine Klassifikation dieser Ansätze vor. Die Analyse zeigt, dass die Mobilitätsmanagementansätze der Vermittlungsschicht am besten dafür geeignet, die Anforderungen der zukünftigen All-IP Netze zu erfüllen. Die Dissertation liefert deshalb einen umfassenden Überblick von diesen Ansätzen zusammen mit einer Diskussion der Vor- und Nachteile jedes Ansatzes. Die Ergebnisse der Analyse zeigen, dass die in der Literatur vorgeschlagenen Ansätze die Performance zu verbessern versuchen, in dem sie entweder die Netzwerktopologie (z.B. Verlangen von einer hierarchischen Topologie, Einführung von neuen Knoten ins Netz, usw.) oder die Mobilgeräte (z.B. Verlangen von zahlreichen Messungen, usw.) mit hohen Anforderungen belasten. Deshalb ist eine neue Technik erforderlich, die das Handoff schnell durchführt ohne die Performance von laufenden Applikationen zu beeinflussen. Die neue Technik darf weder die Netzwerktopologie beschränken, noch die Mobilgeräte mit hohen Anforderungen belasten. Um die gestellten Anforderungen zu erfüllen, schlägt die Dissertation einen Lösungsansatz namens „Mobile IP Fast Authentication Protocol (MIFA)“ vor. MIFA garantiert nahtlose Handoffs und verbessert den Stand der Technik. Die Basisidee sagt, dass die Bewegung von Mobilgeräten in der Realität auf Nachbarsubnetze beschränkt ist. Darauf basierend ermöglicht eine im Voraus realisierte Bereitstellung von ausreichenden, auf Mobilgeräte bezogenen Daten in diesen Nachbarsubnetzen eine schnelle Authentifizierung der Mobilgeräte nach dem Handoff. Die Dissertation spezifiziert den neuen Ansatz sowohl für IPv4 als auch IPv6. Fehlerbehebungsmechanismen sind in den Spezifikationen des Ansatzes sorgfältig definiert. MIFA macht Einschränkungen weder auf die Netzwerktopologie noch auf die Mobilgeräte.

Um den im Rahmen dieser Arbeit vorgeschlagenen Mobilitätsmanagementansatz im Vergleich zu einem breiten Spektrum von Mobilitätsmanagementansätzen evaluieren zu können, schlägt die Dissertation ein generisches mathematisches Modell vor, das die Auswertung von Metriken wie durchschnittliche Handoff-Latenz, Durchschnittszahl von verlorenen Paketen, Standortaktualisierungskosten usw. ermöglicht. Darüber hinaus nimmt das generische mathematische Modell die verlorenen Kontrollnachrichten in Betracht und ermöglicht es, beliebige Netzwerktopologien und Mobilitätsszenarien einzusetzen. Um das generische mathematische Modell zu validieren, vergleicht die Dissertation seine Ergebnisse mit Simulationsergebnissen sowie mit Ergebnissen realer Testumgebungen unter denselben Bedingungen. Die Validierung beweist, dass das generische mathematische Modell eine akkurate Auswertung der Performance in unbelasteten Netzen liefert. Die Genauigkeit des Modells bleibt sogar unter hohen Lasten akzeptabel. Simulationsergebnisse liegen in einem Bereich von $\pm 23\%$ von den Ergebnissen des generischen Modells. Die Ergebnisse realer Testumgebungen liegen hingegen in einem Bereich von $\pm 30\%$ gegenüber den Ergebnissen des generischen Modells. Um die Analyse mit Hilfe des generischen mathematischen Modells zu vereinfachen, sind 4 neue Werkzeuge im Rahmen dieser Arbeit entwickelt worden. Sie automatisieren die Parametrisierung von Mobilitätsprotokollen, Netzwerktopologien und Mobilitätsszenarien.

Eine Auswertung des neuen Mobilitätsmanagementansatzes im Vergleich zu bekannten Ansätzen, z.B. Mobile IP, Handoff-Aware Wireless Access Internet Infrastructure (HAWAII), usw. wurde mittels des generischen mathematischen Modells und anhand von Simulationsstudien mit dem Netzwerksimulator 2 durchgeführt. Die Ergebnisse der Auswertung beweisen, dass MIFA ein sehr schnelles Mobilitätsmanagementprotokoll darstellt. Es unterbietet alle analysierten Protokolle im Bezug auf die Handoff-Latenz und Anzahl der verlorenen Pakete pro Handoff. MIFA ist sowohl für langsame als auch für hohe Geschwindigkeiten geeignet. Es gibt keine bedeutsame Auswirkung der Netzwerktopologie auf seine Leistung. Ein Hauptvorteil von MIFA ist seine Robustheit gegen verlorene Steuernachrichten. Es bleibt in der Lage, schnelle Handoffs durchzuführen, selbst wenn Steuernachrichten verloren gehen. Die Leistungsverbesserung ist jedoch auf Kosten der Einführung von extra Steuernachrichten zurückzuführen. Dies führt dazu, dass mehr Standortaktualisierungskosten als bei den

anderen analysierten Mobilitätsmanagementprotokollen produziert werden. Im Bezug auf die Kosten des Sendens von Datenpaketen generiert MIFA genau soviel Kosten wie Mobile IP und weniger als andere Ansätze.

Ein zusätzliches Ziel dieser Dissertation ist die Entwicklung einer adaptiven eLearning Umgebung, die eine Personalisierung der eLearning Inhalte ermöglicht. Die Inhalte dieser Umgebung umfassen die im Rahmen dieser Arbeit behandelten Themen. Unser Ziel ist auf einer Seite die Forscher schnell an Forschung ran zu lassen. Auf der anderen Seite sollen Lernende wie Studenten nicht mit viel detailliertem Inhalt belastet werden. Die Analyse vorhandener eLearning Umgebungen zeigt, dass es einen Mangel an der Adaptivität gibt. Vorhandene Umgebungen konzentrieren sich hauptsächlich darauf, entweder die Navigation oder die Darstellung des Inhalts nach einem oder mehreren ausgewählten Nutzermerkmalen zu adaptieren. Sie ermöglichen es nicht, die Navigation und die Darstellung des Inhaltes gleichzeitig zu adaptieren. Viele Merkmale des Nutzers werden in dem Adaptivitätsprozess vernachlässigt. Deshalb ist es erforderlich, eine neue adaptive eLearning Umgebung zu entwickeln, die die Schwächen vorheriger Umgebungen vermeidet und die im Rahmen dieser Arbeit behandelten Themen passend personalisiert. Die Dissertation stellt deshalb eine neue eLearning Umgebung Namens „Metadata-driven Adaptive eLearning Environment (MAeLE)“ dar. MAeLE generiert personalisierte eLearning Kurse zusammen mit adaptierter Navigation zur Laufzeit. Die Anpassungsfähigkeit basiert darauf, die Inhalte mit Metadaten zu versehen, die in einer eigenen Datenbank gespeichert werden. Die Metadaten beschreiben die Inhalte und indizieren, zu welchen Nutzertypen sie geeignet sind. Die neue Umgebung ist prototypisch implementiert.

Contents

<u>1.</u>	<u>INTRODUCTION.....</u>	<u>1</u>
1.1.	MOBILE COMMUNICATION NETWORKS	1
1.2.	PROBLEM STATEMENTS	6
1.3.	OBJECTIVES AND CONTRIBUTIONS	7
1.3.1.	OBJECTIVES.....	7
1.3.2.	CONTRIBUTIONS.....	7
1.4.	OUTLINE.....	8
<u>2.</u>	<u>MOBILITY IN MOBILE COMMUNICATION NETWORKS.....</u>	<u>10</u>
2.1.	OVERVIEW	10
2.2.	IP-BASED NETWORKS	11
2.3.	MOBILITY MANAGEMENT REQUIREMENTS.....	13
2.4.	MOBILITY MANAGEMENT APPROACHES.....	14
2.4.1.	LINK LAYER MOBILITY.....	14
2.4.2.	NETWORK LAYER MOBILITY.....	16
2.4.3.	TRANSPORT LAYER MOBILITY.....	18
2.4.4.	SESSION LAYER MOBILITY.....	21
2.4.5.	APPLICATION LAYER MOBILITY.....	21
2.4.6.	HYBRID APPROACHES.....	23
2.5.	CONCLUSION.....	24
<u>3.</u>	<u>NETWORK LAYER MOBILITY MANAGEMENT</u>	<u>26</u>
3.1.	TERMINAL-BASED MOBILITY MANAGEMENT.....	26
3.1.1.	TERMINAL-BASED MACRO MOBILITY MANAGEMENT APPROACHES.....	26
3.1.2.	TERMINAL-BASED MICRO MOBILITY MANAGEMENT APPROACHES.....	39
3.2.	NETWORK-BASED MOBILITY MANAGEMENT	59
3.2.1.	NETWORK-BASED MACRO MOBILITY MANAGEMENT APPROACHES.....	59
3.2.2.	NETWORK-BASED MICRO MOBILITY MANAGEMENT APPROACHES.....	61
3.3.	CONCLUSION.....	64
<u>4.</u>	<u>MOBILE IP FAST AUTHENTICATION PROTOCOL (MIFA).....</u>	<u>71</u>
4.1.	BASIC IDEA.....	71
4.2.	MOBILE IP FAST AUTHENTICATION PROTOCOL FOR IPv4 (MIFAv4).....	72
4.2.1.	OPERATION OVERVIEW.....	72
4.2.2.	INITIAL REGISTRATION PROCEDURE.....	73
4.2.3.	INITIAL AUTHENTICATION EXCHANGE PROCEDURE.....	75
4.2.4.	INFORMATION DISTRIBUTION PROCEDURE.....	76
4.2.5.	OPERATION IN REACTIVE MODE.....	77
4.2.6.	ERROR RECOVERY MECHANISMS IN REACTIVE MODE.....	79
4.2.7.	OPERATION IN PREDICTIVE MODE.....	82
4.2.8.	ERROR RECOVERY MECHANISMS IN PREDICTIVE MODE.....	84

4.2.9.	<i>SECURITY CONSIDERATIONS</i>	86
4.2.10.	<i>FORMAL SPECIFICATION WITH SDL</i>	89
4.3.	MOBILE IP FAST AUTHENTICATION PROTOCOL FOR IPV6 (MIFAV6)	97
4.3.1.	<i>OPERATION OVERVIEW</i>	97
4.3.2.	<i>INITIAL REGISTRATION PROCEDURE</i>	98
4.3.3.	<i>INITIAL AUTHENTICATION EXCHANGE PROCEDURE</i>	99
4.3.4.	<i>INFORMATION DISTRIBUTION PROCEDURE</i>	100
4.3.5.	<i>OPERATION IN REACTIVE MODE</i>	101
4.3.6.	<i>ADDRESS AUTO-CONFIGURATION AND DUPLICATED ADDRESS DETECTION</i>	103
4.3.7.	<i>ERROR RECOVERY MECHANISMS IN REACTIVE MODE</i>	104
4.3.8.	<i>OPERATION IN PREDICTIVE MODE</i>	105
4.3.9.	<i>ERROR RECOVERY MECHANISMS IN PREDICTIVE MODE</i>	108
4.3.10.	<i>SECURITY CONSIDERATIONS</i>	110
4.4.	CONCLUSION	111
5.	<u>ANALYSIS OF MOBILITY MANAGEMENT PROTOCOLS</u>	112
5.1.	BASIC ASSUMPTIONS	112
5.2.	MODELING OF NETWORK TOPOLOGIES	114
5.3.	MODELING OF MOVEMENTS PATTERNS	115
5.4.	PERFORMANCE ANALYSIS	117
5.4.1.	<i>BREAK-BEFORE-MAKE MOBILITY MANAGEMENT PROTOCOLS</i>	118
5.4.2.	<i>MAKE-BEFORE-BREAK MOBILITY MANAGEMENT PROTOCOLS</i>	121
5.5.	COST ESTIMATION	126
5.5.1.	<i>LOCATION UPDATE COST</i>	126
5.5.2.	<i>PACKET DELIVERY COST</i>	127
5.5.3.	<i>TOTAL COST</i>	128
5.6.	APPLICATION OF THE GENERIC MATHEMATICAL MODEL TO MOBILITY MANAGEMENT PROTOCOLS	128
5.6.1.	<i>APPLIED NETWORK TOPOLOGY</i>	129
5.6.2.	<i>APPLIED MOVEMENT MODEL</i>	130
5.6.3.	<i>APPLICATION TO BREAK-BEFORE-MAKE MOBILITY MANAGEMENT PROTOCOLS</i>	131
5.6.4.	<i>APPLICATION TO MAKE-BEFORE-BREAK MOBILITY MANAGEMENT PROTOCOLS</i>	138
5.6.5.	<i>PERFORMANCE EVALUATION</i>	143
5.6.6.	<i>COST ESTIMATION</i>	150
5.7.	IMPACT OF MOBILITY SCENARIOS	152
5.7.1.	<i>APPLICATION OF THE GENERIC MATHEMATICAL MODEL TO MOBILITY MANAGEMENT PROTOCOLS</i>	153
5.7.2.	<i>PERFORMANCE EVALUATION</i>	153
5.7.3.	<i>COST ESTIMATION</i>	155
5.8.	PERFORMANCE VS. COST	155
5.9.	VALIDATION OF THE GENERIC MATHEMATICAL MODEL	158
5.9.1.	<i>GENERIC MATHEMATICAL MODEL VS. SIMULATION</i>	159
5.9.2.	<i>GENERIC MATHEMATICAL MODEL VS. REAL TESTBEDS</i>	163
5.9.3.	<i>SUMMARY</i>	168
5.10.	GRAPHICAL TOOLS SUPPORTING THE GENERIC MATHEMATICAL MODEL	168
5.10.1.	<i>MOBILITY SCENARIOS GENERATOR (MSGEN)</i>	168
5.10.2.	<i>NETWORK GENERATOR (NETGEN)</i>	169
5.10.3.	<i>PROTOCOL DESIGNER (PROTDES)</i>	170
5.10.4.	<i>COMPARATIVE ANALYSIS OF MOBILITY MANAGEMENT PROTOCOLS (CAMP)</i>	172
5.11.	CONCLUSION	172
6.	<u>SIMULATIVE PERFORMANCE EVALUATION</u>	175

6.1.	NETWORK SIMULATOR 2 (NS2)	175
6.2.	SIMULATION SCENARIOS	176
6.3.	PERFORMANCE EVALUATION UNDER DYNAMICALLY CHANGING NETWORK CONDITIONS	177
6.3.1.	<i>HANDOFF LATENCY</i>	177
6.3.2.	<i>NUMBER OF DROPPED PACKETS PER HANDOFF</i>	179
6.3.3.	<i>AVERAGE CONGESTION WINDOW SIZE</i>	182
6.3.4.	<i>SUMMARY</i>	182
6.4.	IMPACT OF NETWORK TOPOLOGY	182
6.4.1.	<i>IMPACT OF NETWORK TOPOLOGY ON THE PERFORMANCE OF MIFA</i>	183
6.4.2.	<i>IMPACT OF NETWORK TOPOLOGY ON THE PERFORMANCE OF MIP</i>	184
6.4.3.	<i>IMPACT OF NETWORK TOPOLOGY ON THE PERFORMANCE OF HAWAII</i>	185
6.4.4.	<i>SUMMARY</i>	186
6.5.	IMPACT OF NETWORK LOAD	187
6.5.1.	<i>IMPACT OF NETWORK LOAD ON THE PERFORMANCE OF MIFA</i>	187
6.5.2.	<i>IMPACT OF NETWORK LOAD ON THE PERFORMANCE OF MIP</i>	190
6.5.3.	<i>IMPACT OF NETWORK LOAD ON THE PERFORMANCE OF HAWAII</i>	192
6.5.4.	<i>COMPARATIVE ANALYSIS</i>	195
6.5.5.	<i>SUMMARY</i>	200
6.6.	IMPACT OF MN SPEED	201
6.6.1.	<i>IMPACT OF MN SPEED ON THE PERFORMANCE OF MIFA</i>	201
6.6.2.	<i>IMPACT OF MN SPEED ON THE PERFORMANCE OF MIP</i>	207
6.6.3.	<i>IMPACT OF MN SPEED ON THE PERFORMANCE OF HAWAII</i>	209
6.6.4.	<i>COMPARATIVE ANALYSIS</i>	211
6.6.5.	<i>SUMMARY</i>	214
6.7.	CONCLUSION	214
7.	<u>ADAPTIVE ELEARNING: NEW OPPORTUNITIES FOR LEARNING</u>	217
7.1.	INTRODUCTION	217
7.2.	eLEARNING PLATFORMS	218
7.2.1.	<i>CONTENT MANAGEMENT SYSTEM (CMS)</i>	218
7.2.2.	<i>LEARNING MANAGEMENT SYSTEM (LMS)</i>	219
7.2.3.	<i>LEARNING CONTENT MANAGEMENT SYSTEM (LCMS)</i>	219
7.3.	eLEARNING STANDARDS	221
7.3.1.	<i>COOPERATION NETWORK OF THE STANDARDIZATION CONSORTIUM</i>	221
7.3.2.	<i>SHAREABLE CONTENT OBJECT REFERENCE MODEL (SCORM)</i>	222
7.3.3.	<i>LEARNING OBJECT METADATA (LOM)</i>	223
7.4.	ADAPTIVITY IN eLEARNING STANDARDS AND SYSTEMS	223
7.4.1.	<i>ADAPTIVITY AND ADAPTIVE HYPERMEDIA SYSTEMS (AHSS)</i>	224
7.4.2.	<i>ADAPTIVITY SUPPORT IN EXISTING eLEARNING STANDARDS</i>	224
7.4.3.	<i>ADAPTIVE eLEARNING SYSTEMS</i>	225
7.4.4.	<i>SUMMARY</i>	228
7.5.	METADATA-DRIVEN ADAPTIVE eLEARNING ENVIRONMENT (MAELE)	229
7.5.1.	<i>REQUIREMENTS OF MAELE</i>	229
7.5.2.	<i>ARCHITECTURE OF MAELE</i>	230
7.5.3.	<i>ADAPTIVITY PROCESS IN MAELE</i>	231
7.5.4.	<i>MODELING OF THE USER</i>	232
7.5.5.	<i>LEARNING CONTENTS METADATA AND THEIR RELATION TO USER CHARACTERISTICS</i>	234
7.5.6.	<i>LEARNING CONTENTS STRUCTURE AND PERSONALIZATION</i>	236
7.5.7.	<i>CASE STUDIES</i>	240
7.5.8.	<i>PROTOTYPE IMPLEMENTATION</i>	241
7.6.	CONCLUSION	244
8.	<u>CONCLUSIONS AND OUTLOOK</u>	246

8.1.	CONCLUSIONS	246
8.2.	OUTLOOK	250
<u>A.</u>	<u>MIFA CONTROL MESSAGES</u>	<u>252</u>
<u>B.</u>	<u>ESTABLISHMENT OF L3-FHRS</u>	<u>257</u>
<u>C.</u>	<u>SDL SPECIFICATION OF MIFAV4</u>	<u>261</u>
<u>D.</u>	<u>PARAMETERS USED IN THE GENERIC MATHEMATICAL MODEL</u>	<u>289</u>
<u>E.</u>	<u>MODELING OF ASYMMETRICAL NETWORK TOPOLOGIES</u>	<u>316</u>
<u>F.</u>	<u>GRAPHICAL TOOLS SUPPORTING THE GENERIC MATHEMATICAL MODEL</u>	<u>320</u>
<u>G.</u>	<u>IMPACT OF NETWORK TOPOLOGY</u>	<u>336</u>
	<u>BIBLIOGRAPHY</u>	<u>339</u>

Abbreviations

ADL	Advanced Distributed Learning
AFA	Anchor Foreign Agent
Agnt_Adv	Agent Advertisement
Agnt_Sol	Agent Solicitation
AH	Authentication Header
AHA	Adaptive Hypermedia Architecture
AHS	Adaptive Hypermedia System
AICC	Aviation Industry Computer-Based Training Committee
ALE	Adaptive Learning Environment
AN	Access Network
ANG	Access Network Gateway
ANP	Anchor Point
ANSI	American National Standards Institute
AP	Access Point
APeLS	Adaptive Personalized eLearning Service
AR	Access Router
ARIADNE	Alliance of Remote Instructional Authoring and Distribution Networks for Europe
AS	Autonomous System
AuC	Authentication Center
BA	Binding Acknowledgement
BAR	Brain Access Router
BARWAN	Bay Area Research Wireless Access Network
BCMP	BRAIN Candidate Mobility Protocol
BMG	Brain Mobility Gateway
BS	Base Station
BSC	Base Station Controller
BSS	Base Station Subsystem
BSS	Basic Service Set
BTS	Base Transceiver Station
BU	Binding Update
BUnode	Binding Update node
BW	Binding Warning
CAM	Content Aggregation Model
CAMP	Comparative Analysis of Mobility Management Protocols
CBT	Core Based Trees
CBT	Computer-Based Training

CDMA	Code Division Multiple Access
CIP	Cellular IP
CLS	Carrying Load Status
CMI	Computer Managed Instruction
CMS	Content Management System
CN	Corresponding Node
CoA	Care of Address
ContNode	Control Node
CP	Content Packaging
CSD	Circuit Switched Domain
CTS	Current Tracking Status
DA	Domain Address
DAD	Duplicated Address Detection
DE	Decision Engine
DeReg PBU	De-Registration PBU
DHCP	Dynamic Host Configuration Protocol
DHMIP	Dynamic Hierarchical MIP
DNS	Domain Name System
DS	Distribution System
ECS	Eager Cell Switching
EGP	Exterior Gateway Protocol
EIR	Equipment Identity Register
eNB	evolved NodeB
EPC	Evolved Packet Core
ESP	Encapsulating Security Payload
ESS	Extended Service Set
eTIMIP	enhanced Terminal Independent Mobile IP
E-UTRAN	Evolved UTRAN
FA	Foreign Agent
F-Back	Fast Binding Acknowledgement
F-BU	Fast Binding Update
FEC	Forwarding Equivalence Class
FHCS	Fast Hinted Cell Switching
FMIPv6	Fast Mobile IP version 6
F-NAadv	Fast Neighbor Advertisement
GCoA	Global Care of Address
GFA	Gateway Foreign Agent
GGSN	Gateway GSN
GMSC	Gateway MSC
GPRS	General Packet Radio Service
GRE	Generic Routing Encapsulation
GSM	Global Standard for Mobile communication

GUI	Graphical User Interface
GW	Gateway
HA	Home Agent
HA_Ack	Home Agent Acknowledgement
HA_Not	Home Agent Notification
HAck	Handoff Acknowledgement
HAWAII	Handoff-Aware Wireless Access Internet Infrastructure
HCS	Hinted Cell Switching
HD	Handoff Decision
HI	Handoff Initiate
HLR	Home Location Register
HMIPv6	Hierarchical Mobile IPv6
HMSIP	Hierarchical Mobile Session Initiation Protocol
HN	Handoff Notification
Hn_Ack	Handoff Acknowledgement
Hn_Not	Handoff Notification
HNP	Home Network Prefix
HORqst	HandOff Request
HRply	Handoff Reply
HRqst	Handoff Request
IDMP	Intra Domain Mobility management Protocol
IEEE LTSC	Institute for Electrical and Electronic Engineers Learning Technology Standards
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IMEI	International Mobile Equipment Identity
IMS	Instructional Management System
InNode	Intermediate Node
Int_Ack	Initial Acknowledgement
IP	Internet Protocol
IPSec-SA	IPsec Security Association
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
I-TCP	Indirect TCP
KDC	Key Distribution Center
L2-LD trigger	Layer 2 Link Down trigger
L2-LU trigger	Layer 2 Link Up trigger
L2-trigger	Layer 2 trigger
L3	Life-Long Learning
L3-FHR	Layer 3 Frequent Handoff Region
LA	Local Address

LCMS	Learning Content Management System
LCoA	Local Care of Address
LCS	Lazy Cell Switching
LERS	Localized Enhanced Routing Schemes
LIP	Learner Information Package
LM	Location Manager
LMA	Localized Mobility Anchor
LMS	Learning Management System
LO	Learning Object
LOM	Learning Object Metadata
LSP	Label Switched Path
LTE	Long Term Evolution
LTSA	Learning Technology System Architecture
LTSC	Learning Technology Standards Committee
M_P_Ack	Movement Probability Acknowledgement
M_P_Not	Movement Probability Notification
MA	Mobility Agent
MAeLE	Metadata-driven Adaptive eLearning Environment
MAG	Mobile Access Gateway
MaISAM	Mobility management aware next step In Signaling for All-IP Mobile
MAP	Mobility Anchor Point
MCONind	MIFA CONnection indicator
MCONreq	MIFA CONnection request
MDISind	MIFA DISconnection indicator
MDISreq	MIFA DISconnection request
MEHROM	Micro-mobility support with Efficient Handoff and Route Optimization
Mem_Join_Resp	Member Join Response
Mem_Join_Rqst	Member Join Request
MIFA	Mobile IP Fast Authentication
MIFAv4	Mobile IP Fast Authentication protocol for IPv4 networks
MIFAv6	Mobile IP Fast Authentication protocol for IPv6 networks
MIPRR	Regional Registration for MIPv4
MIPv4	Mobile IP version 4
MIPv6	Mobile IP version 6
MITHv4	Mobile-Initiated Tunneling Handoff mechanism for IPv4
MME	Mobility Management Entity
MMP	Multicast for Mobility Protocol
MMSP	Mobile Multimedia Streaming Protocol
MN	Mobile Node
MNF	Multicast Non-Forwarding
Moodle	Modular Object-Oriented Dynamic Learning Environment
MPLS	Multi-Protocol Label Switching

MR	Mobility Router
MS	Mobile Station
MSC	Mobile services Switching Center
mSCTP	mobile SCTP
MSF	Multiple Stream Forwarding
MSGen	Mobility Scenarios Generator
MSOCKS	Mobile SOCKeTS
M-TCP	Mobile TCP
M-UDP	Mobile UDP
NAT	Network Address Translation
NetGen	Network Generator
NETLMM	NETwork-based Localized Mobility Management
NS2	Network Simulator 2
NSIS	Next Step In Signaling
NSS	Network SubSystem
NTP	Network Time Protocol
OMC	Operations and Maintenance Center
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OSS	Operation SubSystem
OTcl	Object-oriented Tool Control Language
PAA	Proxy Agent Architecture
PAI	Paging Area ID
PAPI	Public And Private Information
PBAck	Proxy Binding Acknowledgement
PBU	Proxy Binding Update
PCoA	Proactive Care of Address
PCoA-D	Proactive Care of Address-Disable
PCoA-E	Proactive Care of Address--Enable
PDN	Public Data Network
PFA_Ack	Previous FA Acknowledgement
PFA_Not	Previous FA Notification
P-MIP	Paging MIP
Pr_Rt_Adv	Proxy Router Advertisement
Pr_Rt_Sol	Proxy Router Solicitation
ProtDes	Protocol Designer
PrRtAdv	Proxy Router Advertisement
PrRtSol	Proxy Router Solicitation
PSD	Packet Switched Domain
QoMIFA	QoS-aware Mobile IP Fast Authentication
QoS	Quality of Service
QoS-NSLP	QoS NSIS Signaling Layer Protocol

R ² CP	Radial Reception Control Protocol
RA	Router Advertisement
RCP	Reception Control Protocol
RDC	Routing Domain Controller
REACH	Roaming-Enabled ArCHitecture
Reg_Rply	Registration Reply
Reg_Rqst	Registration Request
RFA	Regional Foreign Agent
RLO	Reusable Learning Object
RNC	Radio Network Controller
RNode	Routing Node
RS	Router Solicitation
RSS	Radio SubSystem
RSVP	resource ReSerVation Protocol
RSVP-TE	resource ReSerVation Protocol-Traffic Engineering
RTE	Run-Time Environment
RTP	Real Time Protocol
RtSolPr	Router Solicitation Proxy
SA	Security Association
SAP	Service Access Point
Scast	Simulcast
SCO	Sharable Content Object
SCoAT	Soft CoA Tuple
SCORM	Shareable Content Object Reference Model
SDL	Specification and Description Language
SGSN	Serving GPRS Support Node
SIGMA	Seamless IP diversity-based Generalized Mobility Architecture
SIP	Session Initiation Protocol
SLM	Session Layer Mobility Management
S-MIP	Seamless Mobile IP
sMIPv4	surrogate Mobile IP version 4
SN	Sequencing and Navigation
Soff	Simulcast off
SPS	Synchronized-Packet-Simulcast
SSF	Single Stream Forwarding
STA	Mobile stations
SubA	Subnet Agent
TCP	Transmission Control Protocol
TCP-R	TCP Redirection
TDMA	Time Division Multiple Access
TeleMIP	Telecommunication Enhanced Mobile IP
TIMIP	Terminal Independent MIP

UDP	User Datagram Protocol
UE	User Equipment
ULS	User Location Server
UMTS	Universal Mobile Telecommunication System
UNF	Unicast Non-Forwarding
UPE	User Plane Entity
UTRAN	Universal Terrestrial Radio Access Network
VLR	Visitor Location Register
WAN	Wide Area Network
WBT	Web Based Training
WCMS	Web Content Management System
WLAN	Wireless Local Area Network
WWW	World Wide Web
2G	Second generation mobile communication networks
3GPP	3rd Generation Partnership Project
4G	Fourth generation mobile communication networks

1. Introduction

Mobile communication networks experience a tremendous development clearly evident from the wide variety of new applications way beyond classical phone services. Mobile devices are continuously developed to support not only voice communication, but also other applications, such as Internet browsing, e-banking, video conferencing, peer-to-peer communication, etc. The tremendous success of the Internet along with the demand for always-on connectivity regardless of the users' locations has triggered the development of IP-based mobile communication networks. Deploying these networks requires, however, overcoming many challenges. One of the main challenges is how to manage the mobility between cells connecting through an IP core in a way that satisfies real-time requirements. This challenge is the focus of this dissertation.

This chapter introduces this dissertation starting with a short introduction of mobile communication networks followed by a discussion of the problems addressed in this work. Furthermore, this chapter highlights the objectives as well as the contributions of this dissertation. The chapter is structured as follows: section [1.1](#) provides a brief introduction to mobile communication networks. The problem statements are described in section [1.2](#). Section [1.3](#) lists the objectives and the contributions of this dissertation. Finally, the dissertation outline is presented in section [1.4](#).

1.1. Mobile Communication Networks

Mobile networks can be classified into two primary categories according to their network structure. The most popular and simplest structure is the cellular one, shown in figure 1.1. The second type is the ad hoc network structure, shown in figure 1.2.

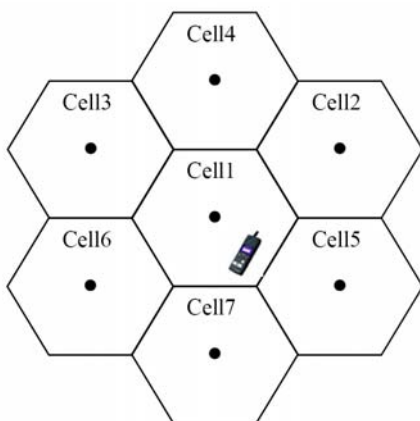


Fig 1.1: Cellular network

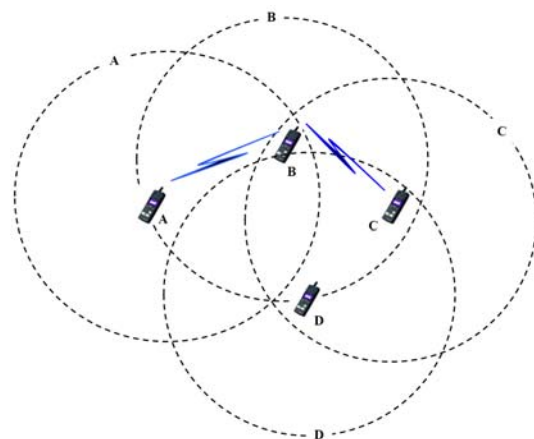


Fig 1.2: Ad hoc network

Cellular networks have a fixed infrastructure consisting of Base Stations (BSs) or Access Points (APs) connected by a wired infrastructure. This infrastructure is connected with other communication networks through special gateways. A Mobile Node (MN) is connected to a BS via a wireless link. The BS can communicate with all MNs in its radio range, called a cell. These MNs can move freely from one cell to another. In contrast to cellular networks, an ad

hoc network is more complex and has no fixed infrastructure. Each node in an ad hoc network may act as a sender, receiver or a potential router between two communicating partners that may not be in direct radio contact with each other. There are many technical challenges that must be addressed to make such networks usable in practice. Primarily, as the nodes can be mobile, the topology may change dynamically. This means that a dynamic routing protocol must be employed to maintain routes between a pair of communicating nodes. Because of bandwidth constraints of wireless links as well as power constraints of MNs, the routing protocol must be efficient in terms of routing overhead.

Cellular networks are widely available and offer connectivity and different kinds of services for users. In the following, common cellular networks will be briefly described.

The Global Standard for Mobile communication (GSM) [ESTI] is the European standard of second generation (2G) mobile communication networks. It is one of the most successful systems and is deployed in more than 220 countries (as of the end of 2008) [Gwo08]. GSM provides wide-area services. Its architecture and specifications can be found in [ESTI96]. Figure 1.3 shows the network infrastructure of a GSM network, which can be divided into three main parts, the Radio SubSystem (RSS), the Network SubSystem (NSS) and the Operation SubSystem (OSS), see [Sch03].

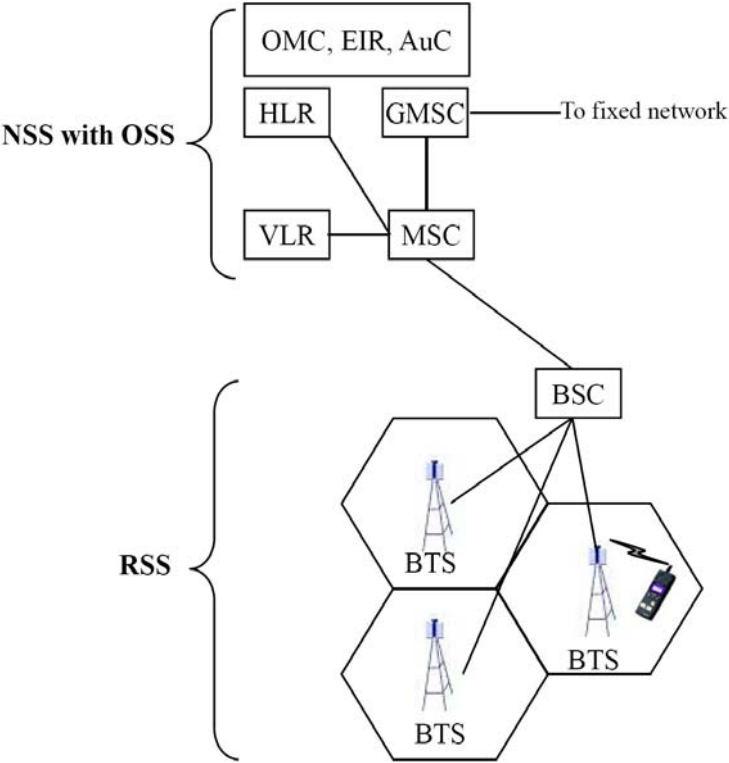


Fig 1.3: GSM network structure

A RSS consists of Mobile Stations (MSs) and a Base Station Subsystem (BSS), which contains Base Transceiver Stations (BTSS) and a Base Station Controller (BSC). A MS presents the hardware and the software required to communicate within the GSM system. Additionally, it carries the subscriber specifications. The GSM network contains many BSSs. Each BSS contains the functions necessary to maintain a permanent radio connection with the MSs. BTSSs focus on radio aspects, i.e. antennas, radio signal and baseband processing, etc. Each BSC controls many BTSSs and provides the functions required to manage the radio resources, support paging and realize handoffs from one BTSS to another controlled by the same BSC.

The NSS connects the BSSs to other common fixed or mobile communication networks, controls the BSSs and supports user localization, accounting and roaming between different mobile communication networks. It contains Mobile services Switching Centers (MSCs), a Gateway MSC (GMSC), a Home Location Register (HLR) and Visitor Location Registers (VLRs). Each MSC controls the BSCs connected to it. The main task of a MSC is to process the signaling required for communication and mobility management including the processing of handoffs and the switching of calls between the GSM system and other common networks, e.g. Integrated Services Digital Network (ISDN), Public Data Network (PDN), etc. The HLR is the most important database in a GSM system and contains the subscribers' data. These data are used to localize users, determine the allowed services, etc. The VLR is a dynamic database and is usually associated with a MSC. It contains important data of the users present in the location area controlled by the specific MSC, to which the VLR is connected. When a new user enters the location area served by the MSC, the VLR requests the respective user-related data from the HLR. These data are used then to control the user without needing to permanently signal the HLR.

The OSS contains the information and the components required to operate and maintain the GSM network. It consists of an Operation and Maintenance Center (OMC), an Equipment Identity Register (EIR) and an Authentication Center (AuC). The OMC observes and controls other network components. Mainly, it observes the traffic of each component and builds status reports for this component. In addition, it manages users' data, produces billing reports of users, etc. The AuC contains users' keys and produces the parameters required for the authentication in the HLR. The EIR is a database for all International Mobile Equipment Identities (IMEI)¹. In this database there are three lists of mobile devices, namely a black list of all stolen mobile devices, a gray list of all mobile devices identified to malfunction and a white list of all other mobile devices.

Following the second generation of mobile systems, the Universal Mobile Telecommunication System (UMTS) [3GPP] (also referred to as a 3G system) has been developed. UMTS provides wide-area services too. However, UMTS uses the Code Division Multiple Access (CDMA) on the radio, whereas GSM uses the Time Division Multiple Access (TDMA).

Figure 1.4 shows the base architecture of a UMTS network [Sch03]. As seen, UMTS extends the architecture of a GSM system. A UMTS network consists of a Universal Terrestrial Radio Access Network (UTRAN) and a core network, which in turn consists of a Circuit Switched Domain (CSD) and a Packet Switched Domain (PSD). The CSD supports classic telephone services and reuses the components of the GSM network (MSC, GMSC, HLR, VLR, EIR and AuC). The PSD is used for packet data transmission. Its main elements are Serving GPRS Support Nodes (SGSNs) and a Gateway GSN (GGSN). HLR, EIR and AuC are utilized by the PSD as well. The UTRAN in UMTS systems deals with radio specifics. This part consists of NodeBs and Radio Network Controllers (RNCs). Similar to a BTS in GSM, the NodeB focuses mainly on radio and baseband processing, leaving higher layer processing to the RNC. Important functions include power control to prevent near-far effects and measurements of wireless link quality and signal strength. The RNC is a central node in a UMTS system. It controls the traffic in the cells and determines whether a new call will be accepted or blocked. In addition, the RNC encrypts data coming from fixed networks before transmitting them over the radio, decrypts data originating from the mobile device before forwarding them to the fixed network and manages radio resources. A User Equipment (UE) is a mobile device comprised of the hardware and software required to communicate with a NodeB. The UE performs measurements of the signal strength to control the transmission power.

¹ IMEI is a unique 17 or 15 digit code used to identify an individual MN to a GSM network.

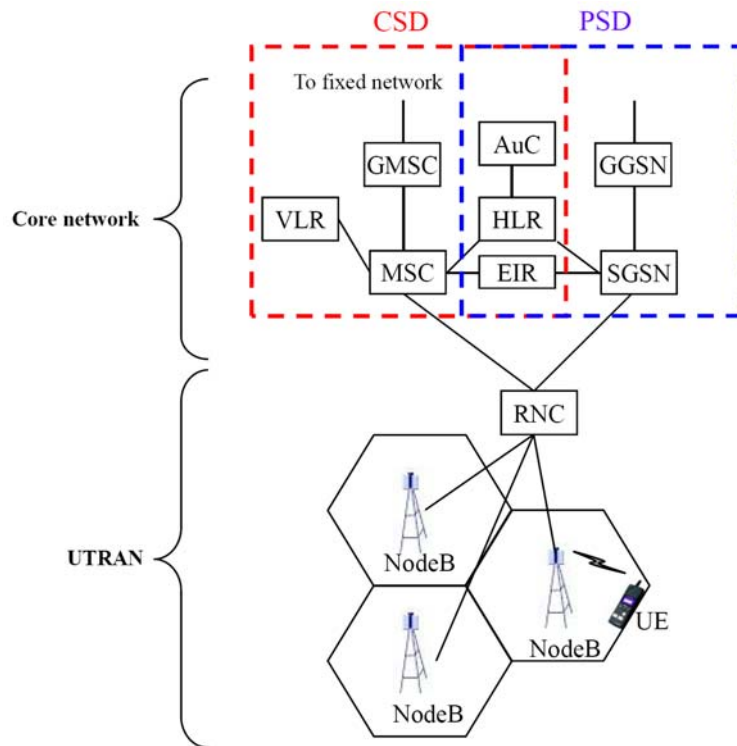


Fig 1.4: UMTS network structure

Currently, the 3rd Generation Partnership Project (3GPP) Long Term Evolution (LTE)¹ [3GPP-L], [Tec06] is focusing on improving the UMTS network to meet the increasing requirements of users, i.e. higher bandwidth and lower delays at lower cost. The goals of LTE include improvements in spectrum efficiency, a reduction of control and user plane latency and lowered costs for operators and users. It also aims at providing higher data rate and average throughput than currently obtained from 3G systems, see [EFK06]. The architecture of LTE includes an Evolved UTRAN (E-UTRAN) on the access side and an Evolved Packet Core (EPC) on the core side, see figure 1.5.

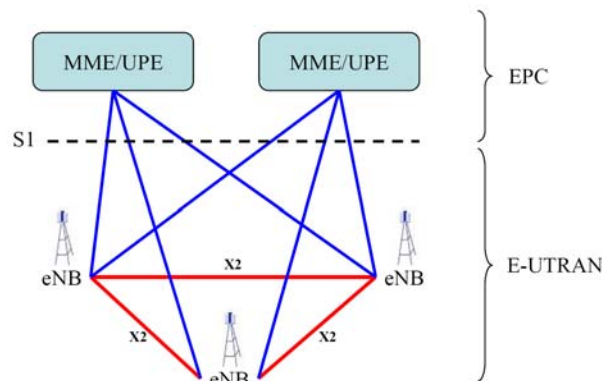


Fig 1.5: LTE network structure

The E-UTRAN consists only of evolved NodeBs (eNBs), which support radio resource management, radio bearer control, radio admission control, connection mobility control and dynamic resource allocation. The eNBs are interconnected with each other by means of X_2 interfaces. It is assumed that an X_2 interface always exists between eNBs that should interconnect to support some functions, e.g. fast handoff management. By means of S_1

¹ 3GPP LTE is the name of the project within 3GPP, which works on the development of LTE.

interfaces, eNBs are connected to the EPC, which contains a Mobility Management Entity (MME) and a User Plane Entity (UPE). The MME distributes paging messages to the eNBs, while the UPE is responsible for processing user data.

In contrast to GSM, UMTS and LTE networks, Wireless Local Area Network (WLAN) [Sch03], [IEEEin] provides local area services and delivers high throughput. IEEE 802.11 is the WLAN standard for license-free usage. The specifications of WLANs make it adequate for rather local and indoor networks, e.g. on a campus, airport, etc. WLAN systems can be operated either in infrastructure or ad hoc mode. Figure 1.6 illustrates the basic architecture of a WLAN system operating in infrastructure mode. A mobile station (STA) contains the mechanisms required to access the medium and build a wireless link with an AP. Each AP along with all STAs in the area covered by this AP build a Basic Service Set (BSS). The APs connects their STAs with other STAs or fixed PCs through a Distribution System (DS). This DS can be one or more LANs, the Internet, etc. The STAs, APs and the DS build an Extended Service Set (ESS), see [Sch03].

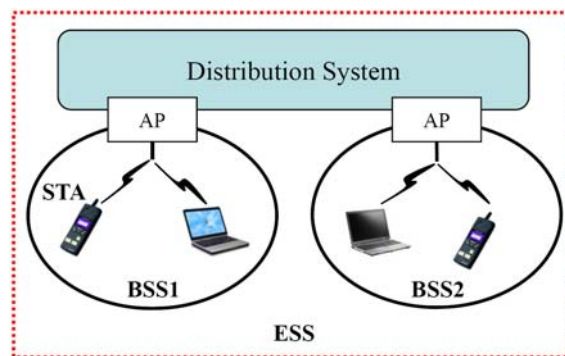


Fig 1.6: WLAN structure (infrastructure mode)

Current research focuses on developing fourth generation mobile communication networks (4G) [Ibr02], [Raa07]. This new generation is intended to complement and replace 2G and 3G systems, see figure 1.7.

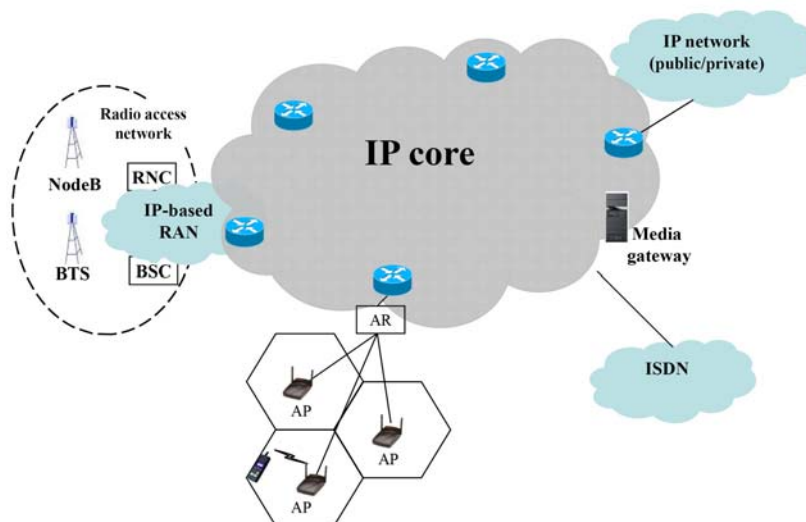


Fig 1.7: All-IP network structure

Ubiquitous access to information and the use of applications anywhere and anytime in addition to the support of large data volumes and minimal delay are the key features of 4G networks. Different from previous networks, 4G networks comprise a set of heterogeneous networks integrating different existing and future systems, e.g. GSM, UMTS, WLAN,

WIMAX, LTE, etc., by means of a common IP core. This new generation will support higher data rates, reduced latencies and smoother handoffs. The focus is on ensuring seamless services across a multitude of wireless systems and networks. 4G is also termed as All-IP and is expected to be widely deployed in the near future, to serve fixed as well as mobile subscribers and to offer any type of service, anytime, anywhere and anyhow under dynamic network conditions. The All-IP architecture aims at lower costs and increased scalability of the network. However, in order to reach these ambitious goals, several challenges must be overcome, e.g. efficient management mobility between cells connected through an IP core, guarantee of Quality of Service (QoS), securing of communication links and content, etc.

1.2. Problem Statements

Today's cellular communication networks, such as GSM or UMTS, are able to offer seamless and fast handoffs. However, the used mobility management techniques are complex and differ from network to network. The current Internet can deliver flexible services at lower cost than those resulting from today's cellular communication networks. It is assumed that the Internet will be a major part of future All-IP networks. The end hosts of All-IP networks will be IP hosts. Furthermore, the various radio access networks will be connected to the global Internet through gateways that represent IP routers. Therefore, mobility management solutions that should be employed in All-IP networks are IP-based. Thus, a MN is represented by an IP address, which also represents the point of attachment to the Internet. This IP address is required to establish a session between the MN and any other nodes in the network. It is likely, however, that the MN will change its point of attachment causing assigning of a new topology-correct IP address. Changing the IP address during an ongoing session may enforce a close-down and re-opening of the session, resulting in a disruption of the communication between the MN and its communication partners. As known, real-time applications are highly affected by any communication disruption during the movement from one cell to another. This disruption will be more critical as the user mobility of IP-based MNs increases and the system cell size decreases. Therefore, the development of adequate mobility management solutions is a big challenge in future All-IP networks. Addressing this challenge is the main motivation of this dissertation, which delivers an in-depth analysis of this problem, highlights the state of the art and develops a new IP-based mobility management approach that advances the state of the art and satisfies the requirements of real-time applications.

This dissertation also focuses on the performance evaluation of mobility management protocols as well. In order to evaluate a mobility management protocol and compare it to others, a mathematical model for each protocol should be developed or the protocols should be simulated or implemented. Implementation and simulation are typically time-consuming. They deliver, however, detailed and accurate results. Mathematical models can be developed more quickly and result in a good performance estimates. Until now, there is no generic mathematical model that allows for the evaluation of a large set of mobility management protocols. The development of such a generic mathematical model is of major interest since it greatly simplifies the analysis of mobility management protocols. In addition, it allows for a comparison of mobility protocols under identical conditions, e.g. mobility scenarios, network topologies, etc. Therefore, the development of such a generic model also represents a main goal of this dissertation.

In addition, the dissertation covers a new adaptive eLearning environment to support studying and dissemination of mobility management issues covered in this dissertation. This eLearning environment allows for a personalization of contents depending on users characteristics. By means of such an environment, researchers can quickly become involved in current research

trends, while students are provided with courses introducing the topics covered in the dissertation to eliminate gaps in their knowledge.

1.3. Objectives and Contributions

1.3.1. Objectives

As described in the previous section, this dissertation addresses the IP mobility management issue. More concrete, it focuses on network layer mobility management. The main goals of this work can be summarized as follows:

1. **Development of a layer 3 mobility management solution** that avoids the drawbacks of existing approaches and minimizes or even eliminates the layer 3 handoff latency.
2. **Development of a generic mathematical model** that can be used to evaluate a wide range of layer 3 mobility management solutions.
3. **Development of an adaptive eLearning environment** that personalizes eLearning contents depending on users characteristics. This environment also helps in conveying the topics covered in this dissertation.

1.3.2. Contributions

The following contributions have been accomplished throughout this dissertation:

1. **An in-depth analysis of the mobility management problem** in future IP-based mobile communication networks.
2. **A classification of mobility management solutions in different layers of the TCP/IP reference model.** The assets and disadvantages of the mobility management implementation in different layers are investigated along with a presentation of the most important approaches in each layer.
3. **A comprehensive review of layer 3 mobility management protocols** along with a discussion of the pros and cons of each protocol.
4. **A qualitative comparison of the described layer 3 mobility management protocols** with respect to handoff management, paging, new nodes that should be introduced to the network to support mobility, nodes that should be updated in the network to enable the mobility management protocol to be employed, used network topology, dependency on layer 2 information, usage of a tunnel, expected handoff performance and load balancing.
5. **Development of a layer 3 mobility management protocol** named Mobile IP Fast Authentication protocol (MIFA), which achieves smooth handoffs without constraining the network topology or introducing new nodes. The developed protocol is specified for both IPv4 and IPv6 networks.
6. **Formal specification of MIFA** for IPv4 using the Specification and Description Language (SDL)¹.
7. **Development of a new generic mathematical model** that allows for the evaluation of a large set of mobility management protocols with respect to the average handoff

¹SDL is a specification language targeted at the specification and description of the behavior of reactive and distributed systems. It is defined by the ITU-T (recommendation Z.100). Originally, it focused on telecommunication systems. However, its current application areas include process control and real-time applications in general.

latency, expected average number of dropped packets, location update cost and packet delivery cost. Parameters of the generic model are set according to characteristics of studied protocols, network topologies and mobility scenarios. The developed model takes the dropping of control messages into account and can be applied to make-before-break as well as break-before-make mobility solutions. To further simplify the analysis using this generic model, a set of tools has been developed to graphically create mobility scenarios, create network topologies and define protocols operation.

8. **Evaluation of MIFA compared to a wide set of well-known mobility management protocols** using the developed generic mathematical model.
9. **A detailed evaluation of MIFA** compared to two well-known mobility management protocols by means of simulation studies, modeled in Network Simulator 2 (NS2)¹ [NS2]. The evaluation comprises studying the impact of network topology, network load and MN speed for real-time and non-real-time traffic.
10. **A literature review and discussion of adaptive eLearning environments.**
11. **Development of a new adaptive eLearning environment** named Metadata-driven Adaptive eLearning Environment (MAeLE), which allows for the personalization of eLearning contents. The architecture, user model and structure of courses created by means of this environment are provided in addition to a prototype implementation.

1.4. Outline

This dissertation is structured as follows: chapter [2](#) provides an overview of basic wireless access scenarios focusing on the definition of a handoff from various points of view. The requirements that should be satisfied by each mobility management approach are presented in this chapter as well. Thereafter, a classification of mobility management protocols regarding their implementation in the layers of the TCP/IP reference model is introduced along with a brief description and classification of well-known approaches for each layer.

Chapter [3](#) focuses on network layer mobility management and highlights the state of the art. A wide range of well-known layer 3 mobility management approaches is presented. Thereafter, a qualitative comparison of the described approaches is provided. This chapter concludes with the main obtained results and the motivation to develop a new layer 3 mobility management solution.

The developed layer 3 mobility management approach, MIFA, is described in chapter [4](#). The chapter presents also the protocol specification for IPv4 and IPv6 networks. Moreover, the SDL specification for the most important parts of MIFA for IPv4 networks is described briefly.

The developed generic mathematical model is described in chapter [5](#). MIFA is evaluated compared to a wide range of well-known layer 3 mobility management protocols using this model. The chapter validates the generic mathematical model by comparing its results to results of simulation as well as real testbeds. Several tools developed to simplify the analysis of mobility management protocols using the generic model are described shortly in this chapter as well.

Chapter [6](#) evaluates MIFA compared to two well-known mobility management protocols via simulation studies modeled in NS2. The evaluation comprises an assessment of the impact of network topology, network load and MN speed using real-time and non-real-time traffic. In

¹ NS2 is a widely-used discrete event simulator targeted at the simulation of wired and wireless networks. A brief description of this simulator will be provided in chapter [6](#).

addition, the behavior of studied protocols under dynamically changing network conditions is evaluated.

The developed adaptive eLearning environment is described in chapter [7](#). This chapter defines the term adaptivity and briefly introduces well-known standards for eLearning. Thereafter, the chapter provides a short description of existing adaptive eLearning environments followed by a detailed description of our developed adaptive eLearning environment. Afterwards, a prototype for the developed environment containing an example course is presented.

Finally, chapter [8](#) concludes the dissertation with the main obtained results and several proposals for future work. In addition to the chapters of this dissertation, several appendixes covering in more detail additional topics supporting the dissertation contributions are included. A list of the messages used in MIFA including a brief description of them is provided in appendix [A](#). Appendix [B](#) discusses three mechanisms used to build groups of neighbors for MIFA. Appendix [C](#) presents a more detailed view of the MIFA SDL specification for IPv4. A list of all used parameters for each protocol analyzed in this dissertation using the generic mathematical model is provided in appendix [D](#). This appendix also contains an example explaining how these parameters can be derived. Appendix [E](#) discusses how asymmetrical network topologies can be considered in the generic mathematical model. The tools implemented to simplify the evaluation of mobility management protocols using the generic model are described in more detail in appendix [F](#). Finally, additional simulation results explaining the impact of network topology are presented in appendix [G](#).

2. Mobility in Mobile Communication Networks

As described in chapter [1](#), All-IP is the vision of mobile communication networks aiming at realizing an always-on connectivity. To achieve this, a suitable IP-based mobility management solution is essential. Therefore, an in-depth analysis of the mobility management issue, especially its requirements and the layer of the TCP/IP reference model where mobility should be implemented, is provided.

This chapter is organized as follows: section [2.1](#) introduces basic wireless access scenarios and discusses the handoff from various points of view. The architecture of IP-based networks is introduced in section [2.2](#). Mobility management requirements are discussed in section [2.3](#). Section [2.4](#) presents a classification of mobility management protocols with respect to the layers of the TCP/IP reference model they are implemented inside. The advantages and disadvantages of mobility support in each layer are discussed in this section along with a brief description and a classification of the well-known approaches in each layer. Section [2.5](#) summarizes the main results.

2.1. Overview

As described before, mobile communication networks are divided into two main parts, a wireless and a wired one. In infrastructure-based systems, each MN communicates with another fixed or mobile terminal through an AP. The wireless network is termed as homogenous when the APs use the same wireless technology and heterogeneous otherwise. When a MN moves outside of the coverage area of a certain AP into the coverage area of a new one, the responsibility for it is transferred to the new AP. This procedure is called a handoff and includes three phases, namely handoff detection, initiation and execution. It includes an exchange of control messages, also termed as signaling messages, between the MN and the network on one side and between the network nodes themselves on the other side.

Three scenarios for the usage of wireless access have been identified [[Wol00](#)]. The first scenario is referred to as a basic wireless access. It is deployed to avoid the installation of cables. MNs can move slowly within the range of a single AP, always the same one. This scenario is a generalization of cordless telephony. The main challenge here is to provide an adequate quality. The second scenario is a nomadic wireless access. In this scenario, MNs are expected to move outside of the current AP's coverage area. However, when a MN has an active session, it should not move to another AP. This means that this MN can change its AP only between two consecutive sessions. The time required to execute a movement to a new AP is usually longer than the session duration itself. Assuring a simple setup in the new environment seems to be a major challenge in such scenarios. An additional challenge is the assurance of security as well as reachability under the original address in the current temporary environment. Finally, the third scenario is called a true mobile access. MNs can move between different APs during an active session. The grade of service continuity is a main quality feature in such scenarios. Service continuity means that there is either no information loss or no observable disruption during the handoff.

The handoff can be categorized according to many criteria, such as number of involved access points, the wireless link used for the handoff operation, initiator of the handoff, changing of

access technology, interconnection network topology, terminal state and service continuity, see [Fes03]. The following discusses these criteria in more detail.

With respect to **the number of involved APs**, the handoff can be classified into **hard**, **soft**, **softer** and **predictive handoff**. Considering the hard handoff, each MN has a connection to one AP only. However, the MN can communicate with more than one AP simultaneously by the soft handoff. Soft handoff requires, of course, that the wireless cells overlap. Handoff from one sector to another controlled by the same AP is called a softer handoff. With predictive handoff, a set of APs, to which the MN may move in the future, is predicted. The current AP, usually referred to as active AP, forwards normally the data sent to the MN in advance to the other APs present in this set, usually referred to as passive APs. Each passive AP buffers the MN's data. When the MN moves to one of these APs, it obtains its data directly after the establishment of the new wireless link. The MN's data buffered in the other passive APs will be deleted after a certain time.

Depending on **the wireless link used for the handoff operation**, the handoff can be either a **backward** or a **forward handoff**. The backward handoff allows the MN to execute a handoff while still being connected. The MN uses the current AP to request a handoff operation from the new one. It is assumed, however, that the handoff as well as the new AP can be predicted. By the forward handoff, the MN executes a handoff through the new AP after the radio link with the old one breaks down.

Considering the **initiator of the handoff**, the handoff can be **terminal-initiated**, **network-initiated** or **network-initiated terminal-assisted**. In the terminal-initiated handoff, the terminal manages the handoff and decides to which AP it should be connected. In contrast, the network takes the main role in the network-initiated handoff. It is assumed, however, that the network can locate the terminal. With the network-initiated terminal-assisted handoff, the network manages the handoff. In addition, the terminal periodically sends reports to the network to assist in taking a handoff decision.

Depending on **changing the access technology**, the handoff can be an **intra-** or **inter-technology handoff**. The intra-technology handoff is a handoff between APs operating in the same wireless technology. The inter-technology handoff, however, is a handoff between APs operating in different wireless technologies.

Taking the **interconnection network topology** into account, the handoff can be a **local** or a **global handoff**. The MN moves inside the same administrative domain in local handoffs. The global handoff expresses a movement to a new AP belonging to another administrative domain.

Depending on the **terminal state**, the handoff is divided into an **active** and an **idle handoff**. The active handoff denotes a handoff during an ongoing session. In contrast, the idle handoff stands for a handoff while the terminal is in idle mode¹.

Depending on the achieved **service continuity**, the handoff is classified as a **lossless**, a **seamless** or a **non-seamless handoff**. Lossless handoff ensures no loss in the continuity in terms of no packet loss. The seamless handoff is unnoticeable to the user, while the user notices the non-seamless handoff.

2.2. IP-Based Networks

The Internet is a collection of interconnected networks, referred to as subnets. Each subnet has its own address space, termed as a network address. The subnet in turn consists of a set of

¹ Terminals in idle mode are terminals that have no sessions with communication partners.

interconnected hosts. A unique address, referred to as a host address, is assigned to each host within the subnet. The combination between the network and the host address identifies the host within the Internet. This address is a permanent address [Hal96].

Multiple subnets are interconnected with each other through special nodes called routers. The Internet Protocol version 4 (IPv4) [Pos81] and version 6 (IPv6) [DHi98], [IPv6WG] are the basic standards used to forward data packets from a given source to a certain destination. Routing is processed in the third layer according to the ISO/OSI reference model¹, it is a connection-less transport of data packets and does not guarantee an in-order delivery of packets. Preserving the sequence of data packets is left to higher layer protocols, such as the Transmission Control Protocol (TCP) [Pos81a].

The global Internet can be seen as multiple Internets interconnected with each other through an IP core. Each Internet, termed usually as an Autonomous System (AS) [Fes03], has its own authority and routing methods. Interior and exterior gateways are used to access the AS. The corresponding routing protocols are the Interior Gateway Protocol (IGP) and the Exterior Gateway Protocol (EGP) [RLi95]. The TCP/IP protocol suite is in principle the set of protocols used for IP-based networks, especially the Internet. The layers of the TCP/IP and the ISO/OSI reference model are shown in figure 2.1.

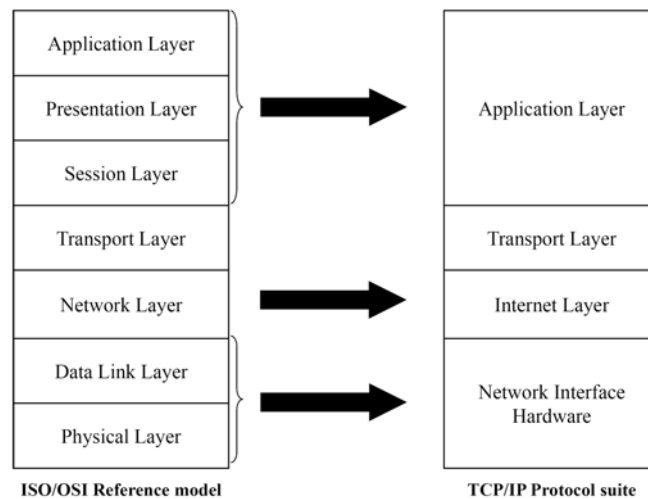


Fig 2.1: TCP/IP and ISO/OSI reference models

Wireless IP-based networks include a set of wireless hosts connected to other hosts through an IP core. Each access network has a gateway to access the global Internet and a set of APs to communicate with the wireless terminals located in the coverage area of this access network. Radio access technologies may differ between different access networks. Each host in these networks is identified by means of a unique IP address reflecting the topological location.

Many challenges have to be solved when using the TCP/IP protocol suite in All-IP networks. Some of them are related to wireless transmission, such as error control, power control, etc. Others are related to the provision of QoS, support of host mobility, security, etc. As mentioned previously, the main issue dealt with in this dissertation is the support of mobility in IP-based networks. There must be mechanisms that enable seamless handoffs. Moreover, these networks should provide personal, terminal and network mobility [WAG03]. Personal mobility expresses the ability of a user to access his personalized network services while being away from his home network. Terminal mobility presents the ability of a network to locate a MN, route incoming or outgoing calls and maintain a connection to this MN while

¹ Open Systems Interconnection (OSI) reference model standardized by the International Standards Organization (ISO)

away from home [ZVT02]. Terminal mobility requires support for two main tasks, namely location management and handoff management [SMM04]. Network mobility expresses the ability of a network to support roaming of an entire subnet or an ad hoc network. This dissertation focuses mainly on terminal mobility, which will be explained in more detail in the next following.

2.3. Mobility Management Requirements

IP applications can be classified into two main groups, namely real-time and elastic applications [Fes03]. Real-time applications, such as video, audio, etc., require a certain minimum of bandwidth to work well. In contrast, elastic applications, such as file transfer, e-mail, Web browsing, etc., use the available bandwidth. If no bandwidth is temporarily available, these applications wait without being severely affected.

Real-time applications can be either two- or one-way applications. The applications that realize two-way communication, such as voice over IP, interactive games, etc., typically require a low delay (100 msec [KR01], [Bla02]) in order to ensure application interactivity. Real-time applications with one-way communication, such as streaming, stored audio/video, etc., require that the end-to-end delay should not exceed a maximum threshold, referred to as a play-out time. These applications typically use a play-out buffer [RKT94] to minimize the variation in the delay, termed as jitter as well. Therefore, they need to know the maximum delay in order to adjust the size of their play-out buffer. Regarding the reliability, real-time applications are loss-tolerant. However, elastic applications require a reliable data transfer. Reliability is offered by a reliable transport protocol, such as TCP.

The main QoS metrics are: delay, jitter, reliability and bandwidth. The delay presents the time required to send a packet from a given source to a certain destination. As mentioned above, the variation in this delay is referred to as jitter. The reliability describes how can applications tolerate a packet loss. The bandwidth expresses the data transmission capability of a network. Terminal mobility highly affects the QoS that can be offered to applications. A mobility scheme may increase the packet delay through forwarding on a non-optimal path, e.g. via the user's home network. As a result, the application notices the handoff, through a disruption of the offered service, and may not be able to tolerate this delay, which may result in packet loss. Notice that packets are received in this case. However, they are discarded due to exceeding the acceptable end-to-end delay. In addition, a mobility scheme may discard the packets in-flight¹ resulting in a packet loss at the receiver or may forward them from the old to the new AP during the handoff resulting in an increase in the end-to-end delay and jitter for these packets compared to the packets received before and after the handoff.

Considering the discussion above, any solution for mobility management in IP-based networks has to take the following requirements into account, see [Fes03] and [Hen03].

1. Keeping a fixed identifier for the MN regardless of its current location.
2. Interworking properly with IP routing and features, such as acquiring a new topological true IP address, etc.
3. Enabling the MN to be located by its peers after the movement.
4. Minimizing impairments on applications.
5. Minimizing the cost for mobility support (signaling cost, packet delivery cost, etc.).

¹ The packets in-flight are the packets forwarded to the MN during the handoff. They will be forwarded normally to the old MN's location until the location of the MN is actualized.

6. Introducing no additional security vulnerabilities to the network.
7. Not affecting the network scalability, but rather, the mobility management scheme should be scalable, robust and deployable.

2.4. Mobility Management Approaches

Mobility management can be implemented in different layers of the TCP/IP reference model. Thus, mobility management solutions can be classified into link layer¹, network layer², transport layer, session layer³, application layer and hybrid layer mobility management solutions. In the following, the motivation behind the support of mobility in the different layers will be given along with a discussion of well-known approaches for each layer.

2.4.1. Link Layer Mobility

As seen in figure 2.1, both the physical layer and the data link layer of the ISO/OSI reference model are combined with each other in one layer in the TCP/IP protocol suite. This layer is responsible for the establishment of a wireless link between the MN and an AP, also referred to as link layer as well as layer 2 mobility in the literature. Link layer mobility deals with the case that the MN moves beyond the range of the current AP and enters the coverage area of a new one. This handoff can be performed below the network layer if the new and the old APs belong to the same subnet. However, a higher layer mobility solution is required, i.e. on the network layer, transport layer or application layer, if the new AP belongs to another subnet.

The link layer handoff comprises four steps [[WOL05](#)]:

1. Recognizing the loss of the connection
2. Search for and detection of a new adequate AP
3. Re-/Authentication with the newly discovered AP
4. Re-/Association with the newly discovered AP

At first, the MN has to recognize that there is a loss of the connection. This can be done either based on the received signal strength or on failed frame transmissions. Depending on the strength of the received signal, the MN declares the out of range and starts the handoff process if the received signal strength goes below a certain threshold. Using the failed frame transmission as an indicator for a handoff needs more time than the time resulting from taking the decision depending on the signal strength. This is because it is difficult to determine the reason for a frame failure, which may be caused by a collision, radio signal fading or due to a movement out of the transmission range. The MN assumes the collision as the first reason for the frame failure and retransmits the frame several times. If the retransmissions stay unsuccessful, the radio signal fading is supposed and, thus, the MN sends probe requests aiming at receiving beacons from the current AP. If no answer is received, the MN declares the AP as out of range and starts scanning the medium for other available APs. To accelerate the layer 2 handoff, the approach presented in [[VKa04](#)] proposes that the MN declares the out

¹ Notice that the TCP/IP reference model does not have a data link layer. However, the handoff implemented in the first layer of the TCP/IP reference model is referred to in the literature as link layer or layer 2 handoff. The name comes originally from the ISO/OSI reference model since the physical and data link layer of the ISO/OSI reference model are combined with each other to form the first layer of the TCP/IP protocol suite.

² This layer is called Internet layer in the TCP/IP reference model. However, the solutions implemented in this layer are referred to in the literature mostly as network layer or layer 3 mobility management solutions.

³ TCP/IP reference model does not have a session layer. However, session layer mobility approaches propose to add a session layer to the TCP/IP protocol suite and to implement the mobility inside.

of range as the first reason for the frame failure and starts scanning the medium. If the reason is radio signal fading, the same AP will be detected.

After recognizing that the link with the current AP is not adequate any more, the MN has to search for and detect a new AP. The methods defined in IEEE 802.11 standard [IEStd] are passive and active scanning methods [WOL05]. Passive scanning works as follows: the MN selects and listens to a channel until it receives a beacon or has listened for beacon duration¹. After that, the MN switches to the next channel and listens again. After the MN has listened to all channels, it selects the new AP depending on the received beacon signals strength. Passive scanning is presented in figure 2.2.

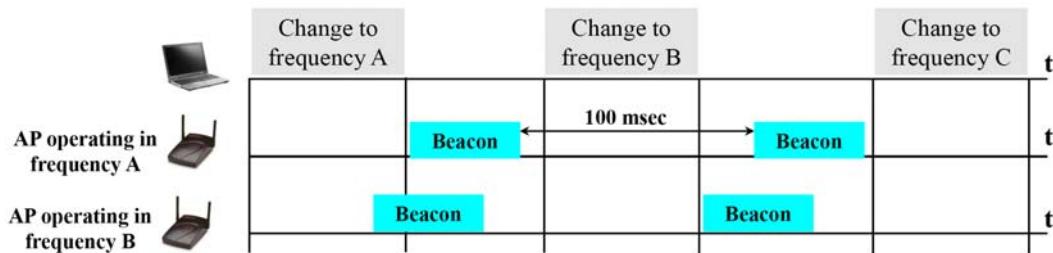


Fig 2.2: Passive scanning in IEEE 802.11 standard

When the MN scans channels actively, it selects a channel and waits for a probe delay to make sure that the selected channel is not active. Afterwards, the MN broadcasts a probe request in order to receive a probe response on this channel and waits for a time called MinChannelTime, see figure 2.3.

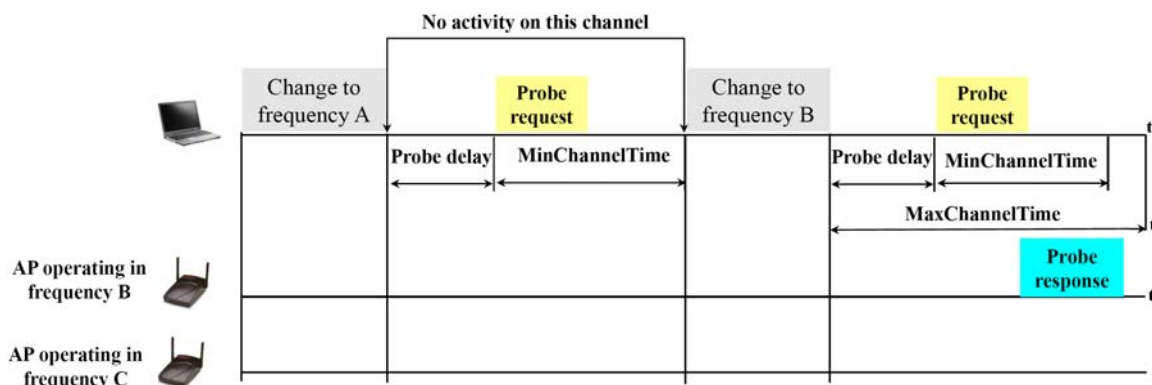


Fig 2.3: Active scanning in IEEE 802.11 standard

If the MN does not notice any activity on the selected channel, it switches to the next channel and starts again. However, if the MN detects that there is traffic sent on the selected channel, it waits for more time called MaxChannelTime aiming at receiving probe responses from the APs working on this channel before switching to the next channel. After the MN has scanned all channels, it selects the new AP depending on the received probe responses strength.

The next step after detecting the new AP is the authentication. The two methods defined in the IEEE 802.11 standard are the open system and shared key authentication, see [IEStd]. The open system is the default authentication method and stands for a null-authentication algorithm, where a non-null-authentication algorithm should be used by the shared key authentication. During the authentication phase, the MN sends an authentication request to the new AP, which is expected to respond by an authentication response indicating acceptance or rejection of the connection. More messages may be exchanged during this phase. Details

¹ The time between two subsequent beacons

depend on the used authentication algorithm. After the authentication phase, the MN has to associate itself with the new AP. This is achieved by exchanging association request and response messages. After that, a wireless link is established between the MN and the new AP. The messages exchanged during the layer 2 handoff are depicted in figure 2.4 for the active scanning case.

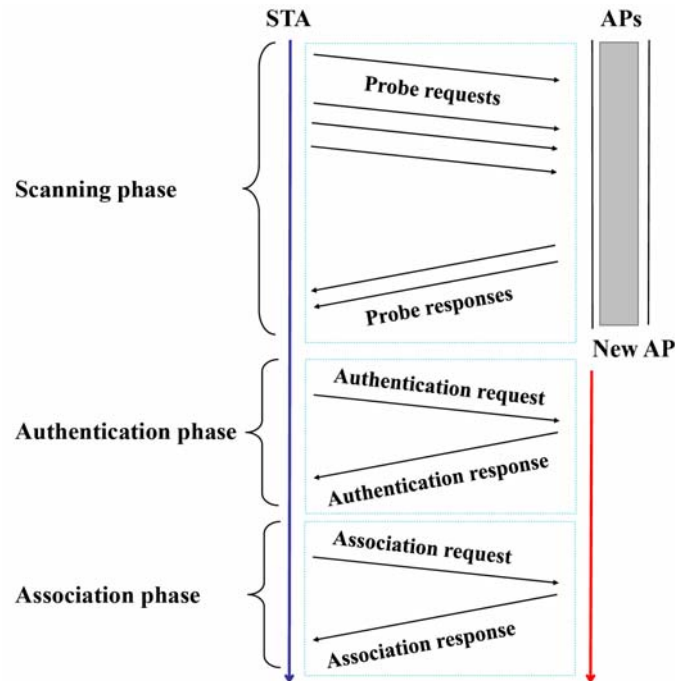


Fig 2.4: Messages exchanged during a layer 2 handoff in IEEE 802.11 standard employing the active scanning

The measurements reported in [MSA02] show that the hardware strongly affects the layer 2 handoff latency. There is a large variation in the layer 2 handoff latency even for the same hardware with the same configuration. Reasons for this variation are various. For example, dropping of beacons or fading results in a large variation in the time required to detect the new AP. Search and detection of a new AP is a dominating factor for the layer 2 handoff latency. This phase accounts for more than 90 % of the overall layer 2 handoff latency. Additionally, it accounts for more than 80 % of the messages exchanged during the layer 2 handoff. This implies that reducing the layer 2 handoff latency effectively requires optimizing the scanning phase.

Many methods have been proposed to reduce the latency resulting from the scanning phase. Periodic scanning [MMN05] aims at reducing this latency through scanning the medium, while the MN is still connected to the old AP. Selective scanning [SFR04] introduces a channel mask to reduce the channels that must be scanned. The caching method, proposed in [SFR04], enables the MN to cache the MAC addresses of the APs discovered during the first full scanning phase. When handing off, the MN tries to associate firstly with one of the APs cached. Other methods [WOL05], [MSA04], [PCh02] try to use information about adjacent APs to accelerate the layer 2 handoff.

2.4.2. Network Layer Mobility

The major benefit obtained from supporting mobility management in the network layer, i.e. the Internet layer of the TCP/IP protocol suite, is the transparency to higher layer protocols, e.g. TCP and the User Datagram Protocol (UDP) [Pos80]. Transparency means that the higher layer protocols should not be affected by the mobility. Therefore, no updates or any other

actions are required at the layer above. The infrastructure, however, is allowed to be changed. Network layer mobility can be supported by achieving a route to the subnet currently being visited by the MN. This can be realized by deploying indirect agents that intercept data packets and forward them to the current visited subnet [Edd04]. The solutions of network layer mobility management can be classified into two main categories, namely terminal-based and network-based mobility management solutions, as shown in figure 2.5. The main difference between these two categories lies in involving the MN in the handoff procedure or not.

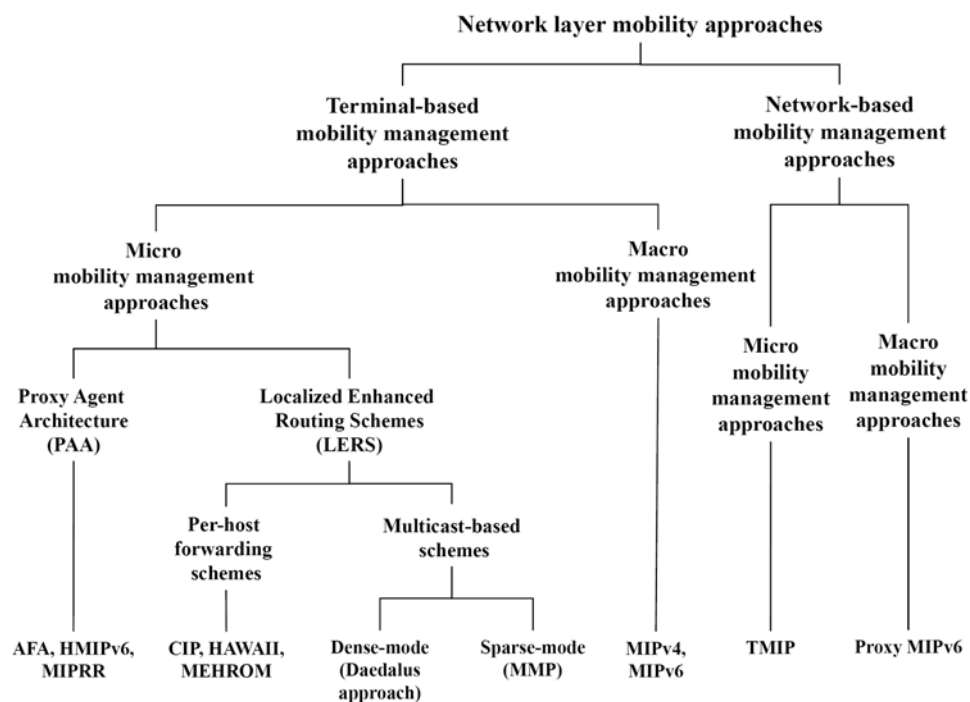


Fig 2.5: Classification of network layer mobility management approaches

Let us consider terminal-based mobility management first. The MN is involved in the handoff procedure and should support mobility functions. The solutions of this category can be further divided into two main groups, macro and micro mobility management solutions. Macro mobility management approaches aim at supporting global mobility. They require updating the MN's location at the MN's home network each time the MN changes the point of attachment. Mobile IP version 4 (MIPv4) [Per02], [mipv4WG] and version 6 (MIPv6) [JPA04], [mextWG] are the well-known examples of these approaches.

Micro mobility management schemes aim at reducing the time required to register with the network by processing the handoff procedure locally. They can be further classified into two subgroups, which are: Proxy Agent Architecture (PAA) and Localized Enhanced Routing Schemes (LERS) [EMS00].

PAA-based approaches, also referred to as tunnel-based schemes as well, extend the MIP principle to manage the mobility locally, e.g. by using a hierarchical network topology and/or intermediate nodes. Anchor FA (AFA) [DYe01], Regional Registration for MIPv4 (MIPRR) [FJP07] and Hierarchical Mobile IPv6 (HMIPv6) [SCM05] are examples of these approaches.

LERSs, also termed as routing-based schemes too, try to support mobility by altering the route to the MN's new location locally. It can be distinguished between per-host forwarding schemes and multicast-based schemes [EMS00]. Per-host forwarding schemes employ a specialized path setup protocol and a location database (routing cache) in the routers existing in the local domain, which, of course, includes many subnets. Routing caches are updated

according to MNs movements. The domain appears as a single subnet to the routers present outside the local domain. There is a special gateway to interconnect the domain with the Internet. Well-known examples of per-host forwarding schemes include Cellular IP (CIP) [Val99], Handoff-Aware Wireless Access Internet Infrastructure (HAWAII) [RLT00], [RLL00] and Micro-mobility support with Efficient Handoff and Route Optimization Mechanisms (MEHROM) [PMD04]. Multicast-based schemes aim at supporting a location-independent addressing and routing by means of point-to-multipoint connections. A multicast address is assigned to the MN and a multicast tree is established and updated according to the MN's movements either after or during the handoff. This can be visualized as a multicast cloud centered on the MN's current location additionally covering neighbor locations, where the MN may move to. Multicast-based schemes can be categorized into two subcategories, dense- and sparse-mode multicast-based approaches. Dense-mode protocols are suitable for densely populated groups. They construct source-based trees, i.e. a separate tree for each source to receiver pair. These protocols use flooding to reach multicast-trees members. The approach proposed in the scope of Daedalus project [SBK97] and the approaches presented in [MBh97] and [TPL99] are examples of these schemes. Sparse-mode protocols are suitable for sparsely populated groups. They use a shared tree for all members to and from a center point, also referred to as a core or rendezvous point. Explicit messages to join the center point are used by sparse-mode protocols. The Multicast for Mobility Protocol (MMP) [MSA00] is a well-known example.

Considering network-based mobility management protocols, the solutions of this category assume that the MN only provides minimal support for mobility or even no support. Thus, the network should execute all tasks related to mobility on behalf of the MN. Network-based mobility management solutions can be further classified into macro and micro mobility management solutions. While network-based macro mobility management solutions aim at global mobility support without involving MNs, network-based micro mobility management protocols localize the handoff processing inside an administrative domain to achieve fast handoffs without any interaction with MNs. Proxy MIPv6 [GLD08] is a well-known example of network-based macro mobility approaches, while Terminal Independent MIP (TIMIP) [GEN01] is an example of network-based micro mobility approaches.

2.4.3. *Transport Layer Mobility*

Mobility management in the transport layer tries to achieve end-to-end mobility management while keeping the Internet infrastructure unchanged by allowing end hosts to take care of mobility. Some transport layer mobility management approaches propose techniques to enable a connection migration from one point of attachment to another during or after the handoff. Other approaches are complete mobility solutions including handoff, connection migration and location management. It is, however, assumed that the MN is able to use the connection after the movement without requiring a reconnection with its communication partner. Transport layer mobility approaches can be classified into four categories [ARe05], namely handoff protocols, connection migration protocols, gateway-based mobility management protocols and mobility management protocols, see figure 2.6.

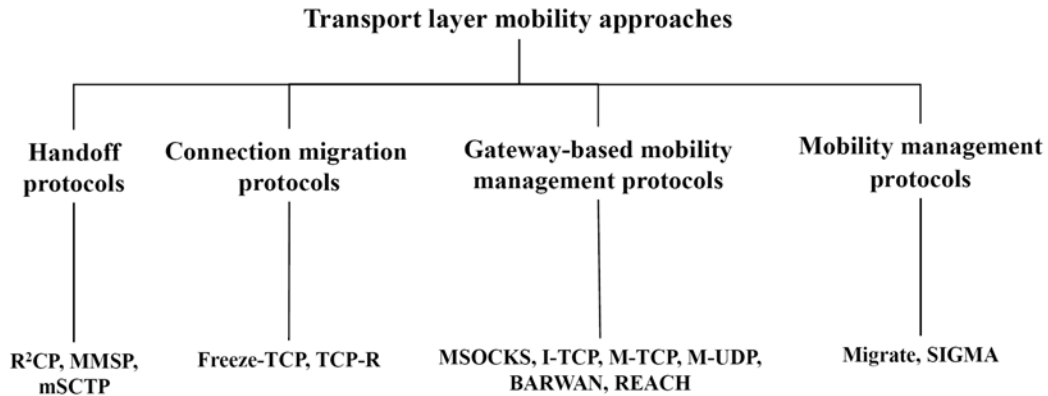


Fig 2.6: Classification of transport layer mobility management approaches

Handoff protocols aim at enhancing transport layer protocols to improve the performance, such as a low latency, less data loss during the handoff, etc. These protocols are not complete mobility management protocols. This is due to the lack of mobility management components, such as location management. Examples of these protocols are the Radial Reception Control Protocol (R²CP) [HKZ03], the Mobile Multimedia Streaming Protocol (MMSP) [MYO03] and mobile SCTP (mSCTP) [KCL04].

R²CP depends on a receiver-centric transport protocol named Reception Control Protocol (RCP) [HKZ03], which is a TCP clone in its general behavior. In other words, RCP is a copy of TCP with minimum changes since it supports improved congestion control, loss recovery and power management through moving the control of these issues from the sender to the receiver. R²CP adds features to RCP aiming at supporting seamless handoff and bandwidth aggregation using multiple interfaces.

MMSP supports multi-homing and soft handoffs using bicasting, i.e. multicasting to two nodes. It assumes that cells overlap and the MN has more than one interface, with an IP address assigned to each of them. When the MN moves into a new cell, the additional interface is assigned a new IP address in the overlapping area. It is assumed, however, that the first interface should retain a valid IP address. The Corresponding Node (CN)¹ is informed about the new IP address, which will be added to the destination address list maintained in the CN. Data packets are bicasted during this phase to the two IP addresses. After the handoff, the old IP address will be inactive. Therefore, the MN informs again the CN, which in turn deletes this address from its destination address list.

mSCTP enables CNs to add or delete an IP address to or from an existing association. Additionally, the primary IP address of an SCTP association can be changed as well. It is assumed that the MN acquires a new IP address while being in the overlapping area between two neighboring cells. The acquired IP address is sent to the CN, which adds it to the existing SCTP association. After the MN moves to the new cell, the newly acquired IP address becomes the primary IP address of the current SCTP association, while the old IP address becomes inactive and will be deleted from the SCTP association.

Connection migration protocols support the migration of connections that have been stopped or put under wait during the handoff. Examples of these protocols are Freeze-TCP [Sch03] and TCP Redirection (TCP-R) [FYT97]. Freeze-TCP supports a connection migration by halting an existing TCP connection during the handoff. This is achieved through advertising a zero window size to the CN. After finishing the handoff, Freeze-TCP unfreezes the TCP connection. Although this scheme reduces packets loss during the handoff, it produces high

¹ CN is the MN's communication partner

delay. TCP-R supports a connection migration by updating the source-destination address pairs of ongoing TCP connections upon a handoff.

Gateway-based mobility management protocols use a gateway mostly to split the connection between the MN and the CN into two. The first connection is between the CN and the gateway, while the second connection is between the gateway and the MN. Movements of the MN affect only the connections between the MN and gateway. Gateway-based mobility management protocols do not provide any details concerning the implementation of location management. Therefore, they do not provide complete solutions for the mobility management problem. Examples of these protocols are Indirect TCP (I-TCP) [BBa95], Mobile TCP (M-TCP) [HA97], Mobile UDP (M-UDP) [BSi96], Mobile SOCKeT (MSOCKS) [MBh98a], Bay Area Research Wireless Access Network (BARWAN) [KBA96] and Roaming-Enabled ArCHitecture (REACH) [ESe08].

I-TCP uses a gateway to split the TCP connection between the MN and the CN. When the MN moves to a new point of attachment, a new TCP connection between the gateway and the MN is established. However, the session between the CN and the gateway remains unchanged. M-TCP uses the same idea as I-TCP. The connection is split in an intermediate node called Mobile-Gateway into two connections. One is between the Mobile-Gateway and the CN (referred to as wireline segment), while the other one is between the Mobile-Gateway and the MN (referred to as wireless segment). What is new in M-TCP compared to I-TCP is the design methodology of the wireless segment. M-TCP is implemented with a different code complexity than I-TCP. More specifically, the higher complexity code is implemented in the Mobile-Gateway, while the lower complexity code is implemented in the MN. Such a design methodology is motivated from the fact that the wireless segment is in principle a single-hop connection. Consequently, many functions of the wireless segment can either be simplified or eliminated, e.g. the flow control will be executed in the Mobile-Gateway and not in the MN, etc. Data packets sent from the CN to the MN are redirected by the Mobile-Gateway. This redirection is unnoticed to both the MN and CN. M-UDP is an implementation of UDP with mobility support, similar to I-TCP and M-TCP.

MSOCKS uses the TCP splice proposal [MBh98] to support a connection migration. It supports multiple IP addresses for multiple interfaces. A proxy is used on the path between the MN and the CN. This proxy splits the TCP connection into two connections, a connection between the CN and proxy as well as between the proxy and MN. The handoff does only have an impact on the connection between the proxy and MN. Similar to MSOCK, BARWAN has a gateway-based architecture. It is mainly a solution for heterogeneous wireless networks. The main difference in BARWAN against the other gateway-based protocols is that the application is aware of mobility. The decision is even made by the application. REACH uses a gateway-based architecture as well. It mainly aims at supporting vertical handoffs and assumes, therefore, that MNs have multiple interfaces with different access technologies. Each MN operating REACH must implement software called REACH-client that communicates with corresponding software implemented on the proxy, also termed as REACH-server. The REACH-client offers many relay plugins responsible for intercepting data packets from applications. Following this, the intercepted data are sent either through stream-based or datagram-based logical links to the REACH-server, which takes care of further forwarding of data towards their destination. Stream-based logical links are established between relay plugins on the REACH-client as well as REACH-server. Moreover, these links are mapped to TCP connections between both REACH-client and REACH-server. Although TCP connections break down during handoffs, logical links remain unaffected. Furthermore, a TCP connection may be used to transfer data of many logical links. Datagram-based logical links use principles similar to those of stream-based logical links. However, they are mapped to UDP associations instead of TCP connections. REACH is capable of

supporting hard, soft and softer handoffs between different interfaces. Clearly, handoffs affect only TCP connections or UDP associations between the REACH-client and REACH-server, while TCP connections as well as UDP associations between the REACH-server and CNs remain unaffected. A main feature of REACH is that it requires updates neither to applications operating on MNs nor to components of the access network operating REACH. This is because all tasks are carried out by the plugins implemented in the MN as well as the proxy.

Mobility management protocols provide true solutions for end-to-end mobility. They support handoffs and location management. Examples of these solutions are Migrate [[SBa00](#)] and the Seamless IP diversity-based Generalized Mobility Architecture (SIGMA) [[FMA05](#)]. Location management in Migrate is supported using dynamic updates to a Domain Name System (DNS). A connection migration is supported by adding a new end-to-end TCP option, which enables the TCP peers to migrate an existing connection to another IP address. Similar to Migrate, SIGMA is a complete mobility management scheme too. It assumes that the MN is a multi-homed node connected through two wireless access networks. After the MN moves into an overlapping area between two cells, it obtains a new IP address for a second interface and informs the CN about this address. Thus, the CN redirects the traffic of the MN to the new IP address. The location of the MN is updated at a Location Manager (LM), which maps between the identity of the MN and the current IP address. After finishing the handoff, the IP address of the interface with the old access network becomes inactive and is deleted from the TCP connection.

2.4.4. Session Layer Mobility

Session layer mobility management solutions extend the current TCP/IP protocol stack by including a session layer, similar to that of the ISO/OSI reference model. Support of mobility in the session layer aims at delivering data to any device the user wants to use [[KMA03](#)]. The delivering of data is referred to as a service migration. Two main tasks must be achieved in any session layer mobility framework, namely session management and location management. When the user wants to use another device, a service migration should be initiated. This migration implies a transport of the session from one device to another. In other words, the session layer mobility management is applied to data streams rather than to mobile users. The Session Layer Mobility Management framework (SLM) [[LLI99](#)] is a well-known example of these approaches.

SLM divides into session management and location management part. The session management part is placed on end hosts. The connections within an end host are controlled by a session management entity. The path between two end hosts is split into three subpaths. The first connects the application on the first end host with a socket connector on the same end host. The second subpath connects the application on the second end host with a socket connector, also placed on this end host. The third subpath connects the connectors in the two end hosts. When a session should be handed over to a new end device, the session management entity on the current end host (caller host) notifies the session management entity on the new device. This results in creating a new connector and notifying the session management entity on the caller host, after that, to hand over the session. The location management part is in charge of localizing MNs. In order to do this, SLM uses a new network entity named a User Location Server (ULS), which stores current locations of MNs.

2.4.5. Application Layer Mobility

The main motivation behind mobility support in the application layer is to provide mobility support without changing the current network infrastructure. In order to do this, the

infrastructure of IP telephony is extended to fulfill mobility requirements. The solutions implemented in this layer can be broadly classified into two categories, as depicted in figure 2.7. The first category aims at supporting global application layer mobility, also referred to as macro mobility management. The second category tries to utilize the IP micro mobility concept to support local mobility management.

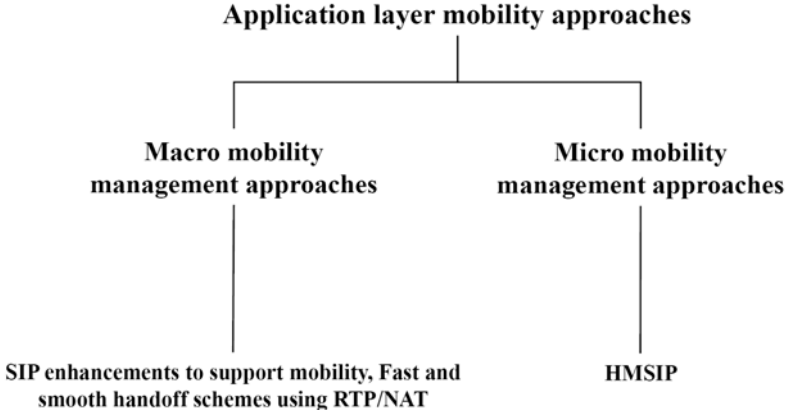


Fig 2.7: Classification of application layer mobility management approaches

Basic approaches belonging to the first category support mobility by using the Session Initiation Protocol (SIP) [HSS99], [SRo99], which allows two or more participants to establish a session consisting of multiple media streams. For this purpose, SIP binds the user identifier to a temporary IP address or host name [SWe00]. When the MN initiates a session, it sends an *INVITE* message to the CN, which acknowledges the request with an *OK* message. The MN acknowledges the *OK* by an *ACK* message and starts the session. When the MN moves to a new point of attachment during an active session, a handoff must be executed. The MN registers the newly acquired IP address with the home SIP server, referred to as home registrar. Additionally, the MN sends another *INVITE* message with the new IP address to the CN. This enables the CN to redirect the session to the new IP address, see figure 2.8. SIP is suitable for real-time applications. However, it is less suitable for TCP-based ones.

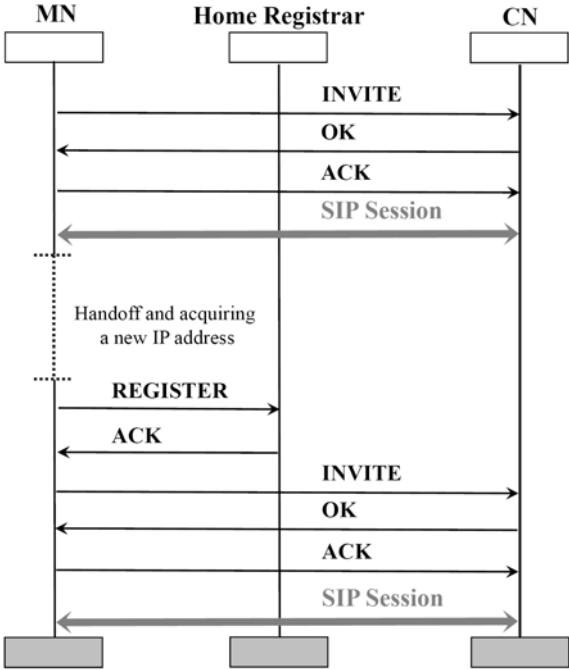


Fig 2.8: SIP-based mobility management

There are some proposals trying to enhance the SIP-based approach to achieve fast and smooth handoffs [DSM02]. The basic idea says that each subnet should be equipped with a Real Time Protocol (RTP) translator [SCF96] or a Network Address Translation (NAT). The RTP translator and the NAT provide mechanisms to enable forwarding of RTP packets, sent to a certain address and UDP port, to a new network address. When the MN moves to a new subnet in the SIP domain, it updates its binding at the home registrar. As known, the update message is sent to the home registrar via the registrar existing in the visited subnet. As a result, this registrar sends a request to the RTP translator of the old subnet. The request forces the old subnet's RTP translator to forward the MN's data packets to the new address.

Solutions of the second category try to use the IP micro mobility concept within SIP environments. Two IP addresses are assigned to the MN, a Local Address (LA) and a Domain Address (DA). The LA specifies the current point of attachment, whereas the DA specifies the address of the gateway controlling the domain. When the MN moves to a new domain, it sends an update message to its home registrar. This message is intercepted by the gateway, which writes the DA in the update message instead of the LA. The update message is forwarded further to the home registrar. If the MN moves inside the same domain, it has to update its LA at the gateway only. Hierarchical Mobile SIP (HMSIP) [VPK03], [VPK03a] is an example of these approaches.

2.4.6. Hybrid Approaches

The mobility management approaches described in the previous sections focus on a single protocol layer. Each layer has, however, positive and negative impacts on the mobility. This has resulted in developing new multi-layer mobility management approaches, which can be divided into two main categories.

The first category tries to optimize the performance of mobility management solutions in a certain layer by using information from other layers. A multi-layer mobility management architecture using cross-layer signaling interactions is proposed in [WAb03]. The key idea is to coordinate between layers to optimize the performance of mobility management approaches. The first layer of the TCP/IP reference model can report current channel conditions and link properties. This information can be used by the network layer to optimize the performance of its mobility management solutions. Low latency methods for MIPv4 [Mal07] and fast handoff methods for MIPv6 [Koo05] are examples of such solutions. They try to accelerate the network layer handoff by using lower layers information¹, referred to as layer 2 triggers. As described previously, terminal mobility can be supported by the network layer. However, with additional information from the application layer, such as the required bandwidth, the minimum delay, etc., a QoS-aware handoff can be achieved. Examples of these approaches are the approaches proposed in [FKK02] and [GRu05]. These approaches try to process the handoff simultaneously with a reservation of the required resources. The scheme presented in [FKK02] is built on HMIPv6 and enables the MN to flexibly choose among a set of APs, so that the AP with the best resources is selected. The approach presented in [GRu05] is built on MIPRR. It integrates this protocol with the resource ReSerVation Protocol (RSVP) [BZB97], so that resources are reserved during the handoff. In addition, many multimedia applications can become adaptive and mobility-aware if the application layer interacts with the network layer. An example is the approach presented in [PCA04] and

¹ This information is given to network layer mobility management solutions, in practical, from the second layer of the ISO/OSI reference model and from the first layer of the TCP/IP protocol suite. The information is considered, however, as layer 2 information in the literature for both reference models since the first and second layer of the ISO/OSI reference model are combined with each other to form the first layer of the TCP/IP protocol suite.

[PCT03]. It uses SIP and MIP to support mobility in IP-based networks. MIP is used for non-real-time applications and TCP-based ones, whereas SIP is used for real-time applications.

Considering the second category's solutions, mobility is supported in more than one layer. However, rather than a hybrid solution, an integrated one is implemented. In other words, the protocols of two or more layers are integrated with each other to replace inter-protocol signaling by intra-protocol signaling. The approach proposed in [WAb03] is an example of such a solution. It integrates SIP and MIP to support mobility in IP-based networks.

2.5. Conclusion

This chapter has introduced the basic scenarios for the usage of wireless access and has presented the definition and different classifications for the handoff. Moreover, the chapter has studied the implementation of mobility management in the layers of the TCP/IP reference model. The main results obtained from this study can be summarized as follows:

1. A movement from one point of attachment to another requires establishing a new wireless link with the new point of attachment. Link layer mobility is responsible for establishing such radio links. No additional procedures are required if the new and old points of attachment belong to the same subnet. However, if the subnet has been changed, additional mobility procedures have to be executed. These procedures can be supported by the network layer, transport layer or application layer or by a hybrid approach involving several layers.
2. Network layer mobility management approaches aim at achieving transparency to higher layer protocols. This is, however, at the cost of extensions to some network nodes. A main advantage of network layer mobility management approaches is that applications implemented on the MN can further communicate with other applications operating on other mobiles or fixed hosts regardless of the current location of the MN and without requiring any modifications to these applications.
3. Transport layer mobility tries to support end-to-end mobility management leaving the Internet infrastructure unchanged. The basic idea is to let the end hosts take care of mobility. Specifically, TCP or UDP should be updated in most transport layer mobility management protocols to support mobility. TCP has emerged due to its congestion control mechanism as the dominant transport layer protocol in wired networks. Extending TCP to work with wireless links is a very hard task. The reasons lie in the complexity of wireless transmissions and the need of any extended TCP implementation to properly interact with standard TCP implementations employed in the wired part of the network. A main disadvantage of mobility support in the transport layer is that most solutions proposed require an update of all hosts involved in the communication with MNs, which is impractical and negatively affects the applicability of such solutions.
4. Application layer mobility leaves the Internet infrastructure unchanged too. The basic idea lies in extending the current infrastructure of IP telephony to support mobility. However, implementing mobility in the application layer forces the applications to be mobility-aware. This requires updating the applications on all hosts communicating with MNs. This is, of course, a big disadvantage of these solutions and negatively affects their applicability.
5. Hybrid approaches allow for the optimization of the performance of mobility management protocols by cooperating between many layers. These solutions require, however, an accurate synchronization between the solutions of more than one layer to optimize the performance. In addition, they violate the separation between the layers.

The conclusion of our discussion is that network layer mobility management solutions seem to be most suited to satisfy the requirements of future All-IP networks. Optimizing the performance of network layer mobility management solutions by using information from other layers is promising too. Therefore, network layer mobility management will be focused on in the next chapter.

3. Network Layer Mobility Management

This chapter reviews previous efforts to support mobility in the network layer. It is organized as follows: sections [3.1](#) and [3.2](#) discuss known terminal-based and network-based mobility management approaches, respectively. Section [3.3](#) summarizes the main results of our review and provides a qualitative comparison of the described approaches. The motivation to develop a new layer 3 mobility management approach is given in this section as well.

3.1. Terminal-Based Mobility Management

As described in section [2.4.2](#), terminal-based mobility management protocols interact with the MN during the handover process. Two categories can be distinguished, macro and micro mobility management.

3.1.1. Terminal-Based Macro Mobility Management Approaches

Macro mobility management techniques aim at supporting global mobility. They require updating the location of the MN at its home network each time the MN changes its point of attachment. Although there are some solutions trying to support fast and smooth handoffs by means of some local processing, they are considered as macro mobility management solutions if the home network is notified of each change in the point of attachment.

3.1.1.1 Mobile IP (MIP)

MIP presents the Internet Engineering Task Force (IETF) standard protocol used to support mobility in IP-based mobile communication networks. It comes in two versions, MIPv4 and MIPv6.

3.1.1.1.1 MIPv4

MIPv4 introduces two new entities, namely a Home Agent (HA) and a Foreign Agent (FA). The HA is a router residing on the MN's home network, it authenticates and offers services to the MN while being inside or outside the home network. The FA is a router in the network being currently visited. The node communicating with the MN is referred to as CN. The CN may be a fixed or a mobile host. An example network operating MIPv4 is illustrated in figure 3.1. Each MN is assigned a unique IP address, i.e. a home address, acting as a static identifier for it. When a MN is away from its home, it obtains a temporary IP address, i.e. a Care of Address (CoA), identifying the current point of attachment. A CoA must be routable from elsewhere. It can be assigned to a certain interface of the MN, i.e. be a co-located CoA [[Dro97](#)], or can be provided by a local router serving the visited subnet, such as the FA. In this case, the CoA is called a FA-CoA. Each FA advertises its existence and properties through a periodic broadcast of Agent Advertisement messages (*Agnt_Adv*). Each *Agnt_Adv* message has a lifetime, which is typically three times the time period between two subsequent *Agnt_Adv* messages. After the MN operating MIPv4 moves to a new AP belonging to a different subnet and establishes a wireless link with it, a layer 3 handoff should be executed. The layer 3 handoff includes two phases, namely movement detection and registration phase.

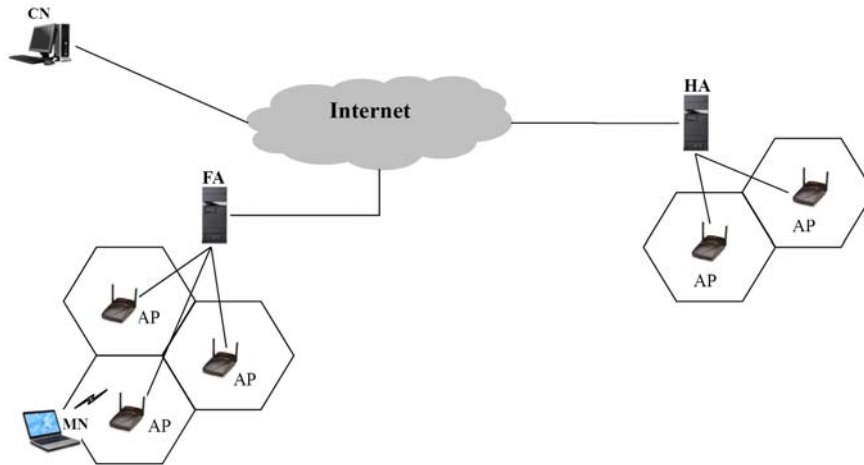


Fig 3.1: An example network operating MIPv4

a) Movement Detection

Movement detection denotes taking a decision that the current serving agent is not reachable any more and discovering the IP address of the new one serving the new subnet, which can be a FA when the MN is away from its home network or the HA when returning back to the home network. Movement detection techniques can be categorized as advertisement-based or hint-based.

Advertisement-based techniques depend on receipt of *Agnt_Adv* messages to detect a movement. Currently, there are two advertisement-based algorithms in use [Per98]. The first one is the Lazy Cell Switching (LCS) algorithm. It aims at avoiding a handoff until it is absolutely necessary. The MN tries to detect a new agent after the expiration of the last received *Agnt_Adv* message's lifetime. The second algorithm is called Eager Cell Switching (ECS) algorithm and operates in a way opposite to LCS. It assumes frequent location changes and, therefore, pursues a handoff immediately upon receiving a new *Agnt_Adv* message.

Hint-based techniques extract information from lower layers to speed up the movement detection [FGo01]. Well-known examples are the hinted and fast hinted cell switching algorithms. Hinted Cell Switching (HCS) depends on link layer information, termed as hints, to determine when a link layer handoff occurs. The MN broadcasts an Agent Solicitation (*Agnt_Sol*) message upon receipt of a link layer hint. *Agnt_Sol* messages force all adjacent FAs to respond with a unicast *Agnt_Adv* message. This enables the MN to determine the identity of the new FA regardless of the advertisement sending interval and lifetime. Fast Hinted Cell Switching (FHCS) expects to determine the new FA identity from information delivered with these link layer hints.

A comparison between advertisement-based and hint-based movement detection algorithms is introduced in [FGo01a]. The main results can be summarized as follows: hint-based algorithms outperform advertisement-based ones. A trade-off between the advertisement rate and the link efficiency should be made when employing advertisement-based algorithms. This trade-off is not required, however, employing hint-based algorithms. The FHCS algorithm performs best compared to LCS, ECS and HCS. It produces a minimal interruption and a stable performance in different movement patterns.

b) Registration

This phase comprises the steps required to notify the HA of the newly acquired CoA. The details are given depicted in figure 3.2. After the MN detects an out of coverage and is

assigned a new CoA, it sends a Registration Request (*Reg_Rqst*) message to the new FA, which in turn records the MN in its visitor list and forwards the *Reg_Rqst* message to the HA. The HA authenticates this message to ensure that it is originated from the MN. If the authentication succeeds, the HA records the MN's CoA in its binding list, generates a Registration Reply (*Reg_Rply*) message and transmits it to the MN. Each registration has a certain lifetime. Expiration of this lifetime forces the MN to re-register with the HA by exchanging the messages described above. To authenticate the MN during the registration process, a shared Security Association (SA) between the MN and the HA should be established. This SA should be distributed off-line and used in cooperation with an authentication algorithm, such as MD5 [Riv92] or HMAC-MD5 [KBC97].

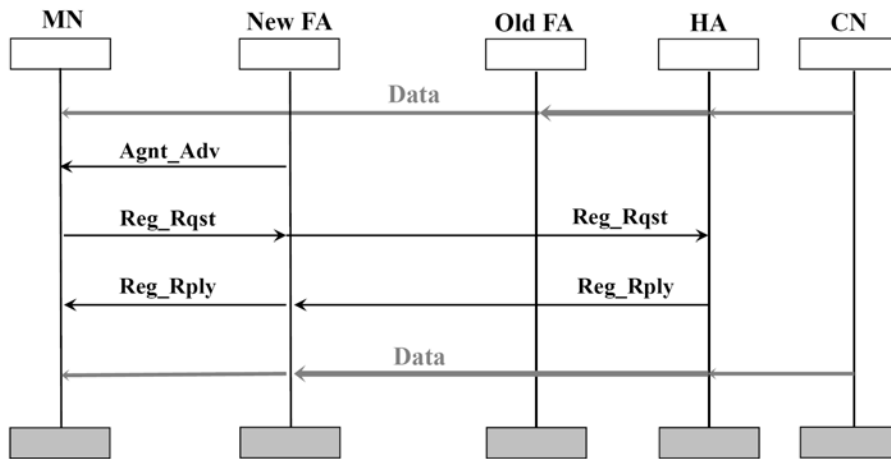


Fig 3.2: MIPv4 handoff procedure

When a CN wants to communicate with a certain MN, it transmits data packets to the MN's home address. The packets are routed to the HA depending on the subnet address. If the MN is away from home, the HA intercepts the MN's data packets, encapsulates and forwards them to the new MN's CoA. Encapsulated data packets will be decapsulated by the FA and forwarded to the MN in case the MN has a FA-CoA. If the MN is assigned a co-located CoA, the tunnel end point will be the MN itself. The standard encapsulation method is IP-in-IP encapsulation [Per96]. Due to the big overhead caused by this method, other encapsulation methods may be preferable, such as minimal encapsulation [Per96a] or Generic Routing Encapsulation (GRE) [FLH00].

3.1.1.1.2 MIPv6

In general, similar assumptions to those of MIPv4 are assumed. The difference is that MIPv6 does not need special routers with mobility support in visited subnets. Regular IPv6 Access Routers (ARs) are sufficient. After the MN moves into the range of a new AR, it waits for a Router Advertisement (*RA*) message and configures its CoA after the receipt of the *RA* message either in a stateful [DBV03] or in a stateless mode [TNa98]. Alternatively, the *RA* can be solicited by sending a Router Solicitation (*RS*) message. After the movement and after acquiring a new CoA, the MN notifies its CN as well as HA of the new CoA. To do this, the MN sends a Binding Update (*BU*) message to the CN and another *BU* message to the HA. A Binding Acknowledgement (*BA*) message should be sent from the CN as well as the HA to the MN as a response, see figure 3.3. When a CN wants to communicate with a certain MN, it first checks for a valid binding for the MN. If a valid binding is found, packets are transmitted directly towards the MN. The CN uses the CoA as a destination address in the IPv6 header, while the MN's home address is inserted into the IPv6 routing header, see [JPA04]. Once the MN receives the packets, it retrieves its home address from the routing header and uses this

address as the final destination address for the packets. If the CN has found no or invalid binding for the MN, data packets will be transmitted to the HA, which in turn tunnels them to the current CoA.

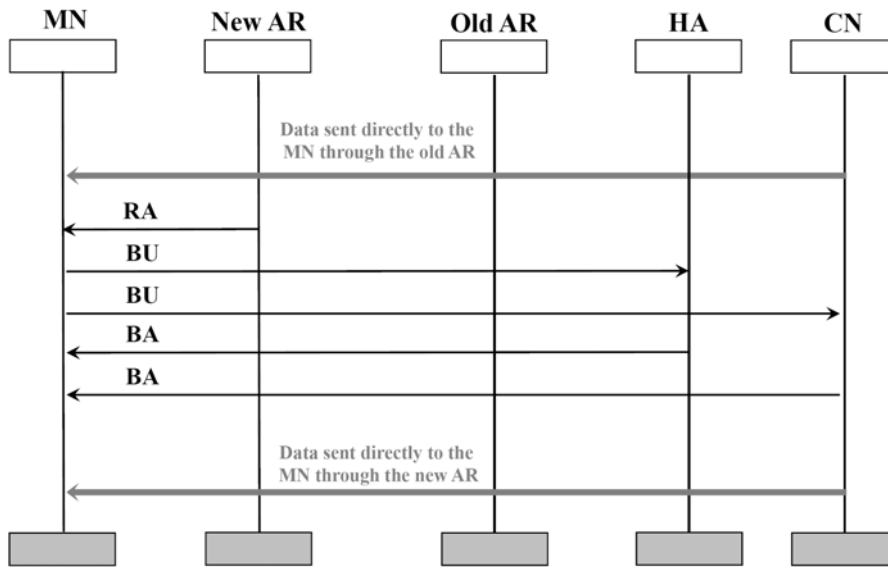


Fig 3.3: MIPv6 handoff procedure

In order to protect the integrity and the authenticity of *BUs* and *BA*s exchanged between the MN and the HA, an IPsec SA (IPsec-SA) [KA98], [IPSec] should be established between them. As a default, they use the Encapsulating Security Payload (ESP) [Ken05] header in transport mode. Additionally, they have to use a non-null payload authentication algorithm to provide data origin authentication, connectionless integrity and optional anti-replay protection. The protection of *BUs* and *BA*s exchanged between the MN and the CN does not require any existing configuration of SAs or any authentication infrastructure. Instead, the return routability procedure is used to generate a binding management key to protect control messages exchanged between the two nodes. The return routability procedure limits potential attackers to those having access to the specific path in the Internet between the home network and the CN. It disables attacks from forged *BUs* from anywhere else in the Internet. Protection of data packets is achieved through the use of IPsec in the same way as by stationary hosts.

MIPv6 introduces the home address destination option, a routing header and tunneling header in the payload. It should, therefore, prevent that these headers are used for attacks. In order to avoid the using of the home address destination option in attacks, MIPv6 limits using of this option to situations, in which the CN has a valid entry for the given MN. Due to using a new type (type 2) for the routing header, this header does not open any new vulnerability. Tunnels between the HA and the MN are protected by a proper use of source addresses and optional cryptographic protection. On one side, the MN verifies that the outer IP address in the packets received from the HA corresponds to its HA. On the other side, the HA verifies that the outer IP address in the packets received from the MN corresponds to the current CoA. In addition, the HA identifies the MN through the examination of the source address present in the inner packet, which is the MN's home address.

3.1.1.1.3 Comparison between MIPv4 and MIPv6

The main differences between MIPv4 and MIPv6 can be summarized as follows:

1. The pool of IP addresses that can be managed by MIPv6 is huge since it manages IP addresses of 128 bits. In contrast, MIPv4 assigns IP addresses of 32 bits only.
2. MNs supporting MIPv4 require the Dynamic Host Configuration Protocol (DHCP) [Dro97] to configure co-located CoAs, whereas MNs operating MIPv6 configure their co-located CoAs automatically using the stateless mode. Of course, DHCP can be applied too.
3. MIPv4 requires supporting mobility in FAs even if co-located CoAs are assigned to MNs. In contrast, MIPv6 does not require any mobility support in ARs.
4. Route optimization is supported by an extension to MIPv4. With MIPv6, this feature is a standard part of the specification. In addition, the MIPv6 route optimization can operate securely even without pre-arranged SAs, which is not supported by the MIPv4 route optimization. Furthermore, the route optimization feature of MIPv6 can coexist with ARs implementing “ingress filtering” [FSe00]. This feature in MIPv4 has, however, problems in working with FAs operating “ingress filtering”.
5. MIPv4 needs IPsec to secure the data communication. This protocol is already integrated, however, within the standard MIPv6.
6. With standard MIPv4, mobility is transparent to CNs. In contrast, CNs are aware of mobility employing MIPv6.
7. With MIPv4, most data packets are tunneled to the MN while a way from home. Opposed to this, the IPv6 routing header is used rather than encapsulation when employing MIPv6. This reduces the forwarding overhead.

3.1.1.1.4 *Shortcomings of MIP*

MIP suffers from many shortcomings making it inadequate for real-time applications. These drawbacks can be briefly summarized as follows:

1. **Triangular routing:** data packets are typically sent from the CN to the MN’s home network, where the HA intercepts and forwards them to the current CoA. This results in sub-optimal routing, known as triangular routing, and increases the end-to-end delay. The measurements presented in [ZCB01] have shown that MIP increases the end-to-end delay by 45 % in a campus network. This negatively affects the offered quality, especially if the MN is far from the HA and close to the CN.
2. **Encapsulation:** the encapsulation of data packets adds more overhead to the traffic. This overhead is remarkable especially for real-time applications with small packet sizes, e.g. as with VoIP. [Fes03] has shown that for a voice codec G.723.1 [HGP00], the IP-in-IP encapsulation increases the packet size by 33 % (voice codec data rate: 5.3 kb/s with a frame size of 20 bytes, IPv4 protocol overhead resulting from encapsulation: 20 bytes, UDP: 8 bytes, RTP: 12 bytes).
3. **Handoff latency:** the handoff latency results from the delay due to movement detection and from the time for registration. The MN has to contact the HA and possibly the CN each time it changes its point of attachment. Clearly, there will be a communication disruption during the handoff. As the user mobility increases, necessary frequent binding updates produce unacceptable disruptions, especially if the FAs or ARs are far away from the HA and the CN.
4. **Signaling:** frequent binding updates at the HA and possibly the CN due to handoffs cause a considerable load resulting from control messages, especially when MNs move at high speeds.

5. **Ingress filtering:** ingress filtering is a security mechanism in routers to check for topological incorrect IP addresses. As known, the MN operating MIPv4 uses its permanent home address as a source address for uplink traffic. These packets will be determined as incorrect packets and discarded due to their non-network conforming source address. A possible solution for this problem can be a reverse tunnel [Mon01], [RBa98] from the FA to the HA for uplink traffic. This results, however, in a sub-optimal routing path. In addition, the MN has to know if there is an ingress filter in the current FA or not. Conversely, MNs operating MIPv6 do not need to bother about the ingress filtering problem. This is due to the fact that MIPv6 uses the acquired CoA as a source address for uplink traffic.

The discussed shortcomings have triggered the development of additional extensions to the standard MIP as well as new mobility management solutions for IPv4 and IPv6 networks.

3.1.1.2 Route Optimization Extension for MIPv4

The route optimization extension [PJo01], [PWa99] enables CNs to maintain a binding cache containing the CoA for one or more MNs. When a CN wants to transmit data packets to a MN, it checks first for a valid entry for the MN in its binding cache. If such an entry exists, the CN tunnels the packets directly to the CoA. If no valid entry is found, the CN transmits data packets to the MN's home address. These data packets will be intercepted by the HA and tunneled to the CoA. Subsequently, the HA notifies the CN of the MN's CoA by means of a **BU** message. The CN records the CoA in its binding cache and tunnels data packets directly to the CoA. The **BU** message does not require to be acknowledged. This is because the CN will send data packets to the HA until it receives a **BU** message. There must be, however, a SA between the CN and the HA to secure the exchanged **BU** messages.

If any FA receives a packet for a MN that does not locate in its range, it sends a Binding Warning (**BW**) message to the HA, which informs the CN about the new CoA. The messages exchanged during the route optimization procedure are presented in figure 3.4.

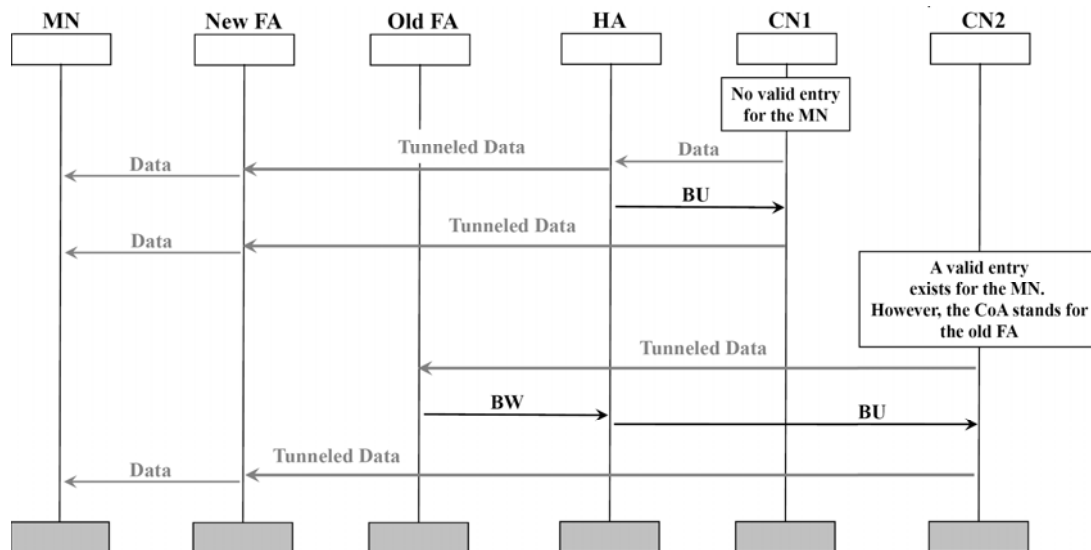


Fig 3.4: Route optimization procedure

The route optimization extension provides means to notify the MN's previous FA of the new CoA, so that the in-flight packets are forwarded to the new CoA, also called smooth handoff. The MN includes a previous FA notification extension in the **Reg_Rqst** message sent to the new FA. The new FA constructs and sends then a **BU** message to the previous one. Although the **BU** message is constructed by the new FA, the data existing in it should be generated and

authenticated by the MN itself. The **BU** message must be acknowledged. Additionally, the MN is responsible for recovering the dropping of it. Figure 3.5 shows the messages exchanged during the smooth handoff procedure.

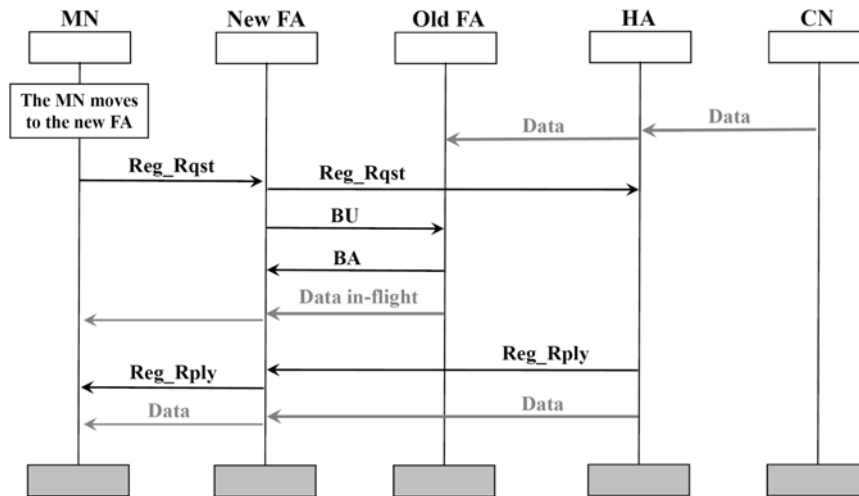


Fig 3.5: Smooth handoff procedure

Route optimization extension eliminates the problem of triangular routing and reduces the number of lost packets during the handoff. However, mobility is not transparent to CNs any more. The HA, FAs, MNs and CNs should be updated to support the route optimization extension. In addition, the HA should have SAs with CNs supporting this extension. This affects negatively the applicability of this approach.

3.1.1.3 Pre- and Post-Registration Methods for MIPv4

Low latency handoff methods proposed in [Mal07] aim at accelerating the handoff by utilizing information resulting from lower layers, e.g. signal strength, signal to noise ratio, etc., to anticipate the network layer handoff, termed as layer 3 handoff too, prior to a break of the current radio link. The information produces indicators called layer 2 triggers. There are three types of these triggers, which are: Layer 2 trigger (L2-trigger), Layer 2 Link Down trigger (L2-LD trigger) and Layer 2 Link Up trigger (L2-LU trigger). The appearance of a L2-trigger means that a handoff will occur in the near future. The MN may fire this trigger when the measurements achieved by the first layer indicate that the signal strength goes below a certain threshold for example. It is assumed that the L2-trigger contains the IP address of the new FA or another address, from which the IP address of the new FA can be derived, e.g. a MAC address. Not only the MN can fire a L2-trigger, but also the current FA or even the new one, as described in [Mal07]. Release of a L2-LD trigger indicates that the radio link with the old FA has just been broken, while firing a L2-LU trigger denotes that the radio link with the new FA has just been established.

The proposed methods are pre-registration, post-registration and a combined method. The appearance of a L2-trigger at the MN prompts it to register with the new FA via the old one when employing the pre-registration method. To do this, the MN sends a Proxy Router Solicitation (**PrRtSol**) message to ask the current FA for sending an advertisement (Proxy Router Advertisement (**PrRtAdv**)) on behalf of the new FA. Upon the receipt of the advertisement by the MN, a **Reg_Rqst** is sent to the new FA via the old one. Subsequently, the MN executes a layer 2 handoff. This prompts a L2-LD trigger at the old FA, which prompts the old FA to forward the MN's data packets directly to the new FA. The **Reg_Rply** is sent to the MN on the new as well as the old link to ensure the arrival of this message.

Figure 3.6 shows the messages exchanged during the handoff employing the pre-registration method.

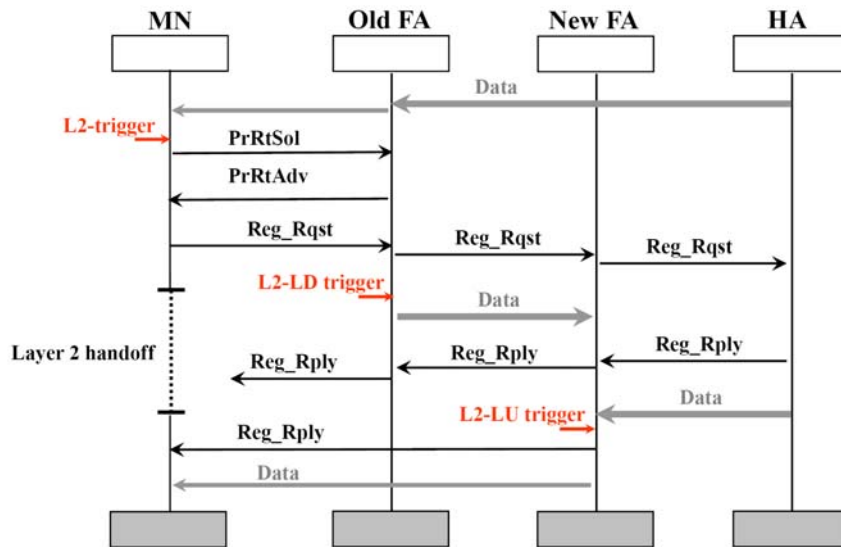


Fig 3.6: Handoff procedure employing the pre-registration method

Considering the post-registration method, the MN executes a layer 2 handoff only after a L2-trigger has been fired, see figure 3.7.

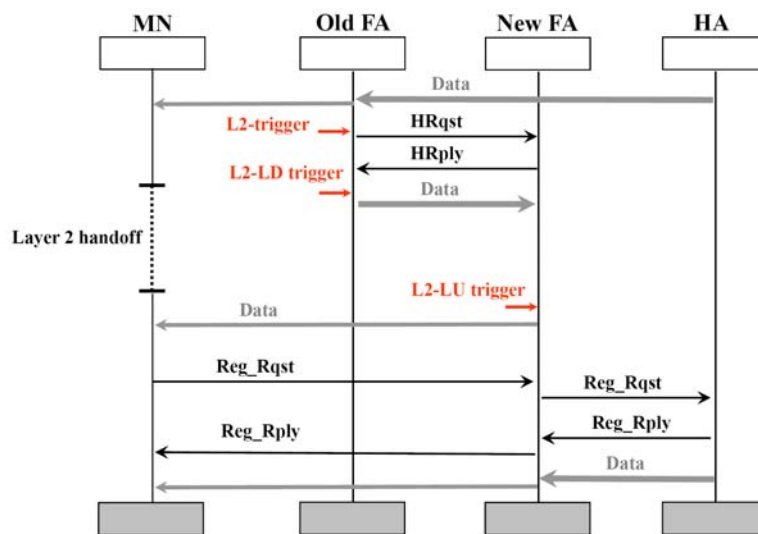


Fig 3.7: Handoff procedure employing the post-registration method

On the network side, a bidirectional tunnel is established between the old FA and the new one. This tunnel is established by exchanging a Handoff Request (*HRqst*) and a Handoff Reply (*HRply*) message between the old and the new FA. When a L2-LD trigger is raised at the old FA, it forwards data packets destined to the MN to the new FA. After execution of the layer 2 handoff, the MN can register with the HA via the new FA while receiving its data packets.

Considering the combined method, the MN tries to use the pre-registration method first, after a L2-trigger is raised. If this fails, the MN employs the post-registration method as described above to ensure a smooth handoff.

Performance studies of pre- and post-registration methods have shown that the timing of the triggers has a major influence on the handoff latency as well as the packet loss rate [BCC03], [CCW03], [BCC03a]. Increased latency results if the L2-trigger for the pre-registration method is delayed. In case the *Reg_Rqst* message gets lost, it is possible that this method

resorts to the standard layer 3 handoff approach, e.g. MIPv4. Even though, employing the post-registration method enables the MN to resume its communication faster than the pre-registration one, delayed triggers have a similar impact as with the pre-registration. As the combined method attempts to employ the pre-registration method first before prompting the post-registration method in case of failure, the combined method inherits the problems of both pre- and post-registration.

In order to reduce the negative impact of the timing of the triggers, an improved approach has been proposed in [PJa02]. In this approach, the MN informs the old FA that a movement will take place and registers with the new FA via the old one, similar to the pre-registration approach. However, the old FA forwards the packets directly to the new FA without waiting for the L2-LD trigger. This reduces the negative impact of layer 2 triggers timing compared to the other methods.

3.1.1.4 MosquitoNet Extensions

The MosquitoNet group has presented two extensions to use MIPv4 more efficiently and more flexibly [Fes03], [ZCB01]. The first mechanism supports multiple packet delivery methods. The adequate delivery method is selected according to the characteristics of each traffic flow. In other words, the MN decides, whether to use the transparent mobility, i.e. triangular routing, support or not. The overhead resulting from the transparent mobility support should be avoided unless it is absolute necessary. In case the transparent mobility support is required, the MN should decide to use triangular routing either employing reverse tunneling, also referred to as bidirectional tunneling, or not. The triangular routing is adequate for incoming traffic initiated by CNs without route optimization support, while bidirectional tunneling is suitable to communicate with FAs operating ingress filters. In order to select the adequate delivery method, a mobile policy table must be implemented in the MN's network layer. The IP route lookup procedure should be updated and has to take this table into account.

The second extension proposed by the MosquitoNet group enables the MN to use multiple interfaces simultaneously. Each interface carries a temporary IP address, to which a flow is bound. The HA in this case has to support multiple CoAs for a single MN. This method is useful for vertical handoffs between different access networks. The same registration procedure employed by MIPv4 is used here too. However, in order to enable the MN to register a certain flow to a certain interface, two new extensions are added to the *Reg_Rqst* message of MIPv4. These extensions are: a flow-to-interface binding and a flow-to-interface binding update extension. The flow-to-interface binding extension is required to register a certain flow to a certain CoA, while the flow-to-interface binding update extension is used to request the HA to redirect an existing flow binding to a different CoA.

MosquitoNet extensions eliminate the ingress filtering problem. The problem of triangular routing is limited to situations requiring transparent mobility support. However, the encapsulation overhead, the long handoff latency and the large signaling overhead remain unsolved.

3.1.1.5 Fast MIPv6 (FMIPv6)

FMIPv6 [Koo05] accelerates the handoff procedure by utilizing layer 2 triggers too. It aims at detecting the new AR and configuring the new CoA while still being connected to the old AR. This allows MNs to reestablish their IP connectivity direct after the layer 2 handoff. The basic operation of FMIPv6 depends on establishing a bidirectional tunnel between the old and the new AR upon firing the L2-trigger. Downlink data packets are tunneled to the new AR via the old one during the layer 2 handoff and until the MN updates its new CoA at the HA and the

CN. Uplink data packets are tunneled after the layer 2 handoff from the new AR to the old AR until the new binding is updated. The old AR is responsible for further forwarding of the packets. FMIPv6 can be operated in two modes, namely a predictive and a reactive mode. The predictive mode is the default operation mode, the MN tries to detect the new AR from the L2-trigger before the layer 2 handoff occurs and, thus, starts the layer 3 handoff in advance. If the MN fails to employ the predictive mode, it performs the layer 2 handoff and employs the reactive mode after that.

The operation of FMIPv6 in predictive mode is shown in figure 3.8. When the MN notices that it has to do a handoff, it sends a Router Solicitation Proxy message (*RtSolPr*) to the old AR. *RtSolPr* should contain some information, from which the old AR can determine the IP address of the new one. As a response, the old AR sends a *PrRtAdv* message to the MN containing enough information, so that the MN can configure its new CoA. The old AR can force the MN to make a handoff through sending an unsolicited *PrRtAdv* with a new subnet-prefix too. After the configuration of the new CoA, the MN sends a Fast Binding Update (*F-BU*) message to the old AR confirming the handoff. As soon as the old AR receives the *F-BU* message, it sends a Handover Initiate (*HI*) message to the new AR to notify it of the incoming MN. The *HI* message should contain the current CoA being used and the new one being requested (new CoA). The new AR first checks if the assigned CoA is valid and not duplicated. A Handoff Acknowledgement (*HAck*) message is sent to the old AR to inform it if the new CoA is valid or not. If the new CoA is valid, the old AR will get ready to forward the MN's data packets to the new CoA. Otherwise, if the new CoA is not valid, the old AR forwards the MN's data packets to the new AR only, which in turn forwards them to the MN using the old CoA. In order to secure the exchange of control messages between the old and the new AR, FMIPv6 requires a trust between the two ARs. How this SA is established is left open in the FMIPv6 specification and is up to the network operator. In order to inform the MN that the layer 3 handoff has been successfully prepared in the new AR, a Fast Binding Acknowledgement (*F-BAck*) message is sent to it through both the old and the new AR. After the MN finishes the layer 2 handoff, it sends a Fast Neighbor Advertisement (*F-NAdv*) to prompt the new AR to resume the flow of packets that may wait in its buffer.

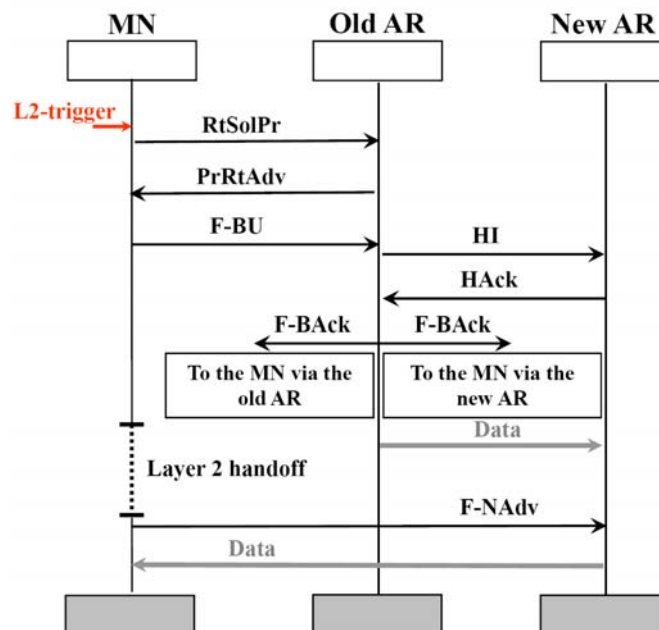


Fig 3.8: FMIPv6 operation in predictive mode

Figure 3.9 presents the operation of FMIPv6 in reactive mode. As mentioned above, this mode is selected when the MN fails in using the predictive mode and could only exchange

RtSolPr and *PrRtAdv* messages with the old AR. After the MN finishes the layer 2 handoff, it sends a *F-NAAdv* message including a *F-BU* message to the new AR. As a response, the new AR exchanges a *F-BU* and a *F-BAck* message with the old AR to verify the MN and to establish a tunnel between the old AR and the new one. In parallel, a *RA* message is sent to the MN, which resumes its communication after a successful exchange of *F-BU* and *F-BAck* messages.

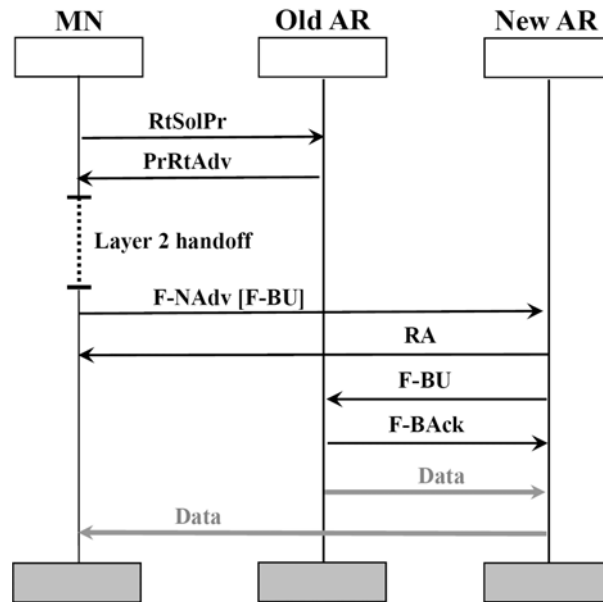


Fig 3.9: FMIPv6 operation in reactive mode

Performance studies have shown that FMIPv6 improves the performance significantly. The predictive mode can even eliminate the layer 3 handoff latency. This operation requires, however, a suitable buffer size to buffer the MN's data packets during the layer 2 handoff. The efficiency of this protocol is clearly reduced if it reverts to reactive mode. Due to the dependency on layer 2 triggers, FMIPv6 suffers from the same problems of the pre- and post-registration methods. The analytical studies presented in [Get08] have shown that there is a high probability for FMIPv6 to operate in predictive mode at reasonable speed of the MN (up to 50 km/h in this study). The MN speed and the time the MN spends inside an overlapping area have a significant impact on the performance. Increasing the MN speed or reducing the lengths of the MN's paths inside overlapping areas will reduce the probability of FMIPv6 operation in predictive mode.

3.1.1.6 Paging MIP (P-MIP)

P-MIP [ZCC02], [ZCC00] defines a set of paging extensions for MIPv4. This work is motivated from the fact that paging enables the handling of a large population of MNs and improves the scalability. Support for paging reduces the signaling overhead associated with location updates and reduces the power consumption of MNs due to the fact that idle MNs do not have to register their accurate location. P-MIP introduces paging areas, for each a Paging Area ID (PAI) is assigned. Paging areas can be configured manually by an administrator or dynamically by an interaction between FAs and paging servers. The *Agnt_Adv* message is extended to include the PAI and a **P** bit to indicate support for paging. In order to indicate that a MN supports paging, the *Reg_Rqst* message is extended to include the **P** bit too.

When an idle MN moves to a new paging area, it has to register with the HA. When packets destined to a certain idle MN reach the HA, it forwards them to the last known FA, referred to as a registered FA. This FA checks for a valid entry for the MN. If a valid entry is found, the

registered FA checks if the MN supports paging or not. In case paging is supported, a paging request is sent to all FAs in the current paging area and broadcasted in the network of the registered FA itself. As soon as the MN receives a paging request, it transits itself to the active state and registers with the HA. After completion of the registration, the MN sends a paging reply back to the registered FA, which forwards the buffered packets to the MN.

The simulation results presented in [ZCC02] show that the number of cells in the paging area has a significant impact on the performance. If the paging area size is smaller than a certain threshold, paging will improve the performance. However, the performance is degraded for paging area sizes bigger than this threshold. Therefore, a good planning of paging areas is essential. In addition, for MNs moving at high speeds, P-MIP performs very well. However, for slow MNs, i.e. at a speed smaller than 5 m/sec in this study, MIP performs better than P-MIP. A main disadvantage of P-MIP is the performance degradation for MNs having a very high session data rate. This is due to the dropping of data packets during the paging procedure. Obviously, the number of dropped packets is proportional to the packet arrival rate.

3.1.1.7 *Proactive Handoff Approaches*

Proactive handoff approaches aim at predicting the candidate subnets that may serve the MN in the future. The handoff is prepared in these candidate subnets in advance before the actual handoff takes place. Some handoff approaches that can be considered as predictive ones have been already explained, namely the pre- and post-registration methods and FMIPv6. Other known proactive approaches will be described shortly in this section.

The approach proposed in [SWC02] introduces a neighbor table maintained in each FA. The MN transfers information about its old FA to the new one. This helps the new FA in maintaining its neighbor table. When the MN recognizes that a handoff may occur in the near future, it informs the old FA to duplicate its packets to all neighboring FAs. After finishing the layer 2 handoff, the MN starts receiving data packets from the new FA and proceeds with the layer 3 handoff. Clearly, this approach reduces the handoff latency and number of lost packets to those resulting from the layer 2 handoff at the cost of extra traffic between FAs.

A Mobile-Initiated Tunneling Handoff mechanism for IPv4 (MITHv4) has been introduced in [GFJ03]. The MN sends a handoff request to the old FA upon detecting that a handoff will take place in the near future. This message triggers the setup of a bidirectional tunnel between the old and the expected new FA, which forwards data packets received from the old FA to the MN after the layer 2 handoff and until the MN completes the MIPv4 handoff procedure. This technique achieves low latency handoff with fewer requirements on layer 2 triggers and access networks than other methods [FRe04]. However, this requires exact knowledge of future FA.

The FA-assisted handoff technique described in [CHK00] proposes extensions to MIPv4 that enable the FA to send handoff messages on behalf of the MN. This technique supposes, however, that layer 2 triggers are available and the IP address of the new FA can be derived from these triggers. Upon firing of a L2-trigger, a *handoff request* and a *handoff reply* message are exchanged between the old and the new FA. Following this, the new FA sends a *Reg_Rqst* message to the HA on behalf of the MN. The HA responds with a *Reg_Rply* message, duplicating the MN's data packets and transmitting them to the old as well as to the new FA until the handoff is finished. This technique requires updating of MIPv4 in the HA as well as FAs.

A proactive handoff method with motion prediction is proposed in [FRe04]. Each MN records a movement history cache to be used for the prediction of the future movement. When the MN is at the boundary of a certain subnet, a L2-trigger indicating a handoff in the near future

is fired. The MN uses a path prediction algorithm to decide to which subnet or subnets it may move, see [FRe04]. The MN sends the CoAs of the discovered FAs in a *forwarding request* message to the current FA, which replies by duplicating data packets and sending them on its wireless network and to the other predicted FAs. After the layer 2 handoff, the MN registers its CoA with the HA using MIPv4. In order to stop the duplication of data packets, the HA sends a *forwarding stop* message to the old FA after the new CoA has been registered. The new movement of the MN is recorded in the history cache to be used by the path prediction algorithm in the future.

A proactive mobility framework for MIPv6 is proposed in [PKi01]. This method supposes that the MN's mobility is nondeterministic in terms of speed and direction. This means that the MN is not aware, where it may move in the future. Therefore, a mobility neighborhood and a routing neighborhood are defined. The mobility neighborhood depends on the geographical location and is defined by the APs, to which the MN may move in the future. The routing neighborhood determines the Routing Domain Controllers (RDCs) controlling the APs located in the mobility neighborhood. Figure 3.10 depicts the mobility and the underlying routing neighborhood. RDCs can be compared with ARs, known from MIPv6, with some additional functions.

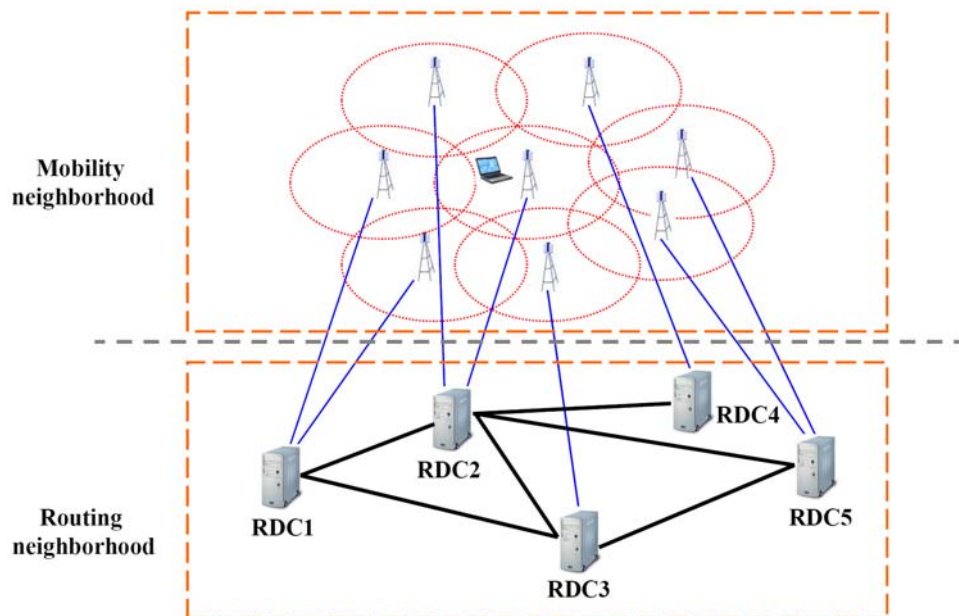


Fig 3.10: Mobility and underlying routing neighborhood

Initially, the MN powers up and registers itself with the HA. Upon completion of the registration, the current RDC generates a Tuple of Soft unicast CoAs (referred to as SCoAT) and sends it to the MN. The SCoAT comprises unique IPv6 addresses, each of which is topologically correct in the range of one of the current RDC neighbors. After that, the SCoAT is mapped onto a single multicast Proactive CoA (PCoA). All RDCs should be, therefore, multicast enabled. The mapping between SCoAT and PCoA is transparent to the HA and the CNs. After that, the current RDC informs the neighboring RDCs to enable the forwarding of the traffic destined to the multicast PCoA group.

When the MN reaches some overlapping areas, it listens to advertisements sent from candidate RDCs over the all-node multicast address. Depending on the received advertisements, the MN enables one or more CoAs from the SCoAT. Afterwards, the MN notifies the current RDC of these candidates by sending a PCoA-Enable (*PCoA-E*) message. The current RDC stops sending on its local network and sends data packets to the MN encapsulated into multicast packets with the PCoA as a destination. After the MN finishes the

layer 2 handoff, it sends a PCoA-Disable (*PCoA-D*) message to the previous RDC via the new one to deactivate the PCoA in terms of forwarding towards the MN.

Proactive handoff methods are promising and can reduce the handoff latency to the latency resulting from the layer 2 handoff. However, these methods produce a significant overhead due to the transmission and processing cost for newly added control messages, the duplication and forwarding of data packets, the movement prediction and the detection of new candidates. The dependence on layer 2 triggers itself produces additional problems, e.g. timing of triggers, power consumption due to doing measurements to detect the layer 2 trigger, etc.

3.1.2. Terminal-Based Micro Mobility Management Approaches

As discussed in section [2.4.2](#), micro mobility management techniques aim at reducing the time required to register with the network. The basic principle says that the MN's mobility should be locally processed as long as the MN moves inside the same administrative domain. We can distinguish between two main categories for these techniques, namely PAA and LERS, see [[EMS00](#)].

PAA-based approaches extend the MIP principle to process the mobility locally, e.g. by using hierarchical network architecture and/or intermediate nodes to locally control MNs movements inside the domain, etc. As the MN moves inside the domain, it updates its mobility binding only at the intermediate node, which tunnels data packets towards the MN.

LERS-based approaches introduce mostly a new dynamic layer 3 routing protocol inside the domain or depend on multicast. The domain's nodes maintain routings entries for MNs. The routing entries are actualized according to MNs movements inside the domain. Instead of tunneling data packets towards the current point of attachment inside the domain, as PAA-based approaches do, data packets are either forwarded hop per hop towards the MN or multicasted to a multicast group, to which the MN has joined. The following summarizes the well-known approaches of PAA and LERS categories along with highlighting their pos and cons.

3.1.2.1. Proxy Agent Architecture (PAA)

As mentioned above, PAA-based approaches extend the MIP principle to provide local mobility processing, e.g. by using a hierarchical network topology, introducing new intermediate nodes, etc. This section summarizes the well-known PAA-based approaches focusing on how they achieve the local processing of mobility along with a discussion of their pros and cons.

3.1.2.1.1. Regional Registration for MIPv4 (MIPRR)

MIPRR [[FJP07](#)] is a domain-based approach used to support micro mobility management in IP-based networks. It uses a hierarchical tree-like structure of two or more levels. FAs form the undermost hierarchy level. A Gateway Foreign Agent (GFA) is located at the top of the hierarchy and controls the domain. Agents with mobility support residing between the FAs and the GFA are called Regional Foreign Agents (RFAs). As a default, each FA advertises its CoA and the CoA of the GFA. When the MN moves into the domain controlled by a GFA, it registers the CoA of the GFA with the HA through exchanging normal *Reg_Rqst* and *Reg_Rply* messages as shown in figure 3.11. Movements between different administrative domains are referred to as inter-domain mobility.

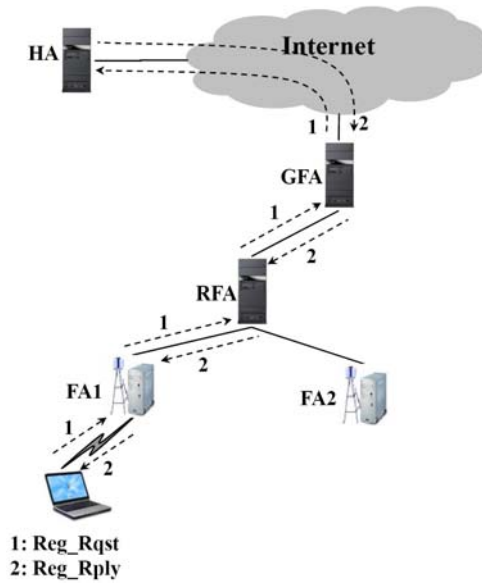


Fig 3.11: Inter-domain mobility employing MIPRR

Subsequent movements inside the domain do not require contacting the HA. Instead, the MN exchanges only a **regional Reg_Rqst** and a **regional Reg_Rply** message with the first crossover agent supporting mobility, which is defined as the agent located on both the path from the GFA to the old FA and from the GFA to the new FA. Clearly, the crossover agent may be a RFA or the GFA in the worst case. An example of this is given in figure 3.12. Movements inside the domain are termed as intra-domain mobility.

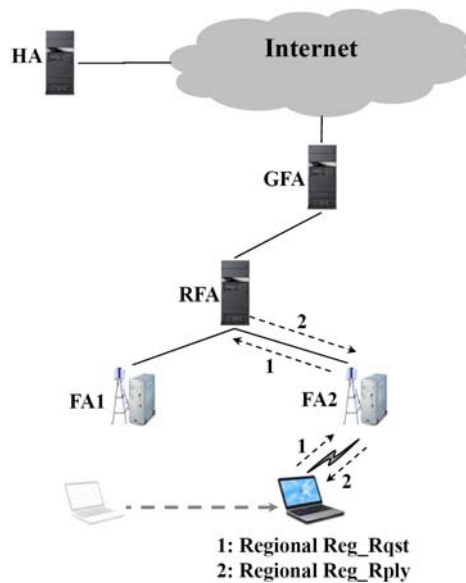


Fig 3.12: Intra-domain mobility employing MIPRR

In general, data packets destined to the MN will be delivered to the MN's home network, where the HA intercepts, encapsulates and forwards them to the MN's registered CoA, i.e. the address of the GFA. As soon as the GFA receives these packets, it de-tunnels and re-tunnels them to the current RFA, which in turn de-tunnels and re-tunnels them to the current serving FA. This FA decapsulates and transmits them to the MN. The delivery of data packets from the MN to fixed or mobile nodes is dealt with by standard IP routing.

The main advantage of MIPRR is the localization of mobility management inside an administrative domain. This reduces the handoff latency, number of lost packets, signaling

traveling towards the HA, etc. However, MIPRR results in restrictions on the network topology since it requires a hierarchical topology. New intermediate nodes are required as well. An additional drawback is the single point of failure. Any error in the GFA strongly affects the performance of MIPRR and may prohibit MNs from using the mobility service. A solution for this problem is presented in [MCN02]. The proposal is based on a backup agent that exchanges messages periodically with the GFA and has a copy of the caches and bindings maintained by the GFA. If the GFA crashes, the back up agent takes its roll by informing all RFAs residing in the domain and the HAs of the MNs registered with the GFA. The performance of MIPRR can be further improved through the utilization of layer 2 triggers. To do this, the pre- and post-registration methods, presented in section 3.1.1.3, can be employed inside the domain.

3.1.2.1.2. Hierarchical Mobile IPv6 (HMIPv6)

HMIPv6 [SCM05] is a micro mobility management protocol for IPv6 networks. It works similar to MIPRR and uses a tree-like network topology too. Each domain is connected to the Internet by means of a Mobility Anchor Point (MAP) that controls many ARs offering IP connectivity to MNs. The address of the MAP is registered with the HA as the MN's CoA, while the address of the AR serving the MN is registered with the MAP as the MN's local address. Figure 3.13 illustrates the network topology of this approach. HMIPv6 uses the same procedure of MIPv6 for inter-domain mobility, while the intra-domain mobility is controlled by the MAP.

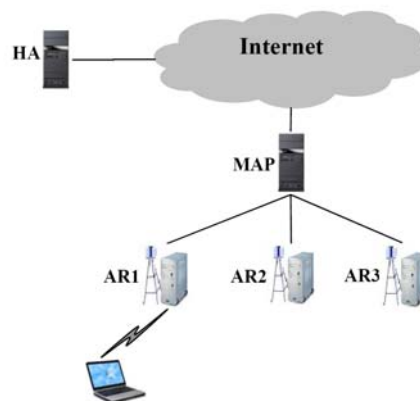


Fig 3.13: HMIPv6 network topology

Similar to MIPRR, HMIPv6 reduces the handoff latency, the number of lost packets and the signaling traveling towards the HA. However, restrictions on the network topology are assumed and an intermediate node (MAP) is required. The single point of failure is a main drawback of HMIPv6 too.

3.1.2.1.3. Dynamic Regional Registration

Dynamic regional registration techniques have no fixed hierarchical topology and do not impose restrictions on the network topology and the geographic location of subnets. The hierarchy is constructed dynamically and even optimized for each MN.

The scheme presented in [JAK02] proposes a distributed dynamic system architecture, where each FA functions either as a GFA or as a normal FA. Which FA will act as a GFA as well as how many FAs should be controlled by it is decided dynamically depending on users' movements. In other words, the role of a FA is specific for a given MN and may differ for

different MNs. The scheme supports only one level of FA hierarchy beneath the GFA. Of course, there must be SAs between the FAs present in the domain.

The optimal number (K_{opt}) of FAs underneath a certain GFA is optimized for each MN. K_{opt} is calculated taking the incoming packet arrival rate (λ) and the mobility characteristics of each user into account. The mobility characteristics and the packet arrival rate may vary from user to user and even over time for the same user. Therefore, the optimal number of FAs (K_{opt}) varies for different users and is adjustable over time for each individual user. The first FA, where the MN registers, acts as a GFA, whereas other K_{opt} FAs act as regular FAs.

When the MN moves initially into a new domain, it records the discovered FA as its GFA. The MN executes a home registration procedure to notify the HA of the acquired CoA. Afterwards, it calculates K_{opt} , which is defined as the optimal number that minimizes the cost function¹. When the MN moves to a new subnet, it records the address of the new FA in its buffer. From the MN point of view, the new FA is controlled by the GFA. This causes the MN to register itself with its GFA via the new FA. Data packets are still forwarded to the GFA that forwards them to the MN's current FA. A home registration occurs only after the MN has visited K_{opt} different subnets. Thus, there is no zigzag effect resulting from movements between borders of different domains. The MN may move back and forth between two subnets and may visit a subnet more than once without adverse effects.

Figure 3.14 shows an example of a network topology. Let us suppose that the MN has switched on in the range of FA1 and has moved and subsequently registered with FA2, FA3, FA4 and FA5. Assuming K_{opt} is 3, FA1 will be the GFA, whereas FA2, FA3 and FA4 will be the FAs belonging to this GFA. FA5 will act as a new GFA for the MN.

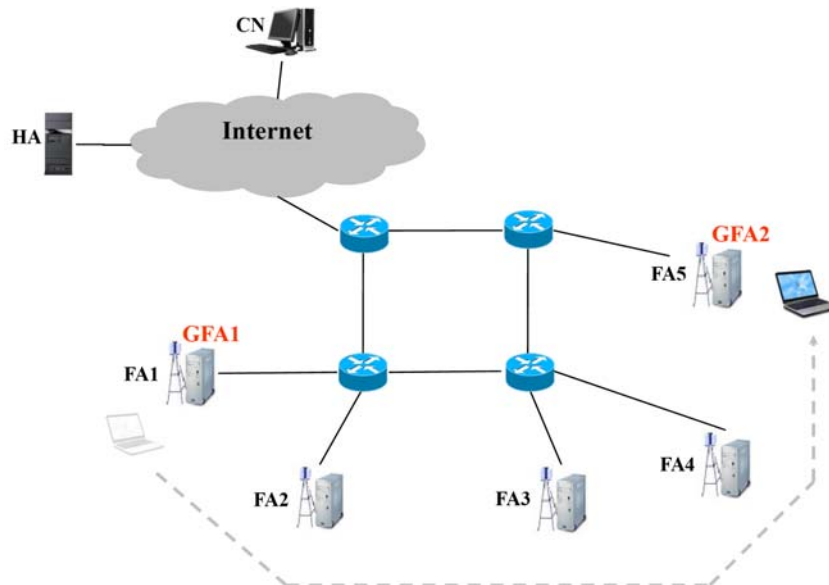


Fig 3.14: An example network topology employing the approach presented in [JAK02]

This scheme localizes the handoff without any restrictions on the network topology. The traffic load in an administrative domain is distributed among all FAs. Thus, the scheme does not suffer from a single point of failure, which improves the system robustness. The fact that each MN has its own optimized system configuration, which is autonomously adapted over time, results in an optimized performance. Clearly, it is required that all FAs are capable to function as a FA and a GFA. The scheme forces the MNs to recalculate their K_{opt} value from

¹ The cost function comprises the location update and packet delivery cost. The location update cost results from transmission of control messages, while the packet delivery cost results from forwarding data packets towards the MN. Cost function will be explained in detail in chapter 5.

time to time. The more accurate the value of K_{opt} , the small time period for its recalculation, the more power is consumed by the MN. Therefore, there must be a trade-off between the accuracy of K_{opt} and the time period for its recalculation.

The approach described in [MFa04] proposes a dynamic hierarchical mobility management technique called Dynamic Hierarchical MIP (DHMIP). The hierarchy is constructed dynamically and optimized for each MN too. However, more than one level of FAs hierarchy may exist underneath the GFA. In order to avoid an excessive packet transmission delay, the number of hierarchy levels should not exceed a certain threshold. This threshold is not the same for all users and even dynamically adjusted for the same user from time to time based on its traffic load and mobility. Exceeding the threshold causes the MN to execute a home registration procedure to register the new FA as a new GFA for the MN. An example of a network topology is shown in figure 3.15.

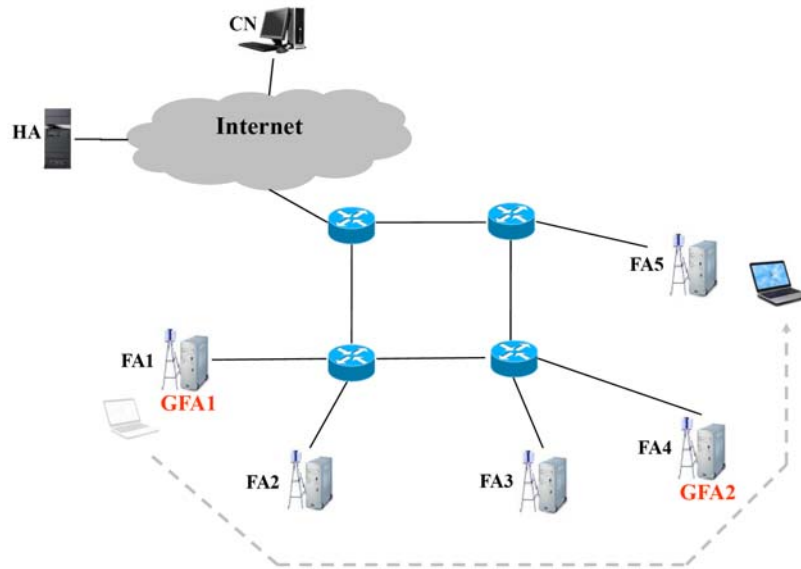


Fig 3.15: An example network topology employing DHMIP

In the figure we assume that the threshold of hierarchy levels underneath a GFA is 2 and that the MN switches on in the range of FA1 and moves to FA2, FA3, FA4 and FA5, respectively. FA1 acts as a GFA for the MN (GFA1 in the figure). When the MN moves into the range of FA2, it updates its CoA at FA1. The movement to FA3 causes the MN to update its CoA at FA2. After the MN enters the region of FA4, it notices that the hierarchy level threshold has been exceeded, executes a home registration and sets up a new hierarchy. This means that FA4 will act as the new GFA (GFA2 in the figure). Upon the occurrence of a handoff to FA5, the acquired CoA has to be updated at FA4. Data packets are re-tunneled from the GFA along the hierarchy towards the MN. For the example, this would mean that downlink data packets arriving at GFA1 while the MN stays at FA3 pass from GFA1 via FA2 to FA3.

The optimal number of hierarchy levels (K_{opt}) can be calculated based on the user's current traffic load and mobility pattern. It may be adjusted from time to time as well. Clearly, the frequent calculation of K_{opt} is not a simple task and consumes the MN's energy. There should be, therefore, a trade-off between the accuracy of K_{opt} and the energy consumption at the MN for its computation. The more often the update of K_{opt} is done, the more accurate is its value, the more the signaling traffic saving and the more the power consumption.

DHMIP localizes the mobility management and, thus, reduces the handoff latency and other problems resulting from this latency, e.g. packet dropping, TCP connection disruption, etc. DHMIP does not suffer from a single point of failure and does not make any restrictions on the network shape. The performance of DHMIP is strongly affected by the user's mobility.

DHMIP is simulated in [MFa04] in two scenarios. The first assumes that the number of hierarchy levels (K) is constant and it is the same for all MNs, while the second scenario optimizes the number of hierarchy levels for each MN, also each MN has its K_{opt} . The simulation results show that DHMIP will never generate more cost, i.e. location update and packet delivery cost, than the IETF MIPRR scheme. Compared to MIPv4, DHMIP generates less cost, regardless if the number of hierarchy levels underneath the GFA is selected as a constant value or as an optimized value, if the user's mobility is high. However, if the user's mobility is low, the cost resulting from DHMIP exceeds the cost resulting from MIPv4 in case the number of the hierarchy levels beneath the GFA is selected as a constant value. This means that the best performance is obtained if K is selected equal to K_{opt} . Therefore, the selection of K is a main task and should be executed accurately. Due to fast and random changes in mobility and traffic characteristics, the calculation of K_{opt} is a complex task that consumes considerable power in the MN.

3.1.2.1.4. Anchor Foreign Agent (AFA)

AFA [DYe01] defines an anchor point for the MN to locally control its movements inside a certain administrative domain. If the MN is away from home, it will be initially registered with the HA. During this registration a shared secret between the MN and the FA ($K_{MN,FA}$) is generated. The FA acts, after that, as an AFA for the MN. This means, after a movement to a new FA in the domain, the MN registers with the AFA instead of the HA. Afterwards, the MN may register the new FA with the HA as a new AFA or still depend on the old one. The HA tunnels data packets to the CoA registered with it, which stands for the AFA. The AFA in turn decapsulates, re-encapsulates and forwards the packets to the new FA, which forwards them to the MN. Data packets sent from the MN should pass through a reverse tunnel between the new FA and the AFA, which de-tunnels the packets and forwards them towards their destination using standard IP routing mechanisms.

If the AFA is not able to authenticate the MN during the registration procedure for any reason, it relays the registration to the HA. In other words, the MN registers itself with the HA via the AFA. This registration is called indirect registration.

Employing this approach, there is no need to establish a tunnel between the HA and the new FA. Instead, an additional bidirectional tunnel from the AFA to the new FA is established. The registration with the AFA reduces the handoff latency, number of lost packets and signal traffic traveling towards the HA. Clearly, this approach does not suffer from a single point of failure. However, forwarding delays on the downlink as well as the uplink increases compared to other approaches. The specification of the approach does not discuss when a new AFA should be registered with the HA to avoid degradation in the performance.

3.1.2.1.5. Seamless Mobile IP (S-MIP)

S-MIP [HZS03] localizes the mobility management by means of a hierarchical network topology that makes use of layer 2 information. It extends the network by adding a Decision Engine (DE) entity and a Synchronized-Packet-Simulcast (SPS) scheme. An example network topology is depicted in figure 3.16. The DE is responsible for making a handoff decision. By means of periodic feedback and movement tracking information from individual ARs, the DE maintains a global view on connection states of MNs present in the domain as well as their movement patterns. This global view enables the DE to support load balancing through instructing the ARs serving fewer MNs to accept new MNs rather than the ARs that currently manage a large number of MNs. Each AR periodically sends Carrying Load Status (CLS)

messages (3 seconds period) to notify the DE of the MNs currently associated with it as well as their IP addresses.

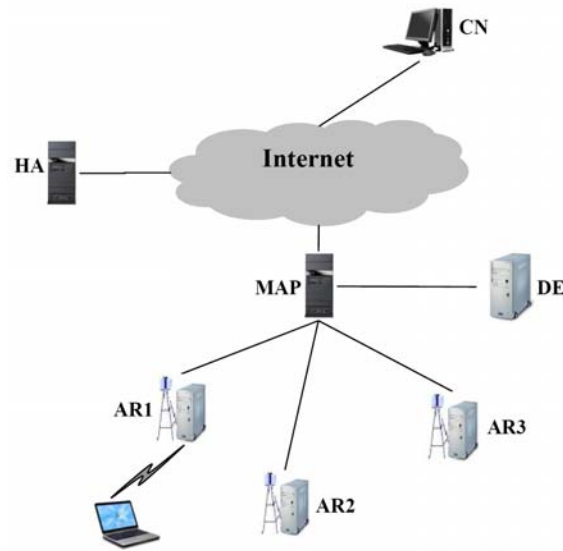


Fig 3.16: S-MIP network topology

Handoffs are implemented as follows: upon receipt of beacon messages from newly discovered ARs, the MN initiates the handoff by sending a **RtSolPr** message, containing the IDs of the discovered ARs, to the old AR. Upon the receipt of the **RtSolPr** message by the old AR, it sends **HI** messages to the new candidate ARs previously provided by the MN in the **RtSolPr** message. Each **HI** message contains the configured new CoA in the range of the new AR and the CoA being currently used by the old AR. Each new AR responds by sending a **HAck** message, which indicates either acceptance or rejection of the new CoA. If a new CoA is accepted by the respective AR, the old AR sets up a temporary tunnel to the new CoA¹. Otherwise, a tunnel is set up to the new AR, which takes care of further forwarding of data packets to the MN using the old CoA temporarily, in case the MN moves to this new AR.

The MN sends a Current Tracking Status (CTS) message to its current AR each time it receives a beacon message from a new AR. A CTS message contains the signal strength and the ID of the detected AR. The signal strength and AR-ID serve as location tracking information for the MN. The old AR forwards the tracking information to the DE periodically (once per second) until a Handoff Decision (**HD**) message is received from the DE. **HD** messages are sent to all participating ARs. After that, the old AR sends a **PrRtAdv** together with a Handoff Notification (**HN**) message to the MN. The **HN** message determines the new AR, to which the MN should hand off.

After receipt of the **HN** message and the new CoA, the MN sends a **F-BU** message to the old AR, which sends a Simulcast (**Scast**) message to the MAP to initiate the simulcasting of packets. As a result, every subsequent packet will be duplicated in the MAP and sent to the old and new AR simultaneously. These packets are marked as **S** packets. As a reply to the **F-BU** message, a **F-Back** message is sent from the old AR to the MN via all currently active interfaces (old and new subnets). Data packets forwarded from the old AR to the MN via the new AR are marked as **F** packets. **S** packets and **F** packets will be stored in different buffers in the new AR, namely S-buffer and F-buffer. After the MN finishes the layer 2 handoff, it sends a **F-NAdv** message to the new AR, which prompts it to start forwarding the buffered MN's data packets. The new AR first transmits the packets stored in the F-buffer followed by

¹ Notice that the MN in this case is the tunnel end-point and not the new AR.

the packets in the S-buffer. In addition, a Simulcast off (*Soff*) message is sent from the new AR to the MAP, which forwards it further to the DE to indicate the completion of the handoff. The DE does not allow the MN to execute a new seamless handoff while the current is still ongoing. The handoff procedure of S-MIP is depicted in figure 3.17.

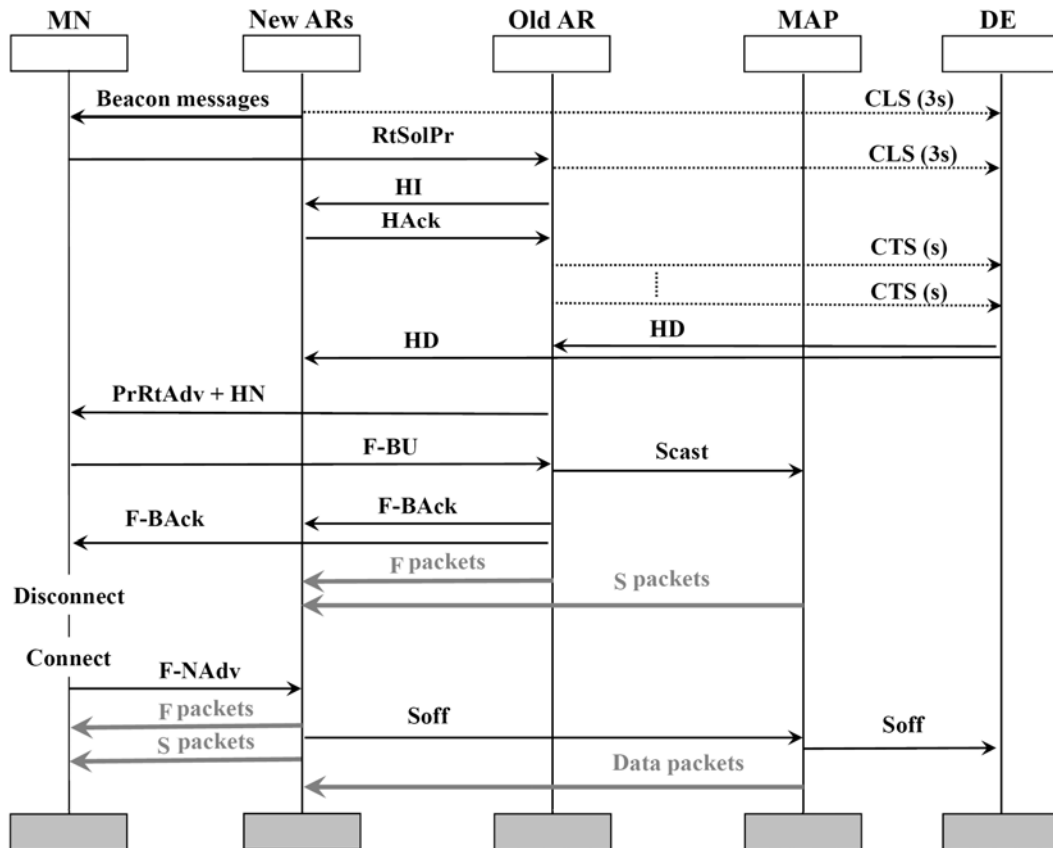


Fig 3.17: S-MIP handoff procedure

The S-MIP handoff procedure varies depending on the MN's movement pattern. If the MN is identified to be in a stochastic moving state, *HD* messages will inform the participating ARs to be in an anticipation mode. In this mode, the old AR maintains the MN's binding in preparation for the MN returning (ping pong). This avoids unnecessary re-setups of resources and time delays. If the MN is identified to be in a stationary state near the boundary between two network coverage areas, *HD* message will instruct the using of multiple bindings between the MN and the participating ARs. In other words, the MN uses more than one CoA simultaneously. See [JPA04] for details. Lastly, if the MN is deemed to be moving in a linear fashion, *HD* messages contain the ID of the AR, to which the MN will go. ARs that are not selected for the handoff will be notified, by means of *HD* messages, to refrain from further participation in the handoff process.

The simulation results presented in [HZS03] show that S-MIP can achieve lossless layer 3 handoffs. The handoff latency is reduced to the latency resulting from the layer 2 handoff. This results in minimizing or even eliminating the disruption of TCP connections resulting from layer 3 handoffs. However, S-MIP assumes restrictions on the network topology. In addition, the protocol requires a new entity (DE) to be added to the network. Due to observing and tracking of MNs in overlapping areas and due to periodic reports sent from MNs as well as ARs (*CTS* and *CLS* messages), a significant signaling overhead is produced. *CTS* messages, sent from the MN to the old AR, consume the MN's power, especially if the MN moves at high speeds. In addition, high-speed movements have a significant impact on the performance of S-MIP, especially if overlapping areas are not large enough. Similar to most

micro mobility management approaches, S-MIP suffers from the single point of failure problem.

3.1.2.1.6. BRAIN Candidate Mobility Protocol (BCMP)

BCMP [KGT01] is proposed by the IST project BRAIN. It is a domain-based protocol used to support micro mobility management. The assumed network topology is shown in figure 3.18.

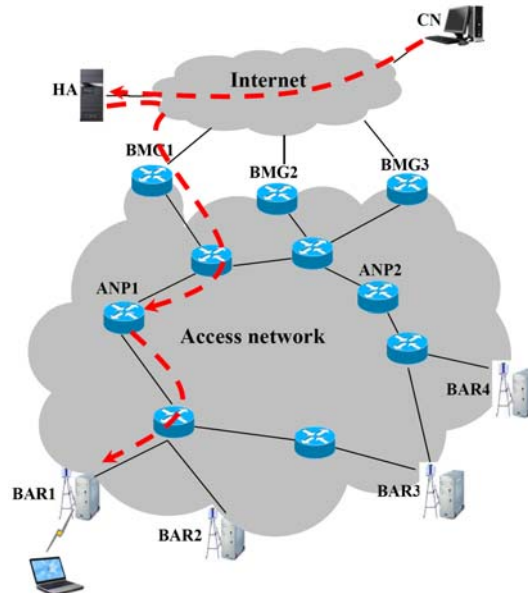


Fig 3.18: BCMP network topology

Mobility-aware functionalities are embedded in key components of the BCMP Access Network (AN), which mainly consists of the BRAIN Access Routers (BARs) and Anchor Points (ANPs). The BARs are located at the edge of the AN. They offer IP connectivity to MNs. The ANPs are located inside the AN. They manage mobility, allocate IP addresses for MNs, authenticate them, maintain their records and tunnel data packets towards them. The location of ANPs in the AN is essential for optimal performance. Tunneled packets are terminated by the BARs and forwarded to MNs. The pool of IP addresses owned by an ANP is advertised using legacy IP routing inside the AN and towards external IP networks. This ensures that data packets, addressed to a MN's locally obtained address, are prefix-based routed to the ANP that has allocated this address. BRAIN Mobility Gateways (BMGs) do not provide any mobility support. They serve as standard border routers, separate the AN from exterior networks and forward incoming traffic to the correct ANPs.

When the MN switches on or moves into an AN, it has to execute a login procedure. The MN sends a **login request** message to the discovered BAR. BCMP assumes that the **login request** contains login and security information for an external AAA procedure. The current BAR selects an adequate ANP according to policy rules and forwards the login request to it. The ANP performs an AAA procedure to identify and authenticate the MN. After a successful authentication, a globally routable IP address and a new session identifier are allocated for the MN. The session identifier and the IP address are sent to the MN in a **login response** message. The acquired IP address stays constant in spite of future handoffs in the AN as long as the ANP remains unchanged. The handover to a new BAR is initiated by the new BAR itself, which notifies the old BAR and the ANP of the incoming MN. The old BAR responds by tunneling the MN's data packets to the new BAR. Upon notification of the ANP of the handoff, the ANP redirects the tunnel to the new BAR and notifies the old BAR, which removes the temporary tunnel to the new BAR. BCMP has an optional handover preparation

phase to ensure fast and smooth handoffs. If the new BAR can be predicted before the actual handoff takes place, a temporary tunnel from the old to the new BAR can be established in advance. The messages exchanged during the login procedure and during the handoff are shown in figure 3.19 and figure 3.20, respectively.

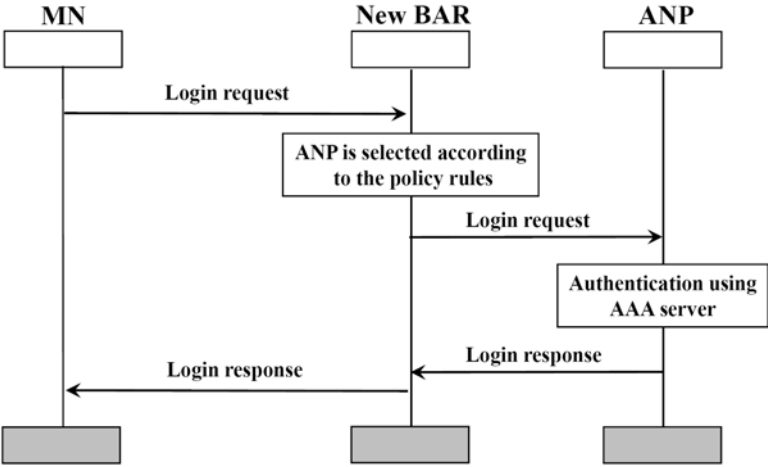


Fig 3.19: BCMP login procedure

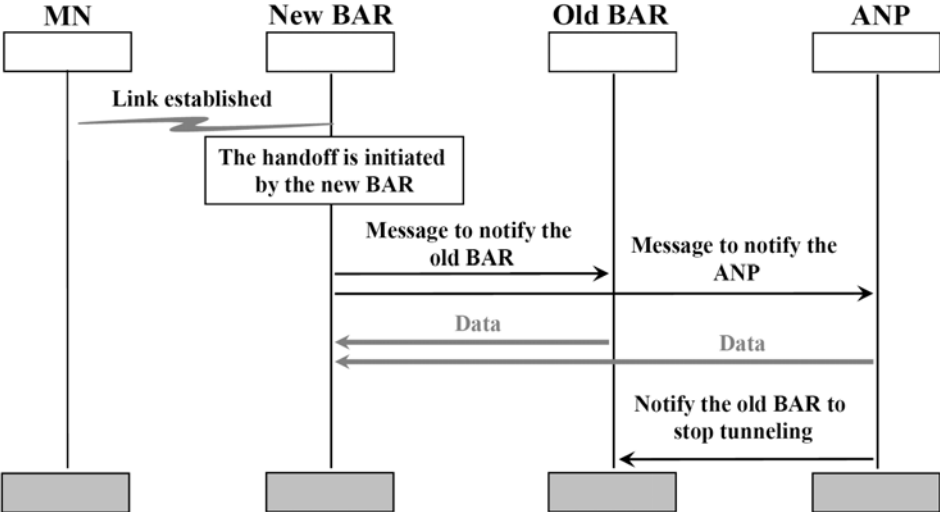


Fig 3.20: BCMP handoff procedure

The tunnel between the ANP and the serving BAR may increase over time if the MN moves far away from its ANP. To avoid such a situation, the MN should change its ANP. This prompts the execution of a global handoff resulting in a change of the ANP and the CoA.

In order to locate idle MNs and to reduce the location update cost inside the AN, paging is supported. Downlink packets are tunneled to the MN’s last known serving BAR, which should know that the MN is idle and initiates the paging process.

Using BCMP, the MN solely communicates with the BAR in all cases and is not aware of the structure of the AN and the protocols employed inside. BCMP has its own message set and does not reuse or extend MIP’s messages. This ensures an independence from the used macro mobility protocol. Existence of many ANPs significantly reduces the impact of a single point of failure. Separation between the functions of the BMGs and the ANPs allows for more flexibility in the selection and deployment of network components. However, in spite of remaining in the same domain, the MN has to execute a home registration, when it moves far away from the current ANP. The location and number of ANPs have a significant impact on

the performance and should be selected carefully. Clearly, this complicates the network design.

3.1.2.1.7. *Telecommunication Enhanced Mobile IP (TeleMIP)*

TeleMIP [DMA00], [CMD01] is based on the same principles as most hierarchical micro mobility management approaches to confine most location update messages within an administrative domain. A new logical entity, called a mobility agent, is introduced at the top of the hierarchy to provide a stable point of attachment in the domain. Location updates inside the domain are solely handled by the mobility agent, while the HA is aware of movements outside the scope of the mobility agent. As can be seen in figure 3.21, TeleMIP administrative domain comprises several subnets. Typically, there are many mobility agents distributed throughout the domain. They are responsible for providing globally reachable CoAs for the registered MNs. Each FA must be associated with at least one mobility agent in the domain.

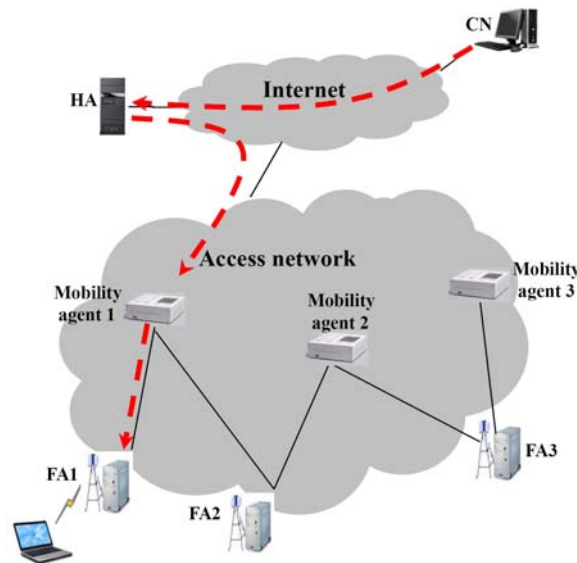


Fig 3.21: TeleMIP network topology

Similar to other micro mobility management solutions, the MN is assigned two IP addresses, a Global CoA (GCoA) and a Local CoA (LCoA). The GCoA is a globally routable address and is typically the address associated with the mobility agent. The GCoA stays unchanged as long as the MN stays within the specific domain or region controlled by the current mobility agent. The LCoA determines the point of attachment from the MN point of view and changes when the MN moves inside the domain. When a CN wants to communicate with a MN, it sends data packets to the MN's home address. These packets will be delivered to the HA, which intercepts and forwards them towards the mobility agent that takes care of further forwarding to the MN's LCoA.

Similar to other micro mobility management techniques, TeleMIP confines mobility processing inside the administrative domain, which results in reducing the handoff latency, the number of lost packets, signaling traffic towards the HA, etc. Due to the presence of many mobility agents in the TeleMIP domain, this technique does not suffer from the single point of failure problem. Because of the assignment of more than one mobility agent to each FA, a load balancing algorithm can be used to optimize the performance, e.g. to register and manage the MNs served by a FA with different mobility agents. An additional benefit is that TeleMIP permits the use of local or private addresses. In other words, it provides a flexible addressing scheme, see [DMA00] for details.

A well-known extension of the intra-domain protocol used in TeleMIP is the Intra Domain Mobility management Protocol (IDMP) [DMD02], [WDM02]. When the MN switches on or moves into a domain, it listens to advertisements from the current Subnet Agent (SubA), which stands for a FA in the previous figure. Upon receipt of an advertisement, the MN performs a subnet-specific registration by sending a *subnet_Reg_Request* message, as depicted in figure 3.22.

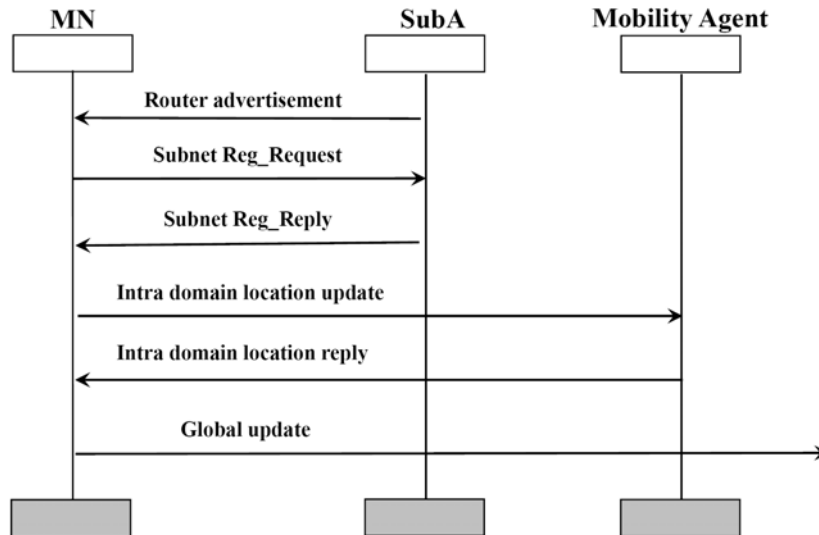


Fig 3.22: IDMP initial registration procedure

The current SubA assigns a mobility agent to the MN and sends a *subnet_Reg_Reply* message as a response. Afterwards, the MN executes an intra-domain location update by communicating its mobility agent, which selects a GCoA and includes it in an *intra-domain location reply* message. Subsequently, the MN executes a global location update procedure to register its GCoA with its home network. Data packets are forwarded to the mobility agent, which intercepts and tunnels them to the current SubA. The packets are then de-tunneled by the current SubA and forwarded to the MN. After the initial intra-domain registration, the MN retains its GCoA as long as it is managed by the same mobility agent. Whenever the MN changes the subnet within the same domain, it carries out a new subnet-specific registration with the new SubA. Clearly, the new SubA does not allocate a new mobility agent for the MN. A new intra-domain location update is executed, after that, to notify the mobility agent of the new LCoA. No global location update procedure is required in this case.

IDMP relies on layer 2 triggers to achieve fast handoffs. The triggers are assumed to be available either to the MN or to the old SubA. After the appearance of a L2-trigger at the MN, it generates and sends a *MovementImminent* message to the mobility agent, which responds by multicasting the MN's packets to a set of neighboring SubAs. Each SubA in this set buffers the packets in a MN-specific buffer to minimize the loss of in-flight packets. When the MN moves to one of the neighboring SubAs, it executes a subnet-specific registration according to IDMP. The new SubA transmits, thereafter, the buffered MN's packets immediately over the wireless interface without waiting for the completion of the intra-domain location update. After the mobility agent has been informed about the new LCoA, it stops multicasting and forwards packets to the new MN's location only.

In order to reduce the intra-domain location update cost for idle MNs, paging is supported. IDMP assumes that SubAs are grouped into paging areas identified by some unique identifiers advertised as a part of advertisement messages. A MN in idle mode can simply detect any change in its current paging area by listening to these unique identifiers in the advertisements. As long as the idle MN does not detect any change in the paging area even if

a change in the SubA is detected, it updates neither its mobility agent nor its current LCoA. However, if the MN detects a change in the paging area, it obtains a new LCoA and executes an intra-domain and a global location update procedure. When a mobility agent receives data packets for a MN with an invalid LCoA, it multicasts a *PageSolicitation* message to all subnets associated with the MN's current paging area and buffers the incoming packets. After the MN responds by registering with the mobility agent, the buffered packets are forwarded to it.

A main advantage of IDMP, additional to the other advantages of TeleMIP, is that IDMP is designed as a standalone approach for intra-domain mobility without assuming MIP as a base protocol for inter-domain mobility. Unlike other fast handoff proposals, the *MovementImminent* message does not specify the IP addresses of the possible new SubAs. This puts less constraints on layer 2 triggers. However, there is a considerable packet delivery cost resulting from multicasting of data packets to all neighbor SubAs. TeleMIP and IDMP require a hierarchical network architecture, which implies restrictions on the network topology. Due to the existence of more than one mobility agent in the same domain, a global registration may occur even if the MN still resides in the same administrative domain.

3.1.2.2. Localized Enhanced Routing Schemes (LERS)

As mentioned in section [2.4.2](#), the approaches belonging to this category try to support mobility by managing the route to the MN's new location locally. These approaches fall into two main categories, namely per-host forwarding and multicast-based schemes [[EMS00](#)].

3.1.2.2.1. Per-Host Forwarding Schemes

Per-host forwarding schemes assume that each router in the domain supports mobility by providing a specific entry in its routing table that defines the output port for each MN in the system. Thus, traditional longest-prefix matching is replaced by an exact matching of IP addresses. The schemes typically employ a specialized path setup protocol along with maintaining routing caches in the routers. The routing caches are updated according to MN's movements. Data packets are forwarded, thereafter, hop per hop towards the MN. In the following, well-known approaches will be summarized including a discussion of their pros and cons.

3.1.2.2.1.1. Cellular IP (CIP)

A CIP-based access network consists of BSs, serving as APs, and a gateway controlling the CIP domain and connecting it to external networks or the global Internet as shown in figure 3.23. IP-based routing of data packets is replaced by a special CIP routing and location management. The MNs attached to a CIP access network select the IP address of the gateway as their CoA and notify their HAs. Inside the CIP access network, packet forwarding to the MNs is based on their home addresses. The gateway broadcasts periodic beacons. The beacons are flooded to all BSs in the access network. Each BS records the neighbor node that has forwarded the gateway's beacons and routes packets towards the gateway always via this node. Notice that all data packets sent from any MN in the domain are routed towards the gateway using this route regardless of their destination. Forwarding of these packets towards the gateway is utilized to refresh the routing entries in the hops in between. Each BS maintains a routing cache, within the IP address of the source MN and the neighbor that has sent the packet. Routing caches are stored in a soft state and refreshed by uplink packets. They are used to route packets towards the MN too. To keep routing caches up to date even if no

uplink packets are transmitted, each MN sends periodic *route-update* packets addressed to the gateway.

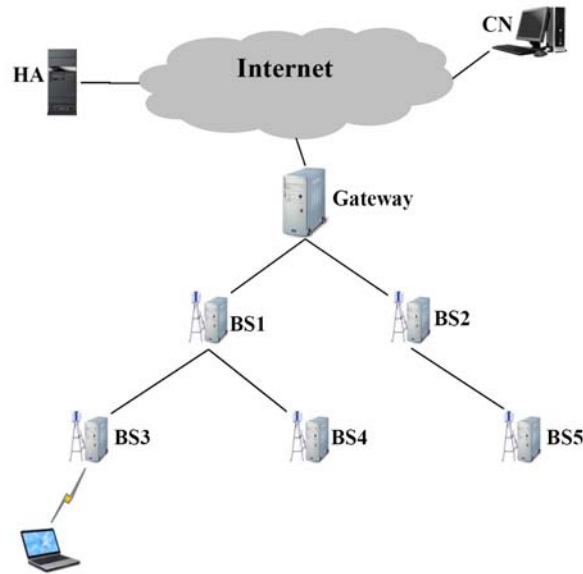


Fig 3.23: CIP network topology

CIP supports hard and semi-soft handoffs. By hard handoffs, the MN listens to beacons from new BSs and initializes the handoff based on signal strength measurements. The radio link with the old BS is broken and a new link with the new BS is established. Afterwards, the MN sends a *route-update* packet towards the gateway to enter or update its entries in the routing caches of the new BS, the gateway and the hops in between. The layer 3 handoff latency equals the round trip time between the MN and the first crossover node. In the worst case, this is the gateway. In the best case, it is the router next to the BS, to which the MN is attached.

The key idea of the semi-soft handoff says that the routing cache mapping must be created in the new BS before the actual handoff takes place. For this purpose, the MN switches to the radio of the new BS and sends a *semi-soft* packet. After that, the MN returns back immediately to listening to the old BS. The purpose of the *semi-soft* packet is to establish a new route between the new BS and the crossover node. After a semi-soft delay, the MN performs a regular handoff and moves to the new BS. At this point, the MN's packets are forwarded to both the old and new BS. Due to the network topology and traffic conditions, the time required to send packets from the crossover node to the old BS may differ from the time required to send the packets to the new BS. This may disrupt some application, e.g. this may produce big jitter fluctuations during the handoff for VoIP traffic. As a solution for this problem, CIP proposes the inclusion of a temporary constant delay along the path between the crossover node and the new BS. The delay can be provided by a simple delay device mechanism. The *semi-soft* packet contains a flag to indicate whether downlink packets should pass through the delay device before being forwarded along the new path. After completion of the layer 2 handoff, the MN sends data or a *route-update* packet towards the gateway. This results in stopping the delay device and forwarding to the old BS.

CIP supports paging to reduce the location update cost and energy consumption of idle MNs. BSs are geographically grouped into paging areas. A paging area identifier is broadcast as a part of beacon messages. Idle MNs have to update their locations only when moving between different paging areas. In addition, they have to transmit *paging-update* packets at regular intervals defined as paging-update-time. This enables them to be always reachable. Each BS may optionally maintain a paging cache that has the same format as a routing cache with a longer lifetime, called paging-timeout.

When a packet addressed to a certain MN is received and the gateway or a certain BS does not find a valid routing cache mapping for the MN, the paging procedure is triggered. If an entry in the paging cache is found, the packet will be forwarded to the BS recorded in the paging cache. However, if no or an invalid entry is found, the packet is forwarded to all interfaces of the gateway or BS. No explicit paging message is used in CIP. Rather, the first received packet represents an implicit paging message. An idle MN that receives a paging packet transmits itself from idle to active state and immediately replies with a *route-update* packet.

Evaluation results presented in [CGK00] and [FKS00] show that due to the processing of the mobility inside the domain locally, CIP reduces the location update cost, the handoff latency and the number of lost packets. Hard handoffs cause packet losses proportional to the packet arrival rate and the round trip time between the MN and the crossover node, while semi-soft handoff may eliminate the packet loss completely. Of course, there must be a sufficient buffer size in the new BS to buffer the packets sent during the semi-soft handoff. Due to paging support, a large number of MNs can be served by the CIP access network, which significantly improves the scalability of the protocol. An additional benefit of CIP is the routing of mobile-to-mobile communication inside the CIP domain via the gateway. The fact that CIP introduces a new layer 3 dynamic routing protocol into the domain strongly affects the TCP performance. More concrete, the throughput of a CIP BS is somewhat lower than the throughput of a standard IP-based BS [CGK00]. In addition to this, CIP assumes that every node in the domain is mobility-aware. A well-known problem of CIP is the single point of failure. This is because all traffic from and to the domain is routed via the gateway.

3.1.2.2.1.2. *Handoff-Aware Wireless Access Internet Infrastructure (HAWAII)*

HAWAII [RLT00], [RVS02] is a domain-based approach to support mobility in IP-based networks. The network architecture of HAWAII is illustrated in figure 3.24. All issues related to mobility management within a certain HAWAII domain are handled by a gateway called a domain root router. This gateway connects the HAWAII domain to other external networks or domains.

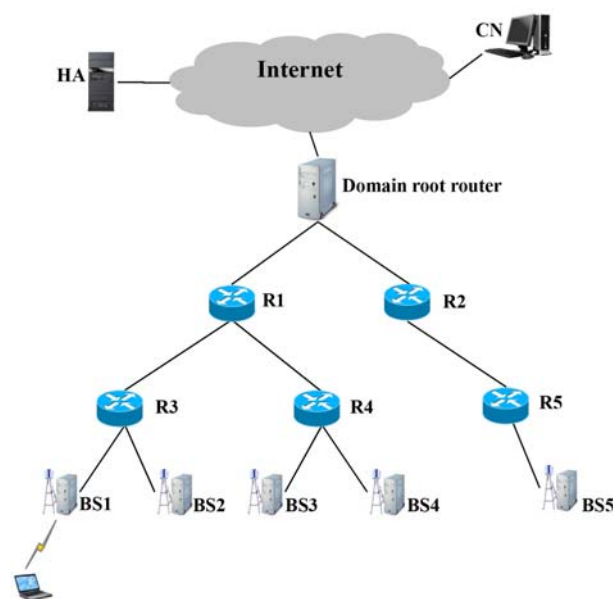


Fig 3.24: HAWAII network topology

When the MN moves to a foreign domain, it is assigned a co-located CoA that does not change as long as the MN stays inside the domain. Data packets sent to a MN are intercepted by the HA, which tunnels them to the domain root router. From there, the packets are routed to the MN using dynamically established paths. Movements between different subnets of the same domain cause only the route from the domain root router to the new BS to be modified.

HAWAII uses three types of messages for the path setup, namely a *power-up*, an *update* and a *refresh* message. When the MN switches on, it sends a *path setup power-up* message. This message establishes host-specific routes for the MN in the domain root router and the intermediate routers on the path towards the MN. To manage user mobility, HAWAII uses *path setup update* messages to establish and to update host-specific routing entries for MNs, so that the packets arriving at the domain root router can reach the MNs with limited disruption. There are four schemes for a path setup after a movement inside a HAWAII domain. The four schemes can be classified into two types, the forwarding and the non-forwarding type, based on the way the packets are delivered to the MN during the handoff. In the forwarding type, packets are forwarded from the old BS to the new one. For this, a Multiple Stream Forwarding (MSF) scheme and a Single Stream Forwarding (SSF) scheme are supported. In the non-forwarding type, packets arriving at the crossover router are delivered to the new BS. For this, a Unicast Non-Forwarding (UNF) scheme and a Multicast Non-Forwarding (MNF) scheme are supported. No forwarding from the old BS to the new one is done in the non-forwarding type.

HAWAII path states are maintained in a soft-state. Each MN sends periodic *path refresh* messages to its serving BS to maintain the host-specific entries. BSs and intermediate routers send periodic aggregated hop by hop *refresh* messages towards the domain root router. In order to reduce the location update cost and power consumption of idle MNs, paging is supported. HAWAII uses IP multicasting to page idle MNs, when data packets destined to them arrive at the domain root router and no recent routing information is available.

Simulations show that HAWAII exhibits smaller disruptions to audio and video traffic compared to MIP schemes, i.e. standard MIPv4 and MIPv4 with route optimization extension, [RVS02]. For stored audio and video applications, where maintaining a small play-out time is not critical, MIPv4 with route optimization extension performs similar to HAWAII schemes. However, HAWAII outperforms MIPv4 with route optimization extension for interactive audio and video applications, where a small play-out time should be maintained. Regarding the different HAWAII schemes, UNF performs best for MNs that can listen to more than one BS at the same time. For MNs that can listen to only one BS at a time, MNF performs best. SSF and MSF are able to achieve lossless handoffs. Regarding long-duration TCP flows, HAWAII schemes clearly outperform standard MIPv4 by 15 % and show a small improvement over MIPv4 with the route optimization extension with respect to the average aggregated throughput of all MNs in the domain. The numerical results presented in [RVS02] show that the processing overhead in the domain root router employing HAWAII is about 10 times lower than the processing overhead in a HA employing MIPv4. This improves the scalability of HAWAII. The requirement that every node in the domain should be mobility-aware can be considered as a drawback that complicates the employment of HAWAII in existing systems. Another drawback is the single point of failure problem.

3.1.2.2.1.3. *Micro-Mobility Support with Efficient Handoff and Route Optimization Mechanisms (MEHROM)*

MEHROM [PMD04] works similar to HAWAII and CIP. However, it does not put any restrictions on the network topology. As shown in figure 3.25 and figure 3.26, a tree-like or a mesh network can be used.

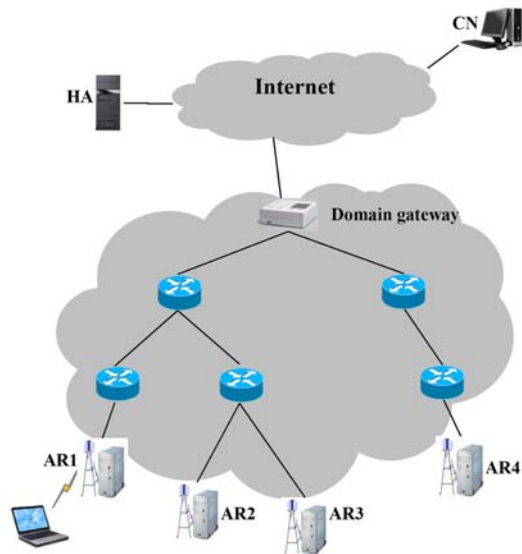


Fig 3.25: MEHROM tree-like network topology

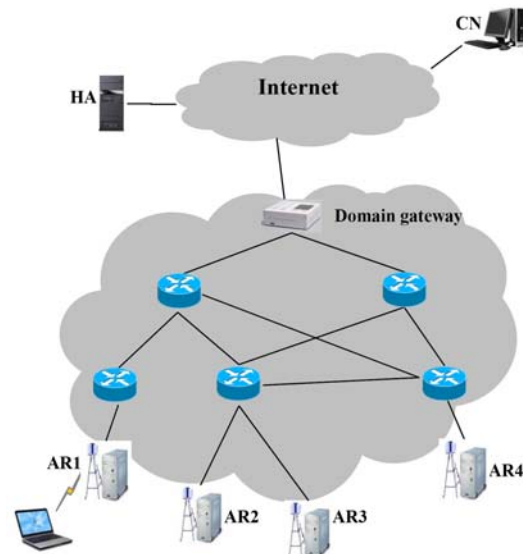


Fig 3.26: MEHROM mesh network topology

MEHROM employs the Open Shortest Path First (OSPF) routing protocol inside the domain [Moy98]. The main motivation behind the selection of OSPF is the link state database it provides, which delivers a view of the topology and the current state of each link. This database can be used to select the optimized path towards the domain's gateway. MEHROM manages the mobility in two phases, a handoff and a route optimization phase. The handoff is executed during the handoff phase, while the path between the current AR and the gateway is optimized during the route optimization phase. The fact that the route from the gateway towards the MN is optimized in a separate phase allows for a flexible optimization considering many criteria, e.g. the delay, the available bandwidth, etc.

Upon receipt of a *Reg_Rqst* message at the new AR, it adds a new entry for the MN and sends a *route update* message hop by hop towards the old AR to create an entry for the MN in the nodes residing in between. If the *route update* message reaches a crossover node, it will not be forwarded further. A *handoff acknowledgement* message is returned to the new AR, which in turn sends a *Reg_Rply* to the MN. In addition to transmitting a *handoff acknowledgement* back to the new AR, the crossover node sends a *route delete* message towards the old AR to delete the entries of the MN. After the MN completes the handoff, the crossover node examines the route between the gateway and the new AR. If the route is not optimal, the route optimization phase is started. The new AR sends a new *route update* message hop by hop towards the gateway to establish an optimal path according to given optimization criteria. While this is done, the MN receives its packets via the old sub-optimal path.

The results presented in [PMD04] can be summarized as follows: considering the packets dropped during the handoff, MEHROM performs similar to CIP and HAWAII for a hierarchical topology. For a mesh access network, MEHROM performs better than CIP and comparable to HAWAII. Taking the path length between the gateway and the new AR into account, MEHROM performs similar to HAWAII and CIP for a tree-like topology. However, for a mesh network, MEHROM performs similar to CIP and better than HAWAII. In contrast to most micro mobility protocols, MEHROM puts no restrictions on the access network topology. The control traffic is concentrated near the involved ARs and a low packet loss and optimal paths are achieved, see [PMD04]. MEHROM suffers, however, from the single point of failure problem. Also, security issues are not discussed.

3.1.2.2.2. Multicast-Based Schemes

As mentioned in chapter 2, multicast-based schemes utilize point-to-multipoint connections to support a location-independent addressing and routing. They can be categorized into two subcategories, namely dense- and sparse-mode multicast-based approaches. Dense-mode approaches are suitable for densely populated groups. They construct source-based trees, i.e. a separate tree for each source to receiver pair, and use flooding to distribute packets to interested hosts. In contrast to dense-mode approaches, sparse-mode protocols are suitable for sparsely populated groups. They use a shared tree for all members to and from a center point, also referred to as core or rendezvous point. Interested hosts join the multicast group explicitly by sending control messages towards the core hence creating or possibly updating the routing tree towards the host. Packets are unicast sent to the core, which converts them to multicast packets and distributes them to interested hosts.

The following describes briefly two well-known examples, the approach proposed in the scope of Daedalus project as an example of dense-mode approaches and MMP as an example of sparse-mode approaches.

3.1.2.2.2.1. Daedalus Proposal

This technique is a dense-mode multicast-based approach [SBK97], [Ses95]. It deploys the same network topology used by MIP. The technique depends on the IP-multicasting and buffering in neighbor BSs to eliminate data loss during handoffs. When the MN is away from home, it is assigned a temporary multicast CoA. The CoA is registered with the HA using MIP. The HA is responsible for tunneling data packets destined to the MN to the associated multicast group that contains the current BS and the BSs locating in the vicinity of the current one. Each MN keeps tracking of its current location and the new BSs locating in its vicinity, e.g. by scanning the medium for available BSs, doing some measurements, etc. Depending on the gathered tracking information, the MN configures the routing between the HA and the discovered BSs. More specifically, the discovered BSs join the multicast group associated with the MN. Joining the multicast group is mobile-initiated. In other words, the MN instructs the BSs by sending specific control messages. The current BS is labeled as a primary BS and forwards packets to the MN. The BSs locating in the vicinity are possible targets, where the MN may move to. They receive data packets sent to the MN too. However, they do not transmit them over wireless links. Instead, they buffer the last few packets. Upon the MN enters the region of one of these targets, it sends a control message containing a list of the last received few packets. The new BS will be a primary one and starts transmitting data packets from its buffer to the MN. Clearly, the new BS does not send the packets existing in the list sent previously from the MN. After transmitting all buffered packets, the new BS starts sending the packets received from the HA.

Utilizing multicast to establish route for the MN in advance greatly reduces the packet loss and minimizes impairments of ongoing applications. The MN controls the forming of the multicast group without joining it. This requires doing lots of measurements, which consume the MN's power. In addition, these measurements are done mainly in the physical layer, which makes this technique technology-dependent. The buffering in neighbor BSs can be seen as an advantage since multicast packets are forwarded on the wireless link only by the primary BS. Clearly, this does not produce any extra bandwidth consumption on wireless links due to the multicast. Consuming of extra resources in terms of bandwidth and buffer space on the wired network is, however, not avoided. The smoother the handoff should be, the more neighbor BSs should join the multicast group, the more resources are consumed. Therefore, there should be a trade-off between the performance and resources consumed.

3.1.2.2.2. Multicast for Mobility Protocol (MMP)

MMP [MSA00] is a domain-based sparse-mode approach. The network architecture of MMP is illustrated in figure 3.27. MMP domain is controlled by a gateway that connects the domain to other external networks or domains as well. This lets the domain appear as a single subnet for hosts locating outside the MMP domain. The Core Based Trees (CBT) [Bal97] is chosen as a sparse-mode multicast protocol inside the MMP domain. It provides fast and efficient tree forming and maintenance methods, see [MSA00]. Mobility between different MMP domains is handled by the standard MIP, while mobility inside the domain is handled by CBT mechanisms. The gateway forms the core of the multicast tree inside the domain and acts as a mapping point between MIP and CBT. The MNs moving inside a MMP domain require supporting MIP only. BSs appear as FAs supporting MIP from the MNs point of view. The BSs use MIP control messages to trigger CBT mechanisms.

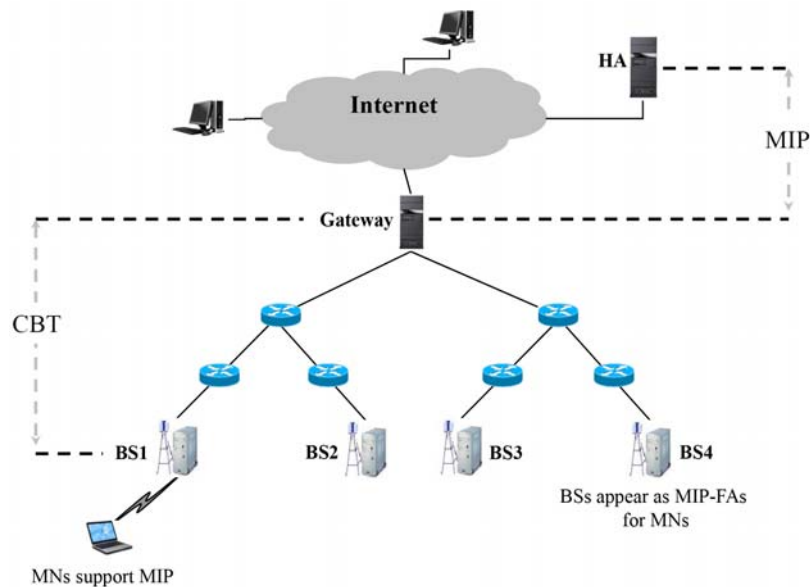


Fig 3.27: MMP network topology

When a MN moves into a new MMP domain, it acquires first a new multicast CoA and transmits a **Reg_Rqst** message constructed according to MIP to the new FA, which is represented through a BS in the figure above. The BS forwards the **Reg_Rqst** to the gateway, which replaces the multicast CoA with its own IP address and forwards the new **Reg_Rqst** message to the HA, see figure 3.28.

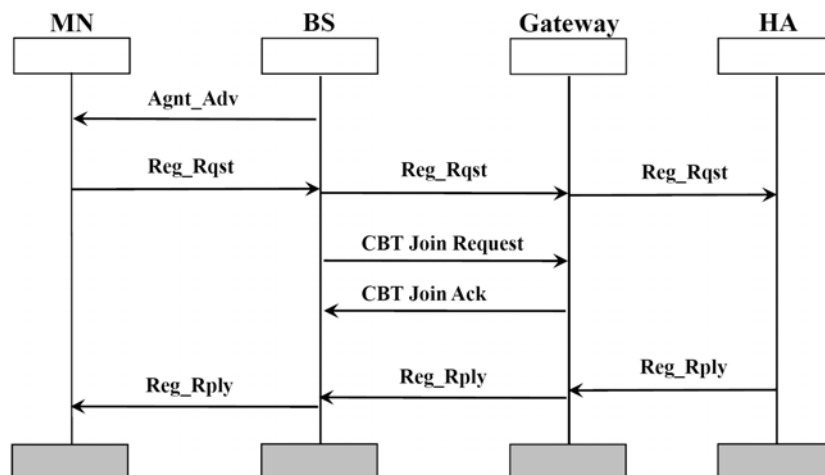


Fig 3.28: MMP initial registration procedure

Apart from forwarding the *Reg_Rqst* to the gateway, the BS starts creating the routing tree inside the MMP domain. For this purpose, the BS transmits a *CBT Join Request* up to the gateway, which replies a *CBT Join Ack* that traverses the same path of the *CBT Join Request* downstream towards the BS. Downlink data packets addressed to the MN are intercepted by the HA, which tunnels them to the gateway. The gateway in turn de-tunnels the packets and tunnels them again in multicast packets transmitted along the formed multicast tree down to the BS, which de-tunnels and forwards them towards the MN. Data packets originating from the MN are dealt with as normal IP packets. Maintenance of the multicast routing tree is achieved by storing soft states in each router indicating that this router is being used. These states have to be refreshed hop by hop periodically, i.e. each router sends *Keepalive* messages to its neighbors participating in the multicast tree.

After the MN detects a movement to a new FA, represented by a BS in the domain as mentioned previously, it sends a new *Reg_Rqst* message to it. This message contains the multicast CoA configured previously since the MN retains this CoA as long as it stays inside the domain. The new BS forwards the *Reg_Rqst* toward the gateway and starts the multicast tree-joining procedure that implies exchanging a *CBT Join Request* and a *CBT Join Ack* messages with the crossover router that is defined as the nearest router shared between the old and the new routing trees. The *Reg_Rqst* message will not be forwarded beyond the gateway. After the BS forms the new routing tree, it sends a *MMP Instruct* message to the old BS forcing it to start the CBT-specific process of replacing the old path from the multicast tree. The handoff procedure is provided in figure 3.29.

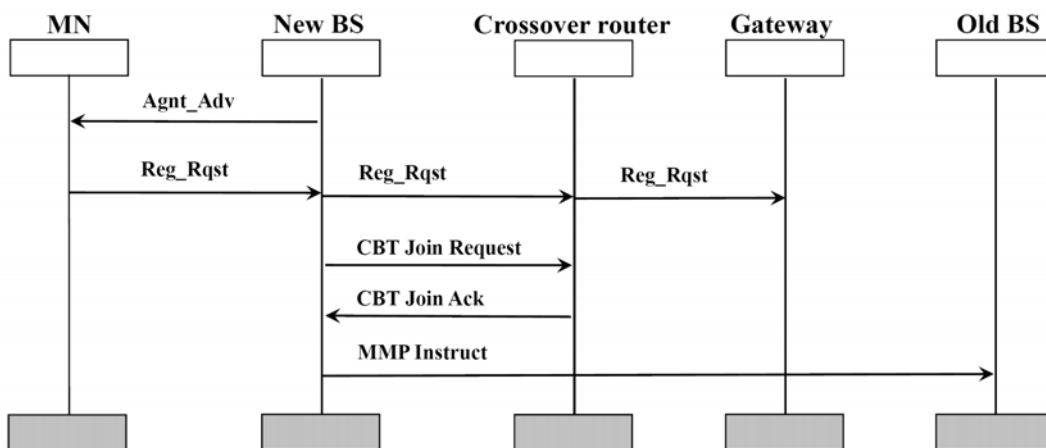


Fig 3.29: MMP handoff procedure

Paging aims normally at reducing the power consumption of idle MNs as well as the signaling overhead associated with the approach. MMP is not concerned with reducing the power consumption of idle MNs since MNs do not participate in MMP mechanisms. Thus, paging aims at reducing the signaling overhead inside the access network. More specifically, it tries to minimize the signaling resulting from refreshing the soft states in each router. Therefore, MMP defines two time values for the soft states stored in each router participating in the multicast tree, namely active and idle value. Each router remains active as long as it is used by at least one MN. If the active state has been expired without receiving any indicator that the router is being used, e.g. flow of data packets, *Keepalive* message, etc., the router switches to idle state and does not send *Keepalive* messages any more.

MMP confines mobility processing inside the administrative domain. The handoff latency is restricted to the time the new BS requires to join the multicast tree. Clearly, this results in reducing the handoff latency and other related problems. However, MMP suffers from the single point of failure problem. In addition, it assumes that MNs conform correctly to multicast CoAs and to no receipt of replies for the transmitted *Reg_Rqst* messages, which are

not forwarded beyond the gateway. This requires, of course, updating the MNs, which complicates the deployment of this approach. All routers locating inside the domain should be multicast-enabled. A main drawback of MMP is the lack of handling security issues. The gateway manipulates *Reg_Rqst* messages before sending them to the HA. This requires a SA to be established between the gateway and the HA. *Reg_Rqst* messages sent during handovers are used to trigger updating the multicast tree only. They are discarded by the gateway that does not have any mechanism defined by MMP to check the replay protection. Clearly, this makes it possible for an attacker to send a replayed *Reg_Rqst* message.

3.2. Network-Based Mobility Management

Network-based mobility management techniques assume a minimal or even no mobility support in MNs. The access network should perform all tasks related to mobility on behalf of the MNs. The fact that no update to MNs' IP stacks is required enables nodes with legacy IP stacks to be mobile. Clearly, this increases the number of MNs that can profit from such solution. The IETF NETwork-based Localized Mobility Management (NETLMM) working group [NETWG] is working on developing network-based mobility management solutions. Techniques applied to realize a network-based mobility management can be either macro or micro mobility management techniques. The following provides an in-depth insight into the both.

3.2.1. Network-Based Macro Mobility Management Approaches

Network-based macro mobility management solutions aim at supporting global mobility without any involvements of MNs. The best-known solution is Proxy MIPv6 developed by the NETLMM working group [GLD08]. The network topology assumed by Proxy MIPv6 is shown in figure 3.30. As shown in this figure, two new entities are introduced, namely a Localized Mobility Anchor (LMA) and a Mobile Access Gateway (MAG). LMA is a router that manages the mobility inside a localized domain and contains mobility bindings of MNs. It is comparable to the HA in MIP. The MAG is a network element that terminates a specific edge link and tracks the MNs' IP-level mobility between edge links. The MAG is comparable to the FA in MIPv4.

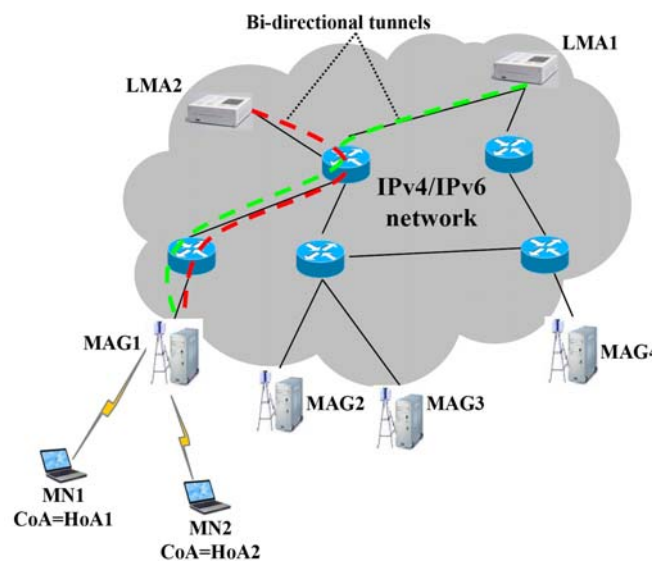


Fig 3.30: Proxy MIPv6 network topology

As soon as the MN powers on or moves into a new Proxy MIPv6 domain, the current MAG identifies the MN and checks if the MN is authorized to use the network-based mobility management service or not. It is assumed that the MAG can check the MN's identity through some AAA [LGG00] procedures. If the MN is authorized to use the network-based mobility management service, an IP address is configured depending on the Home Network Prefix (HNP), the default router address on the link (MAG) and other related configuration parameters. The configured IP address will not be changed even after changing the point of attachment. The MN may operate in IPv4 mode, IPv6 mode or in dual IPv4/IPv6 mode. Based on the operating mode, the MN obtains an IPv4 address, an IPv6 address or dual IPv4/IPv6 addresses. The specifications related to IPv4 support for Proxy MIPv6 are given in [WGu09].

Figure 3.31 shows the control messages exchanged during the first registration with the Proxy MIPv6 domain. As soon as the MN powers on and attaches to a point of attachment in a certain Proxy MIPv6 domain, the current MAG requests the MN-ID and its profile, e.g. from an AAA server. Following this, a Proxy Binding Update (**PBU**) message is sent from the current MAG to the LMA, which checks the identity of the MN and, if successful, responds with a Proxy Binding Acknowledgement (**PBAck**) message including the MN-HNP. Subsequently, the LMA creates a binding cache entry for the MN and establishes a bidirectional tunnel to the current MAG. Upon receipt of the **PBAck** message by the MAG, it sets up a bidirectional tunnel to the LMA. At this point, the MAG will be able to emulate the MN's home link. This is achieved through sending a unicast **RA** message with the MN-HNP as the hosted on-link-prefix. The MN will believe that it is located in its home network and configures its IP address accordingly.

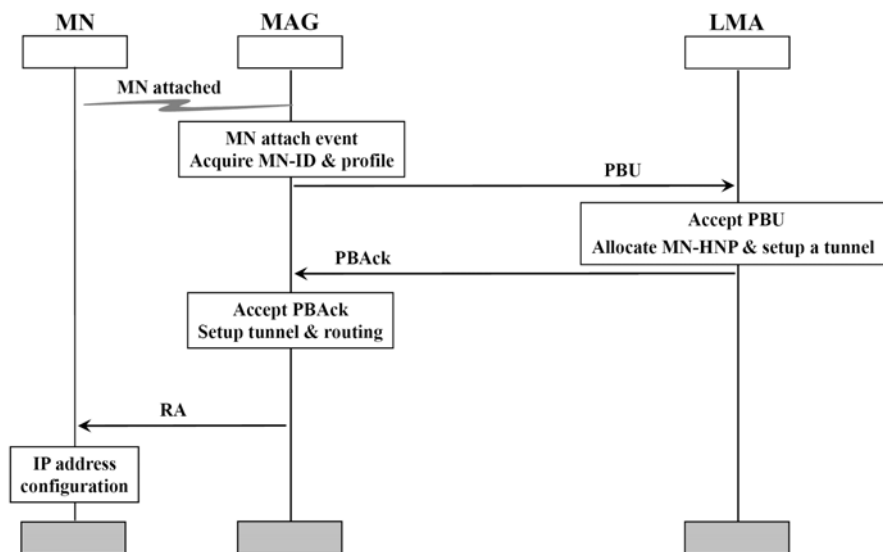


Fig 3.31: Proxy MIPv6 initial registration procedure

Data packets destined to the MN are delivered to the LMA based on the standard IP routing. The LMA intercepts the packets and tunnels them to the current MAG, which decapsulates and forwards them to the MN. Data packets originating from the MN are transmitted to the current MAG that tunnels them to the LMA, which in turn decapsulates and forwards them toward their destination.

When the MN hands off to a new MAG, a new wireless link with the new MAG is established, while the old wireless link is broken down. The old MAG detects the MN's detachment from the link, removes the binding and routing state for that MN and signals the MN's detachment to the LMA by sending a De-Registration PBU (**DeReg PBU**) message as shown in figure 3.32. The **DeReg PBU** message is acknowledged by a **PBAck** message.

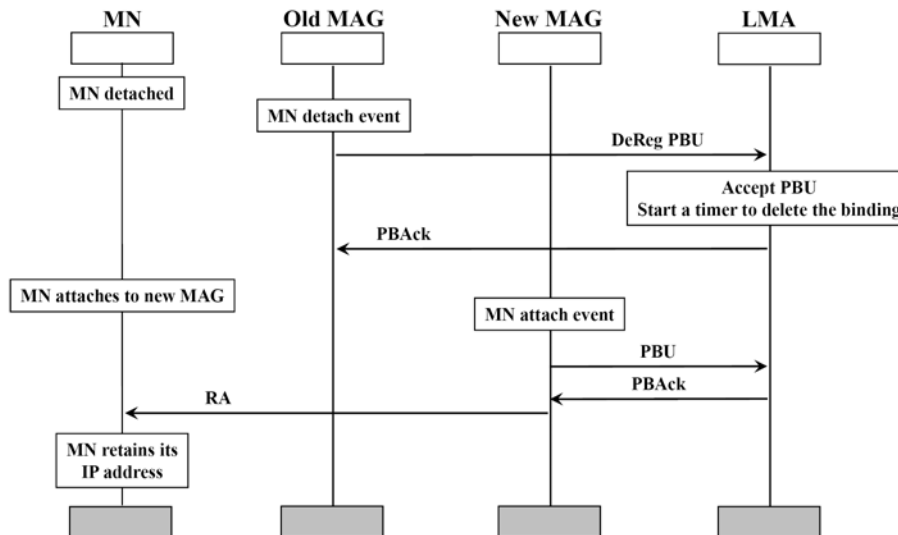


Fig 3.32: Proxy MIPv6 handoff procedure

In order to allow for a smooth handoff, the LMA keeps the MN's binding for some time, after receipt of the *DeReg PBU* message, expecting to receive a *PBU* from the new MAG. As soon as the MN attaches to the new MAG, a *PBU* and a *PBAck* message are exchanged between the new MAG and the LMA. Once signaling is completed, the new MAG sends a unicast *RA* message to the MN with the MN-HNP as the hosted on-link-prefix. This makes the MN believe that it is still using the same link and no address configuration is required.

The main advantage of Proxy MIPv6 is the transparency to MNs, which highly increases its usability. This is the reason why the 3GPP standardization group [3GPP] plans to integrate this protocol as a global mobility management protocol between different access networks in the future LTE standard. An analytical analysis of Proxy MIPv6 compared to FMIPv6 is provided in [DMG08] and [Get08]. The obtained results can be summarized as follows: the focus of Proxy MIPv6 is on supporting a network-based mobility management, not on achieving fast or seamless handoffs. Due to the transparency to MNs, there is no transmission of control messages related to mobility on wireless links. This improves the robustness of this protocol against control messages dropping and reduces the signaling cost resulting from location updates. Proxy MIPv6 requires data packets to be routed to the LMA and subsequently tunneled towards the current MAG, which de-tunnels and transmits the packets to the MN. Terminating the tunnel in the MAG instead of the MN, as is the case with FMIPv6, reduces the data traffic volume sent over the wireless link. In contrast to FMIPv6, route optimization can not be realized with the basic Proxy MIPv6 protocol.

3.2.2. Network-Based Micro Mobility Management Approaches

In addition to managing the mobility without any interaction with MNs, these techniques aim at accelerating the mobility management by localizing the processing of mobility inside the domain. Terminal Independent MIP (TIMIP) [GEN01], [EGV03] is the best-known protocol belonging to this category. It uses a hierarchical network topology and relies on principles similar to those of CIP and HAWAII. The transparency to MNs implies that some network elements are responsible for executing mobility procedures typically executed by the MN itself. The behavior of these network elements is referred to as a "surrogate behavior", see [Gus99]. To provide transparency to MNs, TIMIP couples layer 3 and layer 2 handoff mechanisms at the APs. In order to recognize legacy MNs in a TIMIP domain, all MNs have to be registered off-line with the Access Network Gateway (ANG) that controls the domain. The MNs' registration information is forwarded to all APs in the TIMIP domain. This lets any

AP able to recognize the IP address of the newly associated MN depending on layer 2 information, e.g. the MAC address. When the MN powers on in a TIMIP domain, a routing path is created along the hierarchy of ARs up to the ANG. This procedure takes place as follows, also see figure 3.33:

1. The MN executes a layer 2 handoff and establishes a radio link with the new AP.
2. At the AP, the layer 2 notifies the IP layer of the newly associated MN.
3. The new AP sends a **RoutingUpdate** message to the AR on the next hierarchy level. The AR updates its routing table and replies with a **RoutingUpdateAck** message back to the AP. Simultaneously, it sends a **RoutingUpdate** message to the AR on the next higher hierarchy.
4. The exchanging of **RoutingUpdate** and **RoutingUpdateAck** messages takes place along the hierarchy, up to the ANG.

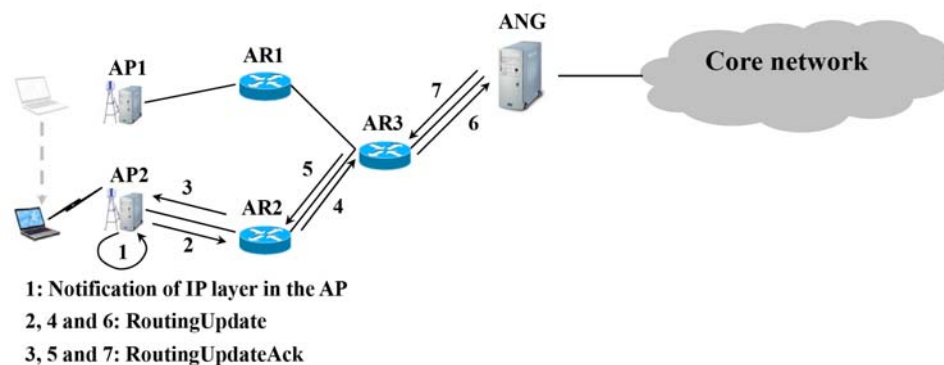


Fig 3.33: TIMIP inter-domain handoff procedure

Routing entries are stored in a soft state and refreshed using data packets sent from MNs. When no data packets are transmitted, the current AR sends an ICMP **EchoRequest** message to the MN upon expiration of the MN's routing entry. The MN answers by an ICMP **EchoReply** message causing the routing entry to be refreshed. If no **EchoReply** message is received, the MN's routing entry will be considered as invalid and will be deleted from the AR.

In case an authentication is required, the MN should implement a special security application. In addition, the MN should use a database of authentication keys for different TIMIP domains, to which the MN is allowed to associate. The authentication takes place during the second step of the power on procedure and immediately after the IP layer is notified of the newly associated MN. The AP sends a **SignatureRequest** message to both the MN and the ANG. The **SignatureRequest** message is authenticated separately by the MN and by the ANG. A **SignatureReply** is sent from the MN as well as from the ANP to the current AP. If the signature of the two **SignatureReply** messages is the same, the AP proceeds with updating the routing entries for the MN.

When the MN moves inside the TIMIP domain, it executes the steps described above (step 1 to 4). However, the **RoutingUpdate** messages are forwarded and processed up to the crossover router only, which will be the ANG in the worst case. Upon notification of the crossover router of the handoff, it sends a **RoutingUpdate** message addressed to the MN hop by hop via the old routing path. Any AR on this path deletes the MN's entry and acknowledges this message by a **RoutingUpdateAck** message back to the AR that has initiated the **RoutingUpdate** message (see figure 3.34).

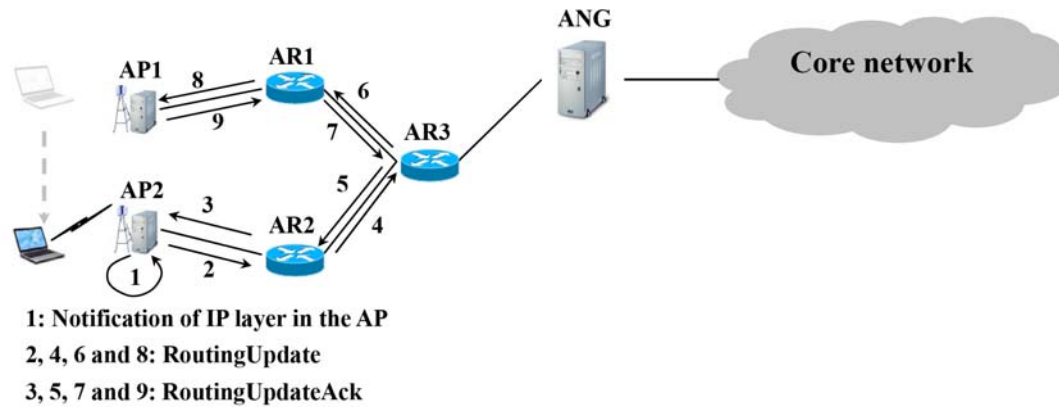


Fig 3.34: TIMIP intra-domain handoff procedure

Data packets are forwarded to the ANG since its address is registered with the HA as the CoA of the MN. The ANG decapsulates the packets and delivers them to the MN according to the routing path established in the TIMIP domain. Data packets originated from the MN are dealt with as normal IP packets.

Similar to most micro mobility management techniques, TIMIP relies on MIP to support macro mobility management between different TIMIP domains. However, in contrast to these techniques, TIMIP takes the terminals without MIP implementations into account. For these terminals, a TIMIP-integrated extension to the MIPv4 architecture, named surrogate MIPv4 (sMIPv4), has been proposed [EGV03]. sMIPv4 extends the functionalities of MIPv4 agents to execute a surrogate mobility management. In other words, the ANG works as a MIPv4 proxy on behalf of terminals without MIP support and generates all required MIPv4 signaling as a MIPv4-supporting MN would do [GEN01].

Like HAWAII, TIMIP forwards data packets always along the shortest path inside the domain. This is especially useful for the MNs locating in the domain and communicating with other MNs in the same domain, i.e. data packets are forwarded hop per hop from the source MN towards the destination MN without bypassing the HA of the destination MN for example. In addition and similar to CIP, TIMIP uses data packets to refresh routing entries inside the network. Explicit signaling is sent to maintain the routing entries in case the MN is in idle mode. Simulation studies presented in [EVN04] show that for the intra-domain mobility, TIMIP and HAWAII perform best. However, due to the existence of out-of-order packets during the handoff when employing HAWAII, TIMIP performs slightly better than HAWAII. The worst performance is experienced by CIP and MIPRR. Concerning inter-domain mobility, TIMIP, CIP, HAWAII and MIPRR show comparable performance. Similar to HAWAII, TIMIP requires all nodes in the domain to be mobility-aware. The single point of failure exists for TIMIP too. A main drawback of TIMIP is the way it handles security. Each MN should implement a special security application and should use a database containing authentication keys for different TIMIP domains, where it is allowed to associate. Clearly, this limits the applicability of TIMIP.

In [Est07] and [EVN06] an enhancement for TIMIP, enhanced TIMIP (eTIMIP), is proposed. Beside the transparency to MNs, already known from TIMIP, eTIMIP is network independent too. It uses an overlay network to provide transparent micro mobility in all existing networks, see figure 3.35.

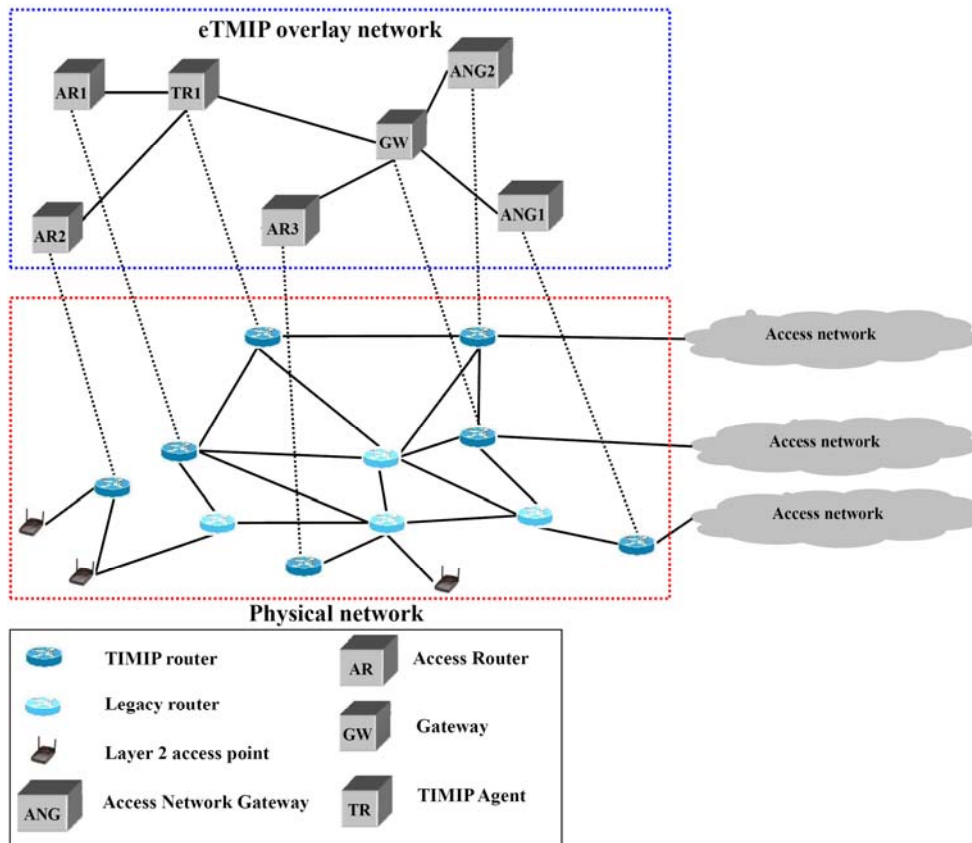


Fig 3.35: eTIMIP overlay network

An overlay network is built on the top of a physical network using software agents maintaining routing information. The software agents build the eTIMIP access network by establishing and maintaining a logical tree among them. Therefore, eTIMIP puts no restrictions on the network shape. The same TIMIP procedures are used to support mobility inside and between eTIMIP domains.

Simulation results presented in [EVN06] show that increasing the number of mobility-aware agents in the domain improves the efficiency of eTIMIP. The location of these mobility-aware agents has a significant impact on the performance. This has in turn a negative impact on the transparency. The conclusion is that there is a trade-off between transparency and efficiency. eTIMIP shows very good performance even if network impairments are present. It inherits, however, the problems known from TIMIP. Notice that not every node in the domain needs to be mobility-aware as is the case by TIMIP.

3.3. Conclusion

The main results obtained from the analysis conducted in this chapter can be summarized as follows:

1. Terminal-based mobility management techniques rely on interactions with MNs to support mobility management.
 - a. The main aim of MIP is to provide support for the TCP/IP protocol suite in a wireless environment. Unfortunately, it tends to suffer from significant performance degradations during handoffs, which make it inadequate for delay-sensitive applications. It is adequate for supporting global mobility management, also called macro mobility.

- b. The fact that MIP does not meet real-time requirements has prompted the development of other macro mobility management techniques. These techniques aim at overcoming the shortcomings of MIP either by use of many interfaces in the MN, utilization of layer 2 triggers or resuming the communication depending on a router on the previous link until completing the handoff. Using of multiple interfaces in the MN, as is the case for MosquitoNet extensions, improves the performance and is mainly adequate for vertical handoffs. Utilizing layer 2 triggers, as is the case for pre- and post-registration methods¹ and FMIPv6, may eliminate the layer 3 handoff latency. Clearly, such solutions violate the separation between the layers of the TCP/IP reference model and depend on the technology. Forwarding of data packets from the old subnet to the new one results in satisfactory performance during the handoff. This requires, however, determining the new subnet accurately. Otherwise, data packets have to be delivered to many candidates, as is the case for proactive handoff methods, which generates lots of overhead.
- c. Another approach to solve the handoff problems present with MIP is to localize the mobility management as implemented by so called micro mobility management protocols, which fall into two categories, PAA and LERS.
 - i. PAA approaches localize the mobility processing inside an administrative domain. In addition, some approaches, e.g. S-MIP, benefit from utilizing layer 2 triggers. Intermediate nodes are introduced to process the mobility inside the domain locally. Thus, the HA is involved only when the MN moves between different administrative domains. These approaches typically depend on a hierarchical network structure. This hierarchy, however, may be dynamic, so that no restrictions are put on the physical network structure, as is the case by dynamic regional registration approaches and AFA. A well-known problem of most of these approaches is the presence of a single point of failure, which results from the dependency on one intermediate node in processing mobility and forwarding of data packets from and to the domain. This problem can be avoided either by introducing a backup agent that can replace the crucial intermediate node in the case of a failure, by using multiple intermediate nodes or by supporting a dynamic hierarchical network structure. Localizing the mobility management inside an administrative domain provides satisfactory performance and very good scalability. Very good performance can be achieved by the approaches that benefit from layer 2 triggers.
 - ii. LERS accelerates the handoff by localizing the mobility processing and, with some variants, by using layer 2 triggers. LERS solutions are categorized in per-host and multicast-based mobility management solutions. Per-host forwarding schemes employ a specialized path set-up protocol and a MN-specific location database, i.e. routing caches, in domain nodes. CIP and HAWAII require a hierarchical network topology, while MEHROM can be employed in hierarchical and mesh topologies. The single point of failure is a well-known problem in these approaches too. The performance achieved by these approaches can be good to very good, especially when utilizing layer 2 triggers. Similar to PAA approaches, these approaches scale very well.

¹ Pre- and post registration methods are considered as macro mobility management solutions, if they are used to improve the performance of MIPv4. However, as outlined in section [3.1.1.3](#), these methods can also be used to accelerate the handoff of micro mobility management approaches, e.g. MIPRR protocol. In this case, pre- and post-registration methods can be considered as micro mobility management approaches.

Multicast-based approaches utilize point-to-multipoint connections to support location-independent addressing and routing. The main idea is to build a routing tree and modify it according to MNs' movements. Layer 2 triggers are utilized by some approaches to optimize the performance. Some approaches make restrictions on the network topology, e.g. MMP, while others do not, e.g. the Daedalus proposal. MMP approach suffers from the single point of failure, while the Daedalus approach does not. The performance of these approaches can be good to very good, especially when utilizing layer 2 triggers. Considering the scalability, the domain-based approaches scale well, while others do not.

2. Network-based mobility management is supported by Proxy MIPv6 and TIMIP. The mobility processing is executed without any interactions with MNs. While Proxy MIPv6 is used to support global mobility, TIMIP is a micro mobility protocol. These solutions seem to be promised. However, they are under development and some issues are still open, e.g. security, route optimization, etc.

A detailed comparison between the protocols described in this chapter with respect to the handover management, paging support, new nodes that should be introduced to the network to enable employing a mobility management protocol, nodes that should be updated when a mobility management protocol is employed, network topology deployed, dependency on layer 2 triggers, usage of a tunnel to forward data packets to the new location of the MN, handover performance and load balancing is presented in table 3.1. The handover management can be either local or global. Clearly, mobility management protocols that achieve a local mobility support are faster than those, which achieve a global mobility support. Considering the paging support, protocols supporting paging can serve a large number of MNs. With respect to the new nodes that should be introduced to the network, the less the number of new nodes, the more the practicability of the protocol and the less the cost resulting from employing this protocol. Notice that we consider the nodes that should be introduced to the access network additional to that currently known from MIP, i.e. beyond the HA and FA/AR. For the nodes that should be updated when a mobility management protocol is employed, the protocols requiring fewer updates are better than those requiring doing lots of updates. Notice that the new nodes that should be introduced to the network are not considered here. For the network topology deployed, either a hierarchical topology is used or constraints are put on the topology. Notice that the protocols requiring a hierarchical network topology work properly only deploying this topology. In contrary, the protocols that do not put any restrictions on the topology can be employed in a mesh or in a hierarchical network. However, mostly the best performance is obtained deploying a mesh topology. Taking the dependency on layer 2 triggers into account, the protocols that depend on these triggers are mostly technology-dependent and violate the separation between the layers of the TCP/IP reference model. Tunneling of data packets to the new location of the MN implies adding extra overhead to the data forwarded, which is not desired especially for applications such as VoIP. Considering the handover performance, the protocols achieving better performance are clearly preferable. For the load balancing, the protocols that support load balancing are more robust than others having no load balancing support.

The analysis achieved in this chapter showed that there is lots of prior work attempting to provide scalable and robust techniques able to satisfy real-time requirements. The techniques have tried to improve the performance by making some constraints either on access networks or MNs. Therefore, a new technique is required that completes handoffs quickly without affecting the end-to-end performance of ongoing applications and neither puts restrictions on the topology nor on MNs. The new technique should meet, in addition to the requirements presented in section [2.3](#), the following requirements:

1. Localization of mobility management without putting any restrictions on the physical network topology and without introducing any new intermediate nodes beyond currently known from the standard layer 3 mobility management solution, i.e. MIP.
2. Making use of layer 2 triggers, if available, while keeping in mind that the negative impact of layer 2 triggers timing should be minimized. Moreover, the new technique has to work well even if layer 2 triggers are not available.
3. Providing high robustness by means of fast failure recovery mechanisms, especially for control messages dropping.
4. Achieving a very good performance for the MNs moving at low as well as high speeds. This should be even guaranteed for wireless networks with small cells.

To meet these requirements, we have developed a new network layer mobility management solution (MIFA) [[DMi04](#)], [[DMX04](#)], which is the topic of the next chapter.

Approach	Handover management	Paging support ¹	New nodes that should be added	Nodes that should be updated	Network topology	Dependency on layer 2 triggers	Usage of a Tunnel	Expected handover performance ²	Load balancing
MIPv4 with route optimization extension	<ul style="list-style-type: none"> Locally by the old and new FA 	N	/	MNs, FAs, CNs, HA	No restrictions	N	Y	M:G	N
Pre-registration	<ul style="list-style-type: none"> Globally by the HA 	N	/	MNs, FAs	No restrictions (when used to improve MIPv4)/hierarchical (when used to improve MIPRR)	Y	Y	G:VG	N
Post-registration	<ul style="list-style-type: none"> Locally by the old and new FA 	N	/	FAs	No restrictions (when used to improve MIPv4)/hierarchical (when used to improve MIPRR)	Y	Y	VG	N
MosquitoNet extensions	<ul style="list-style-type: none"> Globally by the HA 	N	/	MNs, HA	No restrictions	N	Y	B:M	N
FMIPv6	<ul style="list-style-type: none"> Locally by the old and new AR 	N	/	MNs, ARs	No restrictions	Y	Y	VG	N
P-MIP	<ul style="list-style-type: none"> Globally by the HA 	Y	/	MNs, FAs, HA	No restrictions	N	Y	B	N
Proactive handoff using neighbor graph	<ul style="list-style-type: none"> Locally by the old FA and neighboring ones 	N	/	MNs, FAs	No restrictions	Y	Y	VG	N

¹ Y: yes, N: no

² B: bad, M: middle, G: good, VG: very good. If this field is written somehow like M:G, this means a variation between middle and good.

MITHv4	<ul style="list-style-type: none"> Locally by the old and new FA 	N	/	MNs, FAs	No restrictions	Y	Y	VG	N
FA-assisted handoff	<ul style="list-style-type: none"> Globally by the HA 	N	/	FAs, HA	No restrictions	Y	Y	G:VG	N
Proactive handoff method with motion prediction	<ul style="list-style-type: none"> Locally by the old FA and predicted ones 	N	/	MNs, FAs, HA	No restrictions	Y	Y	VG	N
Proactive mobility framework for MIPv6	<ul style="list-style-type: none"> Locally by the old RDC and the RDCs locating in the routing neighborhood 	N	RDCs	MNs	No restrictions	Y	Y	VG	N
MIPRR	<ul style="list-style-type: none"> Locally by the GFA 	N	RFAs, GFA	MNs, FAs, HA (if dynamic GFA assignment is supported)	Hierarchal	N	Y	M:G	N
HMIPv6	<ul style="list-style-type: none"> Locally by the MAP 	N	MAP	MNs	Hierarchal	N	Y	M:G	N
Dynamic regional registration	<ul style="list-style-type: none"> Locally by the first visited FA Globally after visiting K_{opt} different subnets 	N	/	MNs, FAs, HA (if dynamic GFA assignment is supported)	No restrictions	N	Y	M:G	Y
AFA	<ul style="list-style-type: none"> Locally by the AFA Globally by the HA for indirect registrations 	N	/	MNs, FAs, HA	No restrictions	N	Y	M:G	Y
S-MIP	<ul style="list-style-type: none"> Locally by the MAP and the DE 	N	MAP, DE	MNs, ARs	Hierarchal	Y	Y	VG	Y

BCMP	• Locally by the ANPs	Y	BARs, ANPs, BMGs	MNs	Hierarchal	Y (when performing a handoff preparation phase)	Y	G:VG	N
TeleMIP/IDMP	• Locally by the mobility agent	Y	mobility agents, SAs (by IDMP)	MNs, FAs (by TeleMIP)	Hierarchal	Y (by IDMP)	Y	G:VG	Y
CIP	• Locally by the gateway	Y	Gateway	MNs, BSs, all routers in the domain	Hierarchal	Y (in semi-soft handoff)	N	G:VG	N
HAWAII	• Locally by the domain root router	Y	Domain root router	MNs, BSs, all routers in the domain	Hierarchal	Y	N	G:VG	N
MEHROM	• Locally by the domain gateway	N	Domain gateway	MNs, ARs, all routers in the domain	No restrictions	N	N	G:VG	N
Daedalus proposal	• Locally updating by the multicast group associated with the MN	N	/	MNs, BSs/FAs, HA	No restrictions	Y	Y	VG	N
MMP	• Locally updating by the routing tree inside the domain	N	Domain gateway	All routers in the domain	Hierarchical	N	Y	G:VG	N
Proxy MIPv6	• Globally by the LMA	N	LMA, MAGs	/	No restrictions	N	Y	M	N
TMIP/eTMIP	• Locally by the ANG	Y	ANG	APs, all routers in the domain by TIMIP and a part of them by eTIMIP	No restrictions (by eTMIP)/hierarchical (by TMIP)	Y	N	VG	N

Tab 3- 1: Comparison of the studied mobility management protocols with respect to the handover management, paging support, new nodes that should be introduced to the network, nodes that should be updated, network topology, dependency on layer 2 triggers, usage of a tunnel, handover performance and load balancing

4. Mobile IP Fast Authentication Protocol (MIFA)

The analysis in chapter 3 demonstrated a need to develop a network layer mobility management solution able to achieve seamless or even lossless handoffs without constraining the network or introducing any new intermediate nodes. This has led to the development of a solution named MIFA, which supports a continuous communication between the CN and the MN while the registration with the HA, and possibly the CN¹, is in progress. This chapter describes our solution and provides the specification for IPv4 (MIFAv4) and IPv6 (MIFAv6) networks. In the rest of this dissertation, the term MIFA is used where issues relevant for both MIFAv4 and MIFAv6 are discussed. Otherwise, the terms MIFAv4 and MIFAv6 are used explicitly.

This chapter is structured as follows: section 4.1 discusses the basic idea of MIFA. A detailed description of MIFAv4 focusing on its operation modes, error recovery mechanisms and security considerations is given in section 4.2. This includes the formal specification with SDL for the most important parts of the protocol. Section 4.3 describes MIFAv6, while section 4.4 summarizes this chapter. For more details and related information, the reader is referred to the appendixes: appendix A provides a brief description of all control messages and extensions used in MIFAv4 and MIFAv6. Appendix B describes three methods that can be applied to establish MIFA neighbor groups. Appendix C provides a more detailed SDL specification for MIFAv4.

4.1. Basic Idea

Use of neighborhood: MIFA utilizes the fact that, in typical networks, the movement of MNs from any subnet is limited to a small set of neighboring subnets. This fact aids in the prediction of MNs movements inside an access network. To illustrate this, let us consider the scenario shown in figure 4.1, which is rather simple and presents a one-way street.

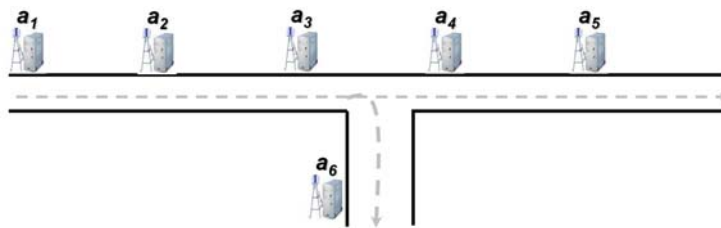


Fig 4.1: An example one-way street scenario

Clearly, MNs will move from the subnet a_1 to a_2 and from a_2 to a_3 . The MNs present in the range of a_3 will typically move either to a_4 or a_6 . Of course, real scenarios are more complex. However, even in complex scenarios and large access networks, MNs movements can, to some extent, be predicted depending on this fact. MIFA aims at utilizing this prediction to accelerate layer 3 handoffs between different subnets. More specifically, the data required to authenticate a MN in neighboring subnets are provided by the current subnet to the neighboring ones to prepare them for the registration of the respective MN.

¹ Considering MIFA for IPv4 networks, the mobility binding is updated only at the HA. However, regarding MIFA for IPv6 networks, the MN should update its mobility binding at the HA as well as the CN.

Due to the availability of these authentication data at neighboring subnets, they can quickly re-authenticate the MN after the layer 3 handoff and, thus, enable the MN to quickly resume its communication. This enables the HA to delegate the authentication of the MN to the new subnet. It should be mentioned, however, that this does not implicate the distribution of the shared secret between the HA and the MN ($K_{MN,HA}$) to the new subnet.

Neighbor authentication: to realize such local authentication, it is useful to build groups of neighboring subnets. A certain subnet is considered a neighbor of the current one if movements to this subnet from the current subnet are possible. MIFA requires each subnet to build a set of neighboring subnets called a Layer 3 Frequent Handoff Region (L3-FHR). A L3-FHR does not necessarily comprise all adjacent subnets, e.g. in the case of physical obstacles preventing movements between adjacent subnets areas. L3-FHRs can be determined either statically (e.g. by means of algorithms such as the neighboring graph algorithm [MSA04a] or others [PCh02]) or dynamically by observing MNs movements inside the access network. More information about how L3-FHRs can be built is provided in appendix B. Clearly, there must be SAs between the subnets of each L3-FHR. The SAs can be established either statically (e.g. by the network administrator) or dynamically (e.g. by the network itself as described in [PJA00] and [PCa01]). Other techniques such as AAA may be used as well. Each subnet is responsible for maintaining its own L3-FHR. This can be achieved simply by exchanging periodic *Hello* messages with other L3-FHR members. If a neighboring subnet is no longer available, it should be removed from the L3-FHR. Because network topologies are rather static, the periodic interval of sending *Hello* messages can be long, perhaps hours. Throughout this chapter, the L3-FHR of the current subnet will be referred to as the current L3-FHR.

4.2. Mobile IP Fast Authentication Protocol for IPv4 (MIFAv4)

4.2.1. Operation Overview

Figure 4.2 shows an example network employing MIFAv4. This figure shows that there are no new nodes introduced to the network beyond the nodes already known from MIPv4, i.e. FAs and a HA.

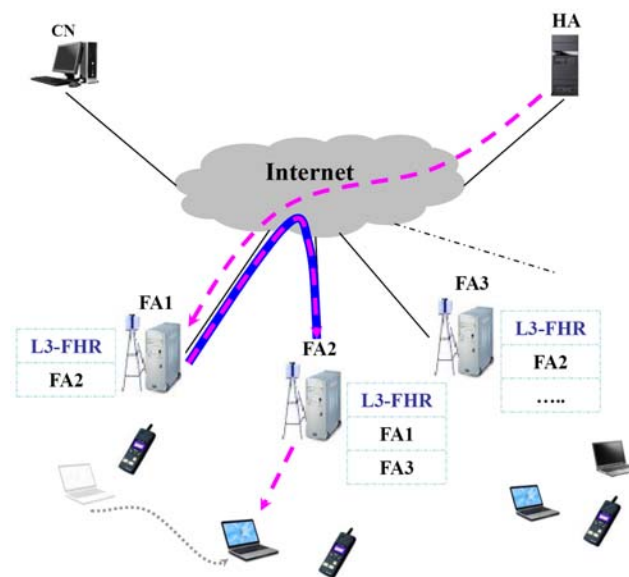


Fig 4.2: An example network that employs MIFAv4

MIFAv4 can either be operated in reactive or predictive mode. The reactive mode is suitable for MNs able to communicate with one AP only. Conversely, the predictive mode is suitable for MNs able to listen to more than one AP simultaneously. If a L2-trigger is raised at the MN before the actual handoff occurs, the predictive mode is employed. Otherwise, MIFAv4 operates in reactive mode.

Predictive mode: when the MN expects a handoff to a new FA in the near future, it initiates the layer 3 handoff in advance. This is achieved simply by sending a *Reg_Rqst* message to the new FA via the old one. Afterwards, the MN executes the layer 2 handoff. Provided that the new FA is a member of the L3-FHR of the old FA, the new FA authenticates the MN based on the authentication information previously received from the old FA. After a successful authentication, the new FA notifies the old one. As a consequence, the old FA begins forwarding data packets destined to the MN to the new FA. After the MN has completed the layer 2 handoff, the new FA replies a *Reg_Rply* message towards the MN. Any packets buffered at the new FA will be forwarded to the MN directly after sending the *Reg_Rply* message.

Concurrently to the procedure described above, the new FA notifies the HA of the MN's new CoA. As a consequence, the HA stops tunneling data packets to the old FA and begins tunneling them to the new CoA. Notice that in our scheme the time required to inform the HA and establish a new tunnel to the new CoA is hidden from the application and has, in practice, no impact on performance. This represents a great advantage of MIFAv4. Although the time required to update the HA has no impact on the performance, updating the HA is necessary to optimize the route and exchange the information required to accelerate subsequent layer 3 handoffs. After completion of the layer 3 handoff, the new FA distributes the information required to authenticate the MN with neighboring FAs to all FAs defined in its L3-FHR.

Reactive mode: the operation of MIFAv4 in reactive mode is similar. However, the MN contacts the new FA directly after the layer 2 handoff has been completed instead of indirectly via the old FA as in predictive mode. The new FA authenticates the MN and subsequently replies a *Reg_Rply* message to the MN, which then resumes its uplink traffic. After the new FA has sent a *Reg_Rply* to the MN, it asks the old FA to establish a temporary tunnel to forward the MN's data packets to the new CoA¹. The forwarding remains in place until the HA is notified and a tunnel to the new FA is established. After finishing the layer 3 handoff, the information required to accelerate the subsequent layer 3 handoff is distributed to all FAs present in the L3-FHR of the new FA.

4.2.2. *Initial Registration Procedure*

When the MN is switched on or wants to connect to the network, it initially uses the regular MIPv4 procedure. The MN waits for an *Agnt_Adv* message from the agent serving the current subnet, which will be a FA in the case that the MN is away from home or the HA in the case that the MN is at home. Alternatively, the advertisement can be solicited by means of an *Agnt_Sol* message. The *Agnt_Adv* message is constructed according to the MIPv4 specification. One bit from the reserved bits in this message is, however, used as a flag to indicate the support of MIFAv4. In the following, this flag will be referred to as a *MI* flag.

Let us now assume that the MN is away from home. As soon as the MN receives the advertisement, it sends a *Reg_Rqst* message to the HA via the discovered FA. Again, the *Reg_Rqst* message is built according to the MIPv4 specification with one of the reserved bits

¹ If the MN has obtained a FA-CoA, the tunnel end-point will be the FA, which de-tunnels data packets and forwards them to the MN. However, if the MN has obtained a co-located CoA, the MN itself will be the tunnel end-point.

used as a *MI* flag. Setting this flag to 1 indicates that the MN prefers to use MIFAv4 in following registrations, see figure 4.3.

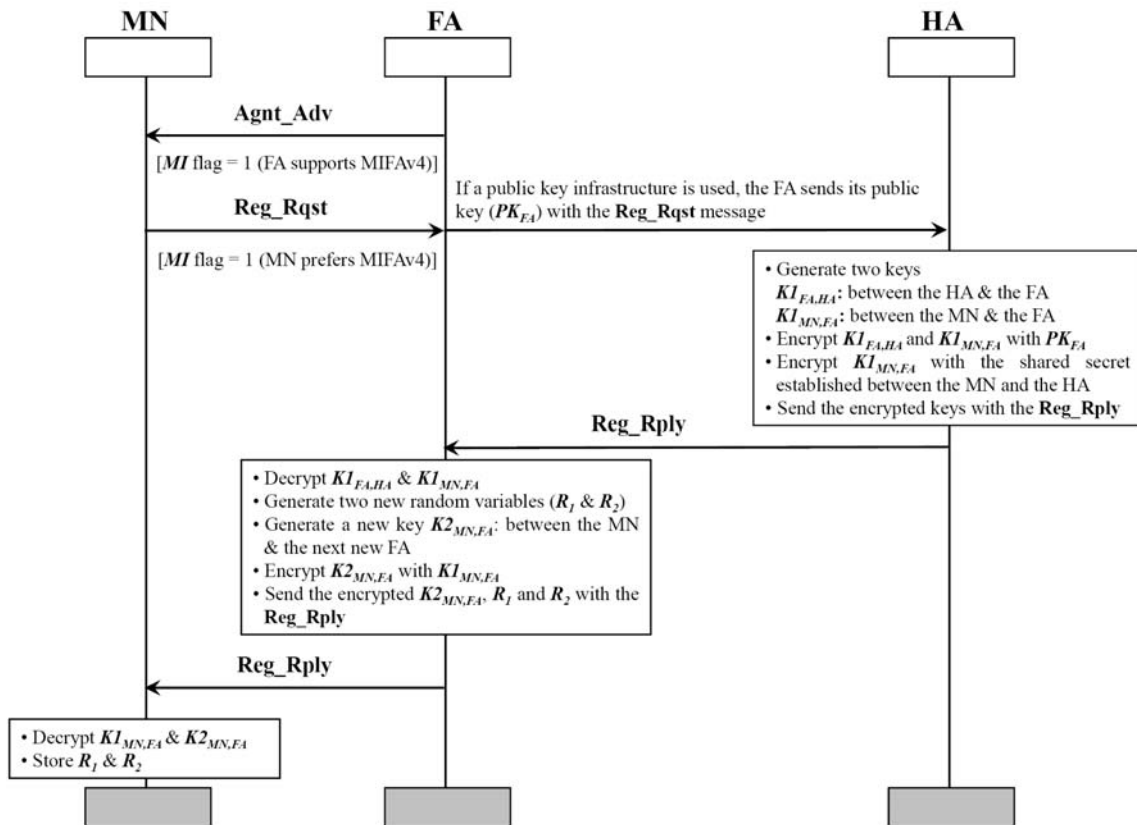


Fig 4.3: MIFAv4 initial registration procedure

Upon receipt of the **Reg_Rqst** message by the HA, the HA responds with a **Reg_Rply** message and by generating two SAs to be used to secure the control messages exchanged

- between the HA and the current FA (the key is $K1_{FA,HA}$) on one side and
- between the MN and the current FA (the key is $K1_{MN,FA}$) on the other side.

Both $K1_{FA,HA}$ and $K1_{MN,FA}$ are distributed to the current FA. $K1_{MN,FA}$ is sent to the MN as well. The keys can be distributed using any adequate key distribution infrastructure or mechanism, e.g. an AAA, the methods presented in [PCa01] and [PJA00], a public key infrastructure, etc. As a default, the keys distributed to the current FA are sent encrypted with its public key (PK_{FA}) as a part of the **Reg_Rply** message¹. However, the key distributed to the MN is also sent encrypted with the shared secret established between the HA and the MN as a part of the **Reg_Rply** message too. Key management is discussed in more detail in section 4.2.9.1.

When the FA receives the **Reg_Rply** message, it derives the two SAs, generates two random variables (R_1 and R_2) and another key ($K2_{MN,FA}$) to be used between the MN and the next new FA. Notice that the new FA in most cases will be a member of the current L3-FHR. $K2_{MN,FA}$ is then encrypted, after that, with $K1_{MN,FA}$ and added along with the generated random variables (R_1 and R_2) in suitable extensions to the **Reg_Rply** message. The new **Reg_Rply** message is then authenticated using $K1_{MN,FA}$ and transmitted to the MN. Upon receipt of the **Reg_Rply**

¹ Of course, this requires the current FA to send its public key to the HA. This can be done by sending the public key as an extension added to the **Reg_Rqst** message, see [PJA00].

message by the MN, the MN decrypts $K1_{MN,FA}$ and authenticates the message. If the authentication is successful and the registration is accepted by the HA, the MN records R_1, R_2 and $K2_{MN,FA}$ and proceeds with the initial authentication exchange procedure.

4.2.3. Initial Authentication Exchange Procedure

The current FA executes this procedure to obtain the data required to locally re-authenticate the MN during the next registration with the next new FA. During this procedure, a SA (the key is $K2_{FA,HA}$) is generated to secure the messages that should be exchanged between the HA and the next new FA, to which the MN may move in the future. The current FA sends, thereafter, a Movement Probability Notification (***M_P_Not***) message to the HA. This message includes the random variables (R_1 and R_2) and $K2_{FA,HA}$ encrypted with $K1_{FA,HA}$, which authenticates this message as well.

The HA then authenticates the ***M_P_Not*** message using $K1_{FA,HA}$. If the authentication is successful, the HA derives $K2_{FA,HA}$ and calculates two authentication values, referred to as $Auth_1$ and $Auth_2$ in the rest of this chapter. $Auth_1$ and $Auth_2$ are calculated by applying a hash algorithm (e.g. MD5, HMAC-MD5, etc.) on R_1, R_2 and additional information related to the MN (e.g. the home address, the HA address, etc.). As a default, the authentication values are calculated as follows:

$$Auth_x = HMAC - MD5 (K_{MN,HA}, R_x, MN - Home\ address, MN - MAC\ address, T_x)$$

$Auth_x$ may be $Auth_1$ or $Auth_2$. *HMAC-MD5* is the default hash function used by MIFAv4. $K_{MN,HA}$ is the shared secret between the MN and the HA. R_x stands for either R_1 or R_2 . T_x stands for a timestamp of the MN and the HA. Notice that because $Auth_1$ and $Auth_2$ are calculated in advance of the handoff, T_x does not indicate the current time of the MN or the HA. Instead, T_x is calculated as follows: $T_x = T_{ini} + N_{Reg} * T_{duration}$, where T_{ini} is the timestamp of the initial registration. N_{Reg} is a number in ascending order referring to the current registration, where the initial registration has the number 1. $T_{duration}$ is a time duration, which may be either constant or variable according to a certain function known to the MN and the HA. Notice that $Auth_x$ is not aimed at achieving replay protection. Such a protection is realized by means of other mechanisms, see section 4.2.9.5. Using timestamps implies, of course, that the MN should synchronize with the HA. In general, some operator-specific information or policies may also be used to generate the authentication values as well. The authentication values are used to authenticate the MN with the HA during the next registration. $Auth_1$ represents the authentication value the MN has to generate after the next movement upon sending a ***Reg_Rqst*** message, while $Auth_2$ represents the authentication value the HA must generate when responding to the MN with a ***Reg_Rply*** message towards the MN.

After the authentication values have been generated, the HA sends them encrypted with $K1_{FA,HA}$ to the current FA as part of a Movement Probability Acknowledgement (***M_P_Ack***) message. In addition to the authentication values, the ***M_P_Ack*** message also contains a HA features extension as well as a replay protection extension. The HA features extension contains some features of the HA (e.g. simultaneous binding, GRE, etc.) that can be

offered to the MN. These features are necessary to enable the new FA to decide whether the MN's requirements can be met or not. The replay protection extension is used to support a replay protection. The M_P_Ack message is authenticated using $K1_{FA,HA}$ and must determine whether the information distribution procedure, presented in section 4.2.4, should be executed or not. The initial authentication exchange procedure is shown in Figure 4.4.

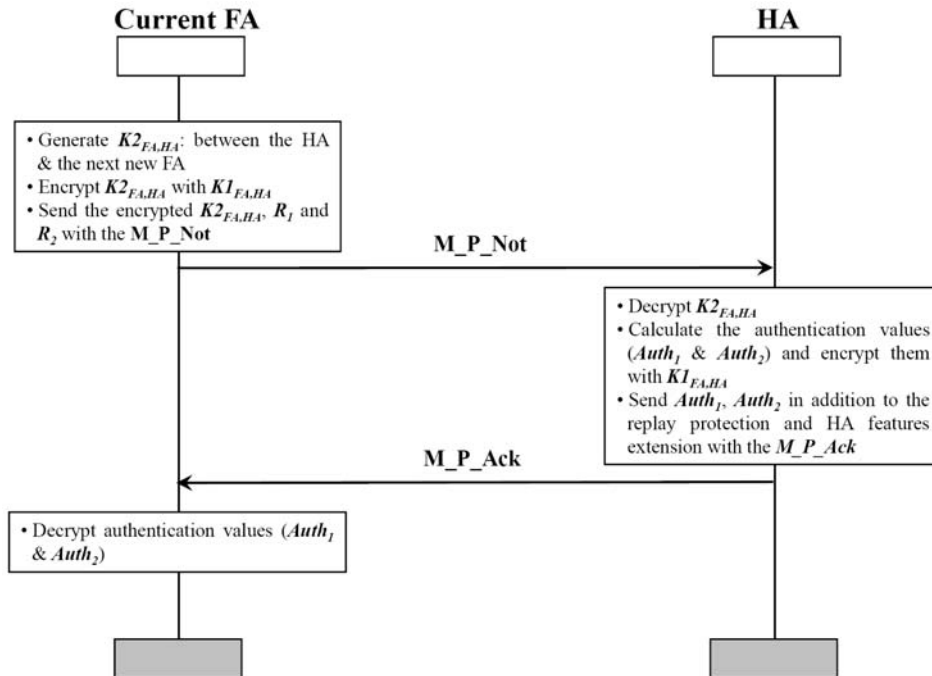


Fig 4.4: MIFAv4 initial authentication exchange procedure

4.2.4. Information Distribution Procedure

In order to notify the FAs in the current L3-FHR of a potential handoff of a certain MN, the current FA sends a M_P_Not message to each FA present in this L3-FHR. The message contains the information required to quickly re-authenticate the MN during registration with the next new FA. This information will be referred to as MN-specific data in the rest of this chapter and contains the SAs between the MN and the current L3-FHR members (the key is $K2_{MN,FA}$) on one side, and the SAs between these members and the HA (the key is $K2_{FA,HA}$) on the other side. In addition to these SAs, the MN-specific data contain the information sent from the HA to the current FA during the initial authentication exchange procedure. $K2_{MN,FA}$ and $K2_{FA,HA}$ are encrypted by means of the SA established between the FAs in the L3-FHR (the key is $K_{FA,FA}$), which also authenticates the M_P_Not messages. The MN-specific data are recorded in a soft state and must be refreshed periodically until the handoff occurs. The MN will be served after the handoff by one of the FAs present in the L3-FHR. This means that $K2_{MN,FA}$ and $K2_{FA,HA}$ will be used by one of the FAs in the current L3-FHR and deleted from the others when the keys lifetime expires. Each neighboring FA may acknowledge the M_P_Not by sending a M_P_Ack message. Sending a M_P_Ack message should be requested explicitly by the M_P_Not message. As a default, M_P_Ack messages are not sent. The motivation for this is to reduce signaling since the probability of messages dropping on wired links are very low. Even in case of dropping, MIFAv4 can handle this failure as discussed in sections 4.2.6 and 4.2.8. In order to refresh the MN-specific data, a *refresh* message is sent from the current FA to each FA in the current L3-FHR before the

expiration of the MN-specific data lifetime. The information distribution procedure is shown in Figure 4.5.

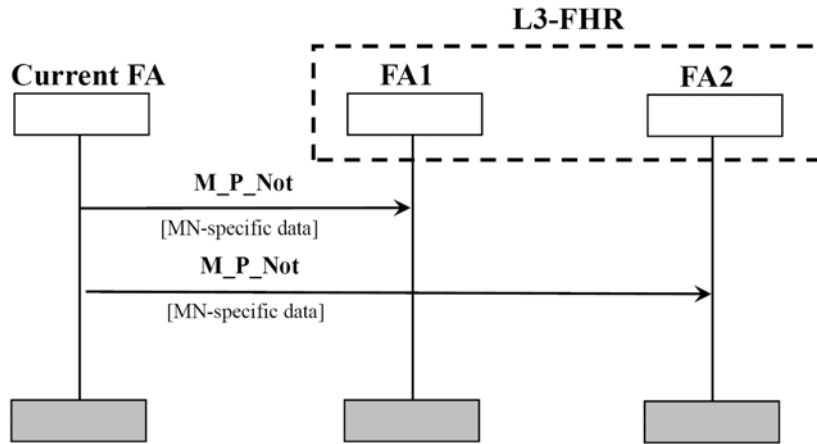


Fig 4.5: MIFAv4 information distribution procedure

The information distribution procedure is optional. As a default, this procedure will be executed. There are some cases, however, where the execution of this procedure will not improve the performance. For example, the information distribution procedure should not be executed for idle MNs or those that have a low packet arrival rate or do not move often, etc. In addition, the execution of this procedure may depend on some operator-specific policies.

4.2.5. Operation in Reactive Mode

Initiation of the layer 3 handoff: when the MN detects that the serving FA is no longer available, it listens to an *Agnt_Adv* message from a new FA. After receiving the *Agnt_Adv* message, the MN checks the *MI* flag. If this flag is set to 1, the MN proceeds with MIFAv4. Otherwise, the MN resorts to MIPv4. Resorting to MIPv4 aims to retain communication even with some extra latency. In case MIFAv4 is supported, the MN transmits a *Reg_Rqst* message to the new FA. This message should contain a MIFA authentication extension including the authentication value $Auth_1$, which is calculated by the MN using the random value (R_1) and the same hash function the HA uses. The *Reg_Rqst* message is authenticated using the SA established between the MN and the new FA (the key is $K2_{MN,FA}$).

Local authentication: the new FA authenticates the *Reg_Rqst* message using $K2_{MN,FA}$. If the authentication is successful, it compares the value of $Auth_1$ sent from the MN with the value of $Auth_1$ calculated previously by the HA and sent from the previous FA during the last executed information distribution procedure. The two authentication values should match, meaning that they were calculated using the same SA. In other words, the MN is trusted by the HA. After that, the new FA checks the replay protection extension to ensure that the *Reg_Rqst* message is freshly generated, see section 4.2.9.5 for details. If this is the case, the new FA examines whether the HA can satisfy the MN's requirements or not. This can be achieved by examining of the features supported by the HA, which are part of the distributed MN-specific data.

Notification of the old FA, MN and HA: if the local authentication is successful, the new FA sends a Previous FA Notification (*PFA_Not*) message to the old FA asking it to forward the MN's data packets to the new location. This message should be authenticated using the SA established between the FAs in the L3-FHR of the old FA (the key is $K_{FA,FA}$). Afterwards, a new key ($K3_{MN,FA}$) and two new random variables (R'_1 and R'_2) are generated. $K3_{MN,FA}$ will

be used to secure the control messages that should be exchanged between the MN and the subsequent new FA. The two new random variables (R'_1 and R'_2) will be used to calculate the authentication values during the next registration with the next new FA. Thereafter, the new FA creates a **Reg_Rply** message containing $Auth_2$, the new random variables and $K3_{MN,FA}$ encrypted with $K2_{MN,FA}$. The **Reg_Rply** message is authenticated using $K2_{MN,FA}$ and transmitted to the MN. The new FA then generates a new key ($K3_{FA,HA}$) to be used in securing the control messages that should be exchanged between the HA and the subsequent new FA. In addition to this, the new FA encrypts $K3_{FA,HA}$ with $K2_{FA,HA}$ and sends a HA Notification (**HA_Not**) message containing the encrypted key and the new random variables to the HA, see figure 4.6.

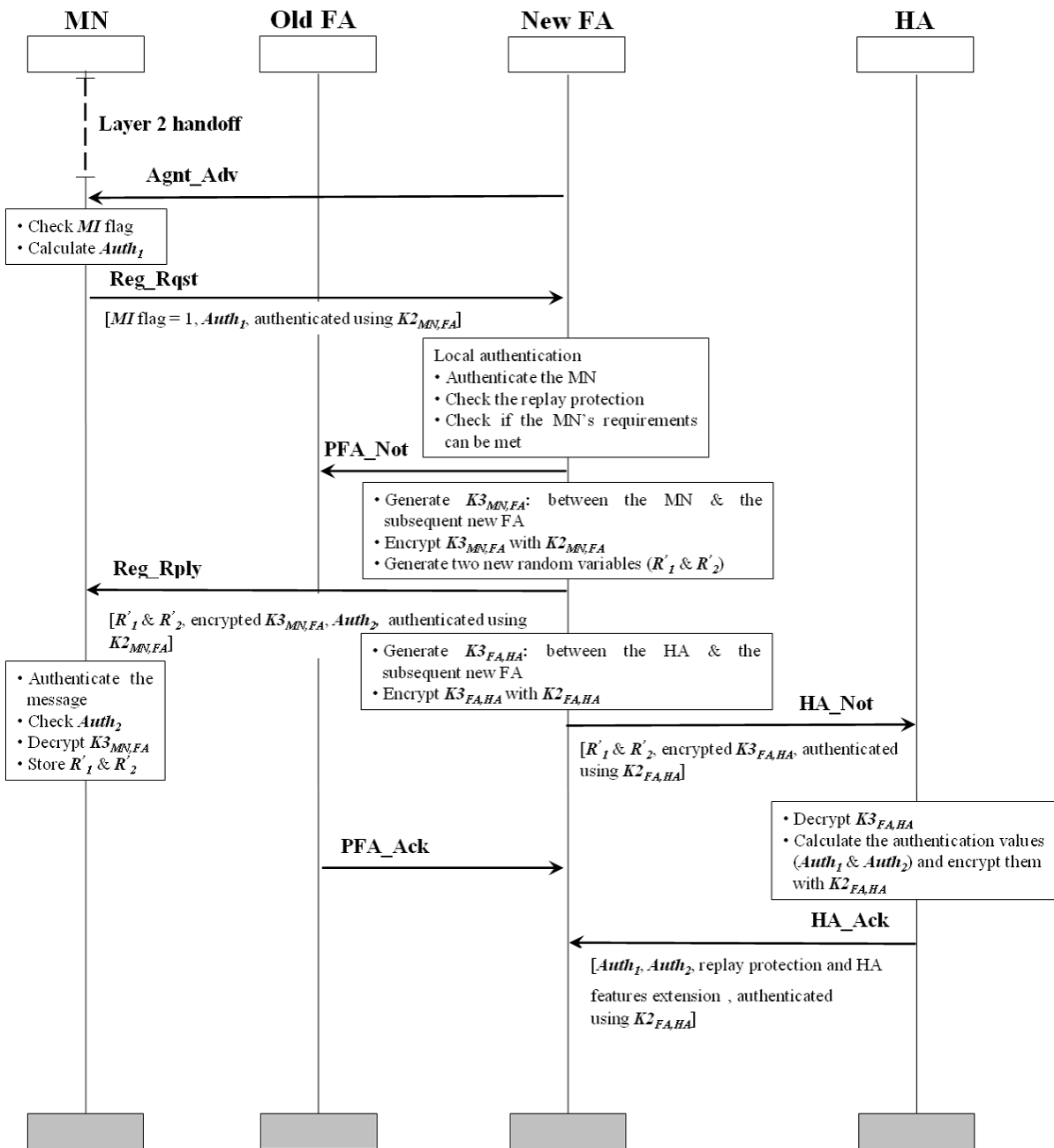


Fig 4.6: MIPv4 operation in reactive mode

Completion of the handoff from the MN point of view: when the MN receives the *Reg_Rply* message, it authenticates this message using $K2_{MN,FA}$. Subsequently, the MN calculates $Auth_2$ using R_2 and the same hash function the HA uses. The calculated value is compared to the value of $Auth_2$ sent from the new FA with the *Reg_Rply* message. As mentioned previously, this authentication value has been calculated originally by the HA and is used to ensure that the new FA is trusted by the HA. If the authentication is successful and the registration has been accepted, the MN decrypts $K3_{MN,FA}$ and stores the two new random variables to be used during the subsequent registration with the next new FA. From the MN point of view, the handoff procedure has been completed and it can resume its communication on uplink.

Forwarding from the old FA: as soon as the old FA receives the *PFA_Not* message, it authenticates and acknowledges this message by sending a Previous FA Acknowledgement (*PFA_Ack*) message back to the new FA. Afterwards, the old FA begins tunneling data packets destined to the MN to the new CoA.

Exchange of authentication information between the HA and the new FA: after the HA receives the *HA_Not* message, it authenticates this message using $K2_{FA,HA}$. If the authentication is successful, the HA derives $K3_{FA,HA}$, generates new values for $Auth_1$ and $Auth_2$, encrypts them using $K2_{FA,HA}$ and sends the encrypted values along with the HA features and replay protection extensions to the new FA with a HA Acknowledgement (*HA_Ack*) message authenticated using $K2_{FA,HA}$. The HA then redirects the tunnel from the old FA to the new one. Notice that during the time required to inform the HA, the MN receives its data packets forwarded from the old FA via the new one. This means that the time required to inform the HA about the new binding and even to establish a new IPsec tunnel [KA198], if required, is hidden from the application and no longer impacts performance.

Distribution of MN-specific data: after completion of the handoff, the MN-specific data should be distributed to neighbor FAs located in the current L3-FHR. This is achieved by executing the information distribution procedure presented in section 4.2.4.

4.2.6. Error Recovery Mechanisms in Reactive Mode

In order to enhance the robustness of MIFAv4, mechanisms to recover from failures must be supported. MIFAv4 in reactive mode distinguishes between the following types of failures:

1. loss of MIFAv4 support,
2. dropping of control messages and
3. moving to a non-member of the old FA's L3-FHR.

Loss of MIFAv4 support: in order to be compatible with MIPv4 and avoid communication disruption due to a loss of MIFAv4 support, the MN should construct its *Reg_Rqst* message according to the new FA's features advertised with the *Agnt_Adv* message. In other words, if the new FA advertises its support of MIFAv4, the MN builds the *Reg_Rqst* according to the specification of MIFAv4. However, if MIFAv4 is not supported, the MN resorts to the standard MIPv4. In addition, the new FA should forward the *Reg_Rqst* message to the HA to be processed there, if necessary. This will be the case if the new FA supports MIFAv4. However, the *Reg_Rqst* message can not be processed according to MIFAv4 for unknown or unpredicted reasons.

Dropping of control messages: MIPv4 sets a timer to retransmit any message dropped on the way to its destination. Let us now discuss the dropping of the *Reg_Rqst* or the *Reg_Rply* message. The MN sets a timer ($T_{timer-1}$) when it sends the *Reg_Rqst* message to the new FA. The default value of $T_{timer-1}$ is set to $2 * RTT_{MN,FA}$, where $RTT_{MN,FA}$ is the round trip time between the MN and the new FA. If the MN does not receive a *Reg_Rply* before $T_{timer-1}$ expires, it retransmits the *Reg_Rqst* message and duplicates $T_{timer-1}$, see figure 4.7. Clearly, the retransmitted *Reg_Rqst* message should be freshly generated.

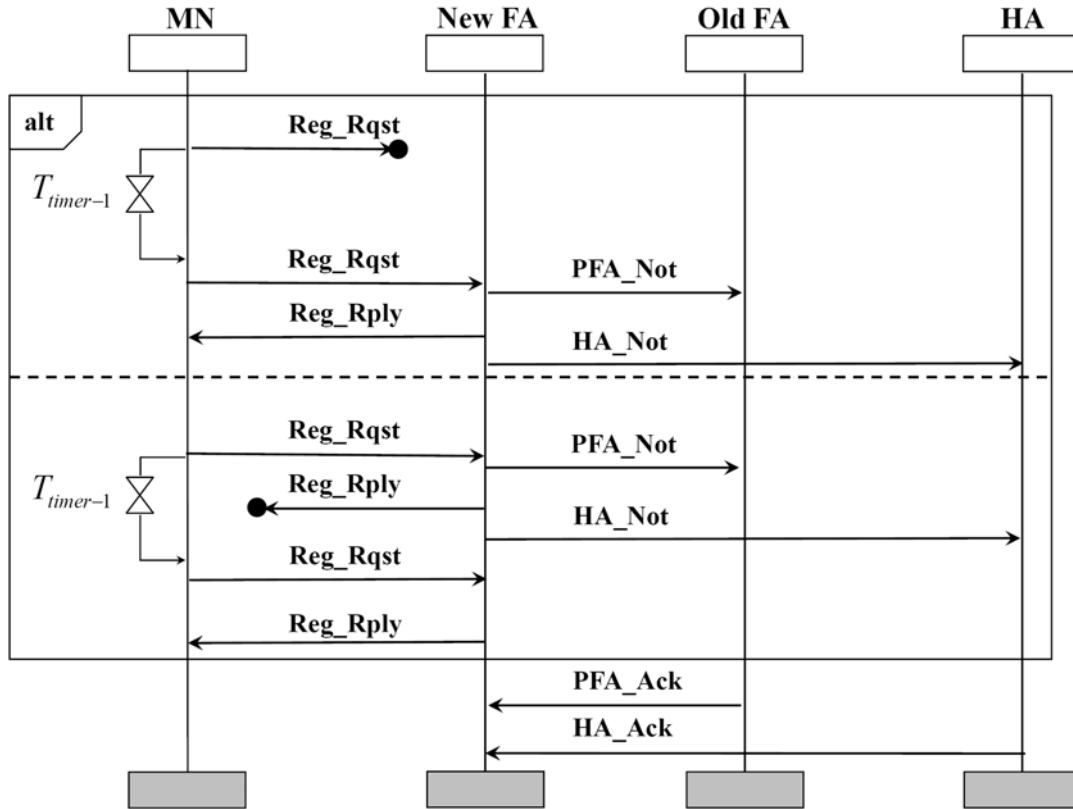


Fig 4.7: Recovery of *Reg_Rqst* or *Reg_Rply* message dropping

When the new FA receives a duplicated *Reg_Rqst* message, it assumes that the *Reg_Rply* has been dropped and retransmits the *Reg_Rply* once more. The retransmitted *Reg_Rply* should not be a copy of the dropped one, but rather freshly generated. Notice that due to the short duration of $T_{timer-1}$, the MN can quickly detect the dropping of the *Reg_Rqst* or *Reg_Rply* on the wireless link.

The dropping of the *PFA_Not* or *PFA_Ack* is recovered in a similar way. The new FA initiates a timer ($T_{timer-2}$) upon sending the *PFA_Not*. The default value of $T_{timer-2}$ is $2 * RTT_{newFA,oldFA}$, where $RTT_{newFA,oldFA}$ is the round trip time between the new and the old FA. If the new FA does not receive a *PFA_Ack* before the expiration of $T_{timer-2}$, it retransmits the *PFA_Not* message and duplicates $T_{timer-2}$. If the new FA receives data packets for the MN without receiving a *PFA_Ack* message, it assumes that the *PFA_Ack* has been dropped. However, there is no need to retransmit the *PFA_Not*.

The dropping of the *HA_Not* or *HA_Ack* is processed in a similar way as well. The new FA uses a timer ($T_{timer-3}$) to detect any dropping of these messages. $T_{timer-3}$ is assumed to be $2 * RTT_{newFA,HA}$, where $RTT_{newFA,HA}$ is the round trip time between the new FA and the HA.

Expiration of $T_{timer-3}$ without receiving the **HA_Ack** forces the new FA to retransmit the **HA_Not** and duplicate $T_{timer-3}$.

When the MN registers with a certain FA, this FA has to notify the FAs existing in its L3-FHR of a potential movement of the MN. This is achieved, as mentioned previously, by sending a **M_P_Not** message to each FA present in this L3-FHR. Sending a **M_P_Ack** message to acknowledge the receipt of the **M_P_Not** message is optional and must be requested explicitly in the **M_P_Not** message. In case the current FA has requested the sending of **M_P_Ack** messages, it can detect the dropping of this message by setting a timer ($T_{timer-4}$) to $2 * RTT_{FA,FA}$, where $RTT_{FA,FA}$ is the round trip time between the current FA and the farthest neighbor FA. Expiration of $T_{timer-4}$ before the current FA receives a **M_P_Ack** message results in retransmitting the **M_P_Not** and duplicating $T_{timer-4}$.

If the transmission of **M_P_Ack** messages is not requested, the current L3-FHR's members will not acknowledge the receipt of **M_P_Not** messages. Assuming a **M_P_Not** message has been dropped on the link between the current FA and a neighbor FA, and assuming the MN has moved to this FA, the MN attempts to register with this FA according to MIPv4. In this case, the new FA forwards the **Reg_Rqst** message to the old FA, which processes it according to MIPv4 specification, see figure 4.8. If the authentication of the MN is successful, the old FA sends a **Reg_Rply** with a **M_P_Not** message to the new FA, which derives the information existing in the **M_P_Not**, forwards the **Reg_Rply** to the MN and acknowledges the receipt of the **M_P_Not** by sending a **M_P_Ack** message. Then, the new FA informs the HA by sending a **HA_Not** message and resumes working according to the specification of MIPv4.

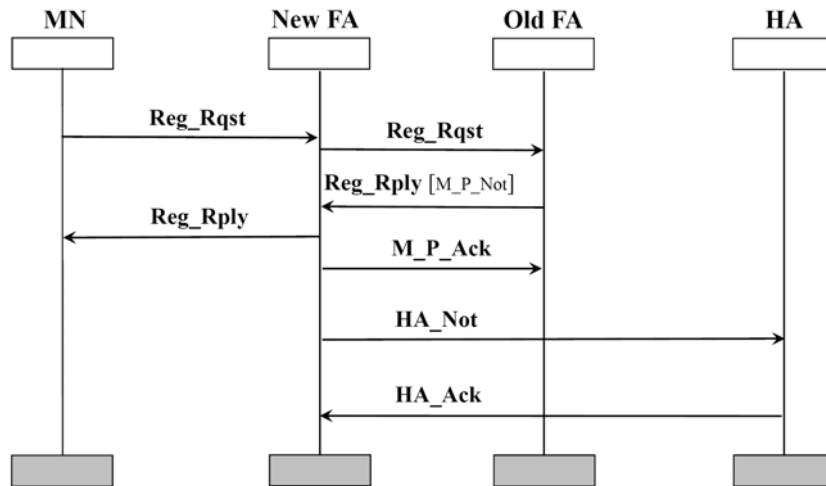


Fig 4.8: Recovery of **M_P_Not** message dropping (**M_P_Ack** is not requested and the new FA is a member of the L3-FHR of the old FA)

Moving to a non-member of the old FA's L3-FHR: if the new FA is not a member of the L3-FHR of the old FA, the new FA forwards the **Reg_Rqst** message to the HA, which treats with this message as an initial registration for the MN, see figure 4.9. The new FA then attempts to join the L3-FHR of the old FA. This is achieved by exchanging a Member Join Request (**Mem_Join_Rqst**) and a Member Join Response (**Mem_Join_Rsp**) message. It is highly recommended that a trust should be established between the new and the old FA before exchanging these messages, e.g. AAA can be used to establish such a trust.

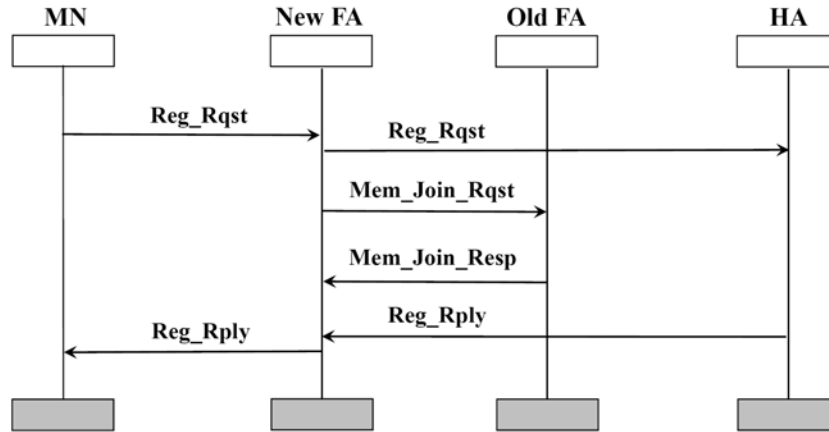


Fig 4.9: Moving to a non-member of the L3-FHR of the old FA

4.2.7. Operation in Predictive Mode

The predictive mode [DMB05a] aims at utilizing layer 2 triggers to anticipate the layer 3 handoff in advance, so that the layer 3 handoff latency is minimized or even eliminated.

In advance preparation for the handoff: when the MN notices that the quality of the current link is deteriorating, it starts scanning the medium for other available APs. If the detected AP belongs to another subnet, a L2-trigger is fired. The L2-trigger is used to identify the next new FA, which is normally a member of the current L3-FHR. This trigger prompts the MN to begin the layer 3 handoff in advance by sending a Proxy Router Solicitation message (*Pr_Rt_Sol*) to the current FA. The *Pr_Rt_Sol* message is similar to the *Agnt_Sol* message known from MIPv4. However, it is used to ask the current FA to send an advertisement on behalf of another FA, the new FA in this case. The current FA responds by sending a unicast Proxy Router Advertisement (*Pr_Rt_Adv*) message. The *MI* flag is included in this message and indicates whether the new FA supports MIFAv4 or not. It is assumed that the FAs of each L3-FHR exchange *Agnt_Sol* and *Agnt_Adv* messages periodically, so that the current FA can directly reply with a *Pr_Rt_Adv* message upon receiving the *Pr_Rt_Sol*.

Start of the layer 3 handoff: following the receipt of the *Pr_Rt_Adv* message by the MN, the MN sends a *Reg_Rqst* message to the new FA via the current one. The *Reg_Rqst* message includes a MIFA authentication extension that contains the authentication value $Auth_1$. The *Reg_Rqst* message itself is authenticated using the SA established between the MN and the new FA (the key is $K2_{MN,FA}$). As soon as the current FA receives the *Reg_Rqst* message, it sends an Initial Acknowledgement (*Int_Ack*) message to the MN. The MN uses this message to ensure that the *Reg_Rqst* message has not been dropped on the wireless link. If the MN-specific data have been distributed to the FAs present in the current L3-FHR, the *Reg_Rqst* message is forwarded to the new FA as it is. If this is not the case, a *M_P_Not* message containing the MN-specific data is built and added in a suitable extension to the *Reg_Rqst* message. The *Reg_Rqst* message is authenticated then using $K_{FA,FA}$ and sent to the new FA.

Local authentication: if the MN-specific data have not been distributed previously, the new FA first authenticates the *Reg_Rqst* message using $K_{FA,FA}$ and then derives the required SAs and information from the *M_P_Not* message sent from the old FA with the *Reg_Rqst* message. After that, the new FA authenticates the MN using $K2_{MN,FA}$. If the authentication is successful, the new FA compares the value of $Auth_1$ calculated by the MN, which is conveyed in the *Reg_Rqst* message, with the value of $Auth_1$ calculated by the HA, which is part of the MN-specific data. If the two values match, the new FA checks the replay protection and

determines whether the HA can satisfy the MN's requirements or not. This is achieved by examining of the replay protection extension and the HA features extension distributed to the new FA as part of the MN-specific data.

Notification of the old FA and HA: if the local authentication is successful, the new FA sends a *Reg_Rply* message to the old FA, which uses this message as an indicator for the success of the handoff as well as for starting tunneling of the MN's data packets to the new FA upon the appearance of a L2-LD trigger. Following this, two new random variables (R'_1 and R'_2) and a new key ($K3_{FA,HA}$) are generated. $K3_{FA,HA}$ is encrypted with $K2_{FA,HA}$ and is used to authenticate the control messages that should be exchanged between the HA and the next new FA. The new FA sends the two new random variables and encrypted key in a *HA_Not* message towards the HA. This message is authenticated using $K2_{FA,HA}$, see Figure 4.10.

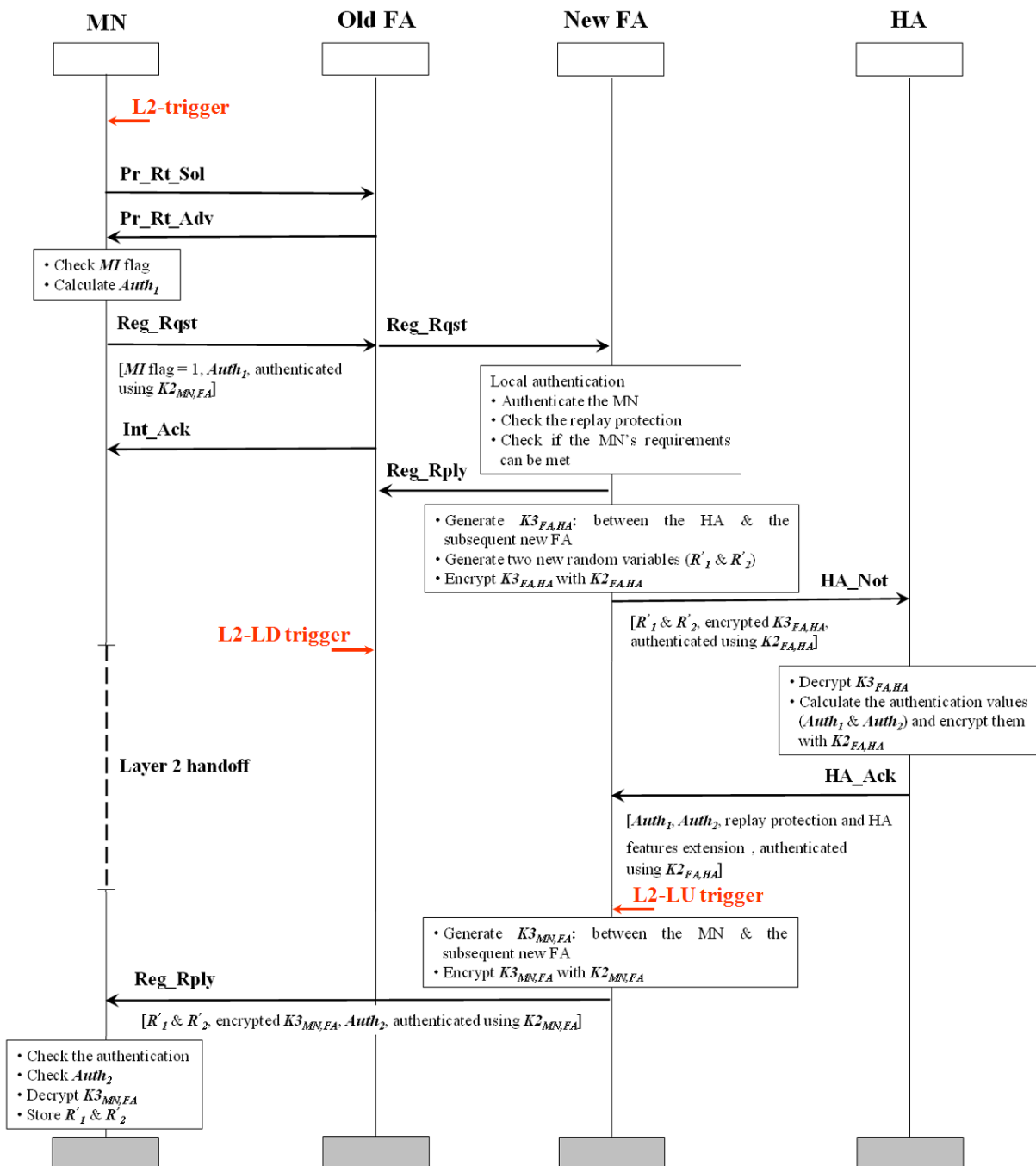


Fig 4.10: MIPv4 operation in predictive mode

Exchange of authentication information between the HA and the new FA: as the HA receives the *HA_Not* message, it authenticates the message using $K2_{FA,HA}$. Afterwards, it derives $K3_{FA,HA}$, calculates the new authentication values ($Auth_1$ and $Auth_2$), encrypts them with $K2_{FA,HA}$ and sends the encrypted values along with the HA features and the replay protection extension to the new FA with a *HA_Ack* message. The HA then tunnels data packets destined to the MN to the new CoA.

Forwarding from the old FA: as a L2-LD trigger is raised at the old FA, the old FA begins forwarding of data packets destined to the MN to the new FA, which in turn buffers them until a L2-LU trigger appears.

Completion of the handoff: following the appearance of the L2-LU trigger at the new FA, a key is generated to be used in authenticating the control messages that will be exchanged between the MN and the next new FA ($K3_{MN,FA}$). This key is encrypted with $K2_{MN,FA}$ and sent to the MN along with the newly generated random variables (R'_1 and R'_2) and $Auth_2$ with a *Reg_Rply* message. Following this, the new FA begins forwarding data packets to the MN. After the MN authenticates the *Reg_Rply* and successfully checks the value of $Auth_2$, it resumes its communication on uplink.

Distribution of MN-specific data: following completion of the handoff, the MN-specific data should be distributed to neighbor FAs located in the current L3-FHR. This is achieved by executing the information distribution procedure, see section [4.2.4](#).

4.2.8. Error Recovery Mechanisms in Predictive Mode

MIFAv4 in predictive mode distinguishes between the following types of failures:

1. loss of MIFAv4 support,
2. dropping of control messages,
3. appearance of layer 2 triggers, i.e. L2-trigger, L2-LD and L2-LU trigger, at inappropriate times and
4. moving to a non-member of the old FA's L3-FHR.

Loss of MIFAv4 support: let us assume that the MN has received a *Pr_Rt_Adv* message before the handoff occurs. As mentioned previously, the *MI* flag contained in the *Pr_Rt_Adv* message indicates the support of MIFAv4 by the new FA. In case the *MI* flag is set to 0, the MN resorts to MIPv4. In other words, the MN executes a layer 2 handoff followed by a layer 3 handoff according to MIPv4 specification. In case the *MI* flag is set to 1, the MN proceeds with MIFAv4 and sends a *Reg_Rqst* message to the new FA via the current one. If the current FA is, for any reason, unable to further process the *Reg_Rqst* message according to MIFAv4, it informs the MN by means of the *Int_Ack* message. As mentioned in section [4.2.7](#), the MN uses this message to ensure that the *Reg_Rqst* message has not been dropped on the wireless link. In addition, this message may contain a handoff possibilities extension that indicates whether the handoff to the new FA using MIFAv4 is possible. If a handoff to the new FA employing MIFAv4 is not possible, the MN resorts to the standard MIPv4.

Dropping of control messages: MIFAv4 in predictive mode works in the same manner as in reactive mode regarding control messages dropping. It sets a timer to retransmit any message dropped on the way to its destination.

Appearance of layer 2 triggers at inappropriate times: considering a delayed firing of the L2-trigger at the MN, one can distinguish between the following cases:

1. The L2-trigger has not been raised or delayed, so that the MN was unable to receive the *Pr_Rt_Adv* message: in this case the MN employs MIFAv4 in reactive mode.
2. The L2-trigger has been delayed and the MN could only receive the *Pr_Rt_Adv* and was unable to send the *Reg_Rqst* message: the MN, again, employs MIFAv4 in reactive mode. However, it does not wait for an *Agnt_Adv* after completion of the layer 2 handoff.
3. The L2-trigger has been fired at an appropriate time at the MN, which could send a *Reg_Rqst* message. However, the *Reg_Rqst* message has been dropped and the wireless link between the MN and the old FA has been broken before the MN could detect the *Reg_Rqst* message dropping: in this case the MN waits a certain amount of time, $T_{timer-5}$, for a *Reg_Rply* message from the new FA after the layer 2 handoff. The default value of $T_{timer-5}$ is set to $RTT_{MN,FA}$, where $RTT_{MN,FA}$ is the round trip time between the MN and the new FA. Expiration of $T_{timer-5}$ without receiving the *Reg_Rply* message forces the MN to employ MIFAv4 in reactive mode.

Let us now consider the appearance of the L2-LD trigger at the old FA. One of the following cases is expected:

1. The L2-LD trigger was raised and the old FA has not received any message from the MN, i.e. the MN could not predict the handoff: in this case, MIFAv4 will be executed in reactive mode as explained in section [4.2.5](#).
2. The L2-LD trigger was raised before the old FA could receive the *Reg_Rqst* message. However, *Pr_Rt_Sol* and *Pr_Rt_Adv* messages have been exchanged with the MN: in this case, MIFAv4 will, again, be executed in reactive mode. However, as the old FA notices this case, it sends a *M_P_Not* message to the predicted new FA in case the information distribution procedure has not been previously executed.
3. The old FA has received the *Reg_Rqst* message from the MN and has forwarded it to the new FA. However, the old FA has received the L2-LD trigger before the receipt of a *Reg_Rply* message: in this case, the old FA may be configured to begin buffering the MN's data packets. These packets are forwarded to the new FA upon the receipt of the *Reg_Rply* message.

Let us now address the appearance of the L2-LU trigger at the new FA. Assuming that no or a delayed L2-LU trigger has been raised at the new FA or the new FA has sent a *Reg_Rply* message to the MN. However, this message has been dropped. The MN will attempt to proceed with MIFAv4 in reactive mode after the expiration of $T_{timer-5}$. This results in sending another *Reg_Rqst* message to the new FA, which either sends or retransmits the *Reg_Rply* again. Of course, the retransmitted *Reg_Rply* message should not be a copy of the dropped one.

Movement to a non-member of the old FA's L3-FHR: provided that the MN has fired a L2-trigger at an appropriate time and has sent a *Pr_Rt_Sol* message to the old FA, the predicted new FA will either be a member or non-member of the current L3-FHR. In the case that the predicted new FA is a member of the old FA's L3-FHR, the old FA proceeds according to MIFAv4 in predictive mode. However, in the case that the predicted new FA is not a member of the old FA's L3-FHR, one can distinguish between the following situations:

1. The old FA does not know the IP address of the new FA: in this case, the old FA does not send a *Pr_Rt_Adv* message, which forces the MN to execute a layer 2 handoff and to attempt to employ MIFAv4 in reactive mode. In this case, the new FA forwards the *Reg_Rqst* message to the HA, which handles it as an initial registration. Assuming

that the new FA supports MIFAv4, it attempts to join the L3-FHR of the old one. For further details, see section [4.2.6](#).

2. The old FA knows the IP address of the new one: in this case, the old FA sends a *Pr_Rt_Adv* on behalf of the new FA and sets the *MI* flag to 1 if MIFAv4 is supported by the new FA.
 - a. If the MN was able to exchange *Reg_Rqst* and *Int_Ack* messages with the old FA, the *Int_Ack* will contain a handoff possibilities extension indicating that a handoff to the detected new FA using MIFAv4 is not possible. In this case, the MN resorts to MIPv4. Of course, the *Reg_Rqst* message will not be forwarded beyond the old FA.
 - b. If the MN was unable to send the *Reg_Rqst* message, it will attempt to register with the new FA using MIFAv4 in reactive mode following the layer 2 handoff. The new FA then forwards the *Reg_Rqst* to the HA, which deals with this case as an initial registration. Subsequently, the new FA attempts to join the L3-FHR of the old FA. For details, see section [4.2.6](#).
 - c. If the MN has sent the *Reg_Rqst* message, but was unable to receive the *Int_Ack* message. The MN assumes to receive a *Reg_Rply* directly after the layer 2 handoff, which, in this case, does not occur. Therefore, the MN attempts to register with the new FA employing MIFAv4 in reactive mode. The new FA in turn forwards the *Reg_Rqst* message to the HA and joins the L3-FHR of the old FA.

4.2.9. Security Considerations

MIFAv4 results in a redirection of the data traffic being forwarded from the HA as well as from the old FA to the new CoA. Therefore, some security considerations should be taken into account to prevent this traffic redirection from becoming vulnerable. An important issue that must be considered is the authentication of control messages by communication partners. The following discusses some security considerations that help in ensuring a strong authentication and a secure traffic redirection.

4.2.9.1. Key Management

It is highly recommended that each administrative domain contains a security infrastructure that can be used to distribute SAs securely. This infrastructure should be used mainly to distribute the SAs during the initial registration. Following this, the generated SAs should be used to generate the required SAs and authenticate the control messages during the initial authentication exchange procedure. Then, each new SA should be generated and distributed encrypted with the old SA. Of course, this requires FAs and the HA to work as Key Distribution Centers (KDCs). It is possible that all SAs are generated and distributed using the existing security infrastructure. However, this should be avoided if it slows down the handoff. Any other mechanism considered as secure can be used to generate and distribute SAs. Moreover, the security infrastructure should be used to distribute SAs between members of each L3-FHR. As mentioned in section [4.2.6](#), when a new FA wants to join a L3-FHR of a certain FA, *Mem_Join_Rqst* and *Mem_Join_Resp* messages are exchanged. To perform this procedure securely, it is highly recommended that a SA should be established between both FAs using the security infrastructure. Clearly, any key distribution method that is declared secure may also be used. Each FA should reject any *Mem_Join_Rqst* message that can not be authenticated. In order to simplify the generation and distribution of SAs, a new mechanism is proposed to address this issue when using MIFAv4. This mechanism can be briefly described as follows.

1. The required SAs are generated using the security infrastructure during the initial registration.
2. It is assumed that all FAs in each L3-FHR have a key (K_{L3-FHR}). This key is used to calculate the shared secrets between the new FA and the MN, on one side, and between the new FA and the HA, on the other side. These shared secrets are calculated as follows:
 - a. The shared secret between the MN and the new FA is a secure hash calculated as follows: $K2_{MN,FA} = HMAC-MD5 (K_{L3-FHR}, MN_{home-add}, MN_{MAC-add}, n)$
 - b. The shared secret between the new FA and the HA is a secure hash calculated as follows: $K2_{FA,HA} = HMAC-MD5 (K_{L3-FHR}, MN_{HA-add}, MN_{MAC-add}, n)$

 $MN_{MAC-add}$, $MN_{home-add}$ and MN_{HA-add} represent the MN's MAC address, home address and HA address, respectively. n is a number in ascending order referring to the number of times the keys are refreshed. This number is sent as a part of the MN-specific data.
 - c. The generated shared secrets are distributed securely to the MN as well as the HA. There is no need to distribute them to the current L3-FHR's members, since they can be derived following the layer 2 handoff. Notice that the MN and the HA do not need to change the keys used for the authentication with the new FA each time the MN changes its point of attachment. The keys will be refreshed after the expiration of their lifetime and distributed to both the MN and the HA.

4.2.9.2. *Selecting Good Random Numbers*

It is highly recommended that generated keys should be pseudo-random and secret (known only to the authorized parties). More information about generating pseudo-numbers can be found in [[ECS94](#)].

4.2.9.3. *Privacy*

Users with sensitive data that should not be viewed by others should use mechanisms to secure the exchanged data, such as IPsec. MIFAv4 is used to securely redirect data packets and not to secure the data themselves.

4.2.9.4. *Ingress Filtering*

Many routers implement a security mechanism called ingress filtering to check for topologically incorrect IP addresses. In MIFAv4, MNs use their permanent home addresses as source addresses for the uplink traffic. From the FA point of view, these packets will be determined as incorrect packets because they appear to originate from outside the subnet and will thus be discarded. To overcome this problem, MNs should use reverse tunnels from their current FAs to their HAs.

4.2.9.5. *Replay Protection*

The new FA should be able to prove that the **Reg_Rqst** message has been freshly generated by the MN and not replayed by an attacker. Such proof requires implementation of a replay protection mechanism. Similar to MIPv4, MIFAv4 uses timestamp as a default replay protection mechanism. An optional nonce-based replay protection may also be employed. In

contrast to MIPv4, MIFAv4 requires the new FA to provide the replay protection rather than the HA. Such a timestamp-based and a nonce-based replay protection will be described briefly in the following.

Timestamp-based replay protection: the basic principle of timestamp-based replay protection can be described as follows: the node that generates a message has to insert its current time of day in the message. The recipient of the message then checks whether the timestamp present in the message matches its current time of day. A difference of T_{Diff} seconds should be used to limit the maximum allowed difference between the two times. For example, according to MIPv4 specification, T_{Diff} equals 7 sec, see [Per02]. Clearly, using timestamp as a replay protection mechanism requires the two nodes to have synchronized time-of-day clocks.

Application of the timestamp-based replay protection in MIFAv4 is achieved as follows: the replay protection extension contains a field of 64 bits termed as identification field and formatted according to the specification of the Network Time Protocol (NTP) [Mil92]. The MN and the HA are normally synchronized. If the FAs of a certain domain are synchronized with the HA, they can simply detect replayed messages. The MN sends a **Reg_Rqst** message including a replay protection extension to the new FA. The MN's timestamp present in the low-order 32 bits of the identification field should be close enough to the FA's time of day and larger than any previous timestamp. If the MN's timestamp is determined valid, the new FA sends a **Reg_Rply** message with a replay protection extension that contains the same identification field sent previously with the **Reg_Rqst** message. If the MN's timestamp is, however, invalid, the new FA copies only the low-order 32 bits from the identification field into the replay protection extension that should be sent with the **Reg_Rply** message. The high-order 32 bits should contain the timestamp of the new FA. Obviously, the registration in this case is rejected. When the MN receives such a rejected **Reg_Rply** message, it first authenticates the message. After that, it checks the low-order 32 bits of the identification field, which should match the timestamp sent previously with the **Reg_Rqst** message, before it uses the high-order 32 bits for synchronization.

If the FAs of a certain domain and the HA are not synchronized, the new FA must use another mechanism to realize the timestamp-based replay protection. Such a replay protection can be achieved as follows: after the MN is initially registered with the HA, the current FA executes an initial authentication exchange procedure, which comprises exchanging **M_P_Not** and **M_P_Ack** messages with the HA. The **M_P_Ack** message should contain a replay protection extension. The identification field existing in this extension should contain the HA's timestamp (TS_{HA}) in the low-order 32 bits and a random number in the high-order 32 bits. As soon as the current FA receives the **M_P_Ack** message, it stores the HA's timestamp and its own timestamp (TS_{FA}). The current FA proceeds with distributing the MN-specific data to the members of the current L3-FHR by sending a **M_P_Not** message to each member. Supposing that the current FA has distributed the MN-specific data after a duration of Δt , the current FA can calculate the current HA's timestamp depending on TS_{HA} , TS_{FA} and Δt . The new HA's timestamp (TS'_{HA}) is then distributed with the **M_P_Not** messages as a part of the MN-specific data. In addition to TS'_{HA} , the identification field that has been used by the MN during the initial registration with the HA (TS_{MN}) is also distributed as a part of the MN-specific data.

Each neighboring FA in turn records the HA's timestamp (TS'_{HA}) and its own timestamp (TS_{neiFA}). When the MN moves to a new FA, it sends a **Reg_Rqst** message containing the replay protection extension. The current FA can check whether the MN's timestamp is close

enough to the current HA's timestamp depending on TS'_{HA} , TS_{neiFA} and the time at which the **Reg_Rqst** message was received.

Nonce-based replay protection: the basic principle of the nonce-based replay protection can be described as follows: the node generating a message must insert a random number, referred to as a nonce, in the message. The node receiving this message should echo this nonce in its next message transmitted to the source node. At the same time, the second node should also include its nonce to be echoed by the source node. In order to use a nonce-based replay protection between the MN and the HA employing MIPv4, the identification field is formatted as follows: the low-order 32 bits stand for the MN's nonce, while the high-order 32 bits represent the HA's nonce. When the MN sends a **Reg_Rqst** message to the new FA, it copies the high-order 32 bits ($Nonce_{HA-1}$) from the identification field existing in the last received **Reg_Rply** message and generates randomly the low-order 32 bits, which present the MN's nonce ($Nonce_{MN-1}$). The new FA has to check the identification field in the replay protection extension sent with the **Reg_Rqst** message. The high-order 32 bits should match the HA's nonce, which is part of the MN-specific data previously received from the old FA. If the check is successful, the new FA sends a **Reg_Rply** message including a replay protection extension to the MN. The MN in turn has to check the identification field, which should contain the low-order 32 bits sent with the **Reg_Rqst** message ($Nonce_{MN-1}$). The high-order 32 bits present a new HA's nonce ($Nonce_{HA-2}$) that should be used during the next registration with the next new FA. The current FA informs the HA by exchanging **HA_Not** and **HA_Ack** message with it. The **HA_Ack** message contains a new HA's nonce ($Nonce_{HA-3}$). The current and new HA's nonces ($Nonce_{HA-2}$ and $Nonce_{HA-3}$) are distributed to the FAs of the current L3-FHR as part of the MN-specific data, so that these FAs can check the replay protection when the MN moves to one of them.

4.2.10. Formal Specification with SDL

As known, SDL [EHS97], [Mit01] is used to describe the behavior of reactive as well as distributed systems. It helps in detecting the errors that may occur and correcting them prior to real implementation. This paragraph briefly introduces the formal specification with SDL for the most important parts of MIPv4. A more detailed description is provided in appendix C. The implemented system consists of 3 FAs, a HA, a MN, 2 wireless channels and 3 wired channels. The HA, MN and each FA is composed of a physical, data link and network layer. The physical layer takes care of sending and receiving data to and from the medium. A simple layer 2 handoff is realized in the data link layer, while the layer 3 handoff using MIPv4 is implemented in the third layer. Figure 4.11 presents the highest level view of the system. As seen in this figure, MIPv4 service is controlled by three signals, MIFA CONnection request (**MCONreq**), MIFA DISconnection request (**MDISreq**) and HandOff Request (**HORqst**). The **MCONreq** signal is used to start MIPv4, while the **MDISreq** signal is used to stop the protocol. The **HORqst** signal simulates a handoff event. More specifically, upon sending this signal, the MN fires a L2-trigger. After successfully establishing a layer 3 connection, a MIFA CONnection indicator (**MCONind**) signal is sent as a response, while a MIFA DISconnection indicator (**MDISind**) signal is sent after the layer 3 connection breaks down.

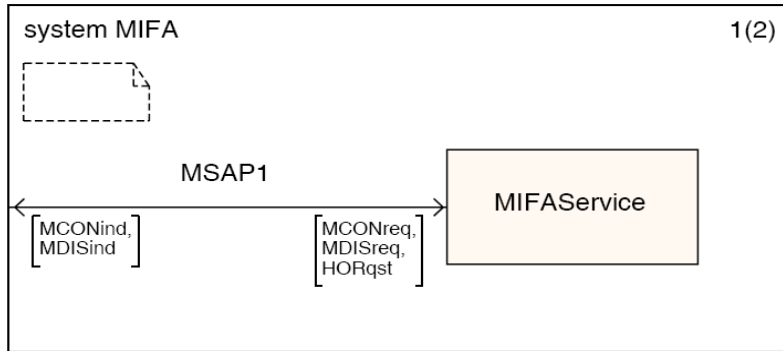


Fig 4.11: MIFAv4 system – high level view

Figure 4.12 shows the contents of the **MIFAService** block. As mentioned above, MIFAv4 is implemented in a network that consists of a HA, 3 FAs and a MN. The three FAs are FA1, FA2 and a neighbor FA (**Neighbor_FA** in the figure). The MN moves between FA1 and FA2, while the neighbor FA is used to simulate a L3-FHR member. The MN is connected to FA1 and FA2 by means of 2 wireless links (**WirelessChannel1** and **WirelessChannel2** in the figure). FA1 is connected with FA2 via a wired channel (**WiredChannel3** in the figure). Another wired channel (**WiredChannel2**) connects FA1 and FA2 with the HA, while a third wired channel (**WiredChannel1**) interconnects between the neighbor FA and FA2. This figure shows that the L3-FHR of FA1 contains only FA2, while the L3-FHR of FA2 includes the neighbor FA and FA1.

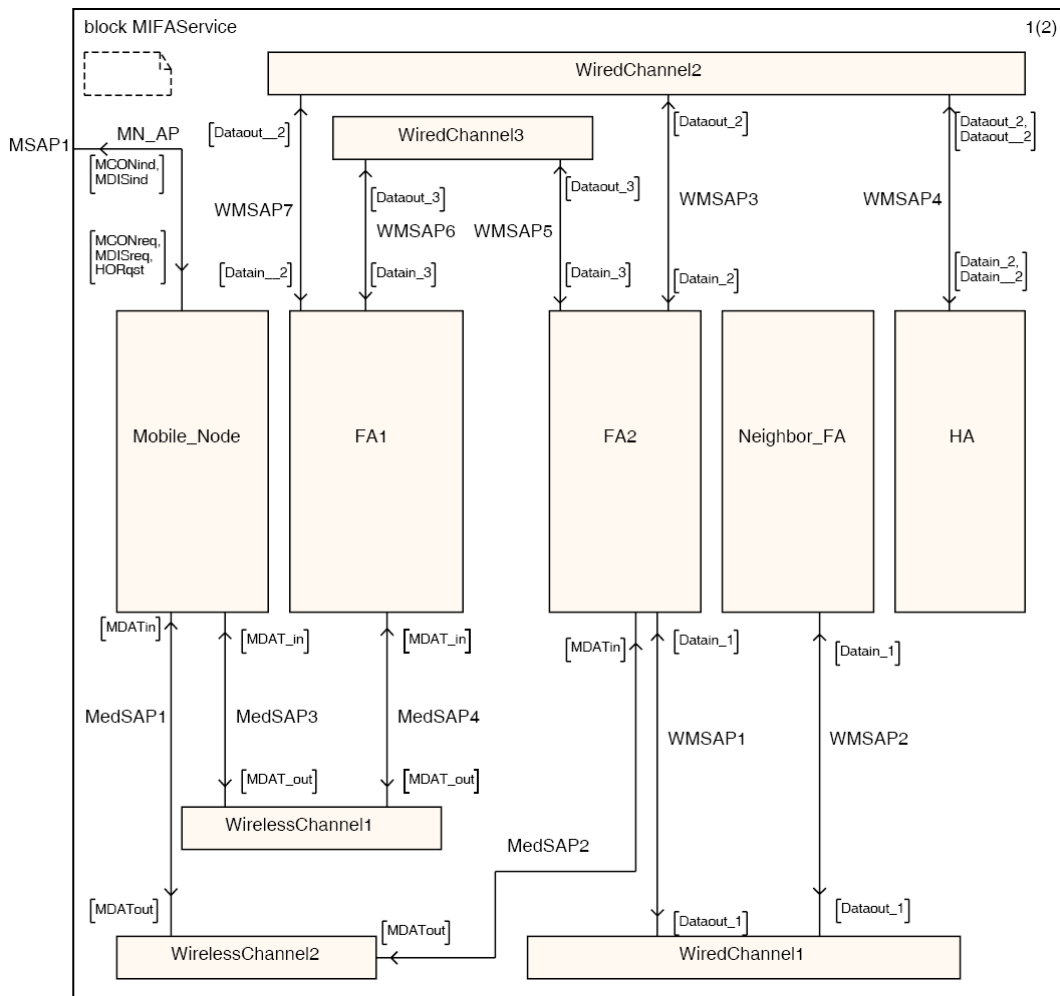


Fig 4.12: **MIFAService** block

Figure 4.13 displays the MN structure. The blocks **MN_Physical_Layer**, **MN_DataLink_Layer** and **MN_Network_Layer** represent the physical, data link and network layer, respectively. Notice that the signals **MCONreq**, **MDISreq**, **HORqst**, **MCONind** and **MDISind** are exchanged between the data link layer and the environment. Of course, this does not match the reality. Moreover, these signals aim mainly at controlling the simulation.

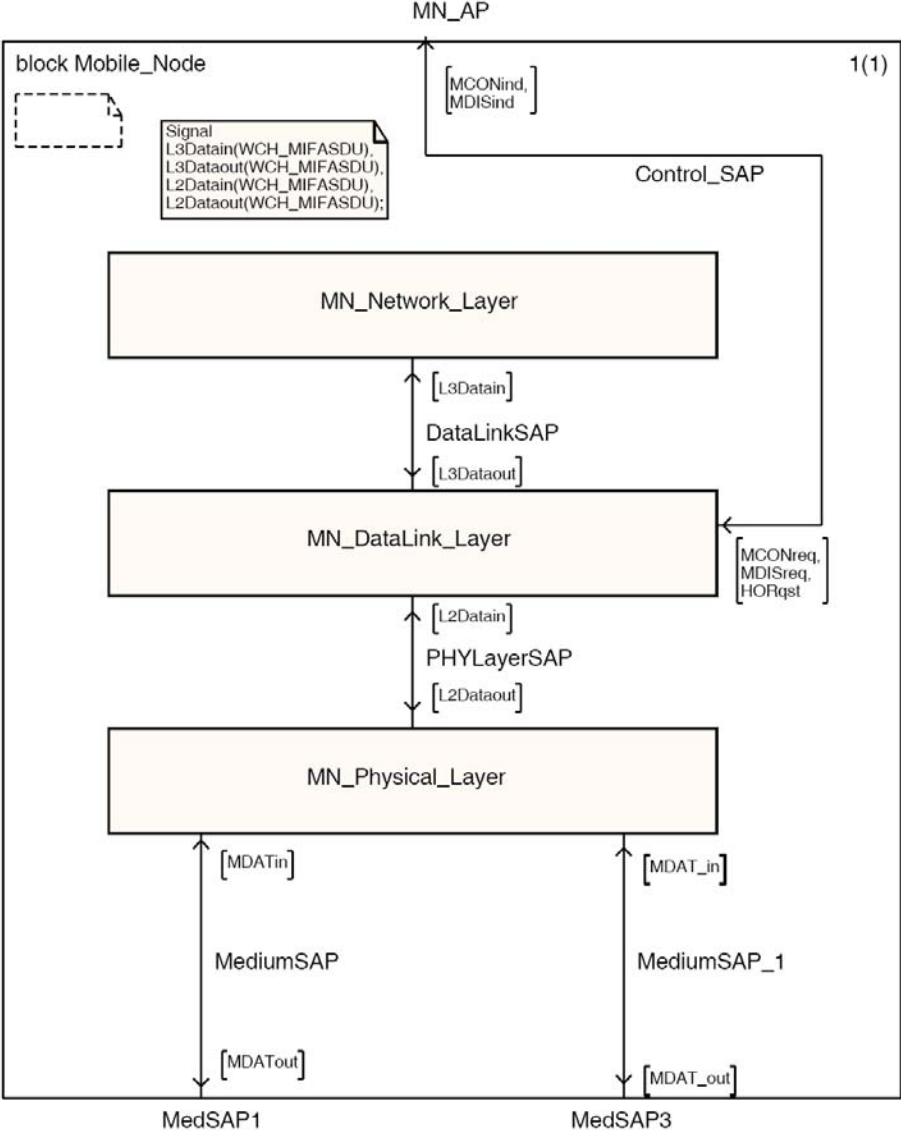


Fig 4.13: Structure of the MN - **Mobile_Node** block

Figure 4.14 shows the structure of the **MN_Physical_Layer** block, which contains only one process named **PHY_Coder**. The **PHY_Coder** process receives data from the wireless channels via two Service Access Points (SAPs), **MediumSAP** and **MediumSAP_1**, and forwards them in suitable form to the **MN_DataLink_Layer** block. In addition, the **PHY_Coder** process takes care of receiving data from the **MN_DataLink_Layer** block and converting them to a suitable form that can be sent over the wireless channels.

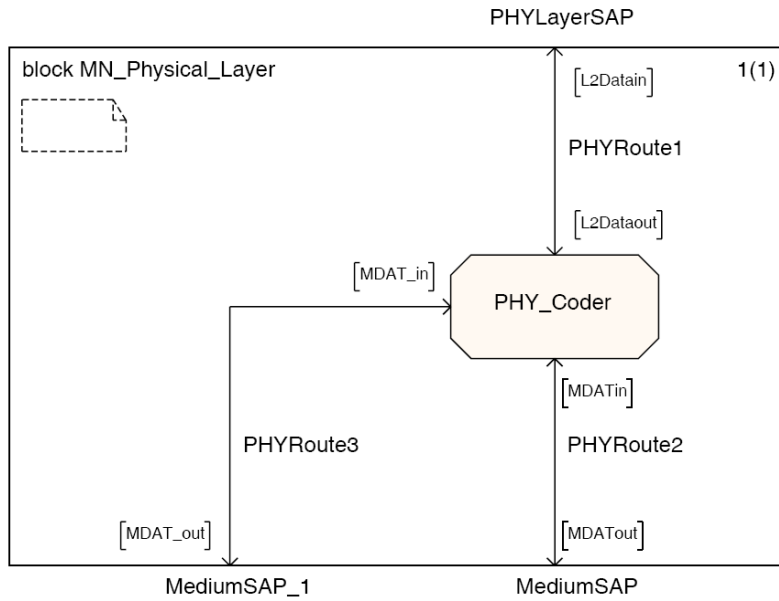


Fig 4.14: **MN_Physical_Layer** block

Figure 4.15 shows the structure of the **MN_DataLink_Layer** block. This block consists of two processes, **L2_Handoff** and **DataLink_Coder**. The **DataLink_Coder** process provides an interface between the first and the third layer. The process receives data from the **MN_Network_Layer** block via the **DataLinkSAP** SAP and sends them either to the **L2_Handoff** process or physical layer. The data coming from the physical layer are injected into the **MN_DataLink_Layer** block through the **PHYLayerSAP** SAP. Again, the **DataLink_Coder** process is responsible for receiving data from the physical layer and either sending them to the **L2_Handoff** process or to the third layer. The **L2_Handoff** process is responsible for the establishment of a radio link between the MN and either FA1 or FA2. In addition, this process is responsible for executing a layer 2 handoff between both FAs. More concrete, the MN initially establishes a wireless link with FA1 and hands off to FA2 upon receiving a **HORqst** signal. Receipt of a **HORqst** signal then forces the MN to switch between FA1 and FA2.

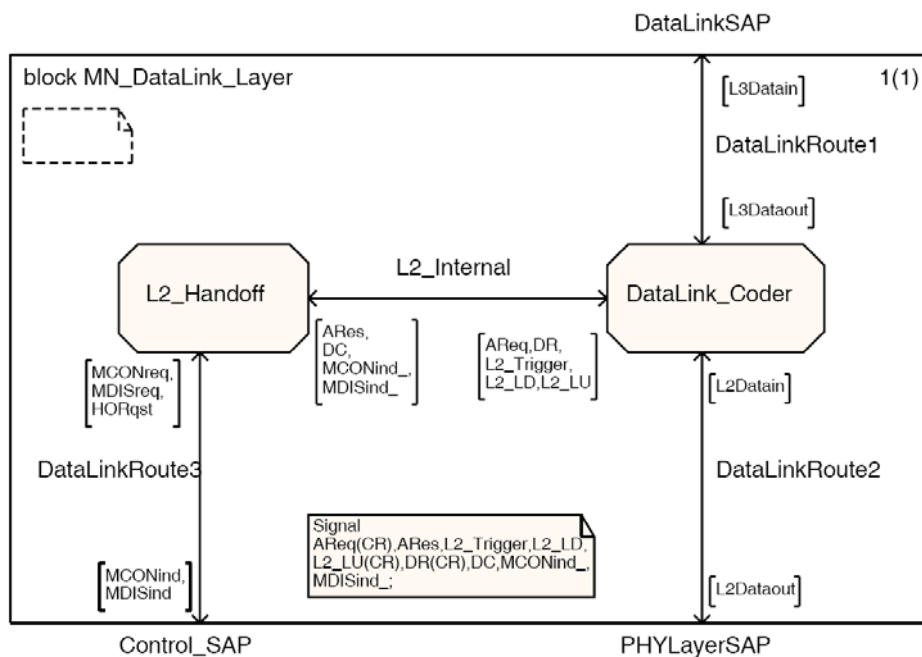


Fig 4.15: **MN_DataLink_Layer** block

Figure 4.16 presents the structure of the **MN_Network_Layer** block. This block contains two processes, **MIFA** and **NetLayer_Coder**. The **NetLayer_Coder** process provides an interface between the third and second layer. This process receives data from the **MN_DataLink_Layer** block via the **DataLinkSAP** SAP and sends them to the **MIFA** process, which is responsible for the establishment of a layer 3 connection between the MN and either FA1 or FA2. The layer 3 handoff between both FAs is executed by this process as well.

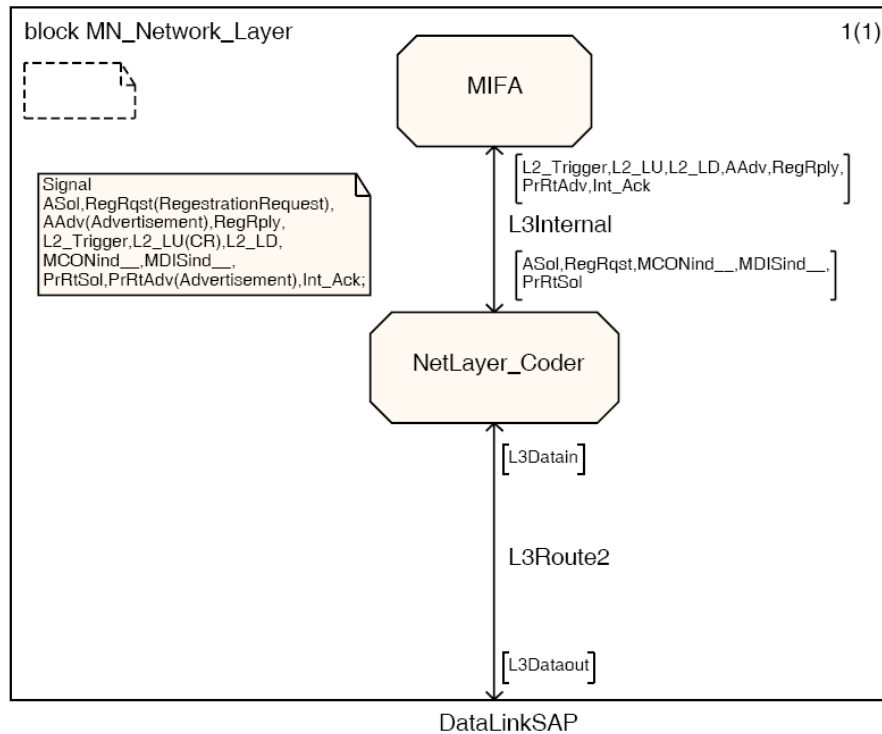


Fig 4.16: **MN_Network_Layer** block

The **MIFA** process is the most important. Therefore, it will be discussed in detail, see figure 4.17, figure 4.18, figure 4.19 and figure 4.20. The MN initiates itself by executing the **Initiating** procedure and transits itself to the **Disconnected** state. Receiving a **L2_LU** signal indicates that the wireless link has just been established successfully. This forces the MN to initiate a timer **T1** and transit itself to the **layer2connected** state. The MN waits in this state for an advertisement from a FA. It may receive in this state an **Agnt_Adv** message (**AAdv(ADV)** in the figures) or a timer expiration signal (**T1** in the figures). Receiving the **AAdv(ADV)** signal causes the timer **T1** to be reset, the mobility binding to be assigned, a **Reg_Rqst** message (**RegRqst(REG)** in the figures) to be built and sent. A timer **T2** is initiated upon sending the **Reg_Rqst** message to recover any dropping of it or of the **Reg_Rply** message. The MN transits itself, after that, to the **Wait** state. The receipt of the signal **T1** while the MN is in the **layer2connected** state results in sending an **Agnt_Sol** message (**ASol** in the figures). After three expirations of the timer **T1**, the MN assumes that the establishment of a layer 3 connection is not possible and sends a **MDISind** signal (**MDISind__** in the figures).

If the MN has employed the predictive mode before executing a layer 2 handoff, it expects to receive a **Reg_Rply** message directly after finishing the layer 2 handoff. This means that the MN expects to receive the **Reg_Rply** message (**RegRply** in the figures) from the **layer2connected** state. If this is the case, the MN executes the **TemporalToCurrentBinding** procedure, which transforms the mobility binding from the temporal to current binding list. Subsequently, the MN sends a **MCONind** signal (**MCONind__** in the figures) and transits

itself to the **L3_connected** state, which indicates that the layer 3 handoff has been completed successfully.

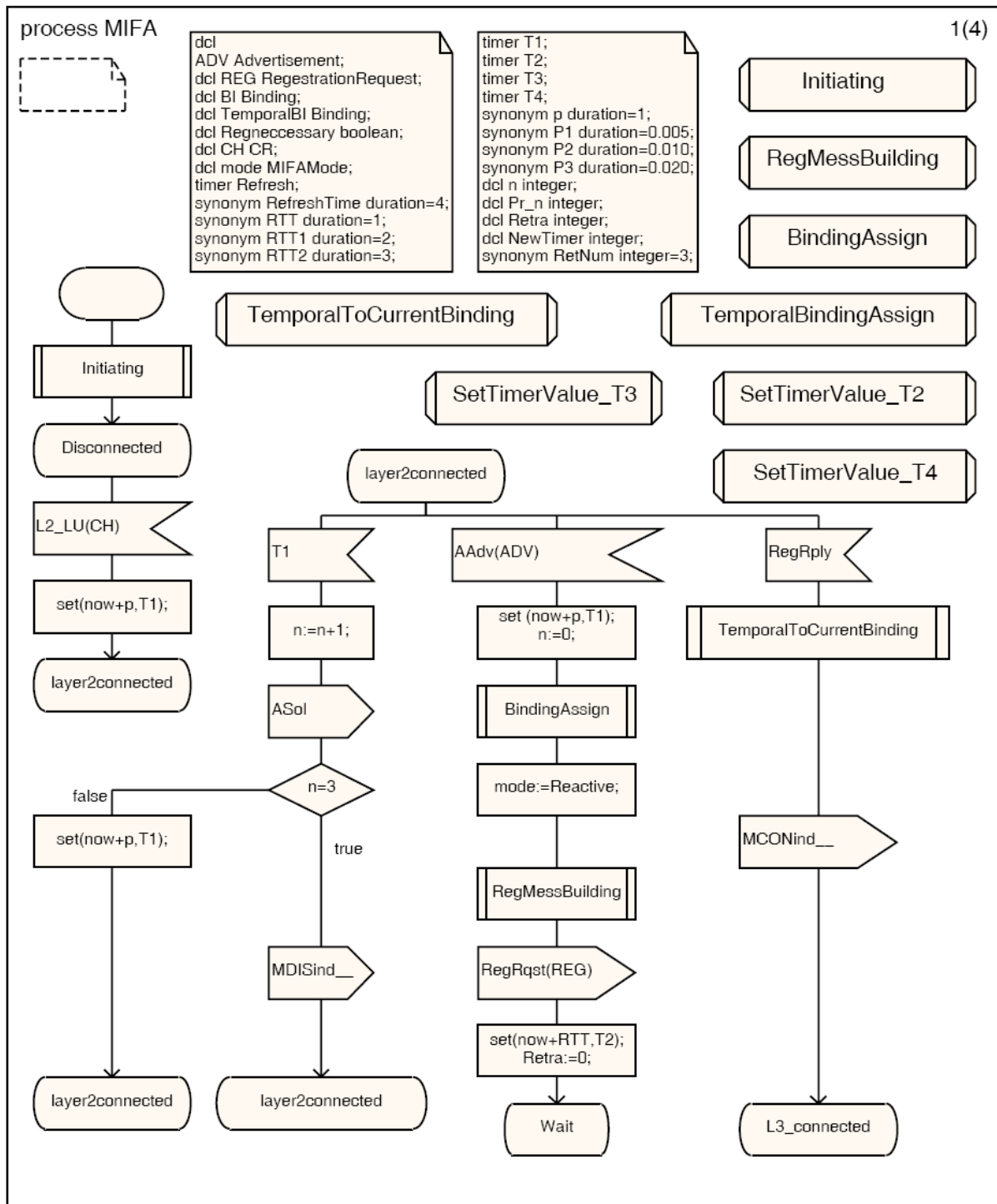


Fig 4.17: MIFA process – (1)

The MN waits mainly in the **Wait** state for a **RegRply** signal indicating a successful registration. When the MN receives this signal, it resets the timer **T2** and initiates another timer, named **Refresh** in the specification. Subsequently, the MN sends a **MCONind_** signal and transits itself to the **L3_connected** state. The timer **Refresh** is initiated to the registration lifetime and is used to renew the registration with the same FA. However, the MN may also receive an **AAdv(ADV)** signal or a timeout for the two timers, **T1** and **T2**, while in the **Wait** state. Expiration of the timer **T2** indicates that the **Reg_Rqst** message or the **Reg_Rply** has

been dropped. Therefore, another **RegRqst** signal is sent and the value of the timer **T2** is duplicated. The **SetTimerValue_T2** procedure is responsible for the duplication of the timer **T2**. After **RetNum** unsuccessful registrations, the MN assumes that the registration with the FA is not possible. This results in sending a **MDISind__** and a transition to the **layer2connected** state. Another time, after three expirations of the timer **T1** while in the **Wait** state, the MN assumes that the establishment of a layer 3 connection is not possible. As a result, it sends a **MDISind__** signal and transits itself to the **layer2connected** state. If the MN receives an **AAdv(ADV)** signal while in the **Wait** state, it reinitiates the timer **T1** and checks whether the advertisement is for the current FA with which the MN currently registers. If this is the case, the MN takes no action. Otherwise, the MN initiates a new registration process.

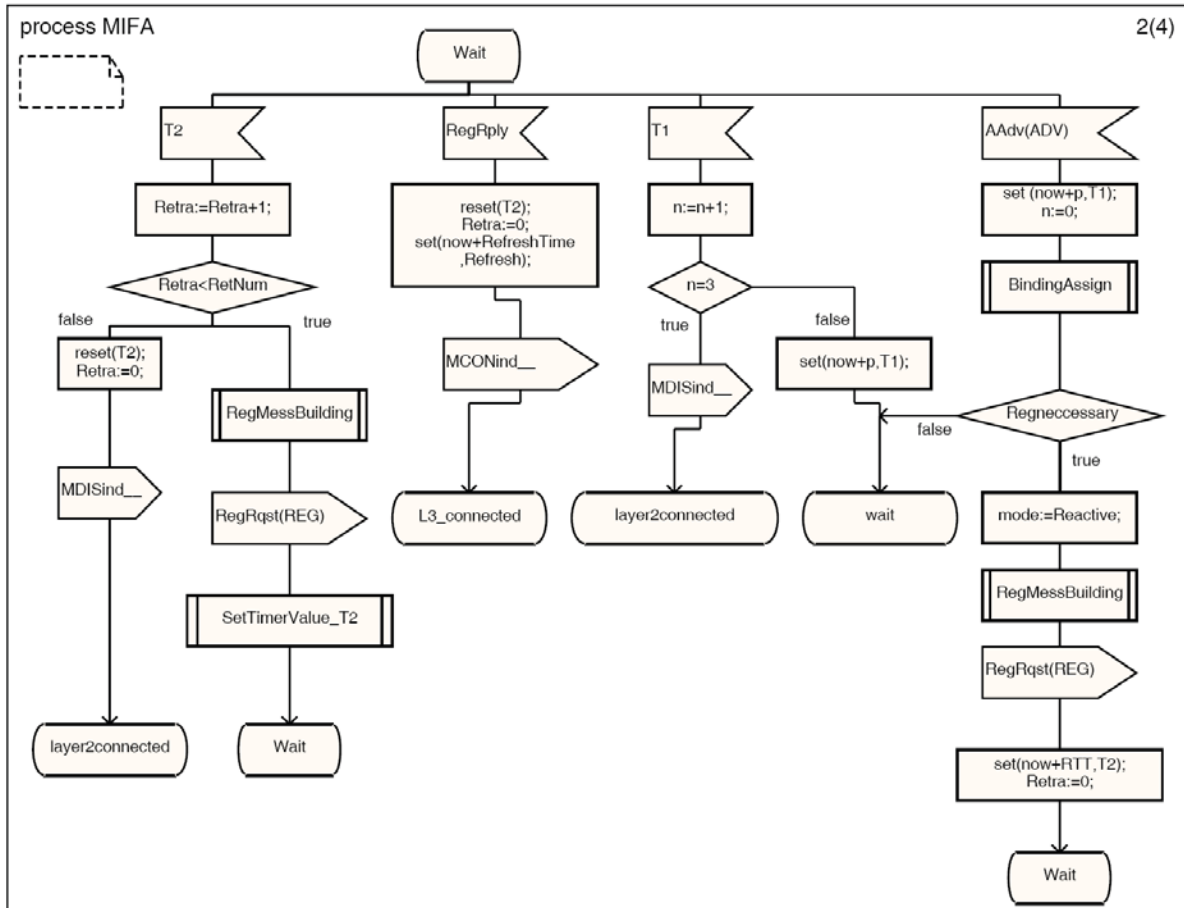


Fig 4.18: MIFA process – (2)

As mentioned above, the **L3_connected** state indicates a successful establishment of the layer 3 connection. The MN should receive **AAdv(ADV)** signals periodically while in this state. Receiving three timeout signals for the timer **T1** forces the MN to assume that the current FA is no longer available, send a **MDISind__** signal and transit itself to the **layer2connected** state with the aim to receive an **AAdv(ADV)** signal from another FA. Receiving the signal **Refresh** indicates an expiration of the current registration lifetime and causes a new registration process to be executed. In other words, the MN sends a **RegRqst** signal and initiates the timer **T2**. The receipt of **T2** signal indicates a dropping of the **Reg_Rqst** or **Reg_Rply** message. This dropping is recovered by a retransmission of the **RegRqst** signal and a duplication of the timer **T2**. Again, after **RetNum** unsuccessful retransmissions, the MN assumes that the current FA is no longer available, sends a **MDISind__** and transits itself to the **layer2connected** state.

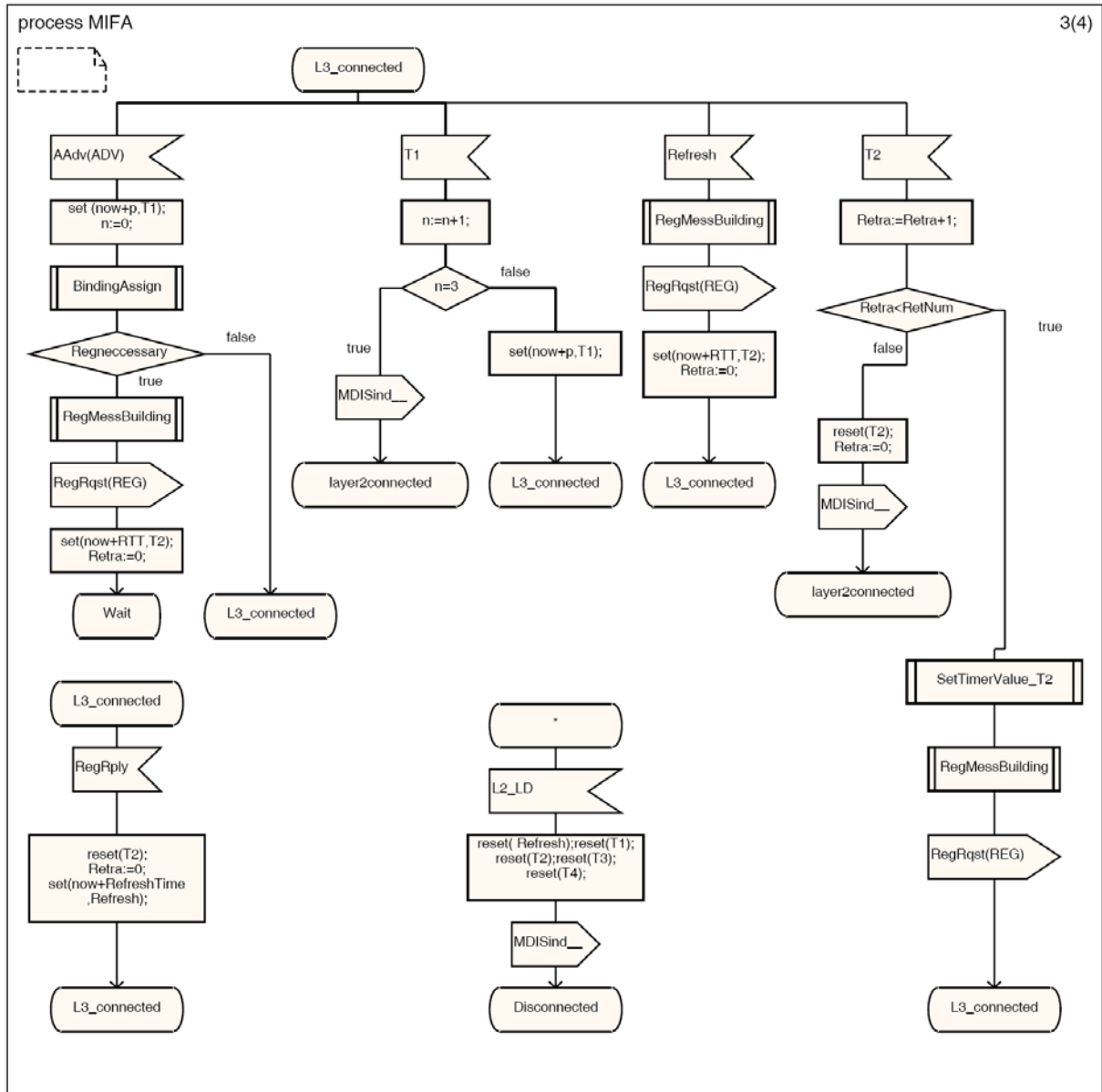


Fig 4.19: MIFA process – (3)

When the MN receives a **L2_Trigger** signal, it sends a **PrRtSol** signal to the current FA and initiates the timer **T3** by executing the **SetTimerValue_T3** procedure. The timer **T3** is used to recover the dropping of the **PrRtSol** or **PrRtAdv** message. In this case, the MN should receive a **PrRtAdv** or **T3** timeout signal from the **L3_connected** state. If the **PrRtAdv** signal is received, the MN resets the timer **T3**, stores the new mobility binding in the temporal binding list by executing the **TemporalBindingAssign** procedure, sends a **RegRqst** signal and initiates a new timer **T4** by executing the procedure **SetTimerValue_T4**. The timer **T4** is used to recover the dropping of the **Reg_Rqst** or **Int_Ack** message. Receiving an **Int_Ack** signal results in resetting the timer **T4** and indicates that the layer 3 handoff in predictive mode has been triggered successfully.

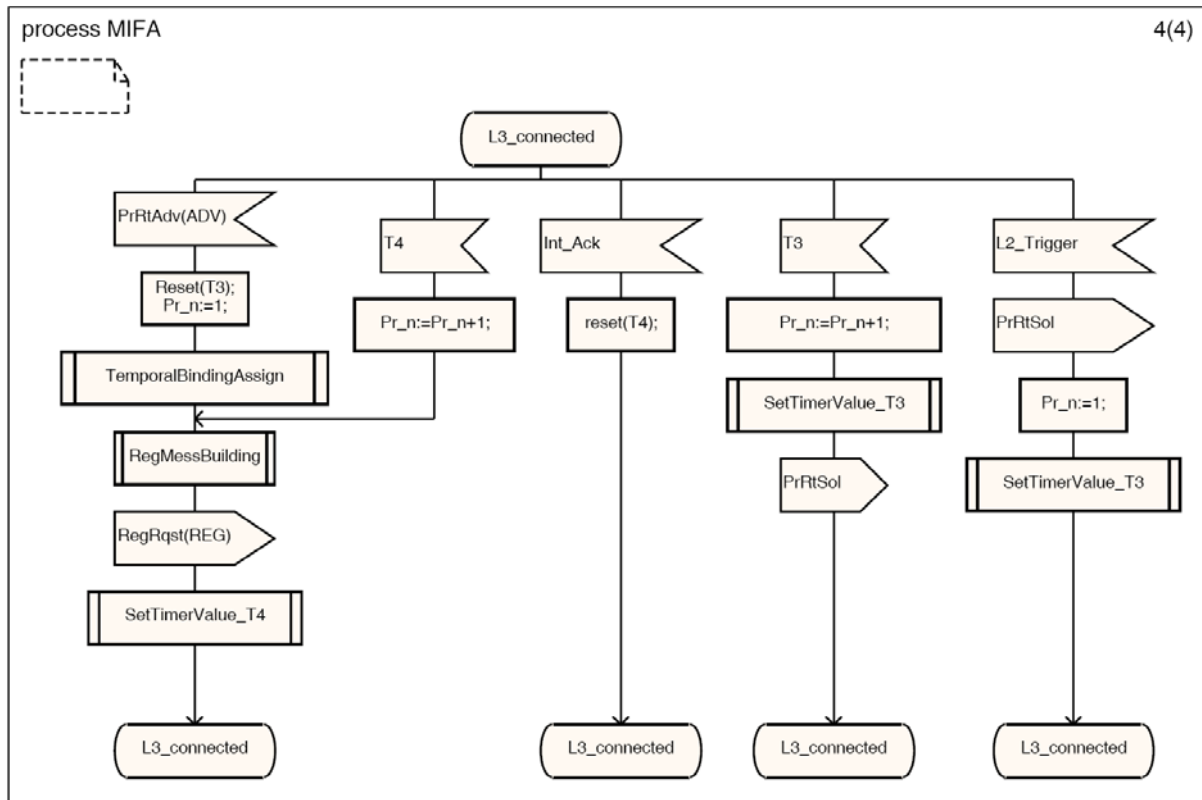


Fig 4.20: MIFA process – (4)

The structure of FA1, FA2, Neighbor_FA and the HA is similar to the structure of the MN. A more detailed description of the SDL specification of the MN as well as other nodes and channels is provided in appendix C.

4.3. Mobile IP Fast Authentication Protocol for IPv6 (MIFAv6)

MIFAv6 aims at achieving seamless mobility in IPv6 networks. Similar to MIPv6, no constraints are made on the network topology. In contrast to MIPv6, however, ARs should understand mobility. The basic idea is the same as for MIFAv4. Achieving seamless handoffs depends on groups of neighbor ARs termed as L3-FHRs. A L3-FHR of a certain AR comprises all ARs, to which movements from the given AR are possible. As before, there must be SAs between the ARs of the L3-FHR. Similar to MIFAv4, the methods presented in appendix B can be used to establish L3-FHRs for MIFAv6 as well. The following delivers a detailed description of MIFAv6 focusing on its operation, error recovery mechanisms and security considerations.

4.3.1. Operation Overview

Similar to MIFAv4, MIFAv6 can be operated in reactive as well as predictive mode. The predictive mode is preferable and is employed if the MN is able to generate L2-triggers in advance of the actual handoff. If MIFAv6 fails to operate in predictive mode, it reverts to reactive mode.

Predictive mode: when the MN expects a handoff, it starts scanning the medium for other available APs. If the discovered AP belongs to a new subnet, a L2-trigger is fired, which triggers the layer 3 predictive handoff procedure. The MN transmits a **BU** message to the new AR via the old one. The old AR acknowledges the receipt of the **BU** message by an **Int_Ack** message and forwards the **BU** message further to the new AR, which authenticates and

authorizes the MN. Following the completion of the layer 2 handoff, the MN receives a **BA** message from the new AR indicating an accepted or rejected registration. In case of success, the MN resumes its communication on both uplink and downlink. Downlink data packets are tunneled from the old AR to the new CoA until the HA, or possibly the CN, is notified, which results in redirecting the tunnel from the HA or CN to the new CoA.

Reactive mode: as mentioned above, if MIFAv6 fails to start the layer 3 handoff in advance, the reactive mode will be employed. After the MN finishes the layer 2 handoff, it sends a **BU** message to the new AR and a **BU** message to the CN. The new AR authenticates the MN and replies a **BA** message towards the MN, which then resumes its communication on uplink in the case of a successful registration. Concurrently to sending a **BA** message to the MN, the new AR notifies the old AR and the HA of the new CoA. Notifying the old AR results in forwarding the MN data packets to the new location until the layer 3 handoff is completed. As soon as the CN receives the **BU** message, it responds by sending a **BA** message and forwarding data packets towards the MN.

After finishing the layer 3 handoff, the new AR distributes the MN-specific data to all ARs belonging to its L3-FHR, so that the MN can further use MIFAv6 in subsequent handoffs. Similar to MIFAv4, the time required to notify the HA as well as the CN does not have any impact on the performance. Contacting the HA and CN is, however, necessary to optimize the route and to exchange the information required to ensure further seamless handoffs.

4.3.2. Initial Registration Procedure

MIPv6 with minimal extension is used for the initial registration with the HA and CN as shown in figure 4.21.

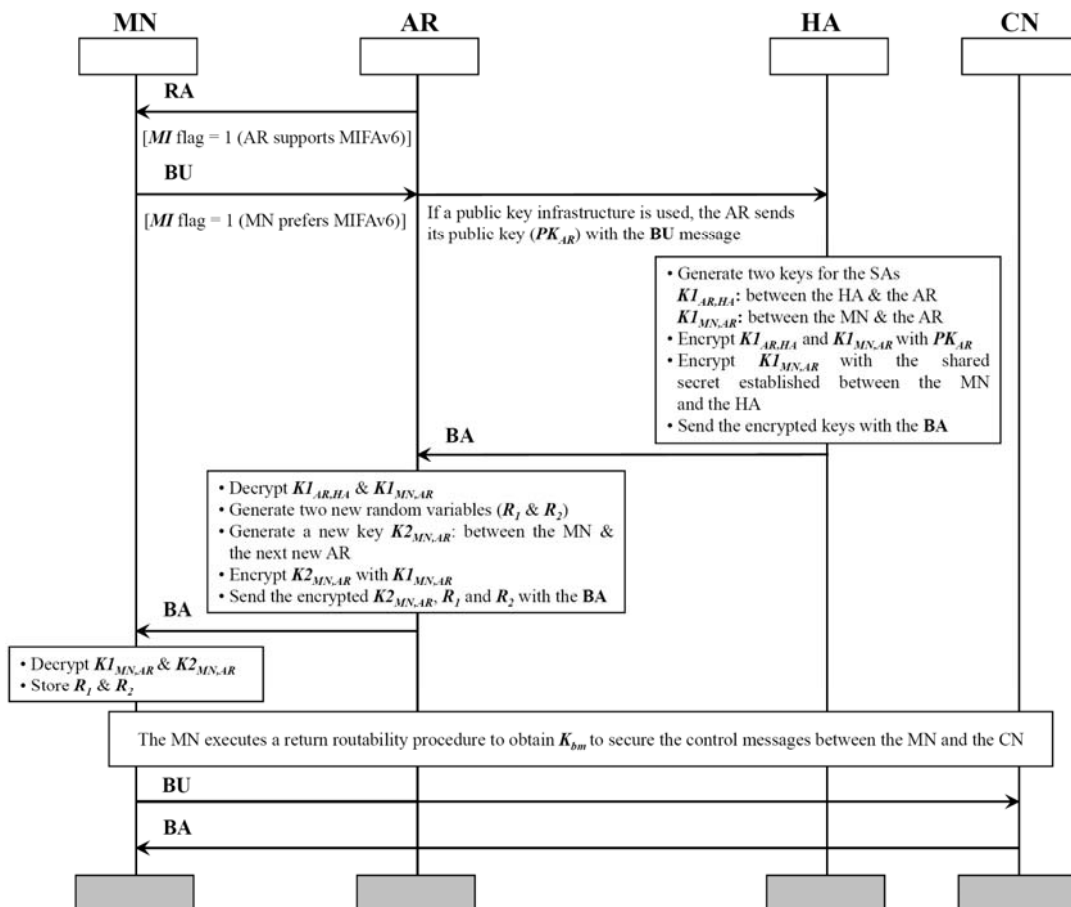


Fig 4.21: MIFAv6 initial registration procedure

Assuming the MN is away from home, the MN waits for a **RA** message from the AR serving the visited subnet. The **RA** message is built according to the specification of MIPv6 with one of the reserved bits as a **MI** flag indicating the support for MIFAv6. As soon as the **RA** message is received, a **BU** message is transmitted to the HA. Again, the **BU** message transmitted to the HA is built according to the standard MIPv6 specification with one of the reserved bits as a **MI** flag indicating that the MN prefers MIFAv6 in subsequent registrations.

As soon as the HA receives the **BU** message, it generates two SAs. The first SA is used to secure the control messages that should be exchanged between the HA and the current AR (the used key is $K1_{AR,HA}$). The second SA is used to secure the exchange of control messages between the MN and the current AR (the used key is $K1_{MN,AR}$). Both SAs have to be sent to the current AR, while the second SA should be sent to the MN as well. The SAs can be sent to both the current AR and the MN in suitable extensions to the **BA** message¹. Similar to MIFAv4, the SAs can be distributed using any adequate key distribution mechanism or technique.

Upon the current AR receives the **BA** message, it extracts the SAs and generates two random variables (R_1 and R_2). After that, another key, $K2_{MN,AR}$, is generated to be used to secure the exchange of control messages between the MN and the next new AR. $K2_{MN,AR}$ is encrypted with $K1_{MN,AR}$ and sent to the MN along with R_1 and R_2 in suitable extensions to the **BA** message, which is authenticated using $K1_{MN,AR}$. When the MN receives the **BA** message, it decrypts $K1_{MN,AR}$ and authenticates the message. If the authentication succeeds, the MN stores $K1_{MN,AR}$, $K2_{MN,AR}$ and the random variables. Subsequently, the MN performs a return routability procedure with the CN to obtain a binding management key (K_{bm}) to be used to authenticate the control messages that should be exchanged between both the MN and CN. Afterwards, the MN transmits a **BU** message to the CN. The **BU** message is built according to the standard specification of MIPv6 without any modification. The CN responds by sending a **BA** message built according to the MIPv6 standard specification as well.

4.3.3. Initial Authentication Exchange Procedure

After the initial registration procedure is completed, the initial authentication procedure should be executed. The current AR executes this procedure to obtain the data required to locally re-authenticate the MN during the next registration with the next new AR. To do this, a new key ($K2_{AR,HA}$) is generated to secure the exchange of control messages between the HA and the subsequent new AR, to which the MN may move in the future. After that, the new AR sends a **M_P_Not** message towards the HA. The **M_P_Not** message should contain the random variables (R_1 and R_2) and $K2_{AR,HA}$ encrypted with $K1_{AR,HA}$. Of course, $K2_{AR,HA}$ may be distributed to the HA using any other security infrastructure or mechanism declared as secure.

The HA then authenticates the **M_P_Not** message using $K1_{AR,HA}$. After a successful authentication, it derives $K2_{AR,HA}$ and calculates the authentication values ($Auth_1$ and $Auth_2$). These authentication values are calculated by applying a hash algorithm, HMAC-SHA1 [NIS95], [KBC97] or other algorithms considered as secure, on the random variables (R_1 and R_2) and additional information related to the MN (e.g. the home address, the MAC address, etc.). In the standard specification of MIFAv6, the authentication values are calculated in a

¹ More concrete, a public key infrastructure is used as a default mechanism same as by MIFAv4.

similar way as by MIPv4, see section 4.2.3. As by MIPv4, the authentication values are used to authenticate the MN with the HA during the next registration with the next new AR. $Auth_1$ stands for the authentication value the MN should generate during the next registration, while $Auth_2$ denotes the authentication value the HA has to generate as a response to $Auth_1$.

The HA sends the authentication values encrypted with $K1_{AR,HA}$ to the current AR with a M_P_Ack message that contains a HA features extension too. The HA features extension contains the features that the HA can deliver to the MN. These features are required to enable the next new AR to decide whether the HA can satisfy the MN's requirements or not. Furthermore, the M_P_Ack message must indicate whether the information distribution procedure should be executed or not. The initial authentication exchange procedure is shown in Figure 4.22.

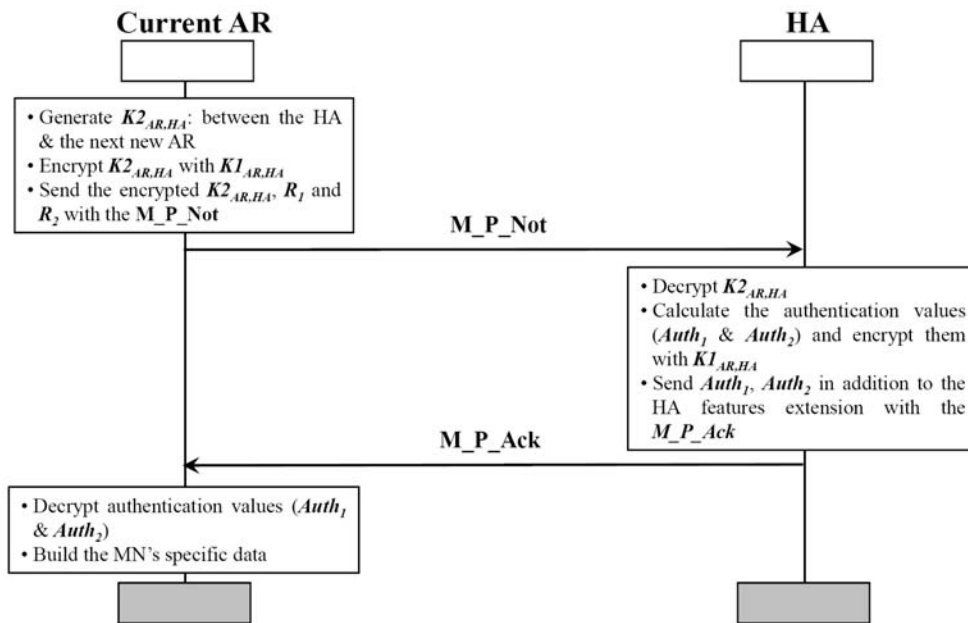


Fig 4.22: MIPv6 initial authentication exchange procedure

4.3.4. Information Distribution Procedure

After the current AR obtains the data required to fast re-authenticate the MN during the registration with the subsequent new AR, it distributes the data, also referred to as MN-specific data too, to all ARs belonging to the current L3-FHR. This is done by sending a M_P_Not message to each member of the current L3-FHR. The MN-specific data include the information sent from the HA to the current AR, the SAs between the MN and the ARs of the current L3-FHR and the SAs between these ARs and the HA. The distributed SAs should be encrypted with the SAs established between the ARs of the current L3-FHR. Each AR stores the MN-specific data in a soft state that must be refreshed periodically until the MN executes a handoff. After the handoff occurs, the MN-specific data will not be refreshed any more, which causes these data to be deleted from the L3-FHR members not participating in the handoff procedure. Each neighboring AR may optionally reply a M_P_Ack message to acknowledge the receipt of the M_P_Not message. Sending a M_P_Ack message should be requested, however, by the M_P_Not message. As a default, M_P_Not messages will not be acknowledged.

Similar to MIPv4, the information distribution procedure is optional. As a default, this procedure will be executed. However, for reasons similar to those presented in section 4.2.4,

the execution of this procedure may not be preferred. The information distribution procedure is shown in figure 4.23.

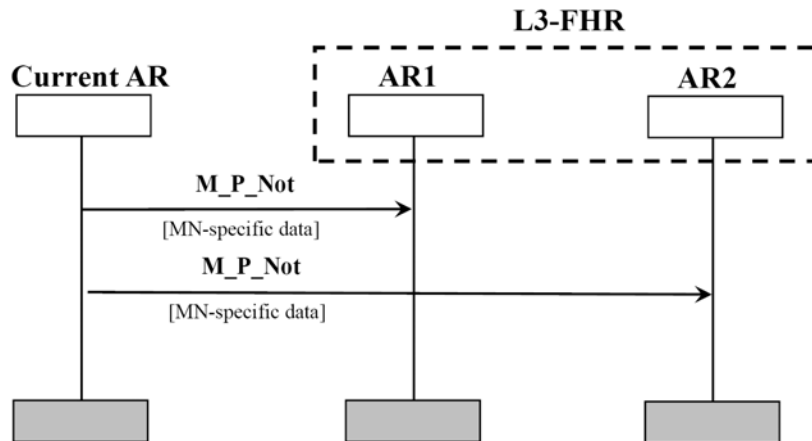


Fig 4.23: MIFAv6 information distribution procedure

4.3.5. Operation in Reactive Mode

Initiation of the layer 3 handoff: after the MN moves to another AR, it waits for a **RA** message first. After the receipt of the advertisement by the MN, the MN checks the **MI** flag. In case the new AR supports MIFAv6, the MN sends a **BU** message built according to the specification of MIFAv6 to the new AR and a **BU** message built according to the specification of MIPv6 to the CN. The **BU** message sent to the new AR contains the **MI** flag and a MIFA authentication extension that includes the authentication value $Auth_1$ calculated by the MN using the random value R_1 and the same hash function the HA uses. $K2_{MN,AR}$ is the shared secret used for the IPsec-SA established between the MN and the new AR.

Local authentication: after the new AR receives the **BU** message and successfully authenticates it, the new AR compares the value of $Auth_1$ sent from the MN with the value of $Auth_1$ that was sent previously from the old AR as a part of the MN-specific data. As known, this authentication value was calculated previously by the HA. The two authentication values will match only if they are calculated using the same shared secret and the same hash function. Subsequently, the new AR checks if the HA can satisfy the MN's requirements or not. This is achieved by examining the HA features extension distributed previously as a part of the MN-specific data.

Notification of the old AR, MN and HA: if the local authentication is completed successfully, the new AR sends a Handoff Notification (**Hn_Not**) message to the old AR causing the MN's data packets to be tunneled to the new CoA. This message should be secured using the IPsec-SA established between the ARs belonging to the L3-FHR of the old AR. After that, a new shared secret ($K3_{MN,AR}$) and two new random variables (R'_1 and R'_2) are generated. $K3_{MN,AR}$ will be used as a shared secret for the IPsec-SA that should be established between the MN and the subsequent new AR. The random values R'_1 and R'_2 will be used to calculate the authentication values during the next registration with the next new AR. Afterwards, a **BA** message containing R'_1 , R'_2 , $K3_{MN,AR}$ encrypted with $K2_{MN,AR}$ and the authentication value $Auth_2$ is transmitted to the MN. Of course, $K3_{MN,AR}$ may be distributed to the MN using other mechanism, e.g. AAA. After the transmission of the **Hn_Not** and the **BA** message, the new AR generates a new shared secret ($K3_{AR,HA}$) and encrypts it

with $K2_{AR,HA} \cdot K3_{AR,HA}$ is used as the shared secret for the IPSec-SA that has to be established between the HA and the subsequent new AR. The new generated random variables and the encrypted $K3_{AR,HA}$ are sent with a **BU** message towards the HA. Another time, $K3_{AR,HA}$ may be distributed to the HA by means of other secure key distribution mechanisms. The handoff procedure employing MIPv6 in reactive mode is shown in figure 4.24.

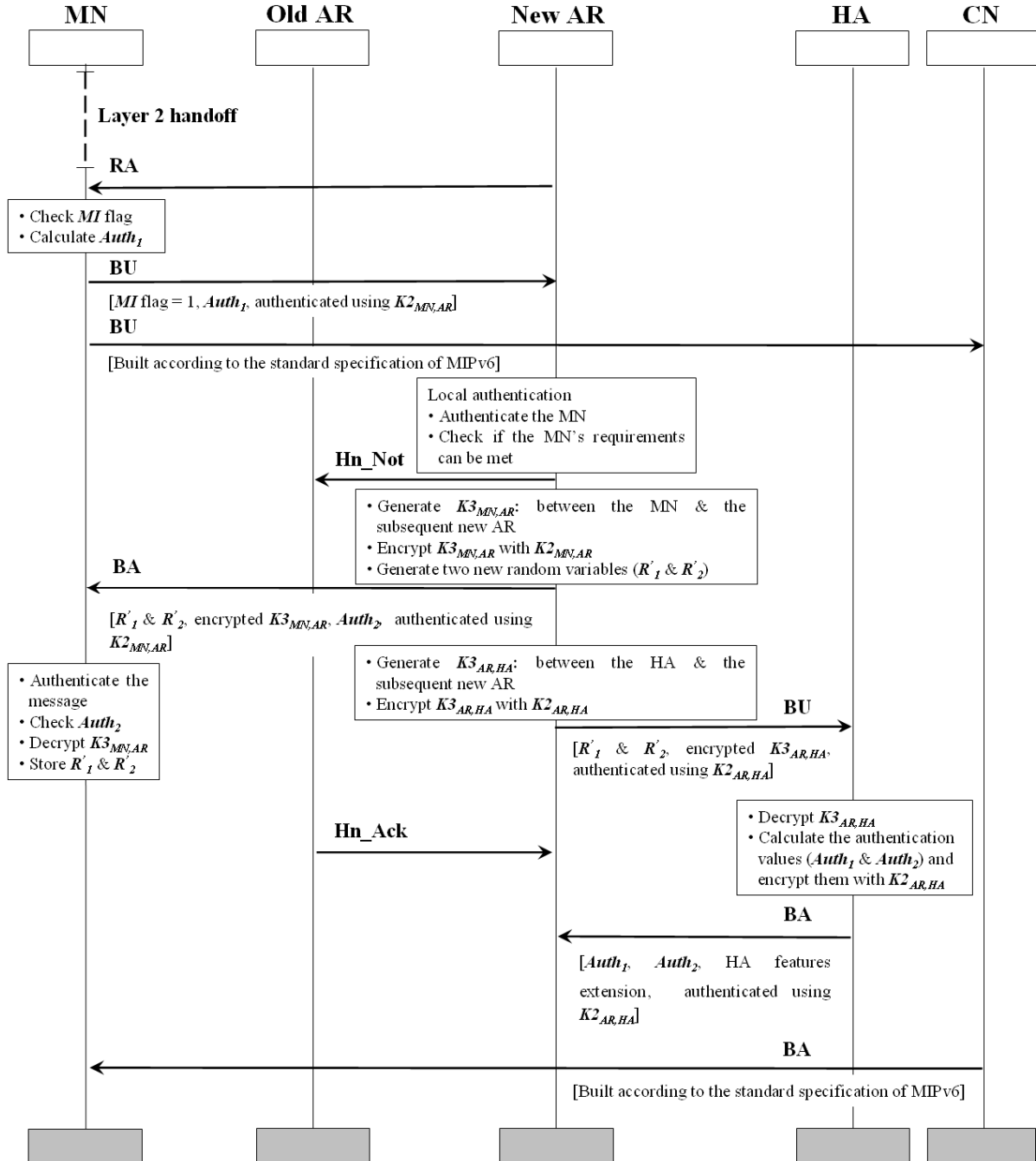


Fig 4.24: MIPv6 operation in reactive mode

Completion of the handoff from the MN point of view: after the MN receives and successfully authenticates the **BA** message, it calculates the value of $Auth_2$ using the random value R_2 , $K_{MN,HA}$ and the same hash function the HA uses. The calculated value should match the value of $Auth_2$ sent from the new AR with the **BA** message. If the registration is successful, the MN records $K3_{MN,AR}$

subsequent registration with the next new AR. From the MN point of view, the handoff procedure has been completed successfully and it can resume its communication on uplink.

Forwarding from the old AR: when the old AR receives the *Hn_Not* message, it responds by sending a Handoff Acknowledgement (*Hn_Ack*) message towards the new AR and begins forwarding MN data packets to the new CoA.

Exchange of authentication information between the HA and the new FA: as soon as the HA receives the *BU* message and successfully authenticates it, the HA records $K3_{AR,HA}$, calculates $Auth_1$ and $Auth_2$, encrypts them with $K2_{AR,HA}$ and sends them to the new AR with a *BA* message, which includes the HA features extension as well. In other words, the *BA* message contains the data required to construct the MN-specific data. In case data packets are tunneled to the MN's CoA using the triangular routing, the HA redirects the tunnel to the new CoA. Notice that because the MN receives its data tunneled from the old AR, the time required to inform the HA does not impact the performance.

Notification of the CN: as mentioned above, as the MN starts the layer 3 handoff, it sends a *BU* message built according to the specification of MIPv6 towards the CN, which responds with a *BA* message built according to the specification of MIPv6 too. If the routing optimization is used, the CN starts forwarding data packets to the new CoA directly. The main motivation behind the usage of MIPv6 for the registration with the CN is to enable all CNs supporting MIPv6 to benefit from MIFAv6 without requiring any updates.

Distribution of the MN-specific data: after completion of the layer 3 handoff, the current AR should distribute the MN-specific data to the ARs present in the current L3-FHR. In order to do this, the current AR executes the information distribution procedure, see section [4.3.4](#).

4.3.6. Address Auto-Configuration and Duplicated Address Detection

As known from MIPv6, MNs configure their CoAs either in stateful or stateless mode. After the configuration of the CoA, a Duplicated Address Detection (DAD) procedure should be executed to ensure that the CoA is not used by another MN. As known, CoA configuration and execution of the DAD procedure slow up the handoff. The duration of the DAD procedure may even exceed 1 sec [[TWY06](#)]. MIFAv6 avoids these sources for extra latency by enabling an in advance configuration of the CoA and execution of the DAD procedure.

Let us first consider the stateless mode. The MN generates a random value (*host-id*), which may be the MAC address of the MN too. This value is sent to the new AR with the *BU* message. After the new AR constructs the MN-specific data, it distributes them to the ARs of its L3-FHR. The *host-id* is distributed as a part of the MN-specific data as well. Each neighbor AR generates a CoA, *net-prefix+host-id*, and executes the DAD procedure in advance. When the MN moves to a new AR and receives the *RA* message, it constructs its CoA using the same *host-id* and the advertised *net-prefix*. Considering the stateful mode, upon the ARs of the current L3-FHR are notified of the MN, they can configure a CoA by means of DHCP servers and also execute the DAD procedure in advance. When the MN moves to one of these ARs, it obtains the pre-configured CoA.

Configuration of the CoA and execution of the DAD procedure in advance are optional. They are useful if the used DAD procedure consumes a long time. Conversely, if the DAD procedure does not consume a long time, it may be more useful not to perform these tasks in advance. Instead, mechanisms similar to those used by FMIPv6 can be used. The ARs support mobility and, therefore, they may have a list of all registered MNs. This enables them to execute the DAD procedure very fast, see [[Koo05](#)].

4.3.7. Error Recovery Mechanisms in Reactive Mode

Similar to MIFAv4, the errors that may happen can be:

1. loss of MIFAv6 support,
2. dropping of control messages and
3. moving to a non-member of the old AR's L3-FHR.

Loss of MIFAv6 support: upon the MN receives the **RA** message from the new AR, it checks if MIFAv6 is supported or not. If MIFAv6 is supported, the **BU** message is built according to the specification of MIFAv6. Otherwise, the **BU** message follows the specification of MIPv6. In case MIFAv6 is supported and the **BU** message could not be processed by the new AR according to MIFAv6 for any reason, the new AR sends the **BU** message to the HA, which takes care of further processing of the **BU** message and employing MIFAv6 in subsequent registrations.

Dropping of control messages: MIFAv6 sets a timer for each message it sends and expects a reply for it. If the reply does not come back before the timer expires, the message is retransmitted and the timer is duplicated. Let us take the dropping of the **BU** or **BA** message on the wireless link as an example. The MN initiates a timer ($T_{timer-1}$) as soon as it transmits the **BU** message to the new AR. The default value of $T_{timer-1}$ is set to $2 * RTT_{MN,AR}$, where $RTT_{MN,AR}$ is the round trip time between the MN and the new AR. If the MN does not receive the **BA** message before the expiration of $T_{timer-1}$, it retransmits the **BU** message and duplicates $T_{timer-1}$, see figure 4.25. Clearly, $T_{timer-1}$ has a short duration, which enables the MN to quickly detect any dropping of the **BU** or **BA** message on the wireless link and, thus, recover this dropping. The dropping of the **Hn_Not** or **Hn_Ack** message on the link between the new and old AR as well as the dropping of the **BU** or **BA** message between the new AR and the HA is processed in a similar way.

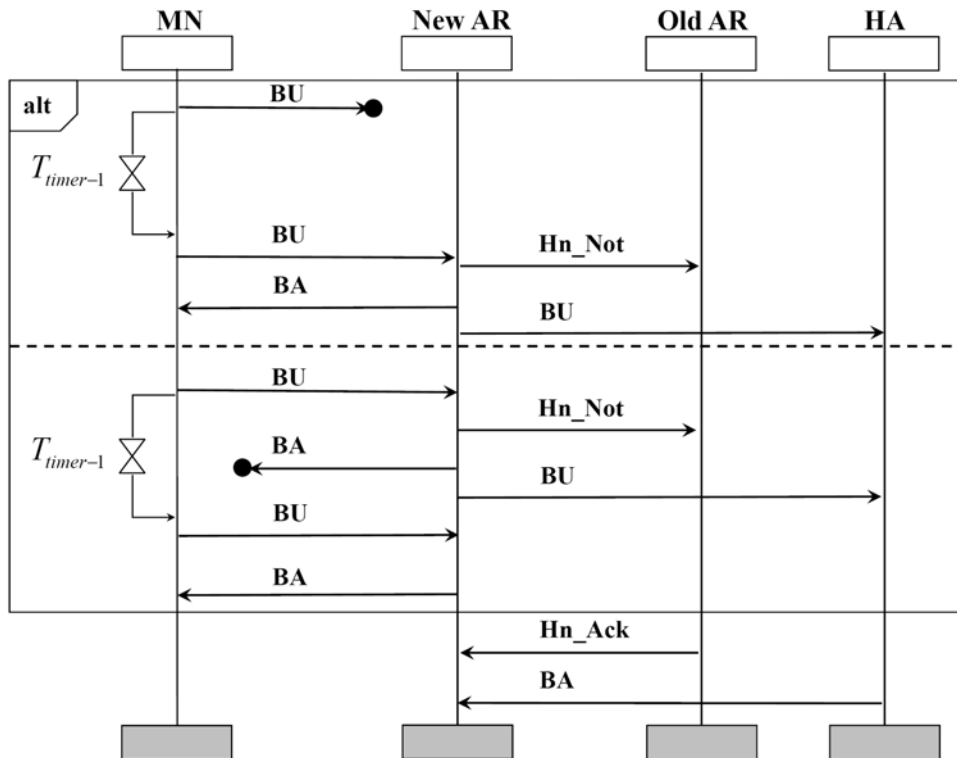


Fig 4.25: Recovery of **BU** or **BA** message dropping on the wireless link

After completion of the layer 3 handoff, the new AR must execute the information distribution procedure. As mentioned previously, this comprises the distribution of *M_P_Not* messages that may optionally be acknowledged by a transmission of *M_P_Ack* messages. In case *M_P_Not* messages have requested to be acknowledged, the new AR can simply detect the dropping of these messages by initiating a timer ($T_{timer-2}$) to $2 * RTT_{AR,AR}$, where $RTT_{AR,AR}$ is the round trip time between the current AR and the farthest neighbor AR. However, if *M_P_Ack* messages are not requested, the new AR can not detect the dropping of any *M_P_Not* message. Let us assume that the new AR has not received any *M_P_Not* message related to the MN. The MN will send a *BU* message to register with the new AR employing MIPv6. The new AR looks for the address of the old AR and forwards this message to it, see figure 4.26.

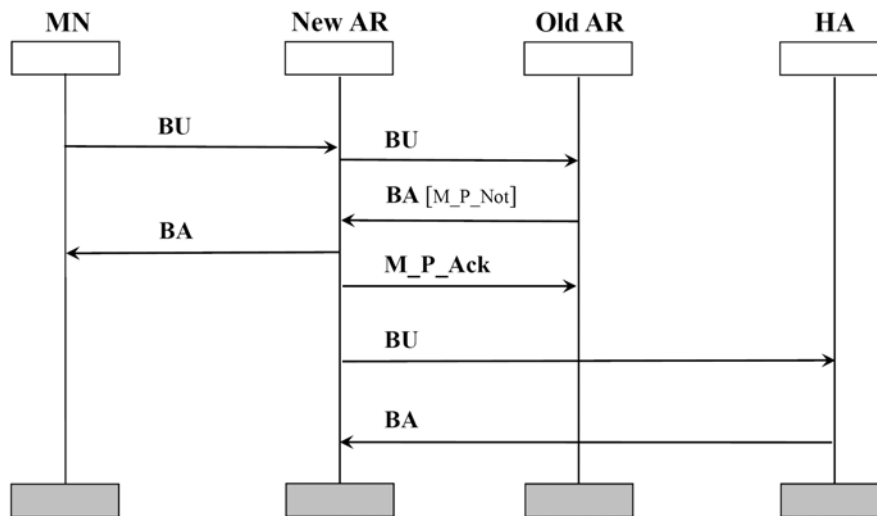


Fig 4.26: Recovery of *M_P_Not* message dropping (*M_P_Ack* is not requested and the new AR is a member of the L3-FHR of the old AR)

The *BU* message forwarded from the new to the old AR should be secured using the IPSec-SA established between the old and new AR. Notice that the new AR is a member of the L3-FHR of the old AR. Upon the old AR receives the *BU* message and successfully authenticates it, the old AR processes this message according to MIPv6 procedures and sends a *BA* message together with a *M_P_Not* message towards the new AR. Afterwards, the new AR extracts the MN-specific data present in the *M_P_Not* message, sends the *BA* message to the MN and acknowledges the receipt of the *M_P_Not* message by sending a *M_P_Ack* message back to the old AR. The new AR takes care, after that, of further processing using MIPv6.

Moving to a non-member of the old AR’s L3-FHR: in this case, the new AR forwards the *BU* message to the HA, which takes care of further processing. After that, the new AR attempts to join the L3-FHR of the previous AR by exchanging a *Mem_Join_Rqst* message and a *Mem_Join_Resp* message with the old AR. Clearly, a trust between the new and the old AR should be established before the exchange of these messages. AAA can be used for example to establish the required trust.

4.3.8. Operation in Predictive Mode

MIPv6 benefits in this mode from layer 2 triggers to anticipate the layer 3 handoff in advance, so that the sum of layer 2 and layer 3 handoff latencies can be minimized to the latency resulting from the layer 2 handoff only. As the MN is switched on or wants to initiate a layer 3 connection, the initial registration, initial authentication exchange and information distribution procedure presented in sections 4.3.2, 4.3.3 and 4.3.4 will be executed.

In advance preparation for the handoff: similar to MIFAv4 in predictive mode, the MN starts scanning the medium when the quality of the current link is deteriorating. If the new detected AP belongs to a new subnet, a L2-trigger is fired resulting in prompting the layer 3 handoff in advance. Of course, the L2-trigger should contain sufficient information, from which the new AR can be accurately determined. The MN sends a *Pr_Rt_Sol* message to the current AR, which replies a *Pr_Rt_Adv* that is similar to the normal *RA* message. However, it is sent on behalf of the new AR. Again, it is assumed that the ARs in each L3-FHR exchange periodic *RA* messages, so that the current AR can directly reply a *Pr_Rt_Adv* message upon receipt of the *Pr_Rt_Sol*.

Start of the layer 3 handoff: after the receipt of the *Pr_Rt_Adv* message, the MN transmits a *BU* message to the new AR via the old one. The *BU* message should contain a MIFA authentication extension that contains the authentication value $Auth_1$. The message is secured using the IPsec-SA established between the MN and the new AR. Upon the old AR receives the *BU* message, it responds by sending an *Int_Ack* message. This message ensures the MN that the *BU* message has not been dropped on the wireless link. In addition, this message may contain a handoff possibilities extension that indicates if the handoff to the new AR employing MIFAv6 is possible or not. Although the MN can know from the *MI* flag present in the *Pr_Rt_Adv* message if the new AR supports MIFAv6 or not, the handoff possibilities extension is required to cover the errors that may happen due to unknown or unpredicted reasons. If the MN-specific data have been distributed previously, the *BU* message is forwarded to the new AR without any modification. If this is not the case, the current AR builds a *M_P_Not* message containing the MN-specific data, adds it to the *BU* message and sends the new message to the new AR secured using the IPsec-SA established between the old and the new AR. Notice that the new AR is a member of the L3-FHR of the old one.

Local authentication: the new AR authenticates the *BU* message using $K_{AR,AR}$ as a shared secret if a *M_P_Not* exists in it. Otherwise, the *BU* message is authenticated using $K_{MN,AR}$ as a shared secret. If the authentication is successful, the new AR compares the value of $Auth_1$ calculated by the MN with the value of $Auth_1$ calculated by the HA, which was received from the old AR as part of the MN-specific data. If the comparison is successful, the new AR determines whether the HA can satisfy the MN's requirements or not.

Notification of the old AR and HA: if the authentication with the new AR is successful, the new AR sends a *BA* message back to the old AR. This message serves as an indicator for the success of the handoff. Subsequently, the new AR generates two new random variables (R'_1 and R'_2) and a new key ($K_{AR,HA}$). $K_{AR,HA}$ is encrypted with $K_{AR,HA}$ and will be used as a shared secret for the IPsec-SA that should be established between the HA and the subsequent new AR. The new AR sends the encrypted key and the random variables with a *BU* message to the HA. The *BU* message is authenticated using the IPsec-SA established between the new AR and the HA.

Exchange of authentication information between the HA and the new FA: after the HA receives the *BU* message and successfully authenticates it, $K_{AR,HA}$ is derived, the authentication values ($Auth_1$ and $Auth_2$) are calculated and encrypted with $K_{AR,HA}$. Following this, a *BA* message containing the authentication values and the HA features extension is sent to the new AR. If the triangular routing is used, the HA starts intercepting the MN data packets and forwarding them towards the new CoA.

Forwarding from the old AR: when a L2-LD trigger appears at the old AR, it intercepts the MN data packets and forwards them to the new CoA. The packets are buffered in the new AR until a L2-LU trigger is raised.

Notification of the MN: following the appearance of a L2-LU trigger at the new AR, the new AR generates a key ($K3_{MN,AR}$) to be used as a shared secret for the IPSec-SA that should be established between the MN and the subsequent new AR. This key is encrypted with $K2_{MN,AR}$ and sent to the MN together with the newly generated random variables with a **BA** message. After the receipt of this message by the MN, the MN resumes its communication.

Notification of the CN: after the MN completes the handoff to the new AR, it has to register with the CN. The registration with the CN is executed according to the standard specification of MIPv6, which implies an exchange of **BU** and **BA** messages. The operation of MIFAv6 in predictive mode is shown in figure 4.27.

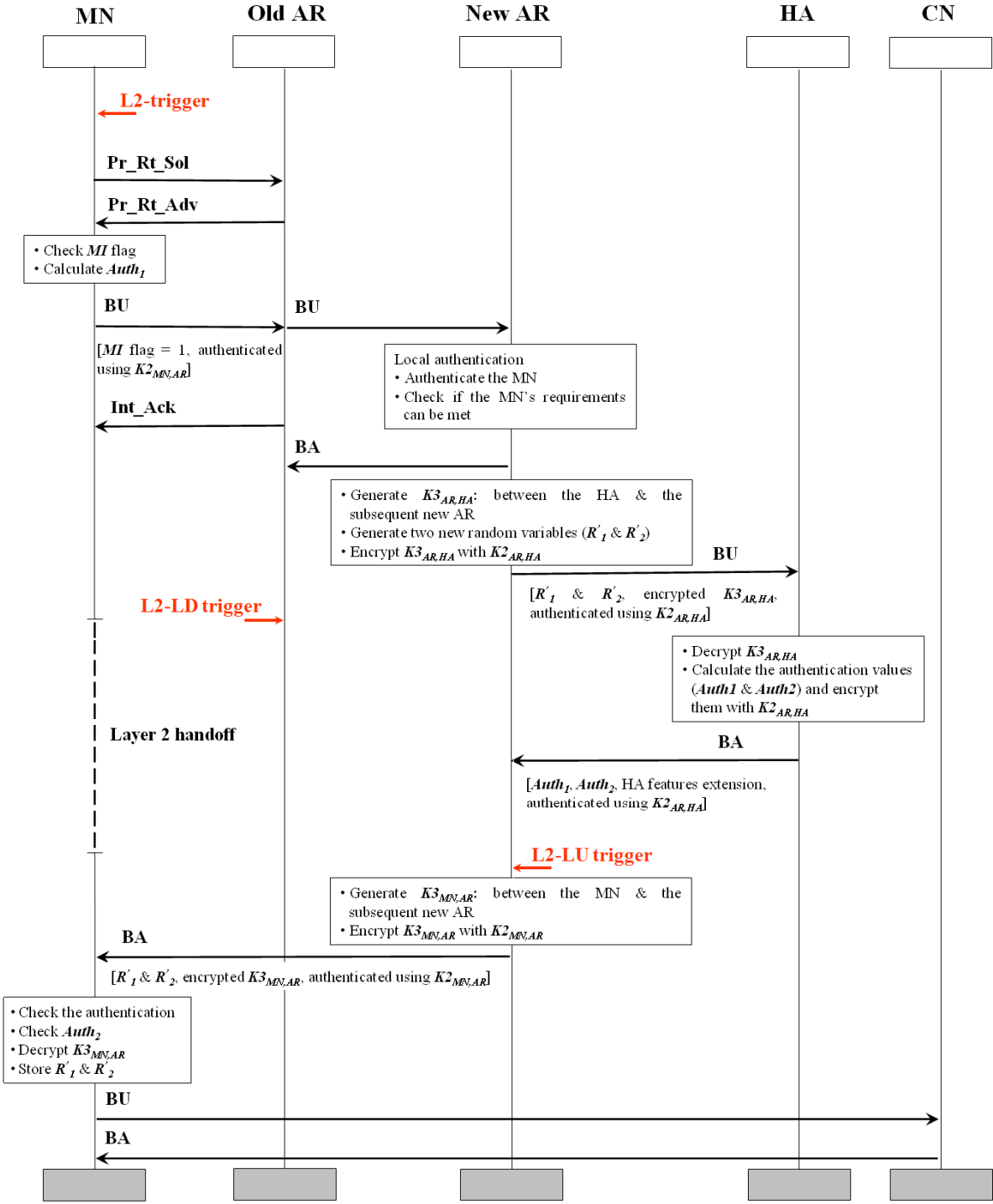


Fig 4.27: MIFAv6 operation in predictive mode

Distribution of the MN-specific data: following the completion of the layer 3 handoff, the new AR should distribute the MN-specific data to the ARs present in its L3-FHR. In order to do this, the new AR executes the information distribution procedure, see section [4.3.4](#).

4.3.9. Error Recovery Mechanisms in Predictive Mode

Similar to MIFAv4 in predictive mode, failures that may happen are:

1. loss of MIFAv6 support,
2. dropping of control messages,
3. appearance of layer 2 triggers at inappropriate times and
4. moving to a non-member of the old AR's L3-FHR.

Loss of MIFAv6 support: when a L2-trigger is raised at the MN, the MN exchanges *Pr_Rt_Sol* and *Pr_Rt_Adv* messages with the current AR. The *Pr_Rt_Adv* contains the *MI* flag, which indicates MIFAv6 support by the new AR. If the *MI* flag is set to 1, the MN constructs the *BU* message according to the specification of MIFAv6. Otherwise, the MN goes to execute a layer 2 handoff followed by a layer 3 handoff employing MIPv6. Assuming that the *MI* flag is set to 1 and the MN has sent the *BU* message constructed according to the specification of MIFAv6. However, the old AR is, for any reason, unable to process the message according to MIFAv6. In this case, the current AR informs the MN by sending an *Int_Ack* message containing a handoff possibilities extension indicating that the handoff to the new AR employing MIFAv6 is not possible. The MN then responds by resorting to MIPv6.

Dropping of control messages: MIFAv6 in predictive mode recovers dropping of control messages in the same manner as in reactive mode. It sets a timer to retransmit any message dropped on the way to its destination.

Appearance of layer 2 triggers at inappropriate times: let us first discuss the delayed appearance of a L2-trigger at the MN. One of the following cases may happen.

1. The MN was unable to receive the *Pr_Rt_Adv* message from the old AR due to a delayed L2-trigger: in this case the MN employs MIFAv6 in reactive mode.
2. The MN could only receive the *Pr_Rt_Adv* and was unable to send the *BU* message: the MN, again, employs MIFAv6 in reactive mode as well. Of course, it does not wait for a *RA* message from the new AR.
3. The MN could send a *BU* message towards the old AR. However, this message has been dropped and the wireless link between the MN and the old AR has been broken down before the MN could detect the *BU* message dropping: in this case the MN proceeds with the layer 2 handoff and waits, after that, for a *BA* message from the new AR. Clearly, the MN should wait for a short time equal to the half of the round trip time between the MN and the new AR. If the MN receives nothing, it employs MIFAv6 in reactive mode.

Let us now discuss the raising of the L2-LD trigger at the old AR. One can expect one of the following cases:

1. The old AR has detected the L2-LD trigger without receiving any message from the MN, i.e. the MN was unable to predict the handoff: in this case MIFAv6 in reactive mode will be employed.
2. The L2-LD trigger was raised at the old AR before the old AR could receive the *BU* message. However, *Pr_Rt_Sol* and *Pr_Rt_Adv* messages have been exchanged with

the MN: in this case, MIFAv6 will, again, be executed in reactive mode. However, the old AR may be configured to start buffering the packets sent to the MN upon detecting the L2-LD trigger. Buffered packets are forwarded to the new CoA as soon as the old AR is notified.

3. The old FA has received the **BU** message from the MN and has forwarded it to the new AR. However, the old AR has detected the L2-LD trigger before the receipt of a **BA** message from the new AR: in this case the old AR works same as in the previous case.

Let us now address the appearance of the L2-LU trigger at the new AR. Assuming that the new AR has detected no or a delayed L2-LU trigger or has sent the **BA** message to the MN and this message has been dropped. The MN will attempt to proceed first with MIFAv6 in reactive mode after waiting for a short time equal to the half of the round trip time between the MN and the new AR. This results in sending another **BU** message to the new AR, which either sends or retransmits the **BA** message. Clearly, the retransmitted **BA** message should not be a copy of the dropped one.

Movement to a non-member of the old AR's L3-FHR: let us assume that the MN has fired a L2-trigger at an appropriate time and has sent a *Pr_Rt_Sol* message to the old AR. In the case that the predicted new AR is not a member of the current L3-FHR, we can distinguish between two situations.

1. The old AR does not know the IP address of the new AR: the old AR does not send a *Pr_Rt_Adv* message to the MN. This forces the MN to execute a layer 2 handoff and to attempt to employ MIFAv6 in reactive mode after that. The new AR forwards the **BU** message to the HA and attempts to join the L3-FHR of the old one, see section [4.3.7](#) for details.
2. The old AR knows the IP address of the new one: provided that the new AR supports MIFAv6, the old AR transmits a *Pr_Rt_Adv* on behalf of this new AR and sets the **MI** flag to 1.
 - a. If the MN was able to exchange **BU** and *Int_Ack* messages with the old AR, the *Int_Ack* contains a handoff possibilities extension indicating that a handoff to the detected new AR using MIFAv6 is not possible. This results, of course, in resorting to MIPv6.
 - b. If the MN was unable to send the **BU** message to the old AR, it will attempt to register with the new AR using MIFAv6 in reactive mode after the layer 2 handoff. The new AR in turn forwards the **BU** message to the HA and tries to join the L3-FHR of the old AR.
 - c. If the MN has sent the **BU** message, but was unable to receive the *Int_Ack* message. The MN assumes that it will receive a **BA** message directly after the layer 2 handoff. Therefore, it wait for a short time equal to the half of the round trip time between the MN and the new AR following the layer 2 handoff aiming at receiving a **BA** message. Of course, the MN will not receive this message in this case. Therefore, it attempts to register with the new FA employing MIFAv6 in reactive mode. The new AR in turn forwards the **BU** message to the HA and attempts to join the L3-FHR of the old AR.

4.3.10. Security Considerations

This section discusses some security considerations that should be taken into account to guarantee a secure exchange of data and control messages between the entities participating in MIFAv6 procedures.

4.3.10.1. Control Messages and Data Exchanged between the MN and the HA

BU and **BA** messages are exchanged between the HA and the MN during the initial registration and in case of errors that may force the MN to resort to MIPv6. Similar to MIPv6, to protect the integrity and authenticity of these control messages, the both MN and HA should use an IPsec-SA. They should use ESP in transport mode and should use a non-null authentication algorithm, such as HMAC-SHA1, to guarantee a data-origin authentication, a connectionless integrity and an optional anti-replay protection. Authentication Header (AH) [KAt98a] may be used as well. The used shared secret between the MN and the HA should be random and unique and must be distributed off-line. Automatic key management approaches, e.g. IKE [HCa98], may be supported too. Regarding the data communication between the MN and the HA, the same security considerations of MIPv6 are used.

4.3.10.2. Control Messages Exchanged between the New AR and the HA

In order to secure the control messages exchanged between the new AR and the HA, an IPsec-SA between the both should be established. This IPsec-SA is similar to the IPsec-SA established between the HA and the MN. However, the used shared secret should be distributed on-line. AAA can be used for the distribution of these shared secrets. IKE or any other key distribution method declared as secure may be used too. In addition, similar method to that presented in section 4.2.9.1 for the generation of necessary keys can be used by MIFAv6 as well.

4.3.10.3. Control Messages and Data Exchanged between the MN and the ARs

To secure the control messages exchanged between the MN and the old as well as the new AR, it is highly recommended to use an IPsec-SA. Similar to MIPv6, ESP in transport mode with a non-null authentication algorithm should be used. An optional anti-Replay may be provided by this IPsec-SA too. The used shared secrets of these IPsec-SAs should be distributed on-line.

After the MN hands off to a new AR, downlink packets are tunneled from the old AR to the MN via the new AR until the HA and possibly the CN is informed. This tunnel is protected by means of the IPsec-SA established previously between the old AR and the MN.

4.3.10.4. Control Messages Exchanged between the Old and the New AR

In order to secure the control messages exchanged between the old and the new AR, it is highly recommended that a similar IPsec-SA to that established between the MN and the HA should be used. As known, the new AR is a member of the L3-FHR of the old one. As described previously, there must be shared secrets between the ARs belonging to a certain L3-FHR. These shared secrets can be distributed off-line by an administrator or on-line by an adequate security infrastructure or mechanism.

4.3.10.5. *Control Messages and Data Exchanged between the MN and the CN*

Update of mobility bindings at CNs as well as data communication with them is executed according to the standard specification of MIPv6. Therefore, the security considerations that should be taken into account are the same as by MIPv6, see [[JPA04](#)].

4.3.10.6. *Ingress Filtering*

There are no problems regarding the ingress filtering in MIFAv6. This is because MIFAv6 extends MIPv6, which does not suffer from this problem at all.

4.4. Conclusion

This chapter has described our proposal (MIFA) for providing seamless mobility in IP-based networks. The fact that future All-IP networks will support both IPv4 and IPv6 has motivated us to specify our proposal for IPv4 as well as IPv6 networks.

MIFA utilizes the fact that MN movements are limited to a small set of neighboring subnets, referred to as L3-FHRs. Thus, contacting these neighbors and providing them in advance with sufficient data related to the MN, also called MN-specific data, enable them to fast re-authenticate the MN after the handoff. This enables the HA to delegate the authentication to the new visited subnet. Thus, the MN only requires contacting its new subnet to be able to resume its communication on uplink. On downlink, the MN relies on a temporal tunnel from the old to the new subnet. Clearly, this makes the handoff performance independent of the delay between the new subnet and the HA or the CN.

MIFA can be operated in predictive and reactive mode. The predictive mode is preferred and will be employed if the MN was able to fire a L2-trigger at sufficient time before the handoff occurs. Failing in starting the predictive mode results in employing MIFA in reactive mode, which implies performing a layer 2 handoff followed by a layer 3 handoff. MIFA is designed to be a robust protocol by including many error recovery mechanisms that cover most failures that may happen. Security issues are studied carefully in this protocol, so that no new vulnerabilities due to MIFA are produced.

We believe that MIFA advances the state of the art and expect that the predictive mode is able to achieve lossless handoffs. Even in reactive mode, seamless handoffs remain possible. Of course, evaluation of our proposal is necessary to show whether these expectations are true or not. The evaluation of the proposal compared to others will be handled in the following two chapters.

5. Analysis of Mobility Management Protocols

Analyses of mobility management solutions focus on performance evaluation and cost estimation. The analyses can be done using mathematical models, simulation studies or real implementations. Implementation and simulation of protocols normally take a long time. However, they deliver detailed and accurate results. Mathematical models can be developed more quickly and result in a good estimation of the performance. Until now, there is no generic mathematical model that allows for the analyses of a wide range of mobility management solutions. Therefore, the development of such a model will be a major contribution that will significantly simplify the analyses of mobility management solutions. This chapter describes such a generic mathematical model.

The parameters of the generic model are set according to the characteristics of the studied protocols, network topologies and mobility scenarios. Performance is analyzed with respect to the average handoff latency and expected average number of dropped packets per handoff taking the dropping of control messages into account. The developed model can be applied for break-before-make as well as make-before-break mobility management protocols. Cost estimation focuses on the estimation of the location update and packet delivery cost. The location update cost results from the update of mobility bindings after movements, while the packet delivery cost results from forwarding data packets from the CN to the new MN's location. The total cost is calculated accordingly as the sum of these two costs using an adequate weighting factor for each. A major contribution of this chapter is the analysis of performance taking cost into account. In other words, discussing which performance gain will be obtained from employing a certain mobility management protocol and what is the cost that should be considered?

This chapter is structured as follows: basic assumptions are provided in section [5.1](#). Modeling of network topologies and mobility scenarios are discussed in sections [5.2](#) and [5.3](#). After that, the mathematical model analyzing the performance of break-before-make as well as make-before-break mobility management protocols is described in section [5.4](#). The cost resulting from mobility management protocols is estimated in section [5.5](#). Section [5.6](#) provides the analytical assumptions and parameters for the used network topology, movement pattern and studied mobility management protocols. The obtained results are discussed in this section as well. The discussion focuses on the evaluation of MIFA compared to the other studied protocols. The impact of mobility scenarios is analyzed in section [5.7](#). Section [5.8](#) deals with the question: which performance gain will be obtained from employing a certain mobility management protocol and at which cost? The validation of our generic model results compared to simulation results as well as results of real testbeds is provided in section [5.9](#). Section [5.10](#) briefly describes several tools implemented to simplify and speed up the analysis of mobility management protocols by means of the developed generic model. Lastly, section [5.11](#) concludes this chapter with the main results.

5.1. Basic Assumptions

It is supposed that the MN moves within one domain. Each node offering IP-connectivity inside the domain represents a Mobility Agent (MA) (i.e. a FA, AR, MAG, etc.). In order to model the performance of mobility management protocols, the parameters listed in table 5.1 are assumed.

Parameter	Definition
Z	Number of nodes offering IP-connectivity inside the domain.
N	Number of neighbors of a node offering IP-connectivity inside the domain.
$t_{x,y}$	Time required for a message to pass through the link from node x to y .
$D_{x,y}$	Distance between two nodes x and y with respect to the number of hops.
$\bar{D}_{x,y}$	Average distance between two nodes x and y with respect to the number of hops.
τ_1	Delay on the wireless link.
τ_2	Delay on a wired link between two hops.
t_0	Time at which the handoff begins.
t_{in}	Time at which the MN enters the overlapping area between two neighboring cells.
t_g	Time at which a L2-trigger is raised at the MN.
t_{LD}	Time at which a L2-LD trigger appears.
t_{LU}	Time at which a L2-LU trigger appears.
T_{L2HO}	Average value of the layer 2 handoff latency.
T_x	Layer 3 handoff latency when updating the mobility binding at node x .
T'_x	Layer 3 handoff duration when updating the mobility binding at node x .
t_x	Time at which the node x is notified of the handoff.
LP_x	Expected number of dropped packets when updating the mobility binding at node x .
a_x	Processing time required to process a control message in node x .
T_{timer}	Value of the timer initiated upon sending a control message for which a response is expected.
S	Number indicating amount of times a certain control message was transmitted.

Tab 5.1: Parameters assumed to model the performance of mobility management protocols

Of course, the MN can not send or receive packets during the handover. In case a control message has been dropped, the value of T_{timer} will be doubled. Throughout this chapter, the control message sent to update the mobility binding is referred to as the *update message*.

Cellular cells are assumed to overlap and MNs able to receive signals from more than one AP/BS. The appearance of t_g causes the layer 3 handoff to be triggered in advance. t_{LD} is raised when the MN moves outside the overlapping area. The appearance of this trigger also means that the MN should start the layer 2 handoff. Therefore, t_{LD} is equals to t_0 .

In order to estimate the location update and packet delivery cost, the terms listed in table 5.2 are assumed.

Parameter	Definition
T_r	Average residence time the MN spends inside the region of each MA.
luc_x	Location update cost experienced when updating the location at node x.
$pd c_{x,y}$	Cost resulting from forwarding data packets from node x to y.
$Tc_{x,y}^S$	Transmission cost of location update message from node x to y.
$Tc_{x,y}^D$	Transmission cost of data packet from node x to y.
a'_x	Processing cost of location update message in node x.
d_x	Processing cost of data packet in node x.
λ	Packet arrival rate.

Tab 5.2: Parameters assumed to estimate the location update and packet delivery cost resulting from mobility management protocols

All delays the packets encounter at different network elements as well as at links are assumed to be deterministic. This assumption is motivated by simulation studies of IP networks with mixed traffic (short voice packets in addition to large data packets). These studies for a managed IP network show that the delay jitter is minimal compared to the overall delay [BBM04]. Especially, the delay is far from being exponentially distributed as assumed by other studies on mobility management protocols [BCC03], [CCW03], [BCC03a]. Thus, a deterministic model represents a good approximation of the problem. Throughout this analytical study, two constant bit rate UDP streams are assumed. The first stream is a downlink UDP stream originating from the CN and directed to the MN. The second stream is an uplink UDP stream sent out from the MN and destined to the CN. λ is the packet arrival rate of both UDP streams.

5.2. Modeling of Network Topologies

The network topology can be either hierarchical or mesh-based. Mobility protocols that require a hierarchical network topology deploy one or more nodes at certain levels of the hierarchy to process the handoff locally. Our model assumes that each domain has one gateway at the first level of the hierarchy, which will be referred to as GW in the rest of this chapter. At the second level of the hierarchy, there are many routers with mobility support, referred to as Mobility Routers (MRs) throughout this chapter. The third level of the hierarchy is composed of the MAs themselves. It is supposed that each (ν) MAs are controlled by one

of (\mathcal{G}) MRs present in the domain. Figure 5.1 illustrates a hierarchical network topology for $Z = 9$, $\mathcal{G} = 3$ and $\nu = 3$.

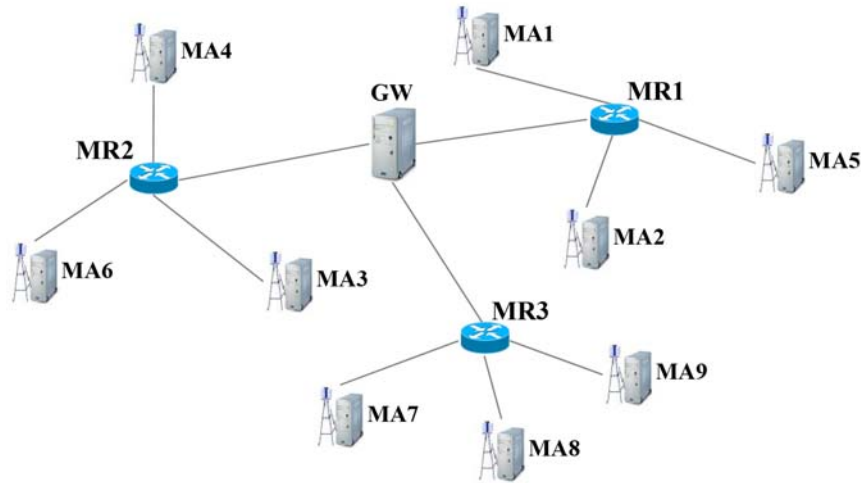


Fig 5.1: An example hierarchical network topology ($Z = 9$, $\mathcal{G} = 3$ and $\nu = 3$)

For the mesh topology, it is assumed that each domain is also controlled by a single GW. Important here is the crossover router, which is defined as the node participating in the path between the GW and the old MA as well as in the path between the GW and the new MA. It is assumed that each MA is connected with all other MAs in the domain through a crossover node different from the GW, in other words through a MR.

For most micro mobility management solutions, mobility bindings are updated either at the GW, MRs, old MA, new MA or a special node in the network. The distances between the current MA and a MR, the GW, the old MA or a certain node in the domain deploying a symmetrical hierarchical topology can be calculated simply by counting the hops on the shortest path. However, for networks with asymmetrical hierarchical or a mesh topologies, these distances vary from movement to movement. Therefore, average values of these distances are assumed in these networks. Notice that the average distances should be calculated taking the used mobility pattern into account. For more details, see appendix [E](#) which discusses how asymmetrical network topologies can be considered.

5.3. Modeling of Movements Patterns

The action the MN executes when moving from one MA to another is defined as a “movement”. In order to model movements between MAs, a probabilistic model is proposed. Besides the already mentioned number of nodes inside the domain (Z), the probability q_i that a MN switches on in the range controlled by MA_i and the transition probability $P_{i,j}$ that the MN moves from MA_i to MA_j are required. These probabilities are described by the matrix P .

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,z} \\ \dots & \dots & \dots & \dots \\ P_{z,1} & P_{z,2} & \dots & P_{z,z} \end{bmatrix} = \begin{bmatrix} 0 & P_{1,2} & \dots & P_{1,z} \\ \dots & \dots & \dots & \dots \\ P_{z,1} & P_{z,2} & \dots & 0 \end{bmatrix}$$

Similar to [\[JAK02\]](#) and [\[DMB05\]](#), the MN does not move from a certain MA to one of the other ($Z - 1$) MAs with an equal probability. In reality, MNs restrict their movement to one of

the MAs located in the geographical neighborhood regardless of the network topology in use. Moreover, MNs movement's patterns are far from being random. They show a high degree of temporal and spatial regularity, thus following simple reproducible patterns [GHB08]. Figure 5.2 shows an example neighbor graph with the movement probabilities for a domain containing 9 MAs. The probabilities not listed there are equal to zero.

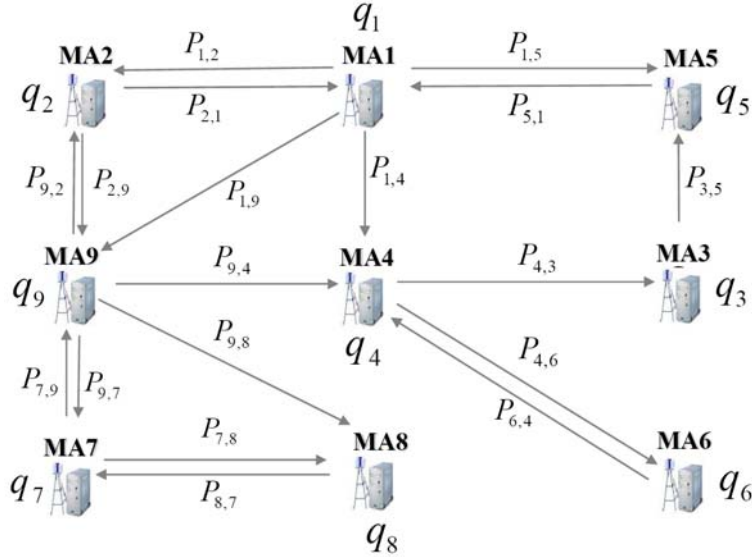


Fig 5.2: An example neighbor graph with the movement probabilities for a domain containing 9 MAs

In order to derive the probability P_i that the MN is attached to MA_i , the start vector $Q^0 = [q_1 \ q_2 \ \dots \ q_z]$ is defined. Using the recursive equation $Q^{n+1} = Q^n * P$, Q is defined as in equation (1).

$$Q = \lim_{n \rightarrow \infty} Q^n = [P_1 \ P_2 \ \dots \ P_z] \quad (1)$$

Q^n contains the probabilities that a MN is attached to each MA after n movements. By multiplying this infinite times with the transition probabilities existing in P , Q is derived. It contains the probabilities that a MN is attached to each MA in the steady state of the system and is defined if all absolute eigenvalues of P are smaller than one.

Let us first take the symmetrical hierarchical topology into account and assume that the MAs shown in Figure 5.2 are structured as in Figure 5.1. By a movement, the MN attaches itself to a new MA, which may be controlled by

- the same MR that controls the old MA,
- a different MR but the same GW or
- a different MR from another domain (controlled by a different GW). This case will not be handled since the analysis will be done within the domain only.

Supposing $P(MR_n)$ is the probability that the MN moves between the MAs controlled by MR_n , we can calculate $P(MR_n)$ as follows.

$$P(MR_n) = \sum_i \sum_j P_i * P_{i,j} \quad \text{where } i, j \in I(MR_n) \text{ and } i \neq j \quad (2)$$

$I(MR_n)$ presents the set of MAs controlled by MR_n . By observing all movements inside the domain, R can be derived from equation (3), where R is the probability that the crossover router will be one of the MRs while moving inside the domain.

$$R = \sum_{n=1}^g P(MR_n) \quad (3)$$

In a similar way, the probability that the GW will be the crossover router during movements inside the domain can be derived. This probability is given in equation (4).

$$G = \sum_i \sum_j P_i * P_{i,j} \quad \text{where } i, j \in Y(MR) \text{ and } i \neq j \quad (4)$$

where $Y(MR)$ is the set of MAs, from which the MN can move to another MA controlled by another MR inside the domain.

Let us now consider the symmetrical mesh topology. As explained in the previous section, it is supposed that all MAs are connected to each other through MRs. Therefore, the term G equals 0, while the term R is derived from equation (5) and is equal to 1.

$$R = \sum_{i=1}^Z \sum_{j=1}^Z P_i * P_{i,j} \quad \text{where } i \neq j \quad (5)$$

Other mobility patterns can be used as well. They should, however, take care of delivering R and G . Appendix [E](#) provides more details explaining how mobility models should be considered for asymmetrical network topologies.

5.4. Performance Analysis

In this section, the functions required to calculate the average handoff latency and expected average number of dropped packets will be derived, see [\[DLM07\]](#). As known, most macro mobility management protocols, e.g. MIPv4, update their mobility bindings at the HA. Others update their mobility bindings at the old MA, as is the case for MIFA in reactive mode regarding handoffs on downlink. The mobility binding may be updated even at the new MA, as is the case for MIFA in reactive mode regarding handoffs on uplink. Most micro mobility management protocols, however, update their mobility bindings at MRs and/or the GW while moving inside the domain. Examples of such protocols are MIPRR and HMIPv6. Some micro mobility management protocols select one or more specific nodes, e.g. MAs, and always update their mobility bindings at these specific nodes, as is the case for dynamic regional registration approaches and AFA. Other micro mobility management protocols update their mobility bindings at the old MA, as is the case for HAWAII. Throughout this chapter, the node at which the mobility binding should be updated will be referred to as the Binding Update node ($BUnode$). Any node that will be selected to be a $BUnode$ and is different from the HA, GW, MR, old MA and new MA will be referred to as an Anchor Point (ANP). Any

node that understands mobility and is different from the HA, GW, MR, old MA, new MA and ANP is referred to as an Intermediate Node (*InNode*).

5.4.1. Break-Before-Make Mobility Management Protocols

5.4.1.1. Handoff Latency

In order to ensure the applicability of the model to the most known mobility management protocols, the handoff latency is defined as in (6) in case the **update message** has not been dropped on the way to the *BUnode*¹.

$$T_{BUnode} = \Delta t + k_1 * t_{MN,MA} + k_2 * t_{currentMA, BUnode} + k'_1 * a_{MA} + k'_2 * a_{BUnode} + k'_3 * ni * a_{InNode} + \gamma \quad (6)$$

In case the **update message** has been dropped on the way to the *BUnode*, the handoff latency is derived from (7).

$$T_{BUnode} = \Delta t + k_1 * t_{MN,MA} + k_2 * t_{currentMA, BUnode} + k'_1 * a_{MA} + k'_2 * a_{BUnode} + k'_3 * ni * a_{InNode} + \gamma + \sum_{i=2}^S 2^{(S-i)} * T_{timer} \quad (7)$$

where k_1 represents the number of messages exchanged on the wireless link within a handoff. k_2 represents the number of messages exchanged between the current MA and the *BUnode*. k'_1, k'_2 and k'_3 stand for the number of times the **update message** has been processed in the current MA, *BUnode* and *InNode*, respectively. Notice that for break-before-make protocols, MNs loose their link first and, after that, hand off to the new MA. The new MA is, therefore, considered to be the current MA in the generic model. ni represents the number of intermediate nodes. γ denotes the extra latency required for a certain mobility protocol, such as authentication time, address auto-configuration, etc. γ varies from protocol to protocol and may not be a constant value, it may even be calculated by a protocol-specific equation. Δt refers to the movement detection time. The term $t_{currentMA, BUnode}$ is equal to $\tau_2 * D_{currentMA, BUnode}$ if delay on the link between each subsequent two hops inside and outside the domain is the same, whereas the term $t_{MN,MA}$ is equal to τ_1 .

Let us now take the mobility model into account. As mentioned above, the *BUnode* varies from protocol to protocol. It can be the HA, GW, MR, old MA, new MA or an ANP. Therefore, a vector $B = [J_{MR} \ J_{GW} \ J_{HA} \ J_{MA-MR} \ J_{MA-GW} \ J_{ANP}]$ is defined. J_x expresses the probability that the MN updates its binding at node x. J_{MA-MR} expresses the probability that the MN updates its binding at the old or new MA when the crossover router is one of the MRs. J_{MA-GW} stands for the probability that the MN updates its binding at the old or new MA when

¹ For terminal-based solutions, the considered dropping is mainly the dropping of the **update message** sent from the MN on the wireless link. However, for network-based mobility management solutions, no messages are sent from the MN. Therefore, the dropping of the **update message** in the backbone is considered.

the crossover router is the GW. It should be distinguished between these two terms in order to model the hierarchical topology. Notice that $J_{MA-GW} = 0$ deploying a symmetrical mesh topology. This is because of the assumptions assumed in section 5.2, which say that there is always a path between the old and the new MA via a MR.

The vector B is written according to the specification of the studied mobility protocol. It should be mentioned, however, that the initial registration with the HA should not be taken into account when writing this vector. Supposing a hierarchical topology, this vector will be $[0 \ 0 \ R+G \ 0 \ 0 \ 0]$ for macro mobility management protocols, where the MN updates its binding only at the HA. For micro mobility protocols, this vector will be $[R \ G \ 0 \ 0 \ 0 \ 0]$ if the mobility inside the domain is controlled by the MRs and the GW. B will be $[0 \ R+G \ 0 \ 0 \ 0 \ 0]$ if the MN updates its binding only at the GW. If the binding should be updated only at the old MA, the vector will be $[0 \ 0 \ 0 \ R \ G \ 0]$. Supposing a mesh topology, the vector B will be $[0 \ 0 \ R \ 0 \ 0 \ 0]$ for macro mobility protocols, where the HA is always the $BUnode$. For micro mobility protocols, this vector will be $[0 \ R \ 0 \ 0 \ 0 \ 0]$ if the mobility inside the domain is controlled by the GW. If the binding should be updated only at the old MA, this vector will be $[0 \ 0 \ 0 \ R \ 0 \ 0]$.

Let us now assume another vector $T = [T_{MR} \ T_{GW} \ T_{HA} \ T_{MA-MR} \ T_{MA-GW} \ T_{ANP}]$. This vector is called a handoff vector and defines the handoff latency experienced when the MN registers with the MR, GW, HA, old or new MA if the crossover router is one of the MRs, old or new MA if the crossover router is the GW and ANP , respectively. It should be distinguished between T_{MA-MR} and T_{MA-GW} for the same reason presented while discussing the B vector. The handoff vector is defined regarding the protocol specification. $T = [0 \ T_{GW} \ 0 \ 0 \ 0 \ 0]$ means that the mobility protocol registers only with the GW while moving inside the access network. Again, the initial registration with the HA should not be taken into account.

Depending on the discussion above, the average handoff latency resulting from employing a certain mobility management protocol can be written as in (8).

$$T_{avr} = B * T^{-1} \quad (8)$$

5.4.1.2. Expected Number of Dropped Packets

Let us first discuss the downlink UDP stream. Similar to the discussion in section 5.4.1.1, t_{BUnode} can be calculated from (9) if the **update message** has been sent successfully to its destination without suffering from any dropping.

$$t_{BUnode} = \Delta t + k_3 * \tau_1 + k_4 * \tau_2 * D_{currentMA, BUnode} + k_4' * a_{MA} + k_5' * a_{BUnode} + k_6' * ni * a_{InNode} + \gamma' \quad (9)$$

In case the **update message** has been dropped and should be, therefore, retransmitted, t_{BUnode} is written as in (10).

$$\begin{aligned}
t_{BUnode} = & \Delta t + k_3 * \tau_1 + k_4 * \tau_2 * D_{currentMA, BUnode} + k'_4 * a_{MA} + k'_5 * a_{BUnode} \\
& + k'_6 * n_i * a_{InNode} + \gamma' + \sum_{i=2}^S 2^{(S-2)} * T_{timer}
\end{aligned} \tag{10}$$

k_3 represents the number of messages required to be exchanged on the wireless link to notify the *BUnode* of the handoff. It should be noticed, however, that the *BUnode* is notified of the handoff upon receiving the **update message**. This does not mean that the handoff has been finished. Handoff completion occurs when the MN has received a control message indicating that the handoff has been finished. k_4 represents the number of messages required to be exchanged between the current MA and the *BUnode* to notify it of the new mobility binding. k'_4, k'_5 and k'_6 stand for how many times the **update message** will be processed at the current MA, *BUnode* and *InNodes* during the exchange of k_3 and k_4 messages. γ' expresses any extra latency required for a certain mobility management protocol. Similar to γ , γ' varies from protocol to protocol and may not be a constant value, In fact, it may even be calculated by protocol-specific equations.

If the *BUnode* is not the old MA, each packet arriving at the *BUnode* belongs to one of the following classes.

- Class 0:** packets arriving at the *BUnode* before t_{BUnode} , these packets are forwarded to the old MA.
- Class 1:** Packets arriving at the *BUnode* after t_{BUnode} , these packets are forwarded to the new MA, which forwards them directly to the MN.
- Class 2:** packets arriving at the old MA before t_0 , these packets are forwarded directly to the MN.
- Class 3:** packets arriving at the old MA at or after t_0 , these packets are lost.

If the *BUnode* is the old MA, data packets belong to one of the following classes.

- Class 4:** packets arriving at the old MA before t_0 , these packets are forwarded to the MN.
- Class 5:** packets arriving at the old MA in the duration between t_0 and t_{BUnode} , these packets are lost.
- Class 6:** packets arriving at the old MA after t_{BUnode} , these packets are forwarded to the new MA, which forwards them to the MN.

Assuming T_{link} is the delay on the link between the old MA and the *BUnode*, this term can be written as follows.

$$T_{link} = \tau_2 * D_{oldMA, BUnode} \tag{11}$$

All packets arriving at the $BUnode$ in the duration between $(t_0 - T_{link})$ and $(t_0 + t_{BUnode})$ are lost. These packets can be calculated according to the equation below.

$$LP_{BUnode} = ((t_0 + t_{BUnode}) - (t_0 - T_{link})) * \lambda \quad (12)$$

Let us now assume a packet dropping vector $LP = [LP_{MR} \quad LP_{GW} \quad LP_{HA} \quad LP_{MA-MR} \quad LP_{MA-GW} \quad LP_{ANP}]$. This vector defines the expected number of dropped packets when the $BUnode$ is the MR, GW, HA, old or new MA when the crossover router is one of the MRs, old or new MA when the crossover router is the GW and ANP, respectively. Once again and for the same reason discussed in section 5.4.1.1, it should be distinguished between LP_{MA-MR} , LP_{MA-GW} and LP_{ANP} . The packet dropping vector is defined according to the protocol specification. $LP = [0 \quad LP_{GW} \quad 0 \quad 0 \quad 0 \quad 0]$ means that the $BUnode$ is the GW and LP_{GW} is the expected number of dropped packets when the MN performs a handoff. The expected average number of dropped packets per handoff for a certain mobility management protocol can be then written as in (13).

$$LP_{avr} = B * LP^{-1} \quad (13)$$

Considering the uplink UDP stream, each packet originating from the MN belongs to one of the following classes.

- Class 7:** packets originating from the MN before t_0 , these packets arrive at the previous MA and are forwarded to the CN.
- Class 8:** packets originating from the MN in the duration between t_0 and $t_0 + T_{BUnode}$, these packets are lost.
- Class 9:** packets originating from the MN after $t_0 + T_{BUnode}$, these packets are forwarded to the new MA, which forwards them to the CN.

Thus, the expected number of dropped packets per handoff on uplink can be derived from equation (14), while the average expected number of dropped packets per handoff on uplink is calculated from (13).

$$LP_{BUnode} = T_{BUnode} * \lambda \quad (14)$$

5.4.2. Make-Before-Break Mobility Management Protocols

In order to analyze make-before-break mobility management protocols, we should distinguish between the following cases.

- Case 1:** the MN has not fired a L2-trigger and, thus, it does not know the new MA. As a result, it performs a layer 2 handoff followed by a layer 3 handoff.
- Case 2:** a L2-trigger was raised at the MN. However, the MN could not start the layer 3 handoff in advance. This will be the case if the L2-trigger has been delayed and the MN has lost the link with the old MA before starting the layer 3 handoff. This will be the case too if the MN has started the layer 3 handoff in

advance. However, the *update message* has been dropped on the old wireless link and the MN could not detect the dropping. Figure 5.3 shows this case for FMIPv6.

In this case, the MN performs a layer 2 handoff followed by a layer 3 handoff. The MN, however, does wait for an advertisement from the new MA or exchange solicitation and advertisement messages with it. In other words, Δt is equal to 0.

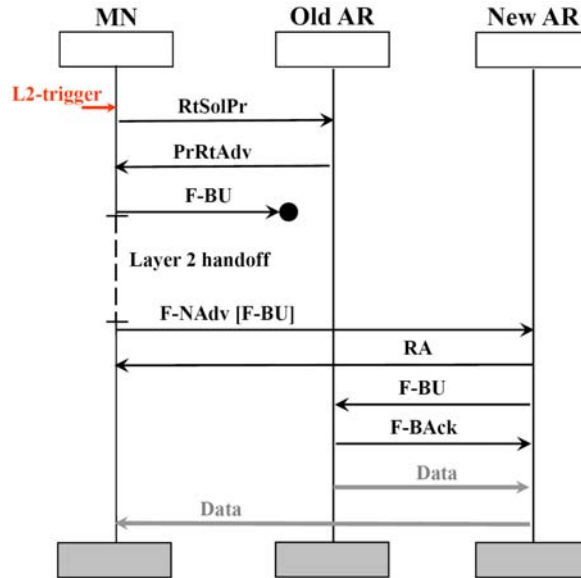


Fig 5.3: Dropping of the *update message* sent on the old wireless link employing FMIPv6 (the MN could not recover the dropping)

Case 3: the MN fires the L2-trigger and still has sufficient time before it loses the radio link with the old MA. In this case, the MN begins the layer 3 handoff in advance. The layer 3 handoff latency is defined as the difference between layer 3 and layer 2 handoff latencies if the layer 3 handoff has not finished before completing the layer 2 handoff. Otherwise, the layer 3 handoff latency is eliminated.

5.4.2.1. Handoff Latency

This section derives the equations required to calculate the handoff latency in the three cases listed above. Considering the first case, there is no impact of layer 2 triggers on the performance. Therefore, the handoff latency is calculated in the same way as in section 5.4.1.1. For the second case, the appearance of a L2-trigger saves only the movement detection time. Thus, the handoff latency is calculated same as in section 5.4.1.1 too. However, the value of Δt is zero in all equations. Layer 2 triggers affect strongly the performance of mobility management protocols in the third case. In order to analyze this case, we distinguish between three categories of mobility management protocols.

Category1: protocols of this category update mobility bindings at the *BUnode* via the old MA. An example is the pre-registration method for MIPv4.

Category2: protocols of this category register only with the new MA via the old one. A complete registration with another *BUnode* may follow the layer 2 handoff. Examples are the post-registration method and FMIPv6.

Category3: protocols of this category switch to the new radio for a short time upon the appearance of a L2-trigger and send an **update message** to the *BUnode* via the new MA. An example is the semi-soft handoff by CIP.

Taking the above discussion into account, equations (6) and (7) deliver the layer 3 handoff duration (T'_{BUnode}). Clearly, the value of Δt is zero in these equations. The layer 3 handoff latency is defined, after that, as follows.

$$T_{BUnode} = \begin{cases} 0 + \gamma_x & \text{when } (t_0 - t_g) + T_{L2HO} \geq T'_{BUnode} \\ (T'_{BUnode} - (T_{L2HO} + (t_0 - t_g))) + \gamma_x & \text{when } (t_0 - t_g) + T_{L2HO} < T'_{BUnode} \end{cases} \quad (15)$$

γ_x results from any extra latency required after finishing the layer 2 handoff, e.g. sending a reply to the MN after the layer 2 handoff. The average handoff latency is calculated in the same way as in section 5.4.1.1.

5.4.2.2. Expected Number of Dropped Packets

The expected number of dropped packets in the first case is calculated in the same way as in section 5.4.1.2. Considering the second case, the expected number of dropped packets is calculated same as in section 5.4.1.2 too. However, Δt is zero in all equations. The expected number of dropped packets depends strongly on layer 2 triggers in the third case. To analyze this case, the Routing Node (*RNode*) is defined as the node that forwards packets to the new MA during the handoff. Assuming t_{RNode} is the time at which *RNode* begins forwarding packets to the new MA, t_{RNode} is written as in (16) in case of no dropping of the **update message**.

$$t_{RNode} = k_3 * t_{MN,MA} + k_4 * t_{currentMA, BUnode} + k_5 * t_{RNode, BUnode} + k_6 * t_{currentMA, RNode} + k_4' * a_{MA} + k_5' * a_{BUnode} + k_6' * ni * a_{InNode} + k_7' * a_{RNode} + \gamma_1' \quad (16)$$

However, in the case of dropping of the **update message**, t_{RNode} is calculated from (17). Notice that the current MA for the protocols belonging to the first and second category, presented in section 5.4.2.1, is the old MA, whereas the new MA is the current MA for the protocols belonging to the third category.

$$t_{RNode} = k_3 * t_{MN,MA} + k_4 * t_{currentMA, BUnode} + k_5 * t_{RNode, BUnode} + k_6 * t_{currentMA, RNode} + k_4' * a_{MA} + k_5' * a_{BUnode} + k_6' * ni * a_{InNode} + k_7' * a_{RNode} + \gamma_1' + \sum_{i=2}^S 2^{(S-2)} * T_{timer} \quad (17)$$

k_3 represents the number of messages that should be exchanged on the wireless link to inform the *RNode*. k_4 stands for the number of messages exchanged between the current MA and the *BUnode*. k_5 denotes the number of messages exchanged between the *RNode* and the *BUnode*.

k_6 stands for the number of messages exchanged between the current MA and the *RNode*. k'_4, k'_5, k'_6 and k'_7 stand for the number of times the *update message* will be processed at the current MA, *BUnode*, *InNode* and *RNode*, respectively. γ'_1 expresses any extra latency required for a certain mobility protocol. Similar to γ and γ' , γ'_1 is protocol-specific.

Considering the downlink UDP stream, if the *RNode* is not the old MA, each packet belongs to one of the following classes.

- Class 0:** packets arriving at the *RNode* before t_{RNode} , these packets are sent to the old MA.
- Class 1:** packets arriving at the *RNode* after t_{RNode} , these packets are forwarded to the new MA, which forwards them to the MN. Notice that the *RNode* may forward the packets not only to the new MA. However, the forwarding to the new MA is only of interest when calculating the lost packets.
- Class 2:** packets arriving at the old MA before t_0 , these packets are forwarded to the MN.
- Class 3:** packets arriving at the old MA at or after t_0 , these packets are lost.

If the *RNode* is the old MA, packets belong to one of the following classes.

- Class 4:** packets arriving at the old MA before t_0 , these packets are forwarded to the MN.
- Class 5:** packets arriving at the old MA in the duration between t_0 and t_{RNode} , these packets are lost.
- Class 6:** packets arriving at the old MA after t_{RNode} , these packets are forwarded to the new MA, which forwards them to the MN.

The expected number of dropped packets per handoff on downlink is calculated then according to equation (18). Notice that packets which arrive at the new MA are buffered until the MN completes the layer 2 handoff.

$$LP_{RNode} = \begin{cases} 0 & \text{when } t_g + t_{RNode} \leq t_0 \\ (t_{RNode} - (t_0 - t_g)) * \lambda & \text{when } t_g + t_{RNode} > t_0 \end{cases} \quad (18)$$

The expected average number of dropped packets per handoff on downlink can be derived from equation (13). Considering the expected number of dropped packets per handoff on uplink, it is derived in the same way as in section [5.4.1.2](#).

5.4.2.3. Movement Inside the Overlapping Area

As mentioned in section [5.1](#), cellular cells are assumed to overlap. The time the MN spends inside the overlapping area has a significant impact on the performance of make-before-break mobility management protocols. Let us assume that a L2-LD trigger appears upon the MN

moves outside the overlapping area. The time $(t_{LD} - t_g)$, which equals to $(t_0 - t_g)$ too, should be long enough to obtain the advertisement of the new MA and to send the **update message**. The length of the MN path inside the overlapping area (ℓ) can be written as in (19), where \mathcal{G}_{MN} is the speed of the MN, see figure 5.4.

$$\ell = \mathcal{G}_{MN} * (t_0 - t_{in}) \quad (19)$$

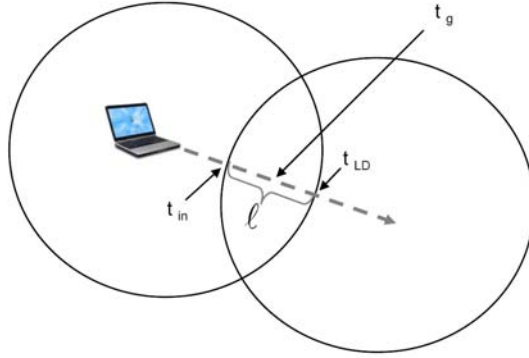


Fig 5.4: An example movement path of a MN inside an overlapping area between two cellular cells

where $(t_0 - t_{in})$ stands for the time duration the MN spends inside the overlapping area. $(t_0 - t_g)$ denotes the time duration, within the MN can start the layer 3 handoff in advance. $(t_g - t_{in})$ stands for the time duration required to fire the L2-trigger. During this time, the MN may scan the medium aiming at receiving beacons from new BSs. The MN can be tracked by the network during this time as well. The value of $(t_0 - t_g)$ required to enable an in advance starting of the layer 3 handoff can be obtained from equation (20) if no dropping of the **update message** on the wireless link has occurred.

$$(t_0 - t_g) \geq \Delta_{Sol / Adv} + k_8 * t_{MN, currentMA} + \gamma_2' \quad (20)$$

In the case of dropping of the **update message**, the time required to detect the dropping (T_{timer}) should be considered. Thus, the value of $(t_0 - t_g)$ required to trigger the layer 3 handoff in advance is written in this case as in (21).

$$(t_0 - t_g) \geq \Delta_{Sol / Adv} + k_8 * t_{MN, currentMA} + \gamma_2' + \sum_{i=2}^S 2^{(S-2)} * T_{timer} \quad (21)$$

where $\Delta_{Sol / Adv}$ represents the time duration required to obtain the advertisement of the new MA (e.g. for FMIPv6, $\Delta_{Sol / Adv}$ is the time required to exchange a **RtSolPr** and a **PrRtAdv** message with the current MA). k_8 expresses the number of messages exchanged on the wireless link between the MN and the current MA to trigger the layer 3 handoff in advance. γ_2' stands for any extra latency required for a certain mobility management protocol.

5.5. Cost Estimation

This section derives the functions required to calculate the location update cost, packet delivery cost and total cost per time unit resulting from mobility management protocols, see [DML08].

5.5.1. Location Update Cost

In order to ensure the applicability of the model to the most known mobility management protocols, the term luc_{BUnode} is defined as follows.

$$luc_{BUnode} = k_1 * Tc_{MN, MA}^S + k_2 * Tc_{currentMA, BUnode}^S + k'_1 * a'_{MA} + k'_2 * a'_{BUnode} + ni * k'_3 * a'_{InNode} + \gamma'' \quad (22)$$

where k_1 , k_2 , k'_1 , k'_2 , k'_3 and ni have the same meaning as in equation (6). γ'' stands for any extra cost resulting during the movement, e.g. movement tracking, notifying the neighbor MAs, etc. γ'' is protocol-specific and may not be a constant value.

The transmission cost of a location update message $Tc_{x,y}^S$ on a wired link is proportional to the distance $D_{x,y}$ with a proportional constant δ_S . Therefore, this transmission cost can be written as follows.

$$Tc_{x,y}^S = \delta_S * D_{x,y} \quad (23)$$

The transmission cost on the wireless link is ρ times more than on a wired link. Thus, this cost can be calculated as in (24).

$$Tc_{MN, MA}^S = \rho * \delta_S \quad (24)$$

Filling in the transmission cost of location update messages specified above, equation (25) is obtained.

$$luc_{BUnode} = \delta_S * (\rho * k_1 + k_2 * D_{currentMA, BUnode}) + k'_1 * a'_{MA} + k'_2 * a'_{BUnode} + ni * k'_3 * a'_{InNode} + \gamma'' \quad (25)$$

Let us now define a location update cost vector $LUC = [luc_{MR} \quad luc_{GW} \quad luc_{HA} \quad luc_{MA-MR} \quad luc_{MA-GW} \quad luc_{ANP}]$. This vector defines the location update cost experienced when updating the binding at the MR, GW, HA, old or new MA if the crossover router is one of the MRs, old or new MA if the crossover router is the GW and the ANP, respectively. This vector is defined regarding the specification of the protocol being analyzed. $LUC = [0 \quad luc_{GW} \quad 0 \quad 0 \quad 0 \quad 0]$ means that the mobility is controlled only by the GW inside the domain. Again, the initial registration with the HA should not be considered when writing this vector.

Depending on the discussion above, the average location update cost per time unit for a certain mobility management protocol can be written as in (26).

$$luc_{TimeUnit} = \frac{B * LUC^{-1}}{T_r} \quad (26)$$

5.5.2. Packet Delivery Cost

The packet delivery cost comprises the transmission cost of data packets and the processing cost required in the participating entities to route, tunnel and de-tunnel the packets. The developed model considers the packet delivery cost a packet incurs on its path from the CN to the MN. This cost per time unit $pdc_{TimeUnit}$ can be derived from the equation below.

$$pdc_{TimeUnit} = pdc_{CN, MN} + Fc_{handoff} \quad (27)$$

where $Fc_{handoff}$ is the packet delivery cost resulting from forwarding data packets during the handoff to ensure a seamless movement, e.g. data packets are forwarded from the old MA to the new one during the handover when using the route optimization extension for MIPv4. Of course, $Fc_{handoff}$ equals 0 if the studied mobility management protocol does not forward any data packets during the handoff.

Let us now define a processing cost vector $d = [d_{MR} \ d_{GW} \ d_{HA} \ d_{oMA} \ d_{nMA} \ d_{InNode}]$. This vector expresses the processing cost of a packet delivery in the MR, GW, HA, old MA, new MA and all *InNodes*, respectively. This vector is defined regarding the protocol specification and the routing path used to forward data packets.

Depending on this discussion, the terms $pdc_{CN, MN}$ and $Fc_{handoff}$ per time unit can be written as in equations (28) and (29), respectively.

$$pdc_{CN, MN} = \sum_{i=0}^5 d[i] + Tc_{CN, MA}^D + Tc_{MA, MN}^D \quad (28)$$

$$Fc_{handoff} = \frac{\Delta * (\sum_{i=0}^5 d[i] + k_9 * Tc_{RNode, newMA}^D)}{T_r} \quad (29)$$

where k_9 stands for the number of MAs to which data packets are forwarded during the handoff. This is necessary to model the protocols that multicast data packets to many candidate MAs during the handoff. Δ is the average time duration, within data packets are forwarded to k_9 MAs. More specifically, Δ equals to the time duration required to inform the node that stops the forwarding in addition to the time duration required to forward the packets in-flight. In the rest of this chapter, the node that stops the packets forwarding to k_9 MAs is referred to as a Control Node (*ContNode*). For example, this node will be the HA employing

FMIPv6. When the HA is notified of the new binding, no packets will be sent on the old path. The *ContNode* may be the *BUNode* itself or another node.

The vector d in $pd_{CN, MA}$ and $F_{handoff}$ may not be the same. Additionally, the elements of this vector may not be constants. They may be calculated by protocol-specific equations.

Similar to the assumptions of the location update cost, the transmission cost $Tc_{x,y}^D$ on a wired link is proportional to the distance $D_{x,y}$ with a proportional constant δ_D . Thus, the transmission cost per time unit can be written as follows.

$$Tc_{x,y}^D = \lambda * \delta_D * D_{x,y} \quad (30)$$

The transmission cost on the wireless link is assumed to be ρ times more than on the wired link. Filling in the transmission cost specified above, equations (28) and (29) can be written as follows.

$$pd_{CN, MN} = \sum_{i=0}^5 d[i] + \lambda * \delta_D * (D_{CN, MA} + \rho) \quad (31)$$

$$F_{handoff} = \frac{\Delta * (\sum_{i=0}^5 d[i] + k_9 * \lambda * \delta_D * D_{RNode, newMA})}{T_r} \quad (32)$$

where $D_{CN, MA}$ depends on the routing path between the CN and the MA, which varies if the route optimization is used or not.

5.5.3. Total Cost

Based on the analysis presented in sections [5.5.1](#) and [5.5.2](#), the total cost function can be obtained using equation (33), where φ is a weighting factor representing the importance of the location update cost against the packet delivery cost.

$$C_{Total} = \varphi * luc_{TimeUnit} + (1 - \varphi) * pd_{TimeUnit} \quad (33)$$

5.6. Application of the Generic Mathematical Model to Mobility Management Protocols

In this section the developed model will be used to evaluate MIFA compared to a set of well-known mobility management protocols. In order to evaluate the performance, we make the following assumptions. For break-before-make mobility management protocols, the MN always sends a solicitation following the layer 2 handoff to obtain an advertisement from the new MA. This means that the movement detection time (Δt) equals 0. If this is not the case (e.g. in the case of protocols that do not require sending a solicitation or when the advertisement is obtained in another way), it will be mentioned and the value of Δt will be

re-defined when discussing these protocols. T_{timer} is set to $2 * RTT$, where RTT is the round trip time between the MN and the $BUnode$. For the protocols that register with more than one $BUnode$ while moving inside the domain, RTT is selected as the round trip time between the MN and the farthest $BUnode$. For example, MIPRR registers with the MRs and the GW. For all movements inside the MIPRR domain, RTT is set to the round trip time between the MN and the GW. Considering the make-before-break mobility management protocols, the value of T_{timer} is assumed to be $2 * RTT$ as well. However, RTT is the round trip time between the MN and the $RNode$. Finally, we assume that t_0 is equal to 0 and λ is equal to 50 packets per second for the uplink as well as the downlink UDP stream.

In order to estimate the location and packet delivery cost, we assume that the cost for the transmission of signaling messages and data packets are available. The cost for a packet processing in MRs, the GW, the HA, MAs and $InNodes$ can be determined as well. As discussed in [JAK02], cost parameters can be expressed as the delay required for messages processing or transmitting. For example, a'_x may represent the delay required to process a location update message in node x. d_x can be seen as the delay required to queue and serve a data packet in node x. $Tc_{x,y}^S$ and $Tc_{x,y}^D$ may stand for the delay experienced when sending a signaling message and a data packet on a particular path, respectively. Other measurements for cost parameters are also possible. For example, other relative costs can be assigned based on some criteria, e.g. available bandwidth, expenses required to operate a particular node, etc.

Regarding the used network topology, this analysis assumes that the best adequate topology for each protocol will be used. If a mobility management protocol requires a hierarchical structure, the hierarchical topology will be applied. Otherwise, a mesh topology is used.

5.6.1. Applied Network Topology

Let us now assume that the hierarchical topology is structured as in figure 5.1 and the domain contains 9 MAs. The parameters of the network topology are assumed to be as shown in tables 5.3 and 5.4 for a hierarchical and a mesh topology, respectively. The parameters displayed in table 5.5 are the same for the both topologies.

$D_{MA,MR}$	$D_{MR,GW}$
2 hops	2 hops

Tab 5.3: Parameters of the hierarchical network topology

$\bar{D}_{MA,GW}$	$\bar{D}_{newMA,oldMA}$	$\bar{D}_{MA,MR}$
3 hops	2 hops	1.5 hops

Tab 5.4: Parameters of the mesh network topology

δ_D	δ_s	ρ	τ_2	$D_{GW,HA}$	$D_{GW,CN}$	$D_{CN,HA}$
0.05	0.5	10	5 msec	5 hops	4 hops	3 hops

Tab 5.5: Parameters for both hierarchical and mesh topology

The value of τ_1 depends on the radio access technology. For example, WLAN has a fast radio access technology. Therefore, τ_1 is assumed to be 2 msec. In contrast, GSM and UMTS have slower radio access technologies, 80 msec in UMTS and 150 msec in GSM according to [DMG08] and [Get08]. In this analysis, a fast radio access technology is assumed. The layer 2 handoff latency is assumed to be 50 msec.

5.6.2. Applied Movement Model

For simplicity, we assume that the MN turns on in the range of each MA in the domain with an equal probability. This means $q_1 = q_2 = \dots = q_9 = 0,11$. In addition, it is assumed that the MN can move from a given MA to $N-1$ others with an equal probability ($\frac{1}{N-1}$), as shown in Figure 5.5.

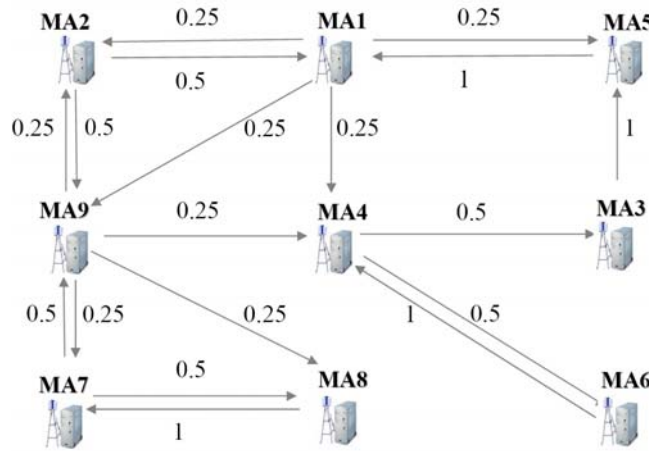


Fig 5.5: The neighbor graph used in the analysis along with the used movement probabilities

Depending on the neighbor graph shown in the figure above, the matrix P can be written as follows.

$$P = \begin{bmatrix} 0 & 0.25 & 0 & 0.25 & 0.25 & 0 & 0 & 0 & 0.25 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0.25 & 0 & 0.25 & 0 & 0 & 0.25 & 0.25 & 0 \end{bmatrix}$$

Depending on this matrix, the matrix Q can be written as follows.

$$Q = [0.1429 \quad 0.0714 \quad 0.0714 \quad 0.1429 \quad 0.1071 \quad 0.0714 \quad 0.1429 \quad 0.1071 \quad 0.1429]$$

Depending on these matrixes, the values of R and G for the hierarchical and the mesh topology can be calculated using the equations (3), (4) and (5). These values are listed in table 5.6.

Hierarchical topology		Mesh topology	
R	G	R	G
0.75	0.25	1	0

Tab 5.6: Values of R and G

5.6.3. Application to Break-Before-Make Mobility Management Protocols

5.6.3.1. MIPv4

MIPv4 always contacts the HA, which represents the $BUnode$. There are no restrictions on the network shape and a mesh topology is normally used. The mobility is processed only in MAs and the HA. There are no $InNodes$. B is $[0 \ 0 \ R \ 0 \ 0 \ 0]$. The handoff vector is $[0 \ 0 \ T_{HA} \ 0 \ 0 \ 0]$. T_{HA} is calculated from equations (6) and (7). Required parameters are listed in table 5.7, where γ is the delay resulting from exchanging a solicitation and an advertisement between the MN and the new MA.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ
2	2	2	1	0	0.1 msec	0.5 msec	0	0	4 msec

Tab 5.7: Parameters required to calculate T_{HA} for MIPv4

The packet dropping vector is $[0 \ 0 \ LP_{HA} \ 0 \ 0 \ 0]$. LP_{HA} is calculated from equations (12) and (14). The other parameters required to calculate t_{BUnode} are listed in table 5.8.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ'
1	1	1	1	0	0.1 msec	0.5 msec	0	0	4 msec

Tab 5.8: Parameters required to calculate t_{BUnode} for MIPv4

The vector LUC will be $[0 \ 0 \ luc_{HA} \ 0 \ 0 \ 0]$. luc_{HA} is calculated from equation (25). The parameters required for this equation are given in table 5.9, where γ'' stands for the cost resulting from exchanging a solicitation and an advertisement with the new MA. Notice that the assignment of a FA-CoA to the MN is assumed.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''
2	2	2	1	0	10	25	0	0	$2*\rho*\delta_s$

Tab 5.9: Parameters required to calculate luc_{HA} for MIPv4

Let us now discuss the packet delivery cost. Packets are forwarded to the MN through a triangular route via the HA. The processing cost vector used to calculate $pd_{CN, MN}$ will be $[0 \ 0 \ d_{HA} \ 0 \ d_{nMA} \ 0]$. d_{HA} and d_{nMA} can be calculated from the following equations.

$$d_{HA} = \eta_1 * \lambda \quad (34)$$

$$d_{MA} = \eta_2 * \lambda \quad (35)$$

where η_1, η_2 are packet delivery processing cost constants in the HA and a MA, respectively. This analysis assumes that $\eta_1 = \eta_2 = 1$. MIPv4 does not forward data packets during the handoff. Therefore, the term $Fc_{handoff}$ equals 0.

5.6.3.2. MIPv6

Similar to MIPv4, a mesh topology is normally used. The MN always updates its mobility bindings at the HA and CN. If the triangular routing is used, the HA will be the *BUnode*. However, if the route optimization is applied, the CN will be the *BUnode*. Mobility is processed only by these *BUnodes* and there are no *InNodes*. B is equal to $[0 \ 0 \ R \ 0 \ 0 \ 0]$ for the triangular routing and $[0 \ 0 \ 0 \ 0 \ 0 \ R]$ for route optimization. Notice that the CN should be determined as an *ANP*. The handoff vector is $[0 \ 0 \ T_{HA} \ 0 \ 0 \ 0]$ for the triangular routing and $[0 \ 0 \ 0 \ 0 \ 0 \ T_{ANP}]$ when employing the route optimization. T_{HA} and T_{ANP} are calculated from (6) and (7). The other required parameters are listed in table 5.7 with the exception of a_{MA} , which is equal to 0, and γ , which is equal to $4 + \Delta_{Auto-Conf} + \Delta_{DAD}$. The value of a_{MA} is set to 0 because no mobility support is required in MAs. $\Delta_{Auto-Conf}$ and Δ_{DAD} stand for the latency resulting from the address auto-configuration and the DAD procedure, respectively, see [JPA04]. As known, the DAD procedure defined in [NNS98] requires a significant amount of time to be completed. It consumes at least one second, see [TWY06]. There are many methods proposed to minimize the time consumed by the DAD procedure. This work does not aim at analyzing the impact of DAD procedure itself. Therefore, it is assumed that the duration of DAD procedure is minimized by using an optimized DAD method, such as optimistic DAD [Moo06], proactive DAD [TWY06], advanced DAD [HCJ03] or others [MDa03]. In this analysis, the stateless address auto-configuration method is assumed. The latency of $\Delta_{Auto-Conf} + \Delta_{DAD}$ is assumed to be 50 msec.

The packet dropping vector is $[0 \ 0 \ LP_{HA} \ 0 \ 0 \ 0]$ for the triangular routing and $[0 \ 0 \ 0 \ 0 \ 0 \ LP_{ANP}]$ for route optimization. LP_{HA} and LP_{ANP} are calculated from (12) and (14). The other parameters required to calculate t_{BUnode} are listed in table 5.8 with the exception that a_{MA} equals 0 and γ' equals $4 + \Delta_{Auto-Conf} + \Delta_{DAD}$.

Let us now discuss the cost resulting from MIPv6. As mentioned above, there are two *BUnodes*, namely the HA and the CN. Whereas one *BUnode* should be taken into account to evaluate the performance, the two *BUnodes* must be considered to estimate the cost. The vector B will be $[0 \ 0 \ R \ 0 \ 0 \ R]$ regardless of the applied routing method. The vector LUC will be $[0 \ 0 \ luc_{HA} \ 0 \ 0 \ luc_{ANP}]$. luc_{HA} and luc_{ANP} are calculated from equation (25). The other required parameters for luc_{HA} are given in table 5.9 with the exception that a'_{MA} is equal to 0 and γ'' is equal to $2 * \rho * \delta_S + \Delta'_{Auto-Conf} + \Delta'_{DAD}$. $\Delta'_{Auto-Conf}$ and Δ'_{DAD} stand for the cost resulting from the address auto-configuration and the DAD procedure, respectively. Again, the stateless

address auto-configuration method is assumed and the cost of $\Delta'_{Auto-Conf} + \Delta'_{DAD}$ is assumed to be 25. The required parameters for luc_{ANP} are displayed in table 5.9 as well. However, γ'' and a'_{MA} are equal to 0. This is because the auto-configuration of the CoA and the DAD procedure are executed only one time.

Let us now discuss the packet delivery cost. Packets can be forwarded to the MN through a triangular or an optimized route. The processing cost vector used to compute $pd_{CN,MN}$ will be $[0 \ 0 \ d_{HA} \ 0 \ 0 \ 0]$ for the triangular routing. The reason for this is that there is extra processing due to the mobility in the HA only. Other nodes deal with data packets as normal IP packets. Considering the route optimization, the processing cost vector used to compute $pd_{CN,MN}$ is $[0 \ 0 \ 0 \ 0 \ 0 \ 0]$. This is because there is no extra cost due to the mobility in the nodes located on the path between the CN and the MN. d_{HA} can be calculated from equations (34). Using MIPv6, there is no forwarding of data packets during the handoff. Therefore, the term $F_{c_{handoff}}$ is equals to 0.

5.6.3.3. MIFAv4 in Reactive Mode

There are no restrictions on network topology, a mesh topology is normally used. MIFAv4 depends on the old MA to forward the MN's data packets until the HA is notified of the new binding. Thus, the *BUnode* is the old MA. This is, however, only the case for downlink traffic. For uplink, MIFA requires registering with the new MA only to be able to resume sending data packets. The new MA is, therefore, the *BUnode* for the uplink traffic. There are no *InNodes* when using MIFAv4. The vector B will be $[0 \ 0 \ 0 \ R \ 0 \ 0]$. The handoff vector is $[0 \ 0 \ 0 \ T_{MA-MR} \ 0 \ 0]$. T_{MA-MR} is calculated from (6) and (7). The other required parameters for the downlink handoff are listed in table 5.7. However, a_{MA} is assumed to be 1 msec and k_1 is equal to 1. Notice that a MA operating MIFAv4 has more tasks to complete than a MA operating MIPv4. For uplink, the required parameters are listed in table 5.10.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ
2	0	1	0	0	1 msec	1 msec	0	0	4 msec

Tab 5.10: Parameters required to calculate T_{MA-MR} for MIFAv4 in reactive mode on uplink

The packet dropped vector is $[0 \ 0 \ 0 \ LP_{MA-MR} \ 0 \ 0]$. LP_{MA-MR} is calculated from (12) and (14) for downlink and uplink traffic, respectively. The other parameters required to calculate t_{BUnode} are listed in table 5.8. However, a_{MA} is equal to 1 msec.

Considering the cost resulting from MIFAv4, this protocol updates the mobility binding at the old MA and at the HA. Therefore, there are two *BUnodes* from the cost point of view. B equals $[0 \ 0 \ R \ R \ 0 \ 0]$. The vector LUC will be $[0 \ 0 \ luc_{HA} \ luc_{MA-MR} \ 0 \ 0]$. luc_{HA} and luc_{MA-MR} are calculated from equation (25). The other parameters required to calculate luc_{MA-MR} are given in table 5.9 with the exception that a'_{MA} is equal to 25 and a'_{BUnode} is equal to 10. This is because the processing cost in the current MA is more than that required in the

old MA, which expresses a $BNode$. The parameters required to calculate luc_{HA} are given in table 5.11.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BNode}	a'_{InNode}	ni	γ''
0	2	1	1	0	10	25	0	0	$a'_{MA} + N_{av} * \delta_S * D_{CurrentMA, neiMA}$

Tab 5.11: Parameters required to calculate luc_{HA} for MIFAv4 in reactive mode

The value of k_1 is 0 because the HA is informed by the new MA and not by the MN. γ'' stands for the cost resulting from informing the neighbor MAs to allow them to quickly re-authenticate the MN. N_{av} is the average number of MAs present in the L3-FHRs. N_{av} can be calculated from the equation below.

$$N_{av} = \frac{\sum_{i=1}^Z N_i}{Z} \quad (36)$$

where N_i is the number of MAs located in the L3-FHR of MA_i . $D_{CurrentMA, neiMA}$ expresses the distance between the current and a neighbor MA. This distance is equal to $\bar{D}_{newMA, oldMA}$ for the assumed mesh topology.

Let us now discuss the packet delivery cost. Data packets are forwarded to the MN through a triangular route via the HA. The processing cost vector used to compute $pd_{CN, MN}$ will be $[0 \ 0 \ d_{HA} \ 0 \ d_{nMA} \ 0]$. d_{HA} and d_{nMA} can be calculated from equations (34) and (35), respectively. Data packets are forwarded, however, from the old MA to the new one during the handoff. The processing cost vector used to compute $F_{c_{handoff}}$ is $[0 \ 0 \ 0 \ d_{oMA} \ d_{nMA} \ 0]$. k_9 is equal to 1. This is because data packets will be forwarded to the new MA only. $ContNode$ is the HA, whereas $RNode$ is the old MA. The average value of Δ is assumed to be 70 msec, which is the time required to inform the $ContNode$ and forward the in-flight packets, see appendix D for details.

5.6.3.4. MIFAv6 in Reactive Mode

MIFAv6 functions similar to MIFAv4. On downlink, MIFAv6 depends on the old MA to forward the MN's data packets until the HA or CN is informed about the new binding. The $BNode$ is, therefore, the old MA. On Uplink, the new MA is the $BNode$. A mesh topology is used and there are no $InNodes$. B , the handoff vector and the packet dropping vector are the same as in MIFAv4. The parameters required to calculate T_{MA-MR} on downlink are listed in table 5.7. However, a_{MA} is assumed to be 1 msec, k_1 equals 1 and γ is equal to $4 + \Delta_{Auto-Conf} + \Delta_{DAD}$. For uplink, the required parameters are listed in table 5.10 with the exception of γ , which equals to $4 + \Delta_{Auto-Conf} + \Delta_{DAD}$. For LP_{MA-MR} , the parameters required to calculate t_{BNode} are listed in table 5.8 with the exception that a_{MA} equals 1 msec and γ' is

equal to $4 + \Delta_{Auto-Conf} + \Delta_{DAD}$. The stateless address auto-configuration is assumed and the value of $\Delta_{Auto-Conf} + \Delta_{DAD}$ is supposed to be 5 msec. The reason for this is that the specification of MIPv6 allows the new MA to quickly detect the duplicated addresses.

Considering the cost resulting from employing MIPv6, not only the old MA and HA are notified of the binding, but also the CN. Therefore, there are three *BUnodes*. Again, there are no *InNodes*. B equals to $[0 \ 0 \ R \ R \ 0 \ R]$. The vector LUC will be $[0 \ 0 \ luc_{HA} \ luc_{MA-MR} \ 0 \ luc_{ANP}]$. luc_{HA} , luc_{MA-MR} and luc_{ANP} are calculated from equation (25). The parameters required to calculate luc_{MA-MR} are the same as in MIPv4 in reactive mode except γ'' , which is equal to $2 * \rho * \delta'_S + \Delta'_{Auto-Conf} + \Delta'_{DAD}$. The cost of $\Delta'_{Auto-Conf} + \Delta'_{DAD}$ is assumed to be 10. The parameters required to calculate luc_{HA} are the same as in MIPv4 in reactive mode. The parameters required to calculate luc_{ANP} are given in table 5.9. However, a'_{MA} and γ'' are 0.

For the packet delivery cost, packets are forwarded to the MN through a triangular or an optimized route. The processing cost vector used to calculate $pd_{CN,MN}$ will be $[0 \ 0 \ d_{HA} \ 0 \ 0 \ 0]$ for the triangular routing and $[0 \ 0 \ 0 \ 0 \ 0 \ 0]$ for route optimization. d_{HA} can be calculated from equation (34). Data packets are forwarded from the old MA to the new one during the handoff. The processing cost vector and k_g required to calculate $F_{c_{handoff}}$ are the same as in the case of MIPv4 in reactive mode. If the triangular route is used, the *ContNode* is the HA and the average value of Δ is assumed to be 70 msec, whereas the *ContNode* is the CN and the average value of Δ is assumed to be 60 msec if the route optimization is used. *RNode* is the old MA. For details, see appendix D.

5.6.3.5. MIPRR

MIPRR requires a hierarchical network topology. The MN updates its mobility binding at RFAs or the GFW while moving inside the domain. The RFAs are represented by MRs in the generic mathematical model, while the GFW is represented by the GW. Thus, there are two *BUnodes*, namely the MRs and the GW. There are no *InNodes* inside the domain. The vector B equals to $[R \ G \ 0 \ 0 \ 0 \ 0]$. The handoff vector is $[T_{MR} \ T_{GW} \ 0 \ 0 \ 0 \ 0]$. T_{MR} and T_{GW} are calculated from equations (6) and (7). The other required parameters are listed in table 5.7. The packet dropping vector is $[LP_{MR} \ LP_{GW} \ 0 \ 0 \ 0 \ 0]$. LP_{MR} and LP_{GW} are calculated from equation (12) and (14) for downlink and uplink traffic, respectively. Other parameters required to calculate t_{BUnode} are listed in table 5.8.

The vector LUC will be $[luc_{MR} \ luc_{GW} \ 0 \ 0 \ 0 \ 0]$. luc_{MR} and luc_{GW} are calculated from equation (25). The parameters listed in table 5.9 are the same for MIPRR. For the packet delivering cost, data packets are forwarded to the MN through a triangular route via the HA, GW, MR and the MA. The processing cost vector used to calculate $pd_{CN,MN}$ will be $[d_{MR} \ d_{GW} \ d_{HA} \ 0 \ d_{nMA} \ 0]$. d_{HA} and d_{nMA} can be calculated from equations (34) and (35). The processing cost in the GW results from the de-capsulation of the tunneled packets forwarded from the HA, checking the visitor list to see if the MN has an entry, re-encapsulating data packets to forward them to the serving MR and managing the routing to this MR. Furthermore,

the processing cost in MRs results from similar tasks. The processing cost in the GW depends on the number of MNs in the domain and on the number of MRs it serves, while the processing cost in a MR depends on the number of MNs served by it and on the number of MAs it controls. Assuming the average number of MNs in each subnet is w , the complexity of the GW visitor list lookup is proportional to w^*g^*v , while the complexity of each MR visitor list lookup is proportional to w^*v , see [JAK02]. The IP routing table lookup is based on the longest prefix matching. Thus, for the wide used traditional Patricia trie [LSV99], the complexity of IP address lookup is proportional to the logarithm of the routing table length [TPr99]. As a result, packet processing cost per time unit in the GW and in each MR can be calculated from equations (37) and (38), respectively.

$$d_{GW} = \xi_1^*g^*\lambda^*(\alpha_1^*w^*g^*v + \beta_1^*\log(g)) \quad (37)$$

$$d_{MR} = \xi_2^*v^*\lambda^*(\alpha_2^*w^*v + \beta_2^*\log(v)) \quad (38)$$

where α and β are weighting factors of visitor list and routing table lookups. ξ is a constant and expresses the bandwidth allocation cost. No forwarding of packets is performed during the handoff. As a result, the term $Fc_{handoff}$ is equal to 0.

5.6.3.6. HAWAII

HAWAII uses normally a hierarchical network topology. It always updates the mobility binding at the old MA. Consequently, the old MA is the *BUnode* for uplink and downlink traffic. The vector B will be $[0 \ 0 \ 0 \ R \ G \ 0]$. The handoff vector is $[0 \ 0 \ 0 \ T_{MA-MR} \ T_{MA-GW} \ 0]$. T_{MA-MR} and T_{MA-GW} are calculated from equations (6) and (7). The other required parameters are listed in table 5.12.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ
2	2	2	1	2	0.1 msec	0.2 msec	0.1 msec	Calculated from (39)	4

Tab 5.12: Parameters required to calculate T_{MA-MR} and T_{MA-GW} for HAWAII

The average number of *InNodes* can be calculated from equation (39). D_1 and D_2 express the distance between the new and old MA via a MR and the GW, respectively.

$$ni = (R*D_1 + G*D_2) - 1 \quad (39)$$

The packet dropping vector is $[0 \ 0 \ 0 \ LP_{MA-MR} \ LP_{MA-GW} \ 0]$. LP_{MA-MR} and LP_{MA-GW} are calculated from equations (12) and (14). The other parameters required to calculate t_{BUnode} are listed in table 5.13. Notice that we should distinguish between the forwarding (MSF and SSF) and non-forwarding (MNF and UNF) schemes when analyzing HAWAII. Although the MN sends a **path steup update** message to the old MA, the crossover router stops sending data packets on the old path after the receipt of the **path setup update** message when employing the non-forwarding schemes. For the forwarding schemes, the old MA sends data packets to the new one after the receipt of the **path setup update** message. Therefore, for the non-

forwarding schemes, the *BUNode* controlling the expected number of dropped packets on downlink is the crossover router, which is a MR for LP_{MA-MR} and the GW for LP_{MA-GW} . For uplink, the *BUNode* still be the old MA, which is also the *BUNode* for uplink and downlink traffic when employing the forwarding schemes.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	a_{BUNode}	a_{InNode}	ni	γ'
1	1	1	1	1	0.1 msec	0.2 msec	0.1 msec	Calculated from (39)	4

Tab 5.13: Parameters required to calculate t_{BUNode} for HAWAII

The vector LUC will be $[0 \ 0 \ 0 \ luc_{MA-MR} \ luc_{MA-GW} \ 0]$. luc_{MA-MR} and luc_{MA-GW} are calculated from equation (25). The other parameters required are listed in table 5.14.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUNode}	a'_{InNode}	ni	γ''
2	2	2	1	2	5	10	5	Calculated from (39)	$2*\rho*\delta_s$

Tab 5.14: Parameters required to calculate luc_{MA-MR} and luc_{MA-GW} for HAWAII

Data packets can be forwarded to the MN through a triangular route or through utilizing the route optimization. Data packets will be sent via the HA and the domain root router to the MN using the triangular routing, while data packets will be sent from the CN via domain root router to the MN using the route optimization. The processing cost vector used to compute $pd_{CN,MN}$ will be $[0 \ d_{GW} \ d_{HA} \ 0 \ 0 \ 0]$ for the triangular routing and $[0 \ d_{GW} \ 0 \ 0 \ 0 \ 0]$ for route optimization. d_{HA} and d_{GW} can be calculated from equations (34) and (37), respectively.

As known, HAWAII uses either forwarding or non-forwarding schemes. There is no forwarding of data packets during the handoff when using the non-forwarding schemes. This means $Fc_{handoff}$ is zero. When forwarding schemes are used, packets are forwarded from the old to the new MA. The processing cost vector used to compute $Fc_{handoff}$ is $[0 \ 0 \ 0 \ 0 \ 0 \ 0]$. This is because no special processing (e.g. encapsulation, de-capsulation, etc.) is required while forwarding the in-flight packets. k_9 is equal to 1. *ContNode* is the crossover node, i.e. a MR or the GW. *RNode* is the old MA. The average value of Δ is calculated from equation (40). For details, see appendix D.

$$\Delta = 2 * \tau_2 * (R * D_{MA-MR} + G * D_{MA-GW}) \quad (40)$$

5.6.3.7. Proxy MIPv6

Proxy MIPv6 uses normally a mesh topology. The MN does not have any mobility support. Mobility bindings are updated at the LMA, which is represented through the HA in our model. There are no *InNodes* between MAGs, which are represented through MAs, and the LMA. The vector B equals $[0 \ 0 \ R \ 0 \ 0 \ 0]$ and the handoff vector is $[0 \ 0 \ T_{HA} \ 0 \ 0 \ 0]$. T_{HA} is calculated from equations (6) and (7), which require the parameters listed in table 5.15.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ
0	2	1	1	0	0.1 msec	0.5 msec	0	0	2

Tab 5.15: Parameters required to calculate T_{HA} for Proxy MIPv6

γ results from sending a **RA** message to the MN. The packet dropping vector is $[0 \ 0 \ LP_{HA} \ 0 \ 0 \ 0]$. LP_{HA} is calculated from equations (12) and (14) for downlink and uplink, respectively. The parameters required to calculate t_{BUnode} are listed in table 5.16.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ'
0	1	0	1	0	0	0.5 msec	0	0	0

Tab 5.16: Parameters required to calculate t_{BUnode} for Proxy MIPv6

The vector LUC will be $[0 \ 0 \ luc_{HA} \ 0 \ 0 \ 0]$. luc_{HA} is calculated from equation (25). The other parameters required are given in table 5.17.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''
0	2	1	1	0	10	25	0	0	$\rho * \delta_s$

Tab 5.17: Parameters required to calculate luc_{HA} for Proxy MIPv6

Notice that γ'' stands for the cost resulting from sending a **RA** message to the MN. Data packets are forwarded to the MN via the LMA. The processing cost vector used to calculate $pd_{CN, MN}$ will be $[0 \ 0 \ d_{HA} \ 0 \ d_{nMA} \ 0]$. d_{HA} and d_{nMA} are calculated from equations (34) and (35), respectively. The parameters of equations (34) and (35) are the same as in MIPv4. As known, Proxy MIPv6 forwards no packets during the handoff, which means that the term $Fc_{handoff}$ is equal to 0.

5.6.4. Application to Make-Before-Break Mobility Management Protocols

5.6.4.1. MIFAv4 in Predictive Mode

As mentioned in section 5.6.3.3, a mesh topology is deployed. MIFAv4 in predictive mode updates mobility bindings at the new MA when a L2-trigger is fired. Therefore, the $BUnode$ is the new MA. Moreover, no $InNodes$ are required. The vector B will be $[0 \ 0 \ 0 \ R \ 0 \ 0]$. The handoff vector is $[0 \ 0 \ 0 \ T_{MA-MR} \ 0 \ 0]$ and the packet dropping vector is $[0 \ 0 \ 0 \ LP_{MA-MR} \ 0 \ 0]$.

So as to analyze the performance of MIFAv4 in predictive mode, the three cases presented in section 5.4.2 should be taken into account. For the first case, MIFAv4 in reactive mode will be employed. Taking the second case into account, MIFAv4 will be employed in reactive mode as well. However, Δt , γ and γ' are equal to zero. Considering the third case, T_{MA-MR} is calculated from equation (15). The values of the parameters required to calculate T'_{MA-MR} are

listed in table 5.18. γ_{ext} is equal to 2 msec, which results from sending the **Reg_Rply** message to the MN after finishing the layer 2 handoff.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ
1	2	2	1	0	1 msec	1 msec	0	0	4

Tab 5.18: Parameters required to calculate T'_{MA-MR} for MIFAv4 in predictive mode

LP_{MA-MR} is calculated from equations (14) and (18) for uplink and downlink, respectively. The parameters required to calculate t_{RNode} are given in table 5.19, where $RNode$ is the old MA.

k_3	k_4	k_5	k_6	k'_4	k'_5	k'_6
1	0	0	0	1	0	0
k'_7	γ'_1	a_{MA}	a_{BUnode}	a_{InNode}	a_{RNode}	ni
0	4	1 msec	1 msec	0 msec	1 msec	0

Tab 5.19: Parameters required to calculate t_{RNode} for MIFAv4 in predictive mode

Let us now discuss the cost resulting from employing MIFAv4 in predictive mode. This protocol updates the mobility binding at the new MA and HA. B is equal to $[0 \ 0 \ R \ R \ 0 \ 0]$. LUC is $[0 \ 0 \ luc_{HA} \ luc_{MA-MR} \ 0 \ 0]$. luc_{HA} and luc_{MA-MR} are calculated from equation (25). The parameters required to calculate luc_{MA-MR} are given in table 5.9 with the exception that k_1 is equal to 3. For luc_{HA} , the required parameters are provided in table 5.11. The term $pd_{CN, MN}$ is the same as in the case of MIFAv4 in reactive mode. As known, data packets are forwarded from the old MA to the new one during the handoff, while the term $Fc_{handoff}$ is calculated in the same way as in the case of MIFAv4 in reactive mode. However, the average value of Δ is assumed to be 91 m sec, see appendix D for details.

5.6.4.2. MIFAv6 in Predictive Mode

As mentioned in section 5.6.3.4, a mesh topology is used. The $BUnode$ is the new MA and there are no $InNodes$. The B vector, handoff vector and packet dropping vector are the same as in MIFAv4 in predictive mode, see the previous section.

Let us now consider the three cases presented in section 5.4.2. In the first case, MIFAv6 in reactive mode will be employed. The same will occur in the second case as well. However, Δt will be zero, while γ and γ' are equal to $\Delta_{DAD} \cdot T_{MA-MR}$ in the third case is calculated from equation (15). The parameters required to calculate T'_{MA-MR} are the same as in the case of MIFAv4 in predictive mode. However, γ is equal to $4 + \Delta_{Auto-Conf} + \Delta_{DAD}$. The value of γ_{ext} is the same as in the case of MIFAv4 in predictive mode. Additionally, LP_{MA-MR} is calculated in the same manner as in MIFAv4 in predictive mode.

Considering the cost resulting from employing MIFAv6 in predictive mode, the mobility binding is updated at the new MA, HA and CN. Therefore, B is equal to $[0 \ 0 \ R \ R \ 0 \ R]$, while LUC is $[0 \ 0 \ luc_{HA} \ luc_{MA-MR} \ 0 \ luc_{ANP}]$. luc_{HA} , luc_{MA-MR} and luc_{ANP} are calculated from equation (25). The parameters required to calculate luc_{MA-MR} are the same as in MIFAv4 in predictive mode. However, γ'' is equal to $2 * \rho * \delta'_S + \Delta'_{Auto-Conf} + \Delta'_{DAD}$. The parameters required to calculate luc_{HA} are the same as in MIFAv4 in predictive mode, while the parameters required to calculate luc_{ANP} are the same as in MIFAv6 in reactive mode. The terms $pd_{CN,MN}$ and $F_{handoff}$ are the same as in the case of MIFAv6 in reactive mode. However, Δ is assumed to be 96 msec if the triangular route is used. For route optimization, Δ is assumed to be 122 msec, see appendix [D](#) for details.

5.6.4.3. Pre-Registration Method

Pre-registration method does not make any restrictions on the network shape, a mesh topology is deployed. The mobility binding is always updated at the HA, which expresses the $BUnode$. The pre-registration method does not require any $InNodes$. The B vector is $[0 \ 0 \ R \ 0 \ 0 \ 0]$, while the handoff vector is $[0 \ 0 \ T_{HA} \ 0 \ 0 \ 0]$ and the packet dropping vector is $[0 \ 0 \ LP_{HA} \ 0 \ 0 \ 0]$.

Considering the three cases presented in section [5.4.2](#), MIPv4 is employed in the first and second cases. In the second case, however, Δt , γ and γ' are equal to zero. T_{HA} in the third case is calculated using equation (15). The values of the parameters required to calculate T'_{HA} are listed in table 5.7. However, γ is set to 4.2 msec, which results from exchanging **PrRtSol** and **PrRtAdv** messages between the MN and the old MA in addition to the processing required in the new MA.

In order to calculate γ'_{ext} , it should be noted that the MN sends a **RegRqst** message to the HA via the old and new MA. The **RegRply** message is sent to the MN over the old and new wireless links. Sending the **RegRply** from the new MA to the MN via the old MA has no impact on handoff duration. This means that the latency resulting from sending the **RegRply** from the new MA to the MN via the old MA should be subtracted from T'_{HA} . Furthermore, transmitting the **RegRply** to the MN on the new wireless link following the layer 2 handoff should be taken into account when calculating γ'_{ext} . Thus, the value of γ'_{ext} is equal to -10.1 msec. LP_{HA} is calculated from equation (14) and (18) in the third case. $RNode$ is the HA and the parameters required to calculate t_{RNode} are listed in table 5.20.

k_3	k_4	k_5	k_6	k'_4	k'_5	k'_6
1	1	0	0	1	1	0
k'_7	γ'_1	a_{MA}	a_{BUnode}	a_{InNode}	a_{RNode}	ni
0	4.1 msec	0.1 msec	0.5 msec	0	0	0

Tab 5.20: Parameters required to calculate t_{RNode} for the pre-registration method

From the cost point of view, the B vector does not change. The LUC vector is $[0 \ 0 \ luc_{HA} \ 0 \ 0 \ 0]$. luc_{HA} is calculated from equation (25), which requires the parameters as listed in table 5.9 except γ , which is set to $3 * \rho * \delta_S + a_{MA}$. The value of γ comprises the cost resulting from exchanging **PrRtSol** and **PrRtAdv** messages between the MN and the old MA in addition to the cost resulting from sending a **RegRply** message from the new MA to the MN as well as the processing cost required in the new MA. The packet delivery cost resulting from the pre-registration method is the same as in MIPv4.

5.6.4.4. FMIPv6

Similar to MIPv6, a mesh topology is used and there are no $InNodes$. B equals $[0 \ 0 \ 0 \ R \ 0 \ 0]$. The handoff vector and the packet dropping vector are $[0 \ 0 \ 0 \ T_{MA-MR} \ 0 \ 0]$ and $[0 \ 0 \ 0 \ LP_{MA-MR} \ 0 \ 0]$, respectively. Considering the three cases presented in section 5.4.2, MIPv6 is employed in the first case, while FMIPv6 operates in reactive mode in the second case. The predictive mode will be employed in the third case.

FMIPv6 in reactive mode notifies the old MA via the new one. The $BUnode$ is, therefore, the old MA. The B vector, the handoff vector and the packet dropping vector are listed above. T_{MA-MR} in this case is calculated from (6) and (7). The other parameters required to calculate T_{MA-MR} are listed in table 5.21.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ
1	2	2	1	0	1 msec	1 msec	0	0	Δ_{DAD}

Tab 5.21: Parameters required to calculate T_{MA-MR} for FMIPv6 in reactive mode

The specification of FMIPv6 assumes that the new MA can detect the duplicated addresses very quickly, see [Koo05]. Therefore, the value of Δ_{DAD} is assumed to be 5 msec.

LP_{MA-MR} is calculated from (12) and (14) for downlink and uplink traffic, respectively. The other parameters required to calculate t_{BUnode} are given in table 5.22.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ'
1	1	1	1	0	1 msec	1 msec	0	0	Δ_{DAD}

Tab 5.22: Parameters required to calculate t_{BUnode} for FMIPv6 in reactive mode

FMIPv6 in predictive mode works as follows: upon the MN notices that a handoff will occur, it registers with the new MA via the old one. Thus, the $BUnode$ is the new MA. The parameters required to calculate T'_{MA-MR} in the third case are the same as in MIPv6 in predictive mode. The value of γ_{ext} is 3 msec, which results from sending a **F-NAadv** message from the MN to the new MA after the layer 2 handoff in addition to the processing required in the new MA. The old MA is the $RNode$ and LP_{MA-MR} is calculated using equations (14) and (18) in the third case. The parameters required to calculate t_{RNode} are provided in table 5.23.

This analysis assumes that the *RNode* buffers data packets after the appearance of the L2-LD trigger if a **F-BU** message has been received by the *RNode*.

k_3	k_4	k_5	k_6	k'_4	k'_5	k'_6
1	0	0	0	1	0	0
k'_7	γ'_1	a_{MA}	a_{BUNode}	a_{InNode}	a_{RNode}	ni
0	4	1 msec	1 msec	0	0	0

Tab 5.23: Parameters required to calculate t_{RNode} for FMIPv6 in predictive mode

Let us now consider the location update cost. LUC will be $[0 \ 0 \ 0 \ luc_{MA-MR} \ 0 \ 0]$ for reactive as well as predictive mode. The parameters required to calculate luc_{MA-MR} for the reactive mode are listed in table 5.9. However, a'_{BUNode} is assumed to be 10 and γ'' is equal to $2 * \rho * \delta'_S + \Delta'_{Auto-Conf} + \Delta'_{DAD}$. For the predictive mode, the required parameters are provided in table 5.9 too. However, k_2 and k'_2 are set to 3. a'_{BUNode} is assumed to be 10, while γ'' is set to $3 * \rho * \delta'_S + \Delta'_{Auto-Conf} + \Delta'_{DAD}$, which comprises the cost resulting from the address auto-configuration and DAD procedure, the cost resulting from exchanging **RtSolPr** and **PrRtAdv** messages between the MN and the old MA and the cost resulting from sending a **F-NAdv** message to the new MA after finishing the layer 2 handoff. After the MN completes the handoff, it has to register again with the HA and CN using MIPv6. The cost resulting from this registration should be considered. The parameters are the same as in MIPv6 except γ'' , which is equal to 0. This is because no solicitation and advertisement messages should be exchanged between the MN and the new MA. In addition, the CoA auto-configuration and the DAD procedure should not be executed again.

The processing cost vector used to compute $pd_{CN,MN}$ is the same as in MIPv6. Data packets are forwarded from the old to new MA during the handoff and even until the HA or CN is notified of the new mobility binding. Therefore, the HA expresses the *ContNode* for the triangular routing, while the CN is the *ContNode* using the route optimization. The processing cost vector used to calculate $Fc_{handoff}$ is $[0 \ 0 \ 0 \ d_{oMA} \ d_{nMA} \ 0]$ for predictive as well as reactive mode. k_9 equals 1 and *RNode* is the old MA. Considering the reactive operation mode, the value of Δ is supposed to be 93 and 83 msec for the triangular routing and route optimization, respectively. It is assumed that the MN informs the HA and CN directly after completion the layer 3 handoff. Notice that the old MA starts forwarding data packets to the new location of the MN after it gets informed and until the in-flight packets are forwarded under the assumption that the HA or possibly the CN has just sent a data packet before the receipt of the **BU** message. For the predictive operation mode, the value of Δ is assumed to be 132 msec for the triangular routing and 122 msec if the data packets are routed using the optimized route from the CN to the MN. The value of Δ results in this case from the sum of the layer 2 handoff latency, the time required to inform the HA and possibly the CN after the layer 2 handoff and the time required to forward the packets in-flight. For more details, see appendix [D](#).

5.6.5. Performance Evaluation

5.6.5.1. Break-Before-Make Mobility Management Protocols

This section evaluates the studied break-before-make mobility management protocols with respect to the average handoff latency and expected average number of dropped packets per handoff. Notice that the handoff latency comprises both the layer 2 and layer 3 handoff latency. Consequently, the expected average number of dropped packets also results from both layer 2 and layer 3 handoffs. In the following, the term MIFA is used where issues relevant for MIFAv4 and MIFAv6 are discussed. Otherwise, the term MIFAv4 or MIFAv6 is used explicitly. The same applies to the use of MIP.

5.6.5.1.1. Average Handoff Latency

Figure 5.6 presents the average handoff latency experienced when employing the studied break-before-make mobility management protocols in case of dropping of the *update message* no and one time. Notice that MIFA performs comparably in IPv4 and IPv6 networks, whereas the latency experienced by MIPv6 is significantly greater than that resulting from MIPv4. The reason for this is the time required for the address auto-configuration and DAD procedure when employing MIPv6. Execution of the DAD procedure when employing MIPv4 is only necessary if the MN obtains a co-located CoA. There is no need for the DAD procedure if the MN is assigned a FA-CoA, which is the case assumed in this analysis. In contrast to MIPv6, MIFAv6 requires only checking the list of the IP addresses registered with the new MA to detect duplicated addresses. Therefore, the DAD procedure is completed very quickly.

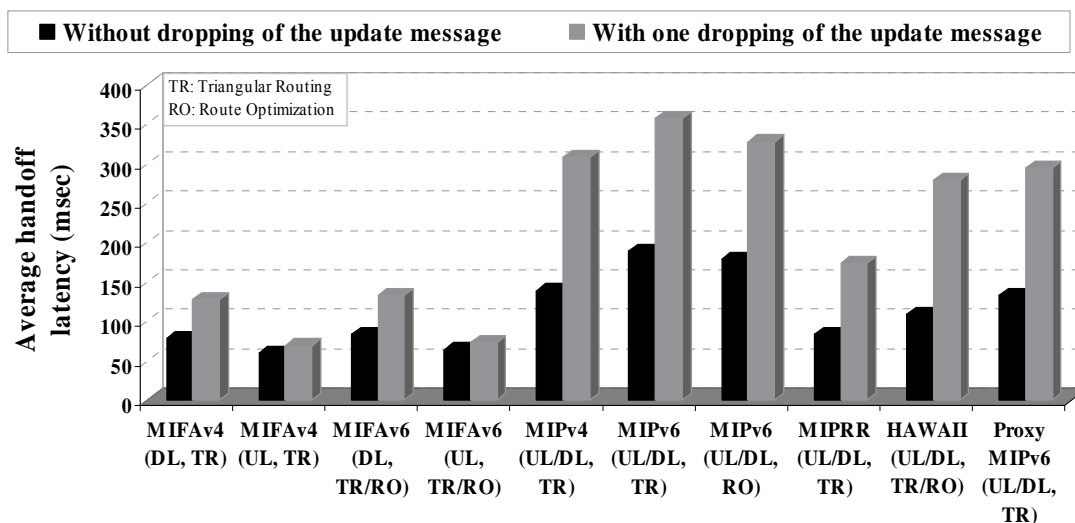


Fig 5.6: Average handoff latency resulting from employing MIFAv4 in reactive mode, MIFAv6 in reactive mode, MIPv4, MIPv6, MIPRR, HAWAII and Proxy MIPv6

Let us first discuss the average handoff latency on downlink without dropping of any control message. According to our analysis, MIFAv4 is 43.4 % better than MIPv4. Compared to MIPv6, MIFAv6 is 55.7 % and 53.2 % better when forwarding data packets towards the MN via a triangular or an optimized route, respectively. The reason for this is that the *BUnode* by MIFA is the old MA, which is only two hops away from the new MA in the assumed mesh topology. In contrast, the *BUnode* by MIP is the HA and possibly the CN, which is farther away than the old MA. Regarding MIPRR, a comparable performance to MIFA can be seen. This is because MIPRR has two *BUnodes*, either a MR or the GW. According to the defined

mobility scenario, the average distance of the *BUnode* by MIPRR is 2.5 hops¹. Although the *BUnode* by MIFA and HAWAII is the old MA considering the downlink handoffs, HAWAII is outperformed by MIFA by approximately 25 %. There are two reasons for this result. The first is the used network topology. MIFA is studied deploying a mesh topology, while HAWAII is analyzed deploying a hierarchical topology. The second reason is the processing required in *InNodes*. As known, all nodes in the HAWAII domain are mobility-aware. Compared to Proxy MIPv6, MIFAv6 clearly performs better as well. This is because Proxy MIPv6 updates the binding at the LMA, represented by the HA in the model. According to our analysis, MIFAv6 performs 37 % better.

Let us now study the average handoff latency on downlink in the case of dropping of the *update message* once on the wireless link. MIFAv4 performs 58.6 % better than MIPv4. MIPv6 is outperformed by MIFAv6 by 63 % and 59.5 % using the triangular route and the route optimization, respectively. Although MIPRR is comparable to MIFAv4 when no dropping of the *update message* occurs, it results in 26.3 % more latency in case of dropping. This is due to the duration of the used timer, which is smaller in the case of MIFAv4 than MIPRR, see section 5.6. For a similar reason, HAWAII is outperformed by MIFA by approximately 53 %, while Proxy MIPv6 produces 55 % more handoff latency.

Considering the average handoff latency on uplink, MIFA is a very fast protocol. This is because the MN only requires contacting the new MA to resume uplink communication. This results in a fast detection and recovery of dropped control messages. In the case of no dropped *update messages* on the wireless link, MIFAv4 is 57.5 % and 29.5% faster than MIPv4 and MIPRR, respectively. MIFAv6 performs 66 %, 64.15 % and 51.7 % better than MIPv6 when sending data packets on a triangular route, MIPv6 employing the route optimization and Proxy MIPv6, respectively. Compared to HAWAII, MIFA is approximately 34 % better. Dropping of the *update message* has a minimal impact on the uplink handoff latency resulting from MIFA, while other protocols experience a significant increase in the handoff latency.

5.6.5.1.2. *Expected Average Number of Dropped Packets Per Handoff*

Figure 5.7 shows the expected average number of dropped packets per handoff on downlink and uplink when employing the studied break-before-make mobility management protocols in case of no and one dropping of the *update message*. The results obtained from this figure are similar to those shown in figure 5.6. Clearly, the best performance is achieved by MIFA.

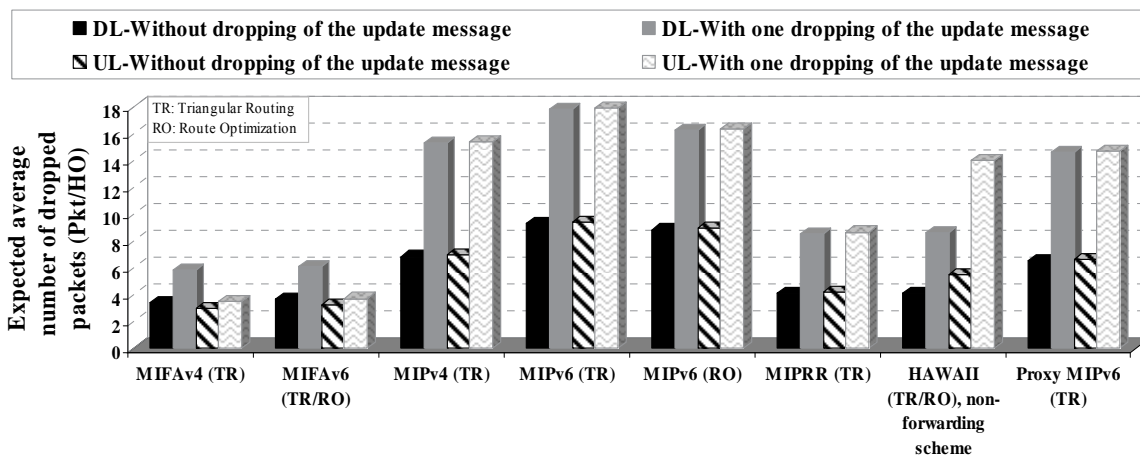


Fig 5.7: Expected average number of dropped packets per handoff resulting from employing MIFAv4 in reactive mode, MIFAv6 in reactive mode, MIPv4, MIPv6, HAWAII (non-forwarding scheme) and Proxy MIPv6

¹ MR is 2 hops away from the new MA, while the GW is 4 hops away. The average value is then calculated taking the values of *R* and *G* into account. For details, see appendix E.

Compared to MIPv4, if the handoff has been completed without any dropping of the *update message*, MIFAv4 is 50.9 % better on downlink and 57.5 % better on uplink. In case the *update message* has been dropped once on the wireless link, MIFAv4 performs 62 % and 77.6 % better on downlink and uplink, respectively. Considering MIPv6, if the MN's data packets are routed through the triangular route, MIPv6 drops 61 % more on downlink than MIFAv6 in the case that the registration has been completed successfully without dropping of any *update message*. In case of dropping, MIFAv6 is 66 % better than MIPv6. On uplink, MIPv6 drops 66 % and 79.3 % more than MIFAv6 in the case of no and one dropping of the *update message*, respectively. If the MN's data packets are routed from the CN directly towards the MN without passing the HA, MIFAv6 drops on downlink 59 % and 62.7 % less than MIPv6 with no and one dropping of the *update message*, respectively. On uplink, MIFAv6 performs 64.1 % and 77.4 % better than MIPv6 in case of no and one dropping of the *update message*.

Compared to MIPRR, if the *update message* has not been dropped on the wireless link, MIPRR drops 17.3 % and 29.5 % more than MIFAv4 on downlink and uplink, respectively. Due to the fast recovery of dropped control messages dropping, MIFAv4 performs in this case 31.9 % and 60.1 % better than MIPRR on downlink and uplink, respectively. For the same reasons highlighted while discussing the handoff latency in figure 5.6, HAWAII is outperformed by MIFA on downlink and uplink. According to the achieved results, MIFA results in about 14 % and 30 % less dropped packets than HAWAII on downlink with no and one dropping of the *update message*, respectively. On uplink, HAWAII produces approximately 43 % and 74 % more dropped packets than MIFA in the case of no and one dropping of the *update message*, respectively.

Regarding Proxy MIPv6, it is outperformed by MIFAv6 by 44.4 % on downlink and 51.7 % on uplink in case the *update message* has not suffered any dropping. In the case of one dropping of the *update message*, MIFA performs 58.3 % better on downlink and 74.8 % better on uplink than Proxy MIPv6.

5.6.5.2. *Make-Before-Break Mobility Management Protocols*

This section analyzes the performance of the studied make-before-break mobility management protocols. Throughout this section, if the results being discussed are related to a certain operation mode, reactive or predictive, this mode will be mentioned explicitly. Otherwise, discussions refer to both methods.

5.6.5.2.1. *Average Handoff Latency*

Figure 5.8 shows the average handoff latency experienced when employing the studied make-before-break mobility management protocols in the three cases presented in section 5.4.2. In the first case, where no L2-trigger appears at the MN, MIFA performs clearly better than the three other protocols. This is due to operating MIFA in reactive mode, while the pre-registration method and FMIPv6 resort to MIPv4 and MIPv6, respectively. A detailed comparison of these protocols was given in section 5.6.5.1.1. Considering the second case, the delayed L2-trigger prohibits either sending the *update message* on the old wireless link, or recovering the dropping of it. The figure shows that MIFAv6 performs on downlink comparable to FMIPv6, while MIFAv4 performs significantly better than the pre-registration method, 44.7 % better according to our results. On uplink, MIFA is faster than the other studied protocols. According to our analysis, MIFAv6 performs 25 % better than FMIPv6, while MIFAv4 results in 59.2 % less handoff latency than the pre-registration method. The reason for this is that MIFA as well as FMIPv6 will be employed in reactive mode. In contrast, the pre-registration method resorts to MIPv4. Clearly, the movement detection time (Δt) is

saved and there is no need to exchange a solicitation and an advertisement message with the new MA.

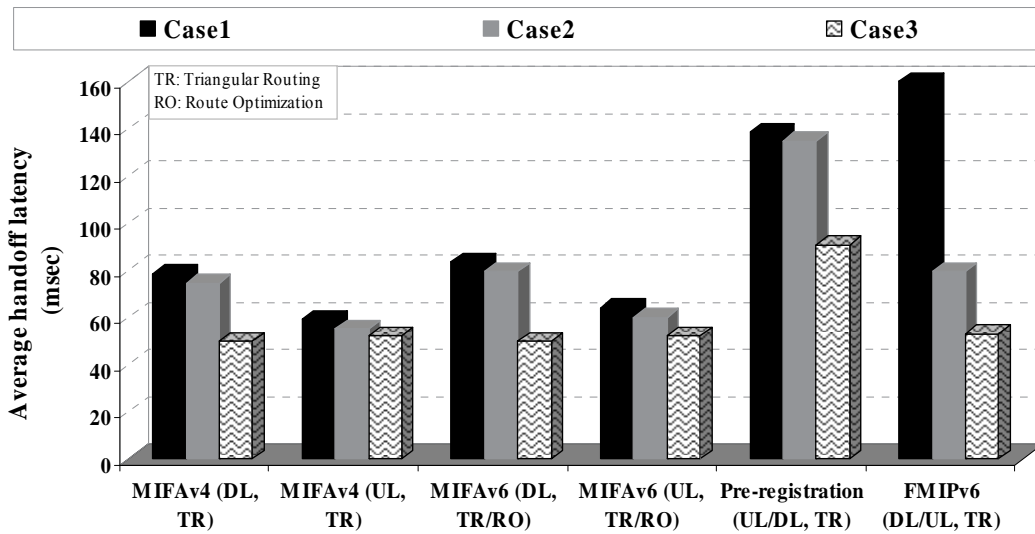


Fig 5.8: Average handoff latency resulting from employing MIFAv4 in predictive mode, MIFAv6 in predictive mode, the pre-registration method and FMIPv6

The best performance is observed in the third case, where the MN benefits from layer 2 triggers. MIFA and FMIPv6 minimize the handoff latency to approximately the latency resulting from the layer 2 handoff. The handoff latency is reduced by the pre-registration method too. However, the MN still requires waiting some time after the layer 2 handoff to complete the layer 3 handoff. 34.5 % of the handoff latency is saved if the pre-registration method was able to initiate the layer 3 handoff in advance. Compared to MIFAv4, the pre-registration method is outperformed by 45 % on downlink and 42.7 % on uplink.

5.6.5.2.2. Expected Average Number of Dropped Packets Per Handoff

Figure 5.9 shows the expected average number of dropped packets experienced when employing the studied make-before-break mobility management protocols in the three cases discussed in section 5.4.2. Similar results to those obtained from figure 5.8 can be derived from this figure as well.

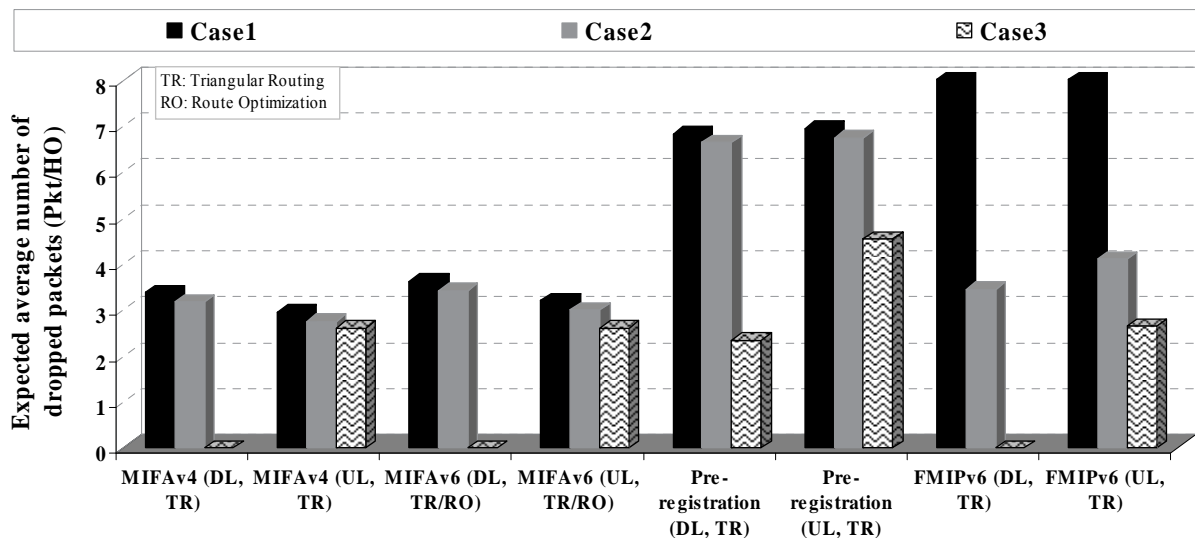


Fig 5.9: Expected average number of dropped packets per handoff resulting from employing MIFAv4 in predictive mode, MIFAv6 in predictive mode, the pre-registration method and FMIPv6

MIFA performs clearly better than the other protocols in the first case. The reason for this was presented while discussing figure 5.8. In the second case MIFAv6 performs comparable to FMIPv6 on downlink and outperforms it by 26.8 % on uplink. Compared to the pre-registration method, MIFAv4 is 52.1 % better on downlink and 59.2 % better on uplink.

MIFAv6 and FMIPv6 are comparable in the third case. Dropping of downlink data packets can be eliminated. This is because the *RNode* in both protocols is the old MA, which forwards data packets to the new MA upon the appearance of the L2-LD trigger. Notice that the new MA should buffer the packets until the MN completes the layer 2 handoff. Regarding uplink traffic, dropping due to the layer 3 handoff can be approximately eliminated¹. Clearly, some packets will get lost during the layer 2 handoff. Because the pre-registration method does not minimize the layer 3 handoff latency to that resulting from the layer 2 handoff, there are some dropped packets on downlink due to the layer 3 handoff. For the same reason, the pre-registration method drops on uplink more than the other studied protocols. In the performed analysis, it drops on uplink 42.7 % more than MIFAv4.

5.6.5.2.3. Impact of MN Speed Inside the Overlapping Area

Figure 5.10 presents the average handoff latency resulting from the studied make-before-break mobility management protocols as a function of the speed of the MN under the assumption that the *update message* sent from the MN on the old wireless link has not been dropped. The analysis assumes that the length of the MN's path inside the overlapping area is 10 meters and $(t_g - t_{in})$ is 300 msec. As known, this time represents the time required to fire the L2-trigger.

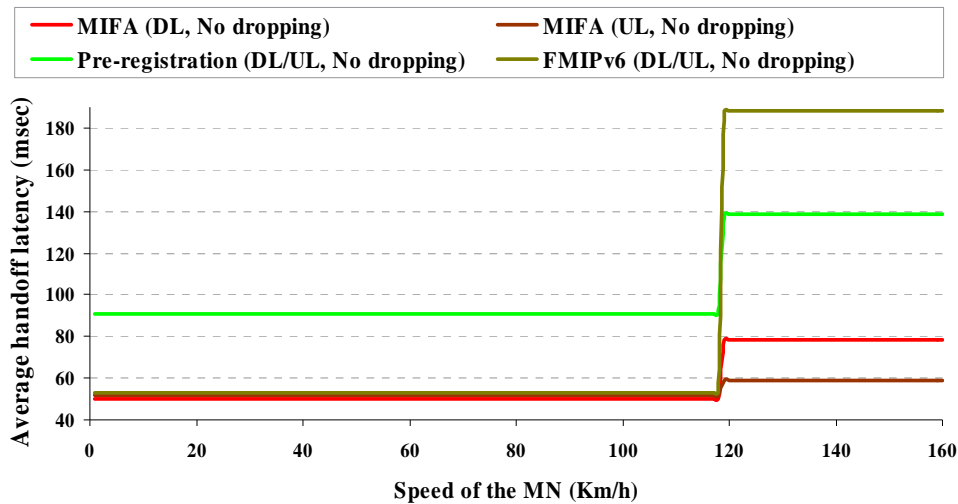


Fig 5.10: Average handoff latency as a function of the speed of the MN employing MIFA in predictive mode, the pre-registration method and FMIPv6 (the *update message* has not been dropped on the old wireless link)

With respect to the average handoff latency on downlink, this figure shows that if $(t_0 - t_g)$ is long enough to start the layer 3 handoff in advance, MIFA in predictive mode and FMIPv6 are comparable. The average handoff latency is minimized to the latency resulting from the layer 2 handoff. Moreover, the MN can move to 118 km/h and is still able to initiate the layer 3 handoff before breaking the old wireless link. If the MN moves faster, the layer 3 handoff can no longer be triggered in advance. In such situations, MIFA remains capable of achieving fast handoffs. In contrast, FMIPv6 will be employed in reactive mode or even resorts to

¹ The MN normally waits for a message (e.g. reply, advertisement, etc.) from the new MA after the layer 2 handoff. The receipt of this message takes some time, which may produce data loss.

MIPv6. As mentioned while discussing figure 5.8, even if the pre-registration method could trigger the layer 3 handoff in advance, the handoff latency is not minimized to that resulting from the layer 2 handoff. Nevertheless, the MN is also capable of utilizing the L2-trigger up to a max speed of 118 km/h.

The main result obtained from this figure is that MIFA in predictive mode is able to guarantee a fast handoff even for high speeds (more than 118 km/h). The same is observed on uplink. An additional advantage here is that the impact of increasing the speed is not even especially noticeable.

Let us now consider the dropping of the *update message* on the old wireless link, see figure 5.11.

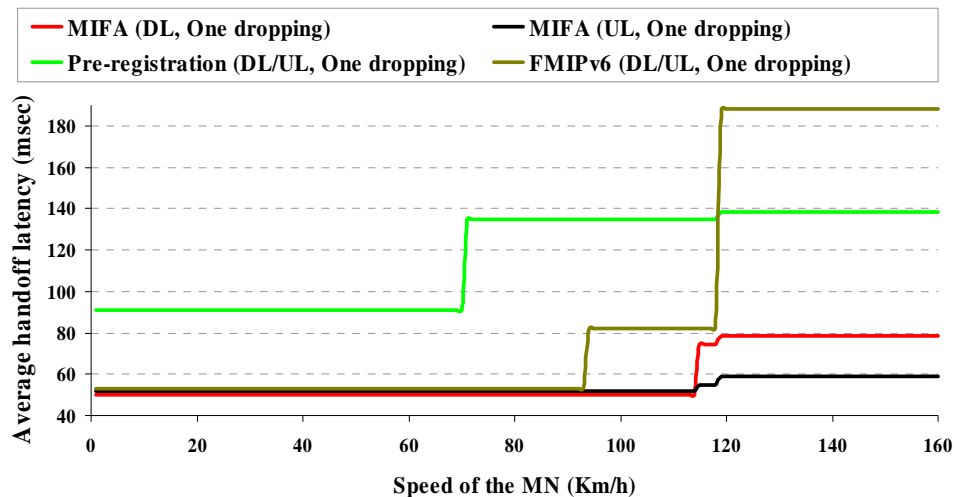


Fig 5.11: Average handoff latency as a function of the speed of the MN employing MIFA in predictive mode, the pre-registration method and FMIPv6 (the *update message* has been dropped once on the old wireless link)

Notice that MIFA in predictive mode can recover the dropping of the *update message* very quickly, which significantly improves its robustness. The MN can move at 114 km/h without suffering from any additional latency resulting from the dropping of the *update message*. Although the layer 3 handoff can not be triggered in advance for speeds in excess of 114 km/h, MIFA remains capable of guaranteeing a fast handoff on downlink and uplink. Regarding FMIPv6, there is a noticeable impact of *update message* dropping on its performance. The MN must move at a maximum speed of 93 km/h to be able to recover the dropping. For speeds greater than 93 km/h and less than 119 km/h, FMIPv6 is employed in reactive mode and, therefore, demonstrates acceptable performance. For speeds greater than 118 km/h, FMIPv6 can operate neither in predictive nor in reactive mode and instead resorts to MIPv6. Considering the pre-registration method, the dropping of the *update message* can be recovered for speeds less than 71 km/h. If the MN moves faster, the pre-registration method resorts to MIPv4.

Figure 5.12 presents the expected average number of dropped packets per handoff on downlink experienced when employing the studied make-before-break solutions as a function of MN speed. This figure shows the impact of MN speed in case of no and one dropping of the *update message*.

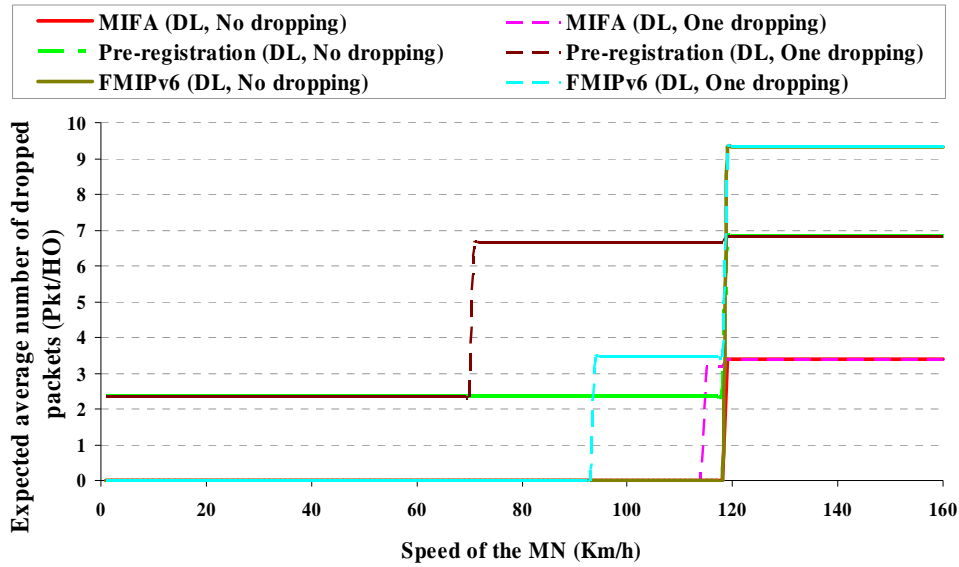


Fig 5.12: Expected average number of dropped packets per handoff on downlink as a function of the speed of the MN employing MIFA in predictive mode, the pre-registration method and FMIPv6

The results obtained from this figure are similar to those derived from figures 5.10 and 5.11. In the case of not dropping of the *update message*, MIFA in predictive mode and FMIPv6 are comparable. Up to a speed of 118 km/h, lossless handoffs can be achieved. The maximum speed can be reached when employing the pre-registration method without missing the L2-trigger is 118 km/h as well. However, lossless handoffs on downlink can be achieved neither for low nor for high speeds. In case of one dropping of the *update message*, the MN can move at a maximum speed of 114 km/h when employing MIFA in predictive mode without suffering data packet loss. When employing FMIPv6, the MN has to move at a maximum speed of 93 km/h to be able to recover the dropping of the *update message*. For the pre-registration method, the MN should not move faster than 70 km/h to be able to recover the dropping of the *update message*.

Similar results can be derived from figure 5.13, which presents the expected average number of dropped packets per handoff on uplink when employing the studied protocols as a function of the MN speed with no and one dropping of the *update message* on the old wireless link.

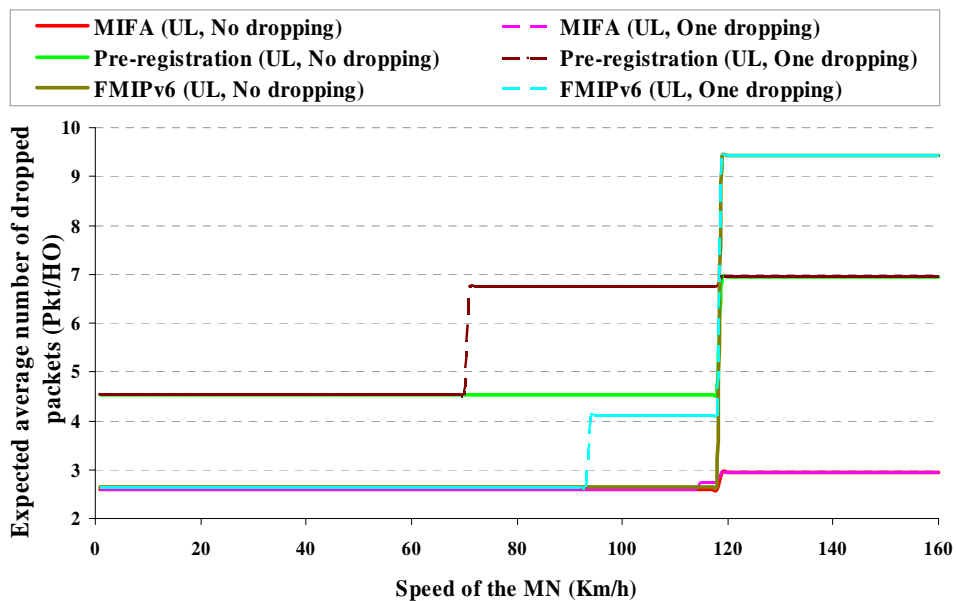


Fig 5.13: Expected average number of dropped packets per handoff on uplink as a function of the speed of the MN employing MIFA in predictive mode, the pre-registration method and FMIPv6

In conclusion, MIFA performs best in predictive mode. It is able to guarantee seamless handoffs at low as well as high speeds even when the *update message* could not be sent successfully after the appearance of the L2-trigger or even when the L2-trigger could not be raised at all. In other words, the negative impact of layer 2 triggers timing is minimized.

5.6.6. Cost Estimation

This section estimates the cost resulting from employing MIFA and compares it to the cost resulting from the other discussed protocols.

5.6.6.1. Location Update Cost

Figure 5.14 shows the location update cost experienced when employing the studied break-before-make mobility management protocols as a function of the residence time (T_r). As shown in this figure, the location update cost as a function of the residence time has a negative exponential distribution. Increasing the residence time reduces the location update cost resulting from all mentioned mobility management protocols. Proxy MIPv6 produces the minimum location update cost. This is because Proxy MIPv6 does not involve MNs in the handoff procedure and, therefore, reduces the amount of signaling on the wireless link. MIPRR produces more location update cost than Proxy MIPv6 and less than others. This is due to the localization of mobility processing inside the MIPRR domain. MIPv4 produces more location update cost than both MIPRR and Proxy MIPv6. HAWAII generates more location update cost than the three mentioned protocols. The reason for this is the processing of control messages in *InNodes*. In spite of this, the location update cost resulting from HAWAII is less than that resulting from MIPv6. The reason is that MIPv6 involves the CN in the handoff and requires execution of the DAD procedure. MIFA is outperformed by MIPv4, MIPv6, MIPRR, HAWAII and Proxy MIPv6. This is mainly due to the distribution of the MN-specific data to all FAs/ARs of the current L3-FHR. In addition, not only the HA is notified after the handoff but also the old FA and possibly the CN as well. Although MIFAv4 and MIFAv6 perform comparably with respect to the handoff latency and number of dropped packets, the location update cost resulting from them is significantly different. This is because MIFAv6 involves also the CN in the handoff and requires execution of the DAD procedure.

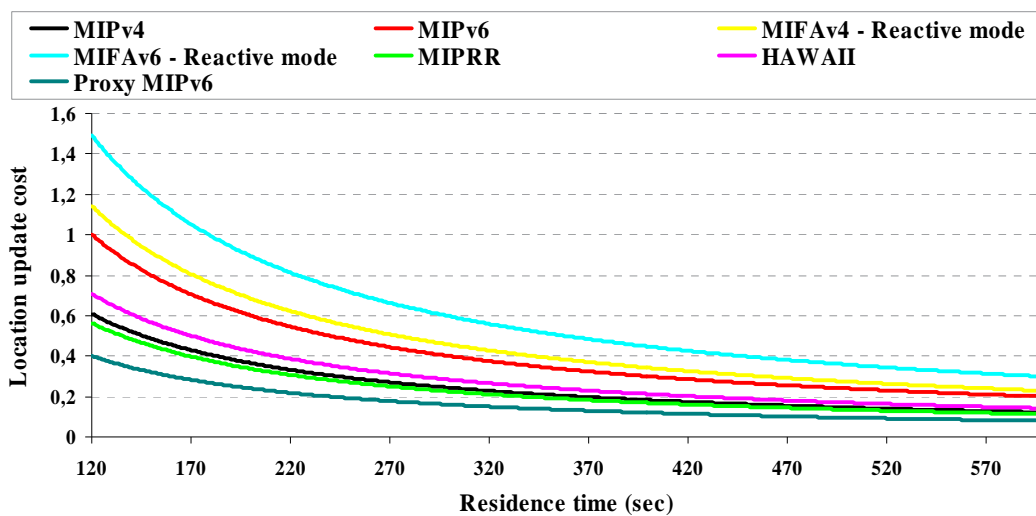


Fig 5.14: Location update cost resulting from employing MIFAv4 in reactive mode, MIFAv6 in reactive mode, MIPv4, MIPv6, MIPRR, HAWAII and Proxy MIPv6

The location update cost experienced when employing the studied make-before-break mobility management protocols is shown in figure 5.15 as a function of the residence time (T_r).

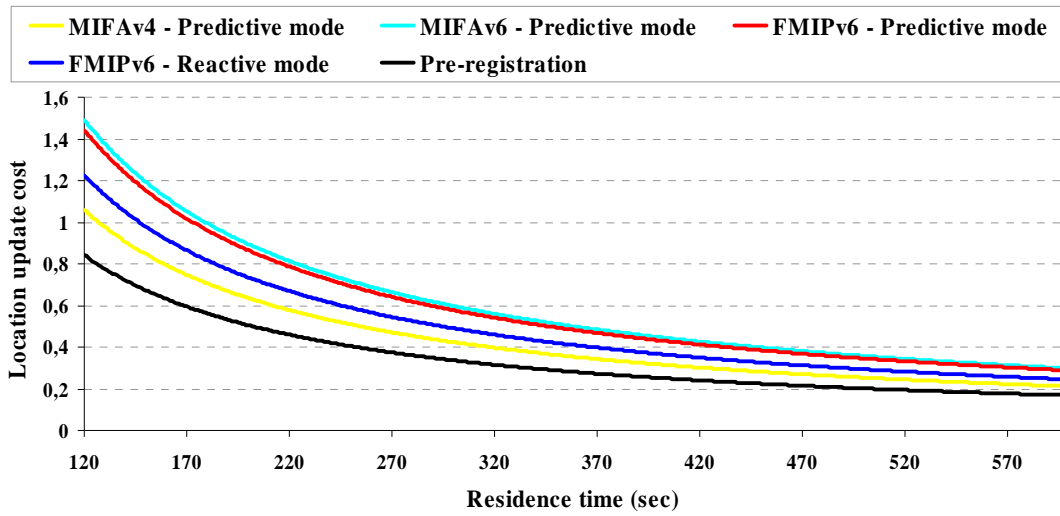


Fig 5.15: Location update cost resulting from employing MIFAv4 in predictive mode, MIFAv6 in predictive mode, the pre-registration method and FMIPv6

This figure shows that the minimum location update cost is produced by the pre-registration method. This is because this method informs the HA only by passing the old and new FA. Because of registration with both the new FA and the HA in addition to the distribution of the MN-specific data to all FAs present in the current L3-FHR, more signaling is generated by MIFAv4 than by the pre-registration method. MIFAv6 and FMIPv6 in predictive mode are comparable in terms of the generated location update cost. In contrast, FMIPv6 in reactive mode produces less signaling cost than MIFAv6 in predictive mode.

5.6.6.2. Packet Delivery Cost

Figure 5.16 shows the packet delivery cost resulting from employing the studied break-before-make mobility management protocols at different packet arrival rates.

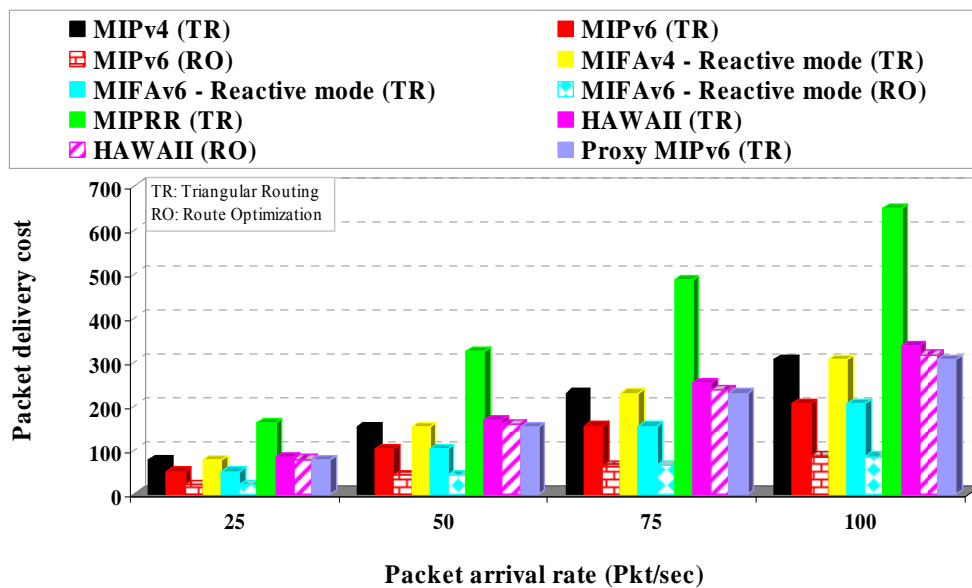


Fig 5.16: Packet delivery cost resulting from employing MIFAv4 in reactive mode, MIFAv6 in reactive mode, MIPv4, MIPv6, MIPRR, HAWAII and Proxy MIPv6

Clearly, the packet delivery cost goes up by increasing the packet arrival rate. MIFA performs comparable to MIP. The reason behind this comparable performance is that MIFA forwards data packets on the same path as MIP. Although MIFA forwards the packets in-flight from the old to the new MA during the handoff, the cost resulting from this forwarding per time unit is negligible compared to the cost resulting from the session itself. Although MIFAv6 produces more location update cost than MIFAv4, it generates less packet delivery cost. The reason for this is the processing cost required in the MAs supporting MIFAv4, which is not necessary in the MAs supporting MIFAv6. Notice that there is additional processing in the MAs operating MIFAv6 when forwarding the packets in-flight from the old MA to the MN via the new MA. However, as mentioned above, this cost is negligible compared to the cost resulting from the whole session.

The worst cost results from MIPRR. This is due to the use of intermediate nodes (the GW and MRs) to forward data packets to the MN. MIFAv4 results in the same packet delivery cost as Proxy MIPv6. This is because the path used in the two protocols to forward data packets is the same, i.e. from the CN to the MN via the HA and the current MA. Due to the fact that MIFAv6 allows the MN be the end point of the tunnel instead of the current MA, the packet delivery cost resulting from MIFAv6 is less than that produced by MIFAv4 and Proxy MIPv6. Employing HAWAII results in more packet delivery cost than that resulting from MIFA and Proxy MIPv6. The reason for this is that HAWAII introduces an intermediate node (the GW) to control the domain.

Figure 5.17 presents the packet delivery cost resulting from employing the studied make-before-break mobility management protocols. The figure shows that MIFAv4 in predictive mode produces the same packet delivery cost as the pre-registration method. This is due to the use of the same path to forward data packets, i.e. from the CN to the MN via the HA and new MA. Regardless of whether the triangular routing or route optimization is used by MIFAv6 in predictive mode and by FMIPv6, the resulting packet delivery cost from both is the same.

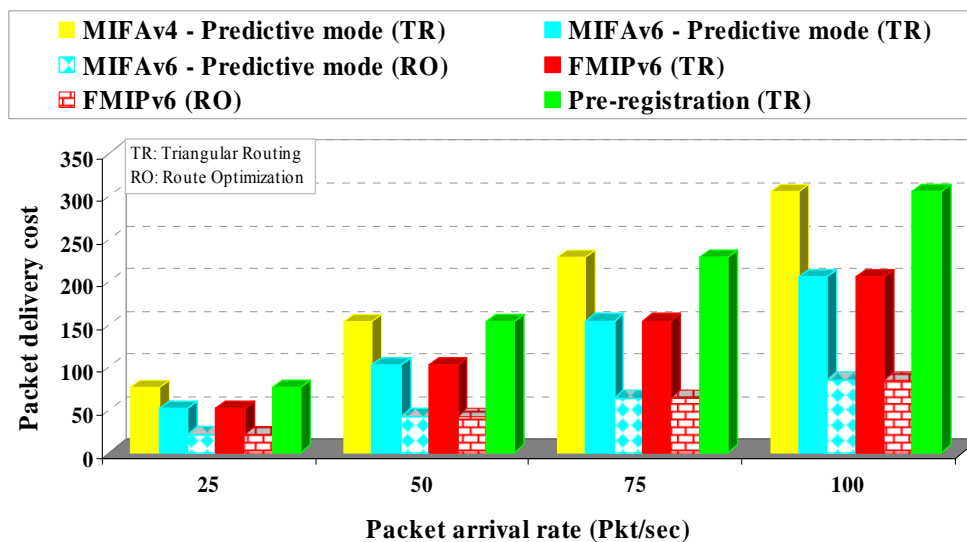


Fig 5.17: Packet delivery cost resulting from employing MIFAv4 in predictive mode, MIFAv6 in predictive mode, the pre-registration method and FMIPv6

5.7. Impact of Mobility Scenarios

This section studies how the mobility of users affects the performance and the cost of mobility management protocols. For this purpose, a hierarchical topology is used. The parameters of this topology is the same as the hierarchical topology used in the previous analysis, see

section 5.6.1. MIPv6, MIFAv6 in reactive mode, Proxy MIPv6 and HAWAII are studied. Mobility scenarios are changed, so that the term R changes from 0 to 1. Notice that the speed of MNs is not studied in this analysis. The point of interest here is the identification of the paths the MNs follow inside the access network and not at which speed they move. In order to evaluate the performance, the average number of dropped packets on downlink as well as on uplink has been calculated. Regarding the cost resulting from these mobility protocols, we focus on the location update cost per time unit.

5.7.1. *Application of the Generic Mathematical Model to Mobility Management Protocols*

5.7.1.1. *MIPv6*

Let us now consider the triangular routing only in this analysis. The HA is the $BUnode$. The parameters presented in section 5.6.3.2 remain the same. The difference is only in vector B , which will be $[0 \ 0 \ R+G \ 0 \ 0 \ 0]$.

5.7.1.2. *MIFAv6 in Reactive Mode*

Once more, we focus only on the triangular routing. Considering the performance, the HA is the $BUnode$. B will be $[0 \ 0 \ 0 \ R \ G \ 0]$. The handoff vector is $[0 \ 0 \ 0 \ T_{MA-MR} \ T_{MA-GW} \ 0]$. The other parameters presented in section 5.6.3.4 remain unchanged. The parameters required to calculate T_{MA-GW} are the same as in T_{MA-MR} , see section 5.6.3.4. Considering the cost resulting from employing MIFAv6, the parameters presented in section 5.6.3.4 remain the same as well. The difference is only in the B and LUC vectors, which will be $[0 \ 0 \ R+G \ R \ G \ R+G]$ and $[0 \ 0 \ luc_{HA} \ luc_{MA-MR} \ luc_{MA-GW} \ luc_{ANP}]$, respectively. The parameters required to calculate luc_{MA-GW} are the same as in luc_{MA-MR} .

5.7.1.3. *Proxy MIPv6*

The parameters presented in section 5.6.3.7 stay unchanged except vector B , which will be $[0 \ 0 \ R+G \ 0 \ 0 \ 0]$.

5.7.1.4. *HAWAII*

There are no changes regarding the parameters required to evaluate the performance and to estimate the cost of HAWAII, see section 5.6.3.6.

5.7.2. *Performance Evaluation*

Figure 5.18 shows the average number of dropped packets per handoff on downlink resulting from employing the above mentioned mobility management protocols under different mobility scenarios. The first result that can be derived from this figure is that MIPv6 and Proxy MIPv6 are not affected by mobility scenarios. The reason behind this behavior is that the MN must always update its mobility binding at the HA. The path between the new MA and the $BUnode$ in the assumed topology is, therefore, always the same regardless of the values of R and G . MIFAv6 and HAWAII are highly affected by mobility scenarios. If the studied MN always moves between MAs connected to different MRs (R equals 0), MIFAv6 performs better than HAWAII. The reason for this is the extra processing delay required in

the *InNodes* present on the path between the old and the new MA when employing HAWAII. When the probability that the crossover router will be a MR begins to increase, the average number of dropped packets per handoff on downlink when employing both protocols starts to decrease accordingly. In addition, the difference between the average number of dropped packets employing HAWAII and MIFAv6 decreases as well. According to the analytical results, this difference is 12.2 %, 11 %, 9.7 %, 8.3 % and 6.7 % when R equals 0, 0.2, 0.4, 0.6 and 0.8, respectively. If the studied MN only moves between MAs controlled by the same MR (R equals 1), MIFAv6 performs slightly better than HAWAII. According to the achieved results, HAWAII drops only 4.8 % more than MIFAv6.

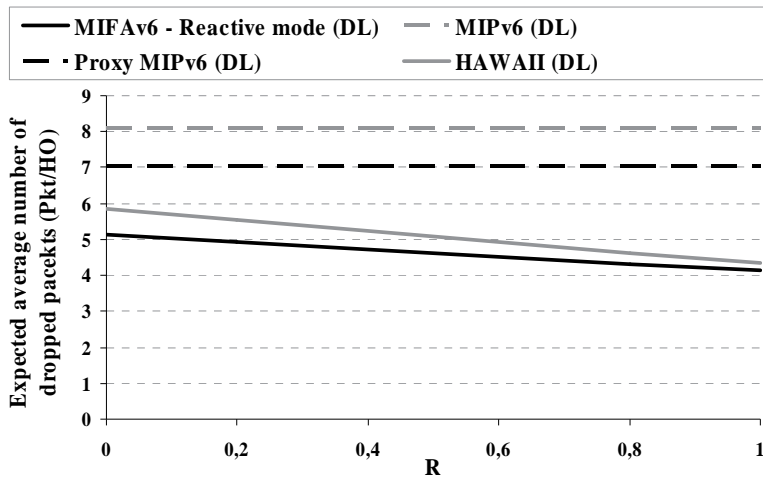


Fig 5.18: Expected average number of dropped packets per handoff on downlink resulting from employing MIFAv6 in reactive mode, MIPv6, Proxy MIPv6 and HAWAII under different mobility scenarios

Let us now address the average number of dropped packets per handoff on uplink, see figure 5.19. Once more and for the same reason mentioned above, MIPv6 and Proxy MIPv6 are not affected. MIFAv6 is also not affected by mobility scenarios. This is, of course, because the MN updates its mobility binding only at the new MA. Regarding HAWAII, it is highly affected by the used mobility scenario. The impact of mobility scenarios on the average number of dropped packets per handoff on uplink is similar as on downlink. Increasing the value of R decreases the average number of dropped packets per handoff on uplink. The best case is seen when R increases to be 1.

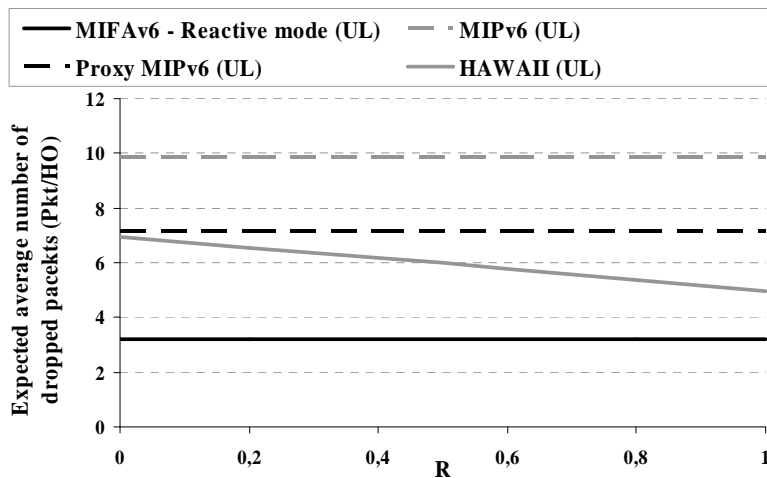


Fig 5.19: Expected average number of dropped packets per handoff on uplink resulting from employing MIFAv6 in reactive mode, MIPv6, Proxy MIPv6 and HAWAII under different mobility scenarios

5.7.3. Cost Estimation

Figure 5.20 shows the location update cost per time unit experienced when employing MIFAv6 in reactive mode, MIPv6, Proxy MIPv6 and HAWAII in the studied topology under different mobility scenarios. The average residence time in the range of each subnet is 600 msec in this analysis.

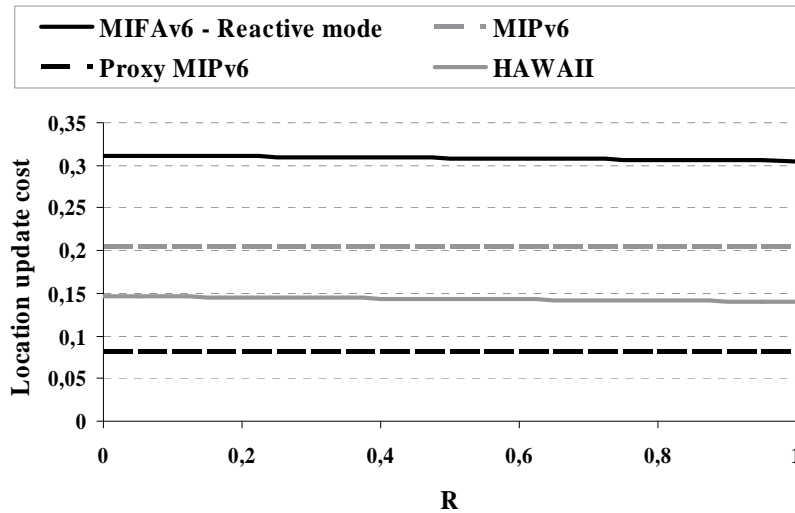


Fig 5.20: Location update cost resulting from employing MIFAv6 in reactive mode, MIPv6, Proxy MIPv6 and HAWAII under different mobility scenarios

Considering MIPv6 and Proxy MIPv6, the MN updates its binding at the HA employing both protocols. The distance between any MA in the assumed domain and the HA is always the same regardless of the applied mobility scenario. Therefore, the applied mobility scenario does not affect the location update cost resulting from these protocols. Let us now consider MIFAv6 in reactive mode and HAWAII. Both protocols update their mobility binding at the old MA. Clearly, they will be affected by the applied mobility scenario. Increasing the value of R slightly decreases the location update cost. For MIFAv6 in reactive mode, the difference between the location update cost when R is 0 and 1 is 2.18 %. Regarding HAWAII, this difference increases to 4.76 %. This means that the impact of mobility scenarios on MIFAv6 in reactive mode is smaller than on HAWAII. The reason for this is that HAWAII updates the binding at the old MA only, while MIFAv6 in reactive mode updates the mobility binding at the old MA, HA and CN. Updating the mobility binding at the HA as well as at the CN is not affected by applied mobility scenarios.

5.8. Performance vs. Cost

Before a network provider decides to employ a mobility management protocol in his access network, protocol performance should be analyzed taking the cost resulting from the protocol into account. The following question should be asked: what performance is gained and at which cost? To achieve such analysis, performance as well as cost metrics should be defined accurately.

Let us assume a performance vector Mx_{per} containing the performance metrics of protocol x . The vector Mx_{per} is defined as follows: $Mx_{per} = [f_{x1} \ f_{x2} \ \dots \ f_{xh}]$, where h is the number of performance metrics. Considering the analysis achieved in this chapter, h is 4. f_{x1} , f_{x2} , f_{x3} and f_{x4} stand for the average handoff latency on downlink, average handoff latency on uplink, average number of dropped packets per handoff on downlink and average number of dropped packets per handoff on uplink, respectively. x can be any mobility

management protocol, e.g. MIPv6, Proxy MIPv6, etc. Let us consider two example mobility protocols, A and B . The performance vectors of A and B are $Ma_{per} = [f_{a1} \ f_{a2} \ \dots \ f_{ah}]$ and $Mb_{per} = [f_{b1} \ f_{b2} \ \dots \ f_{bh}]$, respectively. Next, the gain of protocol A compared to B can be defined in a performance gain vector $G_{per} = [gp_1 \ gp_2 \ \dots \ gp_h]$, where $gp_i = 100 - (f_{ai}/f_{bi}) * 100$. To express the importance of performance metrics, another vector referred to as $F_{per} = [fw_1 \ fw_2 \ \dots \ fw_h]$ must be defined, where fw_i expresses the importance of the performance metric f_{xi} . As a result, the performance gain resulting from mobility protocol A compared to B can be calculated as follows.

$$Gain_{per} = F_{per} * G_{per}^{-1} \quad (41)$$

In a similar way, the cost metrics and their importance are determined. Let us define a new vector Mx_{cost} containing the cost metrics of protocol x . $Mx_{cost} = [c_{x1} \ c_{x2} \ \dots \ c_{xy}]$, where y is the number of cost metrics. Regarding the analysis achieved in this chapter, y is equal to 2. c_{x1} and c_{x2} stand for the location update and packet delivery cost per time unit. To consider the cost gain of mobility protocol A compared to B , a cost gain vector $G_{cost} = [gc_1 \ gc_2 \ \dots \ gc_y]$ is defined, where $gc_i = 100 - (c_{ai}/c_{bi}) * 100$. In order to express the importance of the cost metrics, another vector named $F_{cost} = [cw_1 \ cw_2 \ \dots \ cw_y]$ expressing the importance of cost metrics is defined, where cw_i stands for the importance of the cost metric c_{xi} . The cost gain resulting from employing mobility protocol A compared to B can then be calculated using equation (42).

$$Gain_{cost} = F_{cost} * G_{cost}^{-1} \quad (42)$$

Let us assume a network provider operating an access network structured as in figure 5.1 with the parameters provided in section 5.6.1. Let us further assume that this network provider operates MIPv6 in his backbone and wants to update to MIFAv6 in reactive mode. An analysis showing the performance gain to be obtained as well as the cost to be considered is of major interest. For this purpose, let us use the performance and cost metrics discussed above and assume that F_{per} is $[0 \ 0 \ 0.75 \ 0.25]$. The reason behind the selection of these values is that the user normally receives on downlink more than what it sends on uplink. Regarding the cost, let the vector F_{cost} be defined as $[0.75 \ 0.25]$, which means that the location update cost is more important than the packet delivery cost.

Figure 5.21 presents the performance and cost gains resulting from employing MIFAv6 in reactive mode instead of MIPv6 in the studied access network under different mobility scenarios.

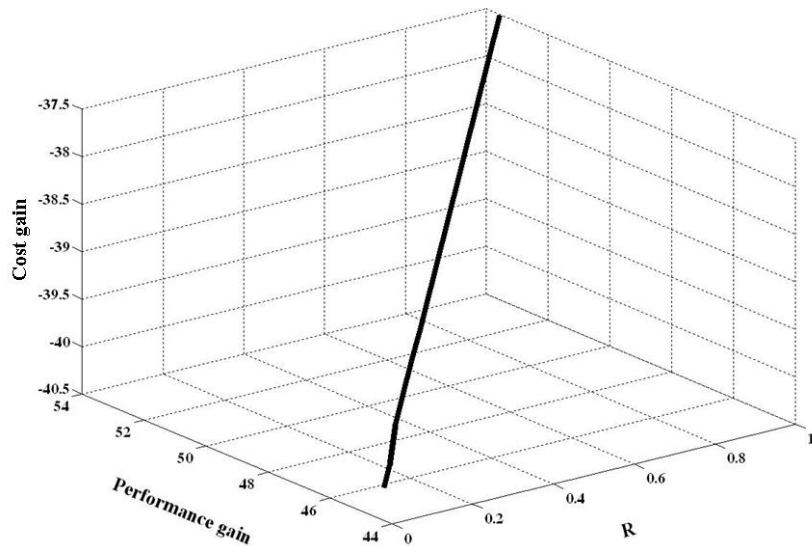


Fig 5.21: Performance and cost gain resulting from employing MIFAv6 in reactive mode instead of MIPv6 in the studied topology under different mobility scenarios

This figure shows that increasing the value of R increases the performance gain while decreasing the cost. The optimal case will be obtained when R is equal to 1, where MIFAv6 produces 53.5 % performance gain. However, this also results in 37.5 % more cost. The worst case is when R is equal to 0. The performance gain does not exceed 44.2 %, while the cost produced by MIFAv6 in reactive mode will be 40.2 % more. Of course, the network operator may not be able to control users' mobility scenarios. He normally has values of R changed in a certain range. Therefore, he should consider the performance gained and the related cost within this range. This may affect his business model.

Let us now conduct the analysis once more and assume that the network provider has Proxy MIPv6 employed in his access network and would like to upgrade to MIFAv6 in reactive mode, see figure 5.22.

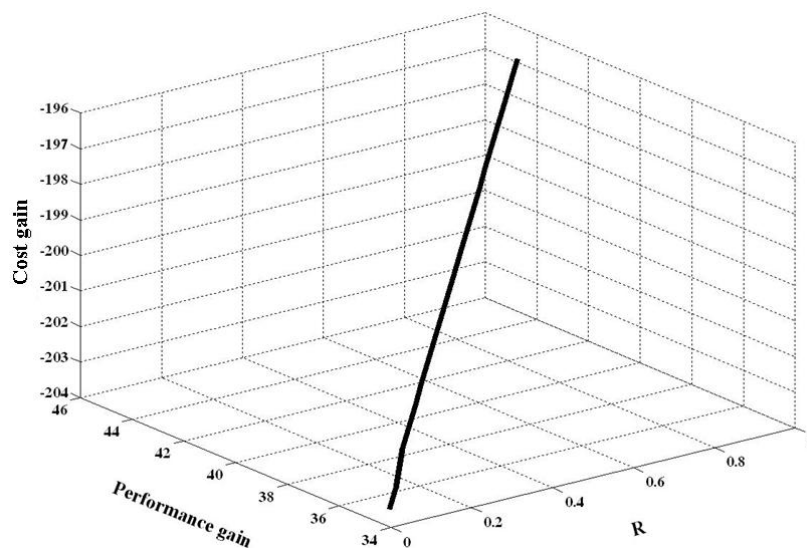


Fig 5.22: Performance and cost gain resulting from employing MIFAv6 in reactive mode instead of Proxy MIPv6 in the studied topology under different mobility scenarios

Similar results are derived from this figure as when upgrading to MIFAv6 in reactive mode from MIPv6. Increasing R increases the performance gain and decreases the cost. The optimal case is obtained when R is equal to 1, while the worst case is seen when R is equal to 0, an additional cost of 203.5 % is related to a performance gain of only 34 %.

Let us now assume that the network provider has HAWAII employed in his network and would like to upgrade to MIFAv6 in reactive mode, see figure 5.23.

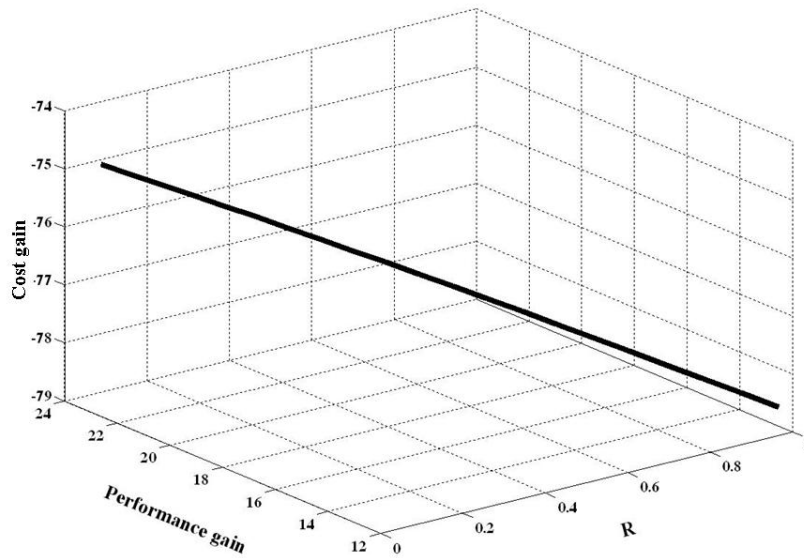


Fig 5.23: Performance and cost gain resulting from employing MIFAv6 in reactive mode instead of HAWAII in the studied topology under different mobility scenarios

The results of the upgrade in this case look different. The performance gain will be better and the extra generated cost is less at small values of R than at higher ones. The best case is obtained when R equals 0, while the worst case is observed when R equals 1. The performance gain changes in a range between 12.5 and 22.6 %, while the additional cost changes in a range between 74.7 % in the best case and 78.7 % in the worst case.

The main results can be summarized as follows: the performance gain resulting from upgrading to a certain protocol should be analyzed while taking the resulting cost into account. A mobility management protocol may be faster and smoother than another one. However, the upgrade to this protocol may not be efficient for a network provider. The performance gain as well as the cost gain depends strongly on the applied network topology and mobility scenarios, which can not be altered in most cases.

5.9. Validation of the Generic Mathematical Model

This section validates the developed generic mathematical model by comparing its results to those of simulation studies and real testbeds. To compare with simulation results, the average handoff latency and average number of dropped packets per handoff on downlink as well as on uplink experienced when employing MIPv4 and MIFAv4 in reactive mode are measured using simulations by means of NS2. After that, the same is calculated using the generic mathematical model. Clearly, the same scenario is used for the simulation as well as mathematical analysis. The results are compared then to each other. In order to validate the generic mathematical model compared to results of real testbeds, the average handoff latency as well as average number of dropped packets per handoff on downlink measured using the testbeds implemented in [Fes03] is considered. The generic mathematical model is applied to the testbeds under the same assumptions assumed in [Fes03] and the same has been calculated. Following that, results of both the generic mathematical model and testbeds are compared to each other.

5.9.1. Generic Mathematical Model vs. Simulation

5.9.1.1 Applied Network Topology

Figure 5.24 presents the network topology used for the simulation of MIPv4 and MIPv4 in reactive mode.

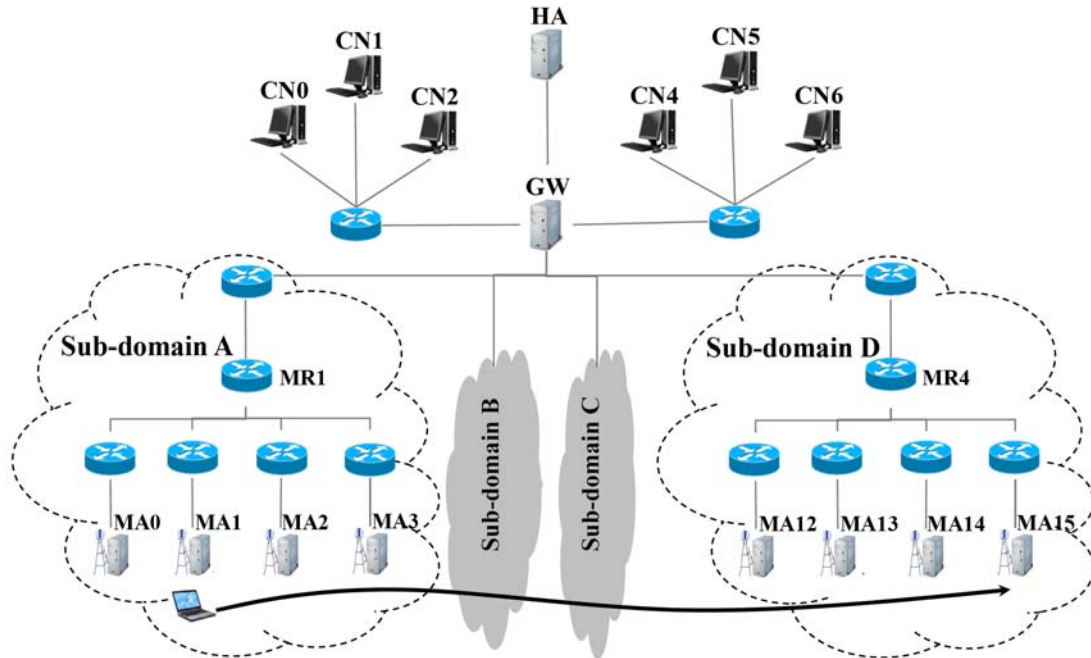


Fig 5.24: Network topology used to validate the generic mathematical model compared to simulation results

A domain of 4 sub-domains having the same structure is used. Each sub-domain contains 4 MAs. A GW interconnects the domain with other nodes outside the domain. The HA is placed outside this domain as well. The distance between each two neighbor MAs is 198 m. The cellular cells in this scenario are overlapped. The distance between the GW and each MA is 4 hops. The delay on each link between each two hops inside the domain is 5 msec. The delay between the HA and the GW is 25 msec, while the delay between the GW and CN0, CN1, CN2, CN4, CN5 and CN6 is 27, 23, 28, 27, 23 and 28 msec, respectively. All links have a bandwidth of 100 Mbit/s. During the simulation, one MN is observed. This MN moves from MA0 to MA15 at a speed of 40 km/h. A downlink as well as an uplink constant bit rate UDP stream with a packet length of 500 bytes and a packet arrival rate of 50 packets per second is exchanged between CN0 and the observed MN.

Many scenarios are considered in the achieved simulation studies. First, it is assumed that there is only one MN in the network. This simulation scenario aims at analyzing the performance under the assumption that there is sufficient bandwidth and the loss of data packets is due to the handoffs themselves and not due to network load. The second scenario aims at studying the impact of network load. To do this, 160 MNs are created in the simulated scenario. The number of active MNs is selected to be 1, 10, 30 and 60 active MNs in each scenario, respectively. The active MNs communicate with the 6 CNs. The load in the network is not changed during the simulation. The third scenario attempts to emulate the behavior of real networks. It depends on the fact that the load in real networks as well as the speed of idle and active MNs changes randomly. Therefore, 59 MNs in addition to the observed one are made active. The other 100 remain in idle mode. The active MNs communicate with the CNs. UDP traffic is used as well. All MNs, except the observed one, move randomly in the network and start sending and receiving data packets at random times and with a random packet arrival

rate for each UDP stream. In order to obtain meaningful results, each scenario is repeated 10 times.

The same network topology is used for the analysis of both protocols using the generic mathematical model. The parameters of the network topology are defined in table 5.24.

$D_{MA,MR}$	$D_{MR,GW}$	$D_{GW,HA}$	τ_1	τ_2
2 hops	2 hops	5 hops	2 msec	5 msec

Tab 5.24: Parameters of the hierarchical topology used to validate the mathematical model

5.9.1.2 *Applied Mobility Pattern*

As mentioned in the previous section, the observed MN moves from MA0 to MA15. This means that the MN executes 15 handoffs during this movement. A MR is the crossover router in 12 of these 15 handoffs. The GW is the crossover router for the remaining 3 handoffs. In other words, a MR is the crossover router for 80 % of the handoffs, while the GW is a crossover router for the remaining 20 %. Therefore, $R=0.8$, while $G=0.2$.

5.9.1.3 *Application of the Generic Mathematical Model to MIPv4*

The parameters of the mathematical model when applying to MIPv4 are detailed in section [5.6.3.1](#). However, the matrix B , γ and γ' should be changed to $[0 \ 0 \ R+G \ 0 \ 0 \ 0]$, 0 and 0, respectively. γ and γ' are equal to 0 because the MN receives an advertisement from the new MA at the defined speed while in the overlapping area, which means that the movement detection time is zero and there is no exchange of solicitation and advertisement messages between the MN and the new MA.

5.9.1.4 *Application of the Generic Mathematical Model to MIPv4 in Reactive Mode*

The parameters required to model MIPv4 in reactive mode are discussed in section [5.6.3.3](#). The same parameters are used here. However, the matrix B will be $[0 \ 0 \ 0 \ R \ G \ 0]$. In addition, γ and γ' are equal to 0 for the same reason discussed in section [5.9.1.3](#).

5.9.1.5 *Comparison to Simulation Results*

Figure 5.25 shows the average handoff latency on uplink and downlink resulting from the simulation and the mathematical analysis employing both protocols. The applied simulation scenario contains only one MN and aims, as mentioned before, at the validation of the generic mathematical model under the assumption that there is sufficient bandwidth and the loss of data packets is due to the handoffs themselves and not due to network load. This figure proves that the results of generic mathematical model are comparable to those resulting from simulation. According to the simulation results, the handoff latency achieved by MIPv4 in reactive mode is 48.6 % less on downlink and 94.2 % less on uplink than the handoff latency resulting from employing MIPv4. The developed generic mathematical model says that MIPv4 in reactive mode performs 46.8 % better than MIPv4 on downlink and 94.7 % better on uplink.

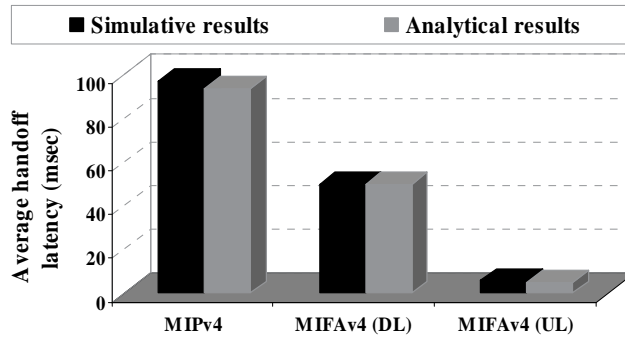


Fig 5.25: Average handoff latency resulting from the simulation and mathematical analysis of MIPv4 and MIFAv4 in reactive mode in a non-loaded network

Let us now study the average number of dropped packets per handoff experienced when employing MIPv4 and MIFAv4 in reactive mode in this scenario, see figure 5.26.

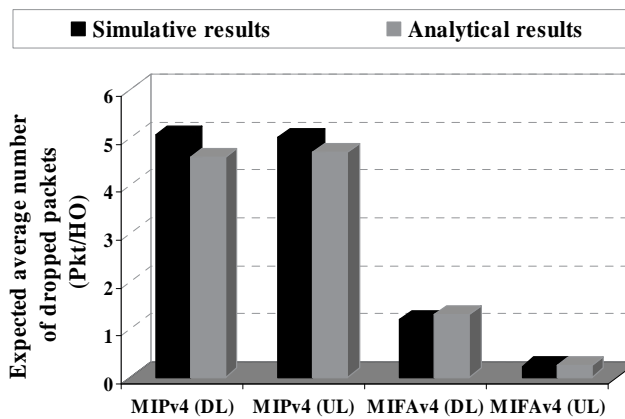


Fig 5.26: Expected average number of dropped packets per handoff on downlink and uplink resulting from the simulation and mathematical analysis of MIPv4 and MIFAv4 in reactive mode in a non-loaded network

Once again, simulation results are comparable to the generic mathematical model results. The results of simulation say that MIPv4 drops on downlink 76.3 % more than MIFAv4 in reactive mode. Regarding the average number of dropped packets per handoff on uplink, employing MIFAv4 in reactive mode results in 95.4 % fewer dropped packets. According to the results of generic mathematical model, MIFAv4 in reactive mode drops 71.7 % less on downlink and 94.7 % less on uplink than MIPv4.

In summary, the results of the generic mathematical model are comparable to those resulting from the simulation under the assumption that there is sufficient bandwidth and any dropping of data packets is due to handoffs and not due to network load.

To analyze the impact of network load, the average handoff latency and average number of dropped packets per handoff on downlink as well as uplink resulting from MIFAv4 in reactive mode and MIPv4 are evaluated under different loads and in a scenario in which the load changes randomly. The results of the simulation and generic mathematical model in addition to the generic mathematical model accuracy are provided in table 5.25 and table 5.26.

The handoff latency on downlink resulting from MIFAv4 in reactive mode compared to its counterpart resulting from MIPv4

Theoretical	Load in simulation scenarios	Simulation results	Accuracy
	1 active MN, 159 idle MNs	55.3 % less	84.65 %

46.81 % less	10 active MNs, 150 idle MNs	53.3 % less	87.82 %
	30 active MNs, 130 idle MNs	55.99 % less	83.60 %
	60 active MNs, 100 idle MNs	60.73 % less	77.08 %
	60 active MNs, 100 idle MNs. Load and speed change randomly	57.77 % less	81.02 %

The handoff latency on uplink resulting from MIFAv4 in reactive mode compared to its counterpart resulting from MIPv4

Theoretical	Load in simulation scenarios	Simulation results	Accuracy
94.68 % less	1 active MN, 159 idle MNs	92.88 % less	98.1 %
	10 active MNs, 150 idle MNs	92.21 % less	97.39 %
	30 active MNs, 130 idle MNs	89.93 % less	94.98 %
	60 active MNs, 100 idle MNs	90.5 % less	95.59 %
	60 active MNs, 100 idle MNs. Load and speed change randomly	89.77 % less	94.81 %

Tab 5.25: Accuracy of the generic mathematical model compared to simulation results with respect to the average handoff latency

Expected average number of dropped packets per handoff on downlink resulting from MIFAv4 in reactive mode compared to its counterpart resulting from MIPv4

Theoretical	Load in simulation scenarios	Simulation results	Accuracy
71.74 % less	1 active MN, 159 idle MNs	70.04 % less	97.63 %
	10 active MNs, 150 idle MNs	69.72 % less	97.18 %
	30 active MNs, 130 idle MNs	67.96 % less	94.47 %
	60 active MNs, 100 idle MNs	60.98 % less	85 %
	60 active MNs, 100 idle MNs. Load and speed change randomly	62.35 % less	86.91 %

Expected average number of dropped packets per handoff on uplink resulting from MIFAv4 in reactive mode compared to its counterpart resulting from MIPv4

Theoretical	Load in simulation scenarios	Simulation results	Accuracy
94.68 % less	1 active MN, 159 idle MNs	94.05 % less	99.33 %
	10 active MNs, 150 idle MNs	92.42 % less	97.61 %

	30 active MNs, 130 idle MNs	88.66 % less	93.64 %
	60 active MNs, 100 idle MNs	94.65 % less	99.97 %
	60 active MNs, 100 idle MNs. Load and speed change randomly	90.36 % less	95.45 %

Tab 5.26: Accuracy of the generic mathematical model compared to simulation results with respect to the expected average number of dropped packets per handoff

It can be clearly seen that the impact of network load is not linear. Simulation results may be more or less than their counterparts resulting from the generic mathematical model. The accuracy of the generic model may exceed 99 % in some measurements and may not reach 78 % in others. The main obtained result says that the accuracy of the generic model lies in a range of ± 23 % for different loads. It should be mentioned, however, that this result can not be simply generalized for all protocols. It is true for the protocols for which the load impact is somewhat similar. Further studies on the load impact are necessary. More specifically, the load should be taken into account when calculating the average handoff latency, average number of the dropped packets, location update cost and packet delivery cost. Further simulations to validate the load impact are necessary as well.

5.9.2. Generic Mathematical Model vs. Real Testbeds

5.9.2.1. Applied Network Topology

Figure 5.27 presents the testbed used to evaluate MIPv4, see [Fes03]. The testbed contains a MN, two MAs functioning as APs as well, HA, CN and two routers. One of the routers works as a Wide Area Network (WAN) emulator capable of adding delay to the link between each MA and the HA. The operation system installed on all nodes is Linux kernel 2.2.18. MIPv4 implementation used is the Dynamics MIP [DyMIP], [FMM00] version 0.6.2. The implementation is RFC 2002 [Per96b] compliant with a minimal modification to enable the MAs sending *Agnt_Adv* messages at higher frequency than once per second. Notice that the wireless links between the MN and MAs are replaced in the testbed by Ethernet links connected via a switch to the MN. Movements are emulated then by switching the MN among these links. Handoffs are advertisement-based, i.e. after the expiration of lifetime of last received *Agnt_Adv* message, the MN declares the out of range and waits for a new *Agnt_Adv* message. As a default, the advertisement lifetime is three times the advertisement sending rate. It is selected to be 300 msec in this study, while the advertisement sending rate is 10 per second. Similar testbed is used to evaluate MIPRR. However, the router connecting the two MAs with other nodes of the testbed works as a gateway, see Figure 5.28. Performance measures are the average handoff latency and average number of dropped packets per handoff on downlink. The average handoff latency resulting from MIPv4 as well as MIPRR was measured in [Fes03] as a function to the round trip time between the HA and the MN. During the experiment, the routing trip time between the MN and HA was varied by changing the delay emulated in the WAN among 0, 10, 20, 30, 40, 50 and 100 msec. To measure the lost packets per handoff, a downlink constant bit rate UDP stream with a packet length of 1024 bytes was exchanged between the CN and MN. The average number of dropped packets per handoff on downlink was measured as a function of the packet arrival rate for a routing trip time between the MN and HA equal to 100 msec.

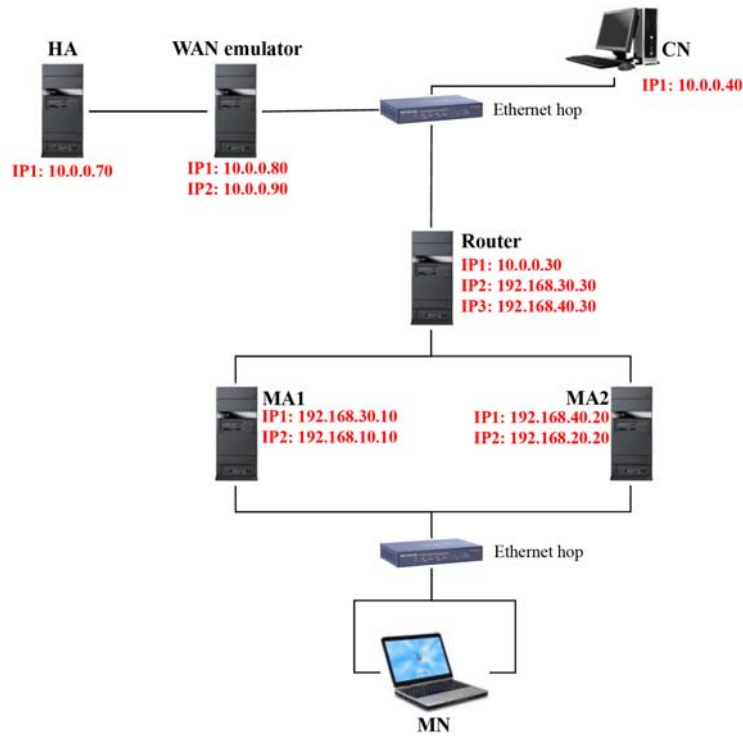


Fig 5.27: Testbed for MIPv4 used to validate the generic mathematical model

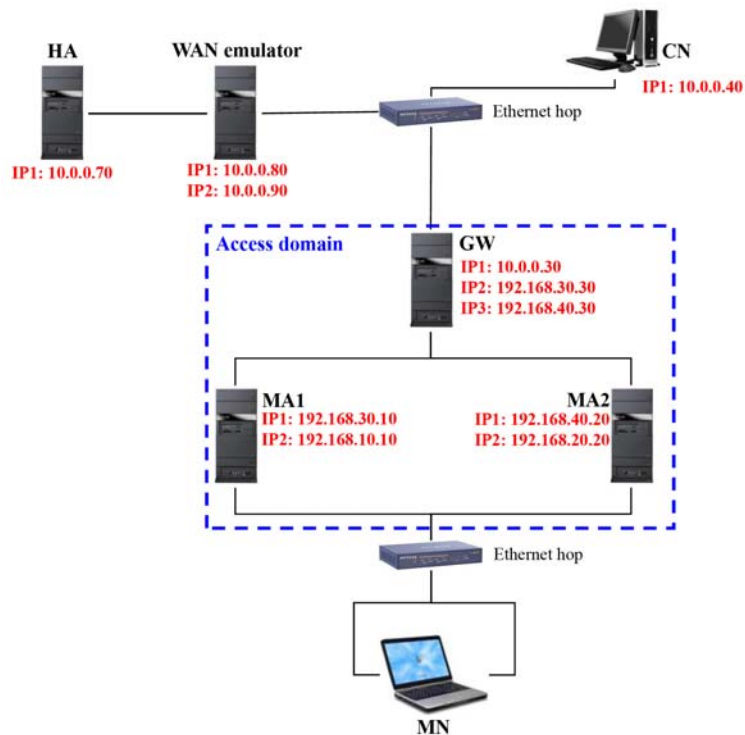


Fig 5.28: Testbed for MIPRR used to validate the generic mathematical model

To apply the generic mathematical model to the testbeds, we use the parameters shown in table 5.27. Notice that τ_1 is equal to τ_2 since Ethernet links are used for both wireless and wired links. $D_{GW,HA}$ is changed among 1, 11, 21, 31, 41, 51 and 101 hops to reflect the delay the WAN emulates in the testbeds. Notice that the WAN is used to add extra delay among 0, 10, 20, 30, 40, 50 and 100 msec to the delay between the HA and each MA. Clearly, the delay

emulated in the WAN is equal to the delay resulting from 0, 10, 20, 30, 40, 50 and 100 hops, respectively, connecting through links with transmission delay of 1 msec. Furthermore, to represent the testbeds for a WAN delay of 0 msec, we set $D_{GW,HA}$ to 1 hop, which stands for the link itself between the GW and HA. The same is for other WAN delays.

$D_{MA,MR}$	$D_{MR,GW}$	$D_{GW,HA}$	τ_1	τ_2
0 hops	1 hops	1, 11, 21, 31, 41, 51 and 101 hops	1 msec	1 msec

Tab 5.27: Parameters of the testbeds used to validate the mathematical model

5.9.2.2. *Applied Mobility Pattern*

As mentioned above, the MN switches between both MAs. There are no MRs in the domain. Thus, the GW is the crossover router for all handoffs. Therefore, $R=0$, while $G=1$.

5.9.2.3. *Application of the Generic Mathematical Model to MIPv4*

Application of the mathematical model to MIPv4 has been discussed in detail in section [5.6.3.1](#). We use the same assumptions except the matrix B , Δ , γ and γ' , which should be changed to $[0 \ 0 \ R+G \ 0 \ 0 \ 0]$, 300 msec, 4 msec and 4 msec, respectively. γ and γ' are set to 4 msec to express the delay resulting from the WAN regardless of the delay it emulates. This is because data packets as well as MIPv4 control messages should be processed first in the router implementing the WAN before passing to the WAN itself to experience the defined delay. Δ is assumed to be 300 msec, which is the advertisement lifetime selected in the testbeds. Notice that the handoff is advertisement-based. Thus, the MN should wait for the expiration of the last received advertisement lifetime before declaring an out of range.

5.9.2.4. *Application of the Generic Mathematical Model to MIPRR*

The parameters required to model MIPRR have been discussed in section [5.6.3.5](#). The same parameters are used here as well except Δ , γ and γ' , which are equal to 300, 4 and 4 msec, respectively. The reasons have been highlighted in section [5.9.2.3](#).

5.9.2.5. *Comparison to Testbeds Results*

Figure 5.29 presents the average handoff latency resulting from employing MIPv4 and MIPRR in the testbeds as well as from applying the generic mathematical model to the both protocols as a function of the delay emulated in the WAN. Notice that the results of the testbeds are presented in [\[Fes03\]](#) as a function of the round trip time measured between the MN and HA at each value of the delay emulated in the WAN. Clearly, the delay emulated in the WAN affects this round trip time. However, the round trip time between the MN and HA may vary even if the delay emulated in the WAN stays the same, e.g. due to processing delays in network nodes, transmission delays, load in network nodes, etc. Therefore, we have selected to present the results as a function of the delay emulated in the WAN instead of the round trip time to be able to apply the same topology of the testbeds to the mathematical model. The figure shows that the handoff latency resulting from the mathematical model follows the same behavior the handoff latency resulting from the testbeds follows. As the delay emulated in the WAN increases, the handoff latency resulting when employing MIPv4

increases linearly (the handoff latency resulting from the testbed can be approximated by a linear function). This is, of course, not the case for MIPRR since handoffs are controlled by the GW. Therefore, the handoff latency remains constant as a function of the delay emulated in the WAN (notice that the WAN is placed outside the access domain). Furthermore, the values of the handoff latency resulting from the generic mathematical model lie in a range of -8.9 to -16.93 % of those resulting from the testbeds.

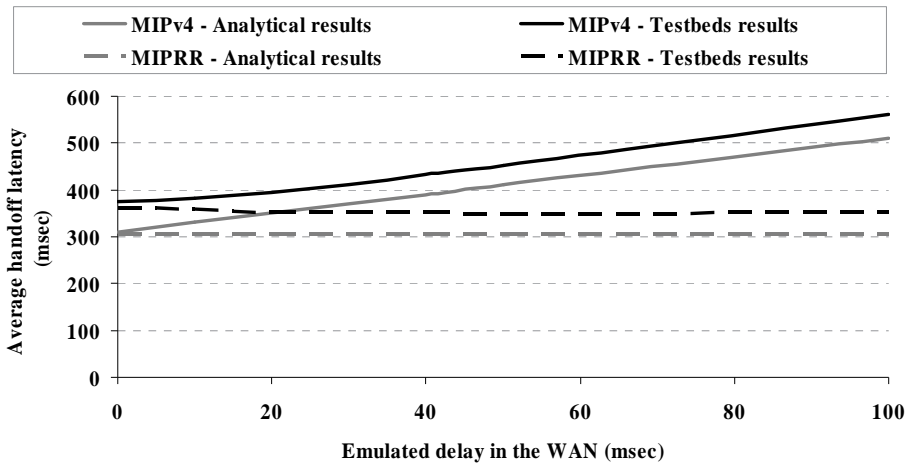


Fig 5.29: Average handoff latency resulting from the testbeds and the generic mathematical model as a function of the delay emulated in the WAN when employing MIPv4 and MIPRR

Similar results are derived from figure 5.30, which shows the average number of dropped packets per handoff on downlink resulting from the generic mathematical model and the testbeds as a function of the packet arrival rate. The delay emulated in the WAN is selected to be 20 msec in this study. The results of the generic mathematical model as well as testbeds are comparable. Moreover, they follow the same behavior, i.e. the average number of dropped packets per handoff on downlink increases linearly while increasing the packet arrival rate. Considering the values resulting from the generic mathematical model, they lie in a range of -4.31 to -29.96 of those obtained from the testbeds.

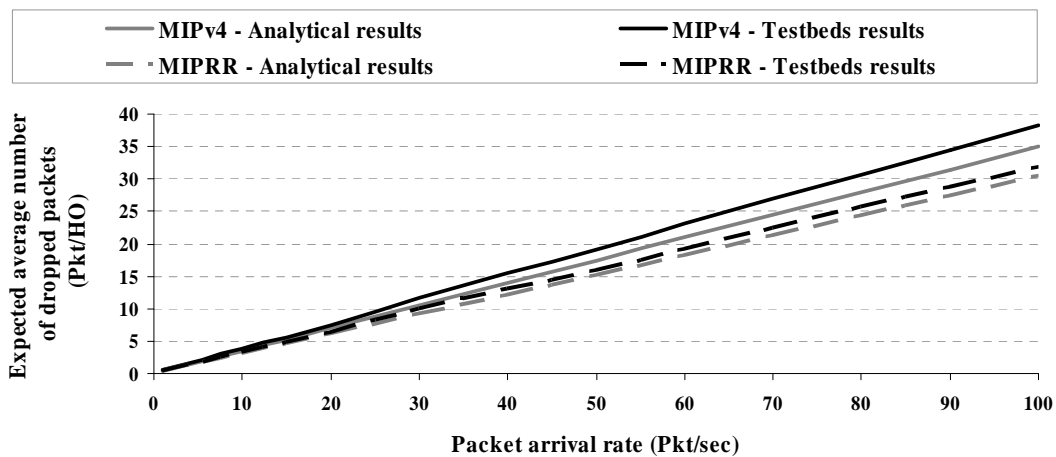


Fig 5.30: Average number of dropped packets per handoff on downlink resulting from the testbeds and the generic mathematical model as a function of the packet arrival rate when employing MIPv4 and MIPRR

Let us now compare the performance of MIPRR to that of MIPv4 by means of the generic mathematical model as well as the testbeds. Table 5.28 and table 5.29 provide the results of the generic mathematical model and testbeds in addition to the accuracy of the generic mathematical model.

Average handoff latency resulting from MIPRR compared to its counterpart resulting from MIPv4			
Delay emulated in the WAN	Theoretical	Testbeds results	Accuracy
0 msec	1.93 % less	3.5 % less	54 %
10 msec	7.86 % less	5.9 % less	70.03 %
20 msec	13.12 % less	10.85 % less	82.72 %
30 msec	17.8 % less	15.12 % less	84.95 %
40 msec	22.01 % less	18.73 % less	85.11 %
50 msec	25.81 % less	23.22 % less	89.96 %
100 msec	40.34 % less	37.29 % less	92.44 %

Tab 5.28: Accuracy of the generic mathematical model compared to testbeds results with respect to the average handoff latency

Average number of dropped packets per handoff on downlink resulting from MIPRR compared to its counterpart resulting from MIPv4			
Packet arrival rate	Theoretical	Testbeds results	Accuracy
1 Pkt/sec	13.16 % less	28.57 % less	46.05 %
10 Pkt/sec		13.16 % less	96.04 %
20 Pkt/sec		16.04 % less	82.04 %
30 Pkt/sec		15.05 % less	87.41 %
40 Pkt/sec		15.72 % less	83.69 %
50 Pkt/sec		17.08 % less	77.02 %
60 Pkt/sec		17.45 % less	75.41 %
70 Pkt/sec		17 % less	77.41 %
80 Pkt/sec		16.68 % less	78.90 %
90 Pkt/sec		16.9 % less	77.86 %
100 Pkt/sec		16.74 % less	78.62 %

Tab 5.29: Accuracy of the generic mathematical model compared to testbeds results with respect to the average number of dropped packets per handoff on downlink

It can be clearly seen that results of testbeds may be more or less than their counterparts resulting from the generic mathematical model. The accuracy of the generic model lies in a

range of ± 30 %. Notice that there are two exceptions. The first is seen by the average handoff latency when the delay emulated in the WAN is set to 0. The accuracy of the generic model is only 54 %. The reason behind this result is that the performance of MIPv4 and MIPRR in this case is comparable (MIPRR is only 1.93 % better according to the generic mathematical model and 3.5 % better according to testbeds results). Thus, the small difference in the performance of both protocols makes the difference between the results of the generic mathematical model and testbeds remarkable. The second exception is seen by the average number of dropped packets per handoff on downlink when the packet arrival rate is 1 Pkt/sec. The accuracy of the generic mathematical model is only 46.05 %. There are two reasons for this result. The first is the comparable performance of MIPv4 and MIPRR in this situation. According to the generic mathematical model, MIPv4 drops on downlink 0.35 packets per handoff, while MIPRR results in 0.30 lost packets per handoff. The results obtained from the testbeds look similar since MIPv4 and MIPRR drop only 0.5 and 0.36 packets per handoff, respectively. The second reason is the small values of the number of dropped packets per handoff, which make any small change remarkable.

5.9.3. Summary

The results obtained from this study can be summarized as follows:

1. Compared to simulation results

- a. The generic mathematical model delivers sound performance evaluation of mobility management protocols in low-loaded networks.
- b. The accuracy of the generic mathematical model remains acceptable even under high loads. According to simulation results, the accuracy of the generic model lies in a range of ± 23 % for different loads. A generalization of this result for all protocols is, however, not possible since it is true when network load has similar impact on studied protocols. Further studies to consider network load in the mathematical model are necessary and will improve the accuracy.

2. Compared to testbeds results

- a. The generic mathematical model results in an accurate performance evaluation of mobility management protocols as well.
- b. The accuracy of the generic mathematical model lies in a range of ± 30 %. To generalize this result, further studies to analyze the impact of network load are also necessary.

5.10. Graphical Tools Supporting the Generic Mathematical Model

This section briefly describes several tools developed to simplify the analysis of mobility management protocols by means of the generic mathematical model. These tools are described in more detail in appendix [F](#).

5.10.1. Mobility Scenarios Generator (MSGen)

MSGen is a powerful Graphical User Interface (GUI) written in Delphi and used to produce mobility scenarios. This tool enables one to upload geographical maps¹ to define paths for

¹ Maps are saved as photos and used as background to define the required mobility scenario.

MNs on them. The paths correspond to streets or ways for MNs. Each path can be parameterized separately, e.g. the probability of moving on the path, speeds of MNs, etc. In addition to the paths, the user can define free zones, where MNs can move freely. Figure 5.31 presents the GUI of MSGen with an example mobility scenario, see appendix F for details.

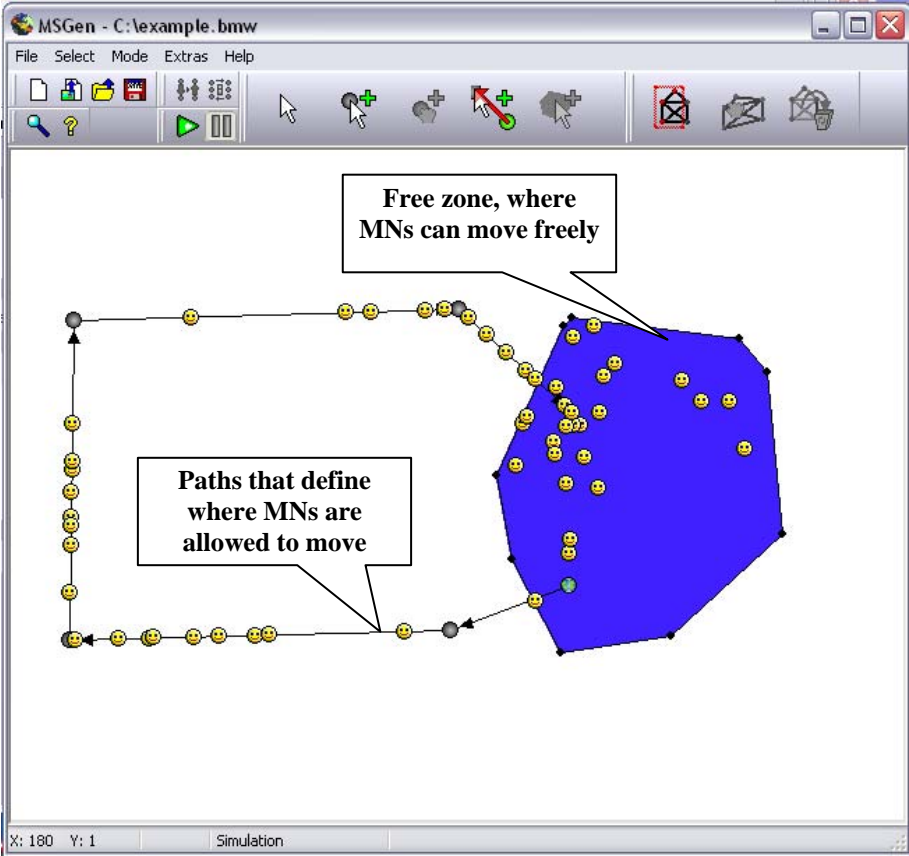


Fig 5.31: GUI of MSGen containing an example mobility scenario

5.10.2. Network Generator (NetGen)

NetGen is a GUI written in Delphi and used to generate network topologies consisting of APs, MAs, MRs, GWs, routers, CNs and MNs. The user uploads the mobility scenario generated previously using the MSGen tool. After that, the network topology is created inside the selected geographical region. Users have the ability to specify different parameters for different network elements, such as link bandwidth, transmission delay, transmission cost, etc. Moreover, users can generate the parameters of the mobility scenario and the network topology automatically. For more information, see appendix F. Figure 5.32 shows the GUI of NetGen containing an example network topology.

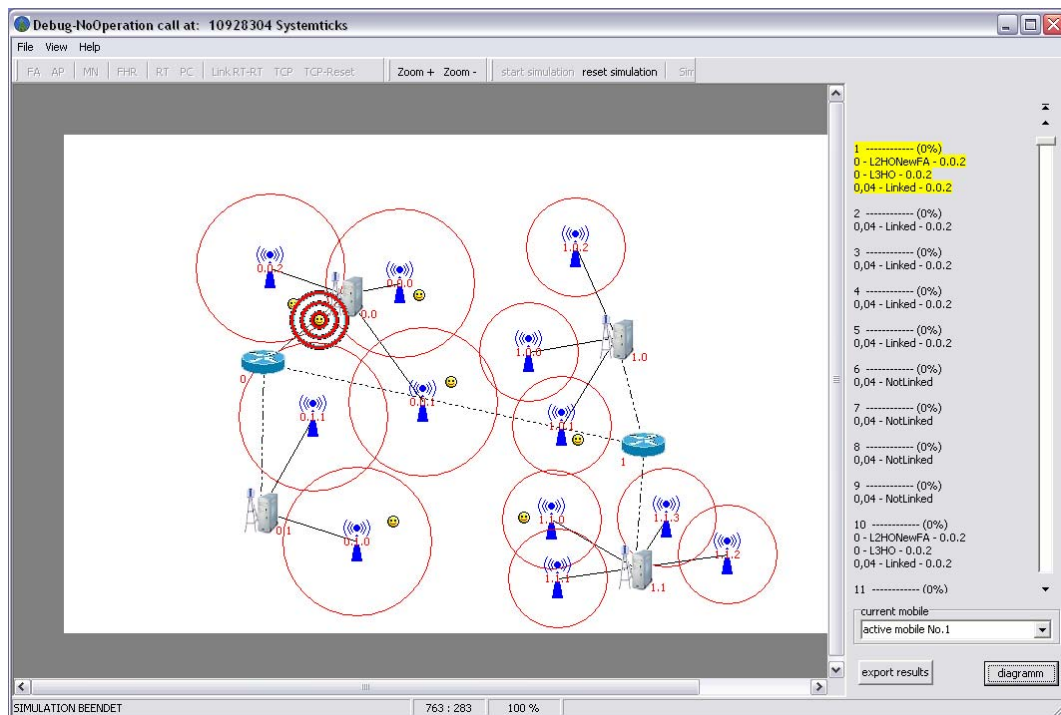


Fig 5.32: GUI of NetGen containing an example network topology

5.10.3. Protocol Designer (ProtDes)

The main idea behind ProtDes [Hei08] is to develop a graphical tool capable of parameterizing mobility management protocols by defining their message sequence charts graphically. At first, the parameters of the applied mobility scenario and network topology are imported from NetGen. Then, the user determines how control messages of the studied mobility management protocol will be exchanged between network nodes (MAs, MR, GW, HA and AP). Afterwards, ProtDes generates a list of all parameters required to analyze the studied mobility management protocol and saves it in a text file. Figure 5.33 shows the GUI of ProtDes containing an example message sequence chart of a mobility management protocol under study (MIPv4 in this example), while Figure 5.34 presents an example of a parameter file.

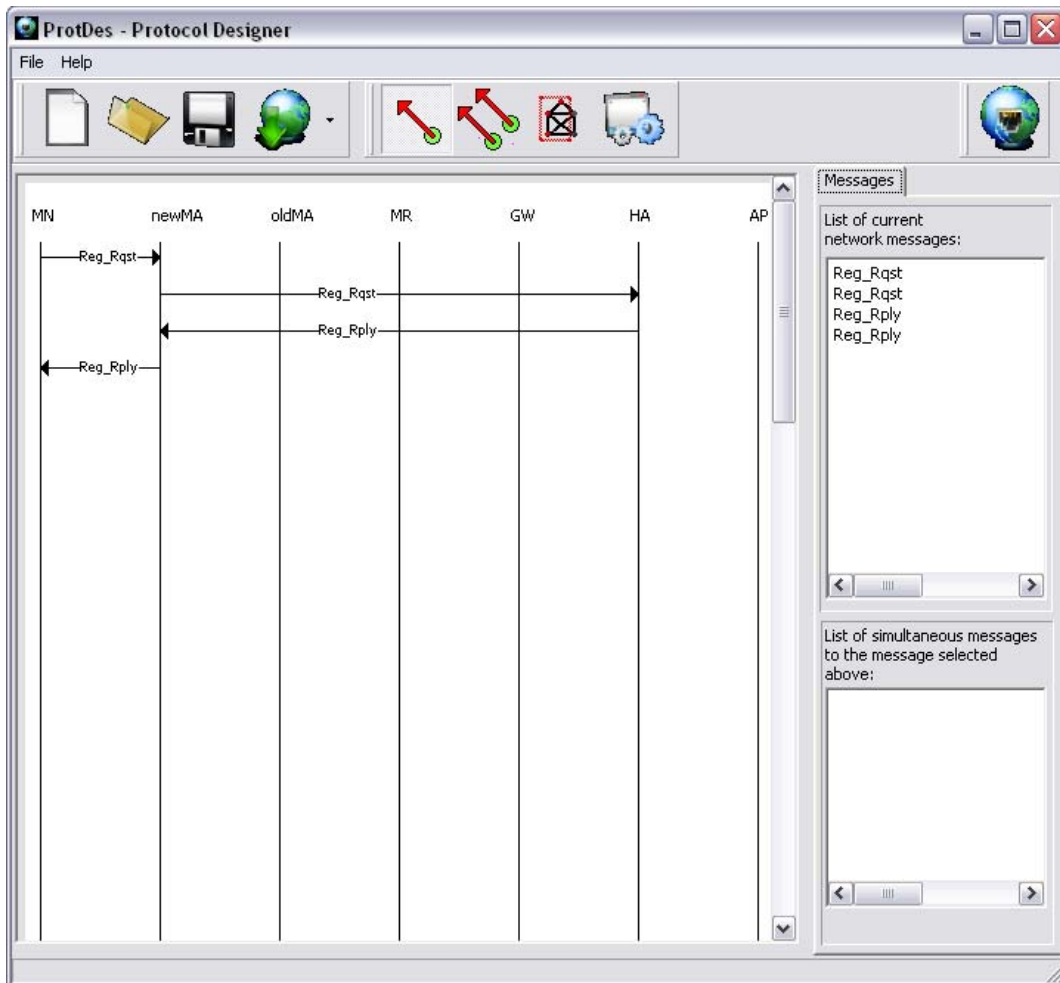


Fig 5.33: GUI of ProtDes containing an example message sequence chart of a mobility management protocol

```

Hierarchical
//*****
//*****
//Timer settings
//*****
//*****
#Timer set
Automatically

#Timer use
No

#Timer value
0
#number of retransmissions
0

//*****
//*****
//Mobility model
//*****
//*****
#R
0.75
#C
0.25

//*****
//*****
//Handoff latency
//*****
//*****
  
```

Annotations in the image point to specific parts of the file:

- 'Comments' points to the line `//Timer settings`.
- 'Parameter name' points to the line `#R`.
- 'Parameter value' points to the line `0.75`.

Fig 5.34: An example parameter file

5.10.4. Comparative Analysis of Mobility Management Protocols (CAMP)

CAMP [DMi07] is a flexible tool for the analysis of mobility management protocols. It has a flexible GUI, shown in figure 5.35, and enables the analysis of the performance as well as cost of mobility management protocols. CAMP is designed in a way that enables a simple integration of new protocols and algorithms, thus ensuring the extensibility of this tool.

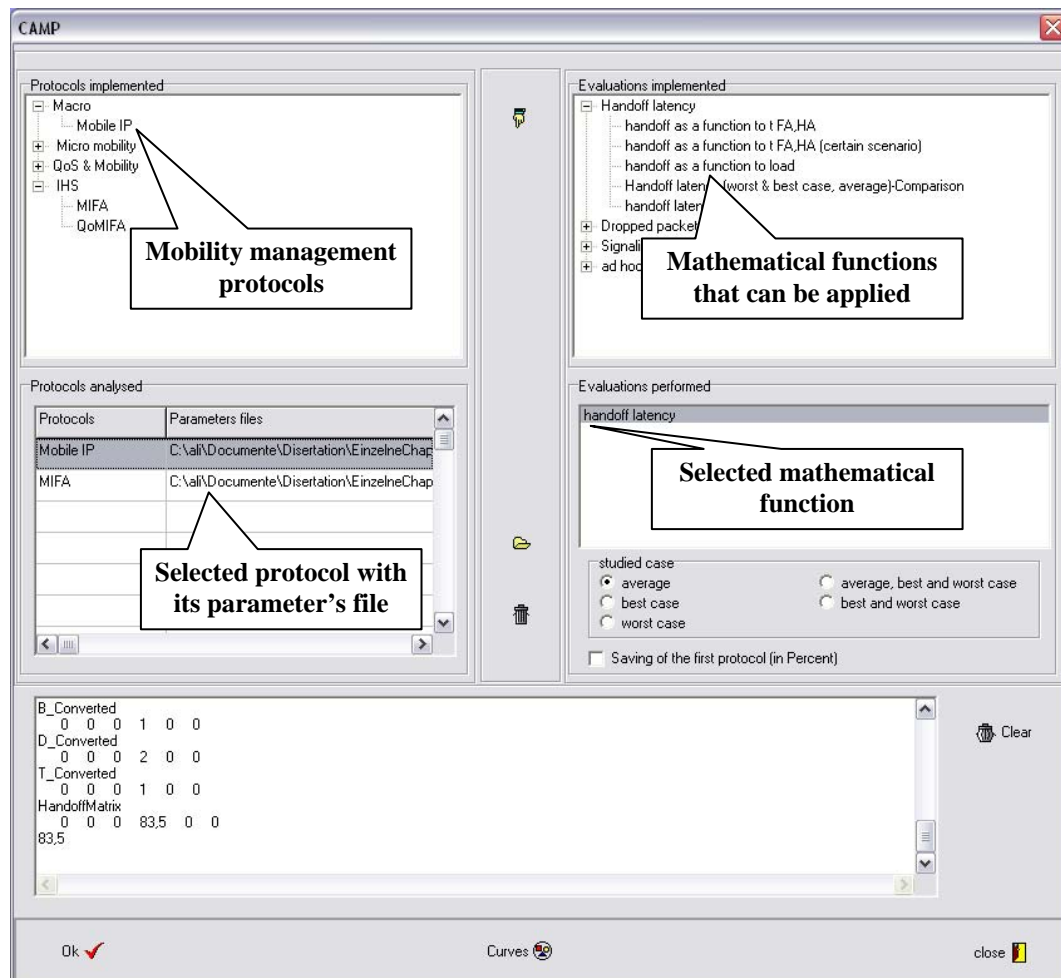


Fig 5.35: GUI of CAMP

At the beginning, the user selects the protocols he would like to analyze. Next, each protocol is linked to a parameters file generated using ProtDes. The user then selects the metric he would like to calculate. Afterwards, CAMP reads the parameters and executes the desired analysis. The results can be shown in CAMP itself or exported to Microsoft Excel for further analysis.

5.11. Conclusion

This chapter has discussed the analysis of mobility management protocols. Due to the long time required to simulate or implement mobility management protocols, a quick mathematical analysis is useful. In order to avoid writing a separate mathematical model for each protocol, a generic model that allows for the analysis of a large set of mobility management protocols has been developed and described in this chapter. The developed model analyzes the performance and estimates the cost resulting from employing mobility management protocols. The parameters of the generic model are set according to protocols characteristics, network topologies and mobility scenarios. Regarding the performance evaluation, the average handoff

latency and the expected average number of dropped packets are calculated taking control messages dropping, different mobility patterns and network topologies into account. The generic model can be applied for break-before-make as well as make-before-break mobility management protocols. With respect to the cost estimation, the location update cost as well as the packet delivery cost is estimated. The total cost is then calculated as the sum of these two costs using an adequate weighting factor for each. So as to evaluate MIFA compared to a wide range of well-known mobility management protocols, the model was applied to MIFA, MIP, MIPRR, HAWAII, Proxy MIPv6, the pre-registration method and FMIPv6.

The analysis has shown that MIFA is a very fast protocol. It outperforms the studied protocols with respect to the average handoff latency and expected average number of dropped packets per handoff. If the L2-trigger can be fired at adequate time, MIFA in predictive mode will be employed. Otherwise, the reactive mode is operated. Employing MIFA in predictive mode minimizes the handoff latency on downlink and uplink to the latency resulting from the layer 2 handoff. The loss in downlink data packets can even be eliminated. The performance of MIFA is comparable to FMIPv6 if the L2-trigger can be fired at an adequate time prior to the handoff. MIFA is suitable for low as well as high speeds. The MN can move at a max speed of 118 km/h without suffering from more latency than the layer 2 handoff latency. Moving faster than 118 km/h forces MIFA to operate in reactive mode. This, however, has a negligible impact on the performance on uplink. Although slightly more latency is experienced on downlink, MIFA remains capable of achieving smooth handoffs. A main advantage of MIFA is its robustness against control messages dropping. It remains capable of achieving seamless handoffs even if dropping of the *update message* occurs.

MIFA produces greater location update cost than most other studied mobility management protocols. This is because MIFA informs all MAs present in the L3-FHR of the current MA regarding a possible movement of the MN. In addition, not only the HA is notified of the handoff, but also the old MA as well. FMIPv6 and MIFAv6 in predictive mode are comparable to each other with respect to the location update cost. Considering the packet delivery cost, MIFAv4 in predictive as well as in reactive mode is comparable to MIPv4, the pre-registration method and Proxy MIPv6, while MIFAv6 generates the same packet delivery cost as MIPv6 and FMIPv6 in both operation modes. The other studied protocols are outperformed by MIFA.

To study the impact of mobility scenarios, the developed mathematical model has been used to analyze MIFAv6 in reactive mode, MIPv6, Proxy MIPv6 and HAWAII deploying the hierarchical network topology under different mobility scenarios. The analysis has shown that MIPv6 and Proxy MIPv6 are not affected by mobility scenarios with respect to the location update cost and average number of dropped packets per handoff on uplink and downlink. MIFAv6 is not affected by mobility scenarios regarding the average number of dropped packets per handoff on uplink. However, mobility scenarios affect MIFAv6 with respect to the average number of dropped packets per handoff on downlink and the location update cost. HAWAII is affected strongly by mobility scenarios with respect to both location update cost and average number of dropped packets on uplink as well as on downlink.

The chapter has pointed out that the performance of mobility management protocols should be evaluated taking their resulting cost into account. The main question that should be asked is: which performance gain will be obtained and at which cost? A mobility management protocol may be faster and smoother than others. However, employing this protocol in an access network of a network provider may not result in the desired results. The performance and cost gains depend strongly on the applied network topology and mobility scenarios, which can not be changed in most cases.

The developed generic mathematical model has been validated by comparing its results to simulation results modeled in NS2. The same scenarios were used for the simulation as well as for the generic model. The validation was comprised of the following scenarios: the first scenario aimed at analyzing the performance under the assumption that there is sufficient bandwidth and the loss of data packets is due to the handoffs themselves and not to network load. The second scenario studied the impact of the load, while the third scenario aimed at validating the performance under dynamically changing network conditions. More specifically, the load in this scenario as well as the speed of MNs changes randomly. The validation showed that the generic mathematical model delivers an accurate evaluation of the performance in low-loaded networks. The accuracy of the model remains acceptable even under high loads. Extending the generic model to take network load into account is very important and will significantly improve the accuracy of the generic mathematical model. This extension is a topic for future research. After the comparison to simulation results, the generic model was validated compared to results of real testbeds. The validation showed that the generic mathematical model provides a sound performance evaluation of mobility management protocols.

6. Simulative Performance Evaluation

This chapter provides a detailed performance evaluation of MIFA compared to MIP and HAWAII¹. The motivation behind selecting MIP and HAWAII is that MIP is the standard macro mobility management protocol, while HAWAII is a well-known protocol for micro mobility management. The evaluation is achieved via simulation studies modeled in NS2 version 2.29, which has been extended to model MIFA. HAWAII and MIP implementations were originally taken from the CIMS v1.0 mobility suite [[CIMS](#)] and upgraded to NS2 version 2.29. Although MIFA aims to employ the predictive mode before attempting to use the reactive one, the two modes are analyzed separately in the achieved studies. This is because the predictive mode can be used by MNs able to receive signals from more than one AP/BS, while MIFA in the reactive mode is employed for MNs able to communicate with only one AP/BS. Our performance evaluation involves assessing the impact of the network topology, network load and MN speed. In addition, the behavior of the mentioned protocols under dynamically changing network conditions² has also been evaluated.

This chapter is structured as follows. Section [6.1](#) briefly introduces NS2, while simulation scenarios used are described in section [6.2](#). Section [6.3](#) analyses the performance of the studied protocols under dynamically changing network conditions. The impact of the network topology, network load and MN speed are studied in sections [6.4](#), [6.5](#) and [6.6](#), respectively. Finally, section [6.7](#) summarizes the obtained results. Additional simulation results explaining the impact of the network topology are provided in appendix [G](#).

6.1. Network Simulator 2 (NS2)

NS2 is a widely used simulator for networks with wired and wireless nodes. It is an object-oriented simulator written in two languages, C++ and the Object-oriented Tool control language (OTcl). NS2 supports two classes hierarchy. The first is a C++ class hierarchy, referred to as a compiled hierarchy, while the second is a similar class hierarchy within the OTcl interpreter, referred to as an interpreted hierarchy. Compiled and interpreted hierarchies are closely related to each other. In fact, there is a one-to-one correspondence between the classes of these two hierarchies. The reason for using two languages is that NS2 requires doing two different kinds of things. First, protocols simulation requires an efficient and quick programming language that runs over large data sets. Therefore, C++ is used to implement the protocols that should be simulated. Second, researchers require a simple and quick reconfiguration of simulation scenarios. OTcl is slower than C++. However, it can be changed quickly, which makes the reconfiguration of simulation scenarios rather simple. Simulation of scenarios can be done as follows: an NS2 user writes the OTcl script describing the scenario and runs the simulation. The required class hierarchies in C++ and OTcl are created allowing configuration parameters to be simply exchanged between them. Finally, simulation results are gathered in certain files to be processed, e.g. visualized using the network animator **Nam** [[Nam](#)] or presented using **XGraph** [[XGra](#)].

¹ UNF scheme is used in the simulation studies presented in this chapter.

² The aim is to make the scenario as realistic as possible. As known, the load and speed of MNs in real networks change randomly. Of course, this is true when observing real networks for a short time, e.g. for several hours in the morning.

6.2. Simulation Scenarios

The simulation studies were achieved using different scenarios that correspond to an Internet domain consisting of a wireless part and an infrastructured backbone. MNs are connected to the network via IEEE 802.11 wireless links, while the infrastructured backbone is composed of wired Ethernet links that form a mesh or hierarchical topology, see figure 6.1 and figure 6.2.

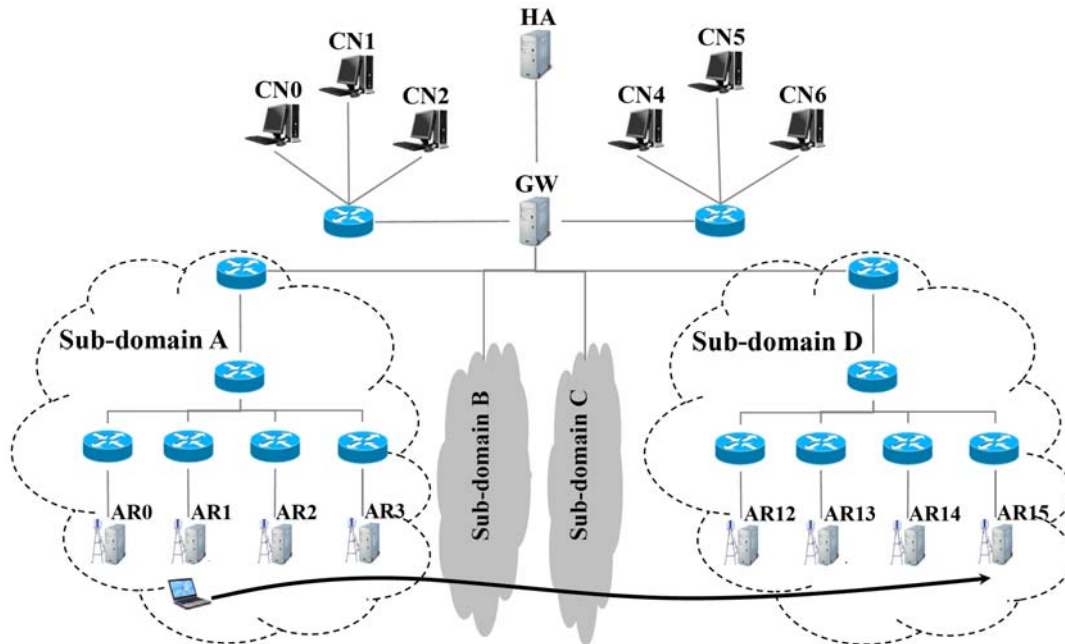


Fig 6.1: Applied hierarchical network topology

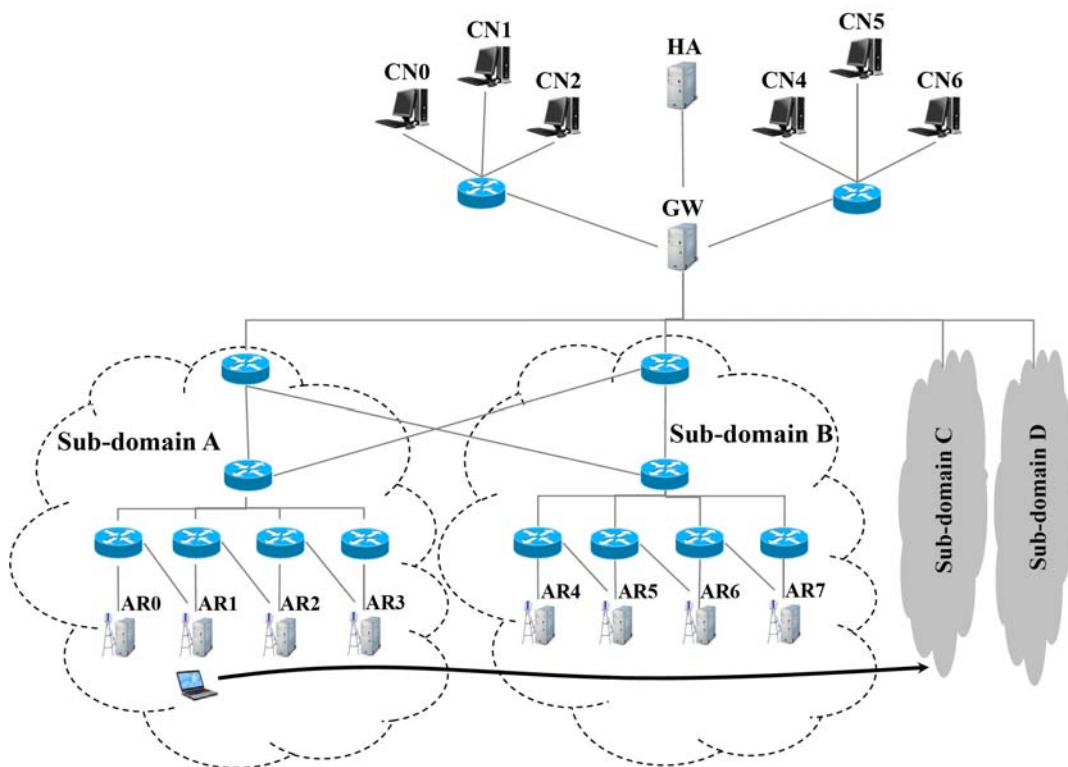


Fig 6.2: Applied mesh network topology

The hierarchical topology is the same used to validate the generic mathematical model, see section 5.9.1. A domain of 4 sub-domains with the same structure is used. Each sub-domain contains 4 ARs. One GW is placed at the first level of the hierarchy and interconnects the 4 sub-domains with each other and with nodes outside the domain. All traffic coming from or going outside the domain pass through the GW. The HA is placed outside the domain. The distance between each two neighbor ARs is 198 m. The cellular cells in this scenario overlap. The distance between the GW and each AR is 4 hops. The transmission delay on each link between each two subsequent hops inside the domain is 5 msec. The delay between the HA and the GW is 25 msec. The delay between the GW and CN0, CN1, CN2, CN4, CN5 and CN6 is 27, 23, 28, 27, 23 and 28 msec, respectively. All links have a bandwidth of 100 Mbit/s. There are 160 MNs in the domain (10 in the range of each AR). The studied scenarios contain active as well as idle MNs. Active MNs communicate with the CNs, while idle ones generate signaling traffic only. One MN that sends and receives data packets while moving from AR0 to AR15 is observed during the simulation. In order to stress the simulation results, several measurements were achieved. More concrete, each scenario was repeated 10 times, which resulted in 150 handoffs for each measurement. The same settings were used for the mesh topology as well.

6.3. Performance Evaluation under Dynamically Changing Network Conditions

As known, the load in real networks as well as the speed of idle and active MNs changes randomly. In order to evaluate the mentioned protocols under these situations, 59 MNs in addition to the observed one are selected to be active, while the other 100 ones remain in idle mode. Furthermore, real-time and non-real-time traffic are used between the active MNs and the CNs. The performance metrics we are interested in measuring include the handoff latency, number of dropped packets per handoff and congestion window. Notice that the studied performance metrics consider the layer 3 handoff only. The layer 2 handoff is excluded from this study. So as to evaluate the studied protocols with respect to the handoff latency and the number of dropped packets per handoff, UDP traffic is used. The observed MN moves at a speed of 40 km/h from AR0 to AR15 while exchanging a downlink and an uplink constant bit rate UDP stream with CN0. The packet length of both UDP streams is 500 bytes, while the packet arrival rate is 50 packets per second. To evaluate the three protocols with respect to the congestion window size, a TCP connection is established between CN0 and the observed MN. The other active MNs move randomly inside the network and start sending and receiving data packets at random times and with random packet arrival rates for UDP traffic. The network topology itself is selected to be hierarchical in this study.

6.3.1. Handoff Latency

The handoff starts either at the time at which the MN will be outside of the old cell's coverage area, or at the time at which the MN receives an advertisement from the new AR while being inside the overlapping area. As soon as the MN receives a control message indicating a successful registration, the handoff is declared completed. Due to the selected scenario, the MN spends 1.35 sec in the overlapping area between each two neighbor ARs and so receives the advertisement from the new AR while still in the overlapping area. Notice that the handoff is controlled by one node only when employing MIP and HAWAII, i.e. the HA using MIP and the old AR using HAWAII. Thus, the handoff latency on downlink and uplink is the same. On the contrary, the handoff is controlled by two nodes when employing MIFA, i.e. the new and the old AR. The MN receives a **BA** message indicating a successful registration

directly after contacting the new AR. On downlink, the old AR should be notified. Therefore, the handoff latency on downlink differs from the handoff latency on uplink.

Figure 6.3 presents the distribution function of the handoff latency resulting from employing HAWAII, MIP and MIFA inside the defined scenario. This figure shows that the layer 3 handoff latency can be eliminated if MIFA is operated in predictive mode. This is, of course, due to the sufficient time the MN spends inside the overlapping area. Even if MIFA can be operated in reactive mode only, a very fast handoff is possible. According to simulation results, 55.7 % and 78.2 % of the executed handoffs on uplink can be completed in less than 10 and 25 msec, respectively. As mentioned previously, the MN requires contacting the new AR only to be able to resume its uplink communication. MIFA in reactive mode performs very well on downlink too. 65.3 % of the achieved handoffs are executed in less than 56 msec, while 89.3 % require no more than 86 msec for the completion. The reason for this is that the new AR has to notifying the old AR only to enable the MN to resume its communication on downlink.

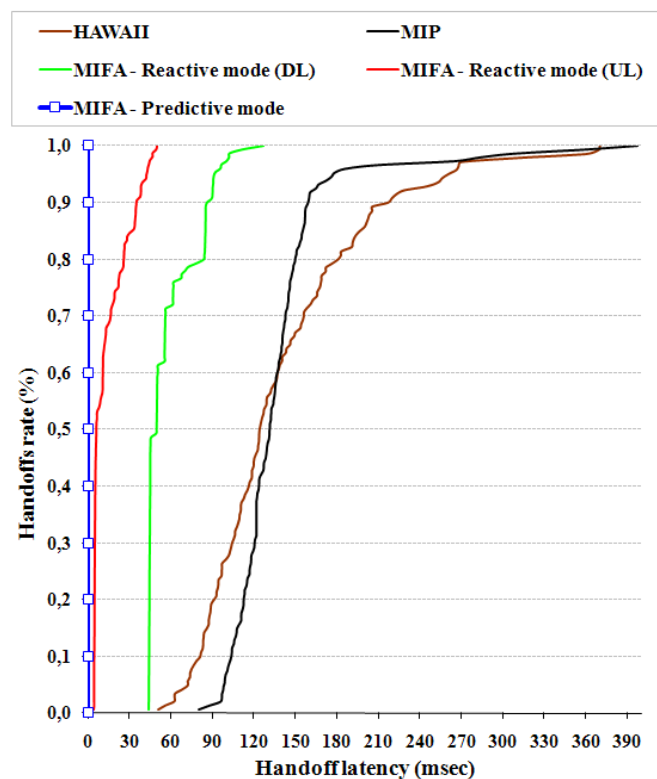


Fig 6.3: Distribution function of the handoff latency resulting from employing HAWAII, MIP and MIFA under dynamically changing network conditions

We will now compare the handoff latency resulting from the three mentioned protocols to each other. As mentioned before, MIFA in predictive mode performs the best. MIFA in reactive mode significantly outperforms the other two protocols on uplink. This is, of course, because MIFA only requires contact with the new AR, while HAWAII and MIP must wait for a response from the old AR and the HA, respectively. Regarding the handoff latency on downlink, although HAWAII updates its binding at the old AR as MIFA in reactive mode does, it suffers from longer handoff latencies than those resulting from MIFA in this mode. The reason for this is that the *update message* has to be processed only by the new and old ARs when employing MIFA in reactive mode, while the *update message* must be processed by the new and old ARs as well as by each hop on the path in between when employing HAWAII. This processing takes a long time if these hops are high-loaded. Therefore, under all situations in the studied scenario, HAWAII is outperformed by MIFA. Regarding MIP, it requires significantly more time to complete handoffs than MIFA. The reason for this is that

MIP requires the completion of updates of mobility bindings at the HA, while MIFA contacts only the old AR, which is almost closer to the new AR than the HA.

When comparing the handoff latency resulting from HAWAII to that resulting from MIP we notice that HAWAII is faster than MIP in 58.5 % of the executed handoffs and comparable only in approximately 1.6 %. As mentioned before, the load in this scenario changes randomly and these handoffs are performed, in principle, in low- to middle-loaded network. For the remaining handoffs, MIP performs better. The reason for this is that these handoffs are executed in high-loaded network and HAWAII, as it will be shown later, is highly affected by network load.

Similar results to those shown in figure 6.3 can be seen in figure 6.4, which shows the average handoff latency resulting from employing the above mentioned mobility management protocols in the studied scenario. Although HAWAII is considered a micro mobility management protocol, it performs comparable to MIP in a network that has dynamically changing conditions. The reason for this has been discussed previously and is mainly due to the high impact of network load on the performance of HAWAII, as it will be shown later. MIFA in predictive mode is the best and results, as mentioned earlier, in eliminating the layer 3 handoff latency. MIFA in reactive mode performs significantly better than MIP and HAWAII. Simulation results showed that MIFA in this mode results in a performance improvement of approximately 58 % on downlink and 90 % on uplink over MIP and HAWAII.

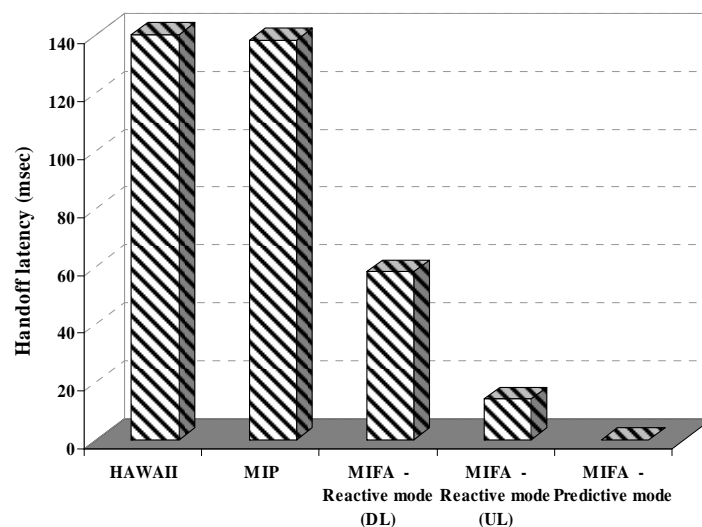


Fig 6.4: Average handoff latency resulting from employing HAWAII, MIP and MIFA under dynamically changing network conditions

6.3.2. *Number of Dropped Packets Per Handoff*

Figure 6.5 and figure 6.6 present the distribution function of the number of dropped packets per handoff on downlink and uplink resulting from employing HAWAII, MIP and MIFA in the studied scenario. Notice that the best performance is achieved by MIFA in predictive mode, which is capable of guaranteeing lossless handoffs. Seamless handoffs are achieved by MIFA in reactive mode as well. On downlink, 50.6 % of all handoffs are executed with a maximum loss of 2 packets per handoff. No more than 4 packets per handoff are lost for 86.6 % of all handoffs. On uplink, there are no more than 3 dropped packets per handoff in all handoffs. Once again, HAWAII and MIP are outperformed by MIFA. The reasons have been discussed while analyzing the handoff latency.

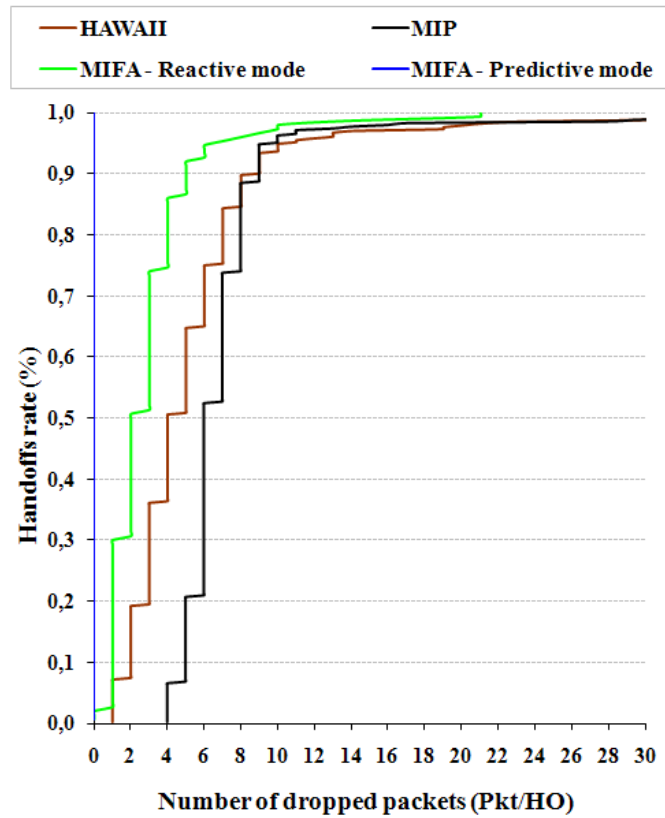


Fig 6.5: Distribution function of the number of dropped packets per handoff on downlink resulting from employing HAWAII, MIP and MIFA under dynamically changing network conditions

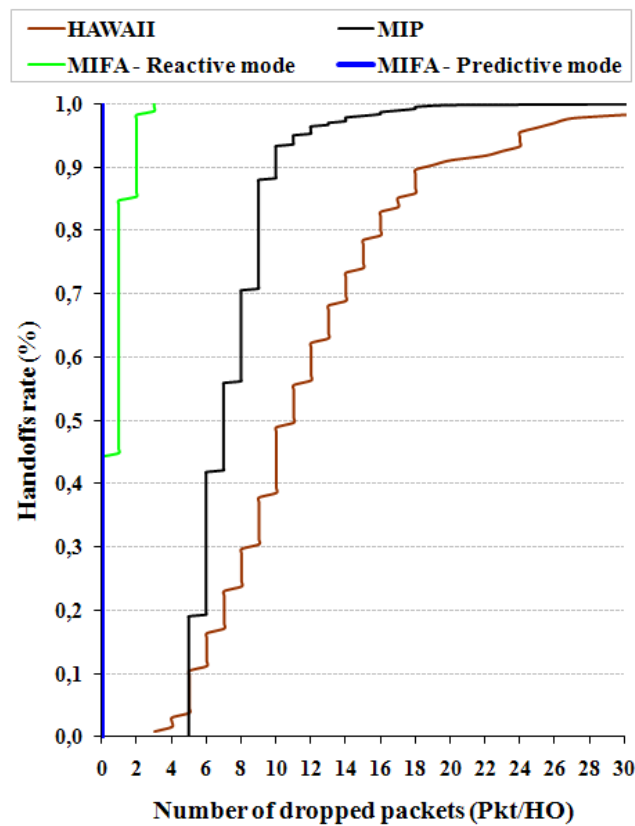


Fig 6.6: Distribution function of the number of dropped packets per handoff on uplink resulting from employing HAWAII, MIP and MIFA under dynamically changing network conditions

Although HAWAII outperforms MIP in only 57.8 % of the handoffs considering the handoff latency, it outperforms MIP in 84.3 % of the handoffs with respect to the number of dropped packets per handoff on downlink. This is because HAWAII starts sending data packets to the new CoA upon the crossover router receives the *update message*. In contrast, MIP takes a long time to inform the HA, which redirects the traffic to the new CoA. In the remaining handoffs, HAWAII is comparable to MIP. For the number of dropped packets per handoff on uplink, the behavior looks totally different. Only in 2.9 % of all handoffs (the first 2.9 % that relate to a very low-loaded network) HAWAII shows better performance than MIP. In the following 7.5 %, HAWAII and MIP are comparable. After that, MIP starts to perform better. This is due to the dependency on the handoff latency, as displayed in figure 6.3, which shows that MIP outperforms HAWAII with respect to the handoff latency in high-loaded networks. Notice that all data packets the MN sends during the handoff are lost. Similar results can be derived from figure 6.7, which shows the average number of dropped packets per handoff on downlink as well as on uplink.

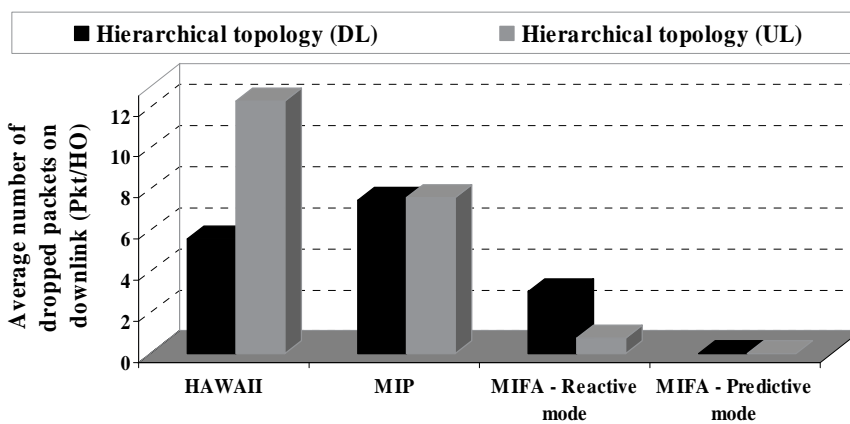


Fig 6.7: Average number of dropped packets per handoff on downlink and uplink resulting from employing HAWAII, MIP and MIFA under dynamically changing network conditions

This figure shows that MIFA results in lossless layer 3 handoffs if it could be operated in predictive mode. HAWAII drops per handoff 46 % and 94 % more than MIFA in reactive mode on downlink and uplink, respectively. Regarding MIP, it is outperformed by MIFA in reactive mode by 59.6 % and 90.4 % on downlink and uplink, respectively.

We will now compare the average number of dropped packets per handoff on downlink to that on uplink for each protocol. HAWAII performs on downlink significantly better than on uplink. Our simulation results have shown that HAWAII drops on uplink 54.7 % more than on downlink. The reason was presented while discussing figure 6.5 and figure 6.6. With respect to MIP, it shows on downlink slightly better performance than on uplink. According to the simulation results, MIP drops on uplink only 2 % more than on downlink. This behavior can be interpreted as follows: when the MN operating MIP moves to a new AR, it should register with the HA. Packets dropped on uplink are the packets the MN sends during the handoff. On downlink, however, dropped packets are the packets forwarded to the MN on the old path after starting the handoff and before the HA is notified of the new CoA. In addition to these dropped packets, the packets forwarded to the MN on the old path before starting the handoff and reaching the old AR after breaking the link with the MN are dropped, as well. In contrast to MIP and HAWAII, MIFA in reactive mode drops on downlink more than on uplink. In the studied scenario, it drops 75.7 % more on downlink than on uplink. This is due to the controlling of the handoff by the old AR on downlink, while the new AR controls the handoff on uplink.

6.3.3. Average Congestion Window Size

Figure 6.8 presents the average congestion window size resulting from employing MIP and MIFA in the studied scenario.

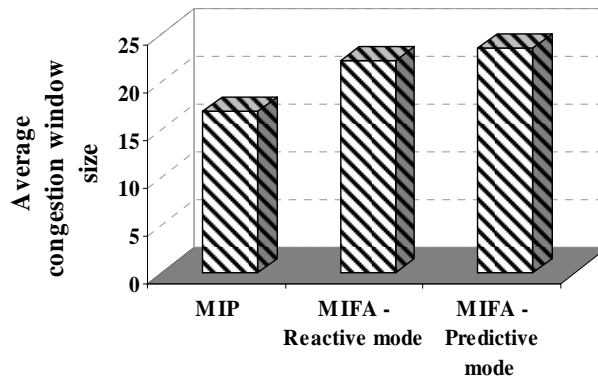


Fig 6.8: Average congestion window size resulting from employing MIP and MIFA under dynamically changing network conditions

Because MIFA in predictive mode produces lossless handoffs, it avoids executing the slow start mechanism due to handoffs. If MIFA in predictive mode can not be operated and, therefore, the reactive mode should be employed, executing the slow start mechanism due to communication disruptions during handoffs may not be avoided in all handoffs. Therefore, MIFA performs in predictive mode slightly better than in reactive mode, 6.3 % better according to the simulation results. Regardless of the employed mode, MIFA clearly outperforms MIP. According to the simulation results, the average congestion window size resulting from employing MIP is 31 % and 39.3 % less than that resulting from MIFA in reactive and predictive mode, respectively. The main reason is the execution of the slow start mechanism each time the MN operating MIP moves inside the network.

6.3.4. Summary

The main results can be summarized as follows: under dynamically changing network conditions, the best performance is achieved by MIFA in predictive mode. If the time the MN spends inside the overlapping area is sufficient to start the layer 3 handoff in advance, lossless layer 3 handoffs can be achieved. In other words, good network planning helps significantly in improving the performance. Even if MIFA can be operated in reactive mode only, it offers fast and seamless handoffs. It clearly outperforms MIP and HAWAII with respect to the handoff latency and the number of dropped packets per handoff on downlink as well as on uplink using real-time traffic. In contrast to MIP and HAWAII, MIFA performs on uplink significantly better than on downlink. When non-real-time traffic is used, MIFA offers very good performance as well. It performs in predictive mode slightly better than in reactive mode and clearly better than MIP with respect to the average congestion window size.

6.4. Impact of Network Topology

In order to analyze the impact of the network topology on the performance of MIFA, MIP and HAWAII, the same scenario applied in section 6.3 deploying the hierarchical topology is applied to the mesh topology shown in figure 6.2. The handoff latency as well as the number of dropped packets per handoff is measured in the two topologies and compared to each other. More results describing the impact of the network topology is provided in appendix G.

6.4.1. Impact of Network Topology on the Performance of MIFA

Figure 6.9 shows the range of the handoff latency as well as the average value on downlink and uplink experienced when employing MIFA in reactive as well as predictive mode in a hierarchical and a mesh network topology. The range of the number of dropped packets per handoff as well as the average value on downlink and uplink employing MIFA in the both topologies is illustrated in figure 6.10.

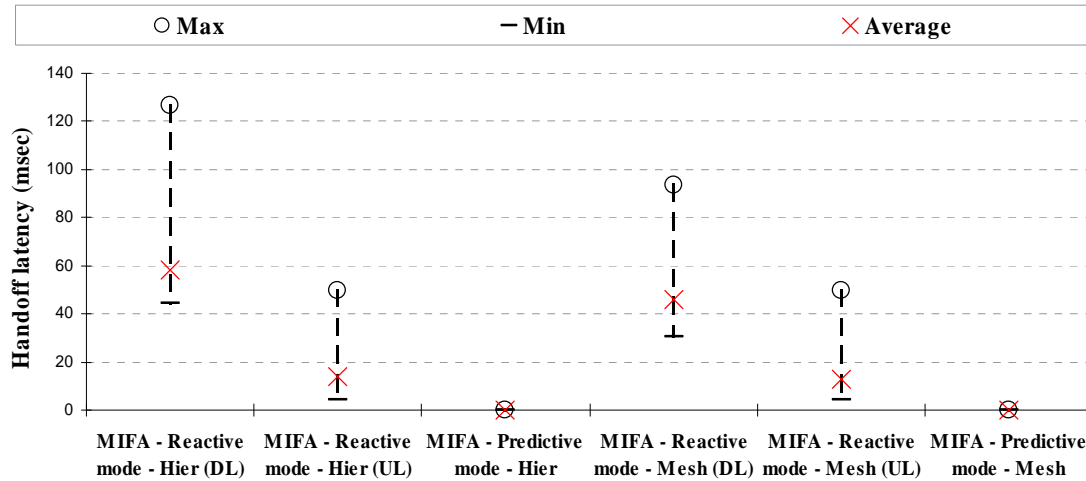


Fig 6.9: Impact of network topology on the handoff latency resulting from employing MIFA under dynamically changing network conditions

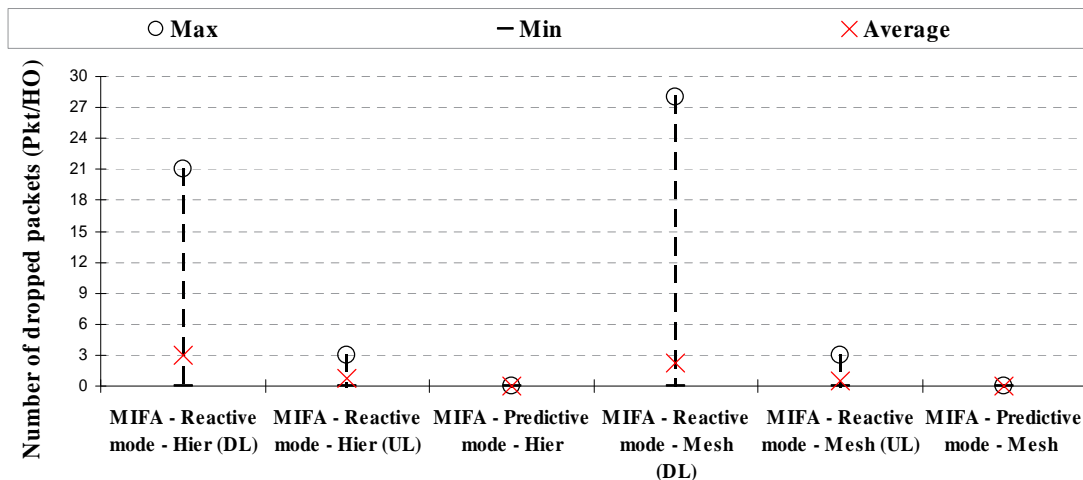


Fig 6.10: Impact of network topology on the number of dropped packets per handoff on downlink and uplink resulting from employing MIFA under dynamically changing network conditions

The two figures show that because MIFA in predictive mode eliminates the layer 3 handoff latency in the studied scenarios and so produces lossless layer 3 handoffs, there is no impact of the network topology on its performance. Considering MIFA in reactive mode, the handoff on uplink is finished from the MN point of view as soon as a **BA** message is received from the new AR. Therefore, there should be theoretically no impact of the network topology in this case. This assumption is validated in figure 6.9 and figure 6.10, which prove that MIFA in reactive mode performs on uplink in a hierarchical topology similar as in a mesh topology. On downlink, there is a clear improvement in the performance deploying a mesh topology. This is due to the dependency on the old AR to forward data packets to the MN until the HA is informed. The path between the new and the old AR is shorter deploying the mesh topology, see figure 6.1 and figure 6.2. The simulation results have shown that there is 21 % performance improvement deploying the mesh topology with respect to the average handoff

latency. Regarding the average number of dropped packets per handoff, the performance is improved by 26.2 % deploying the mesh topology. Notice that the number of dropped packets per handoff varies when deploying the mesh topology in a bigger range than that resulting from deploying the hierarchical topology. This is, of course, due to the dynamically changing network conditions.

Let us now assess the impact of the network topology on handoff duration when employing MIFA in predictive mode, see figure 6.11.

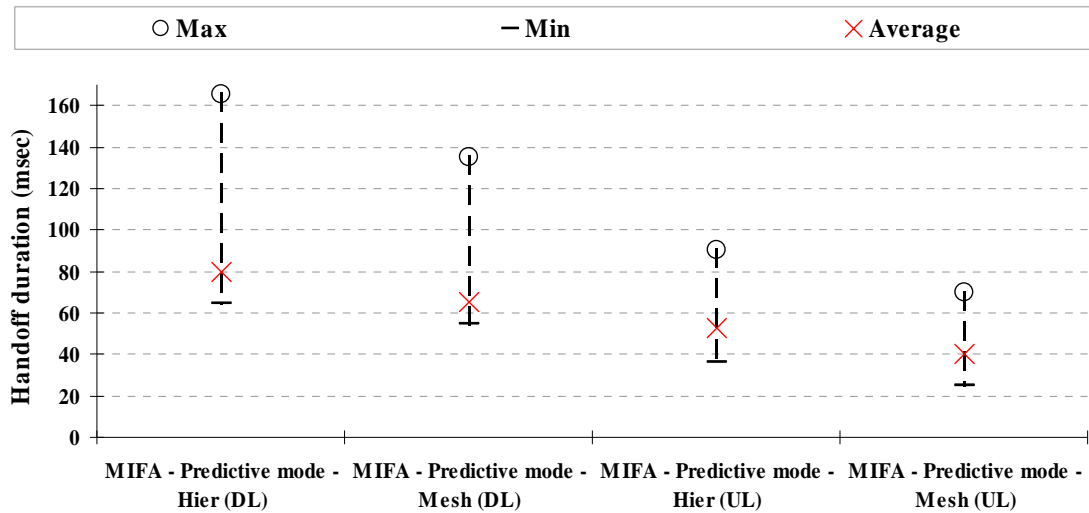


Fig 6.11: Impact of network topology on the handoff duration on downlink and uplink resulting from employing MIFA in predictive mode under dynamically changing network conditions

The handoff duration on downlink is defined as the time duration between the time at which the MN detects that a handoff will occur and the time at which the old AR receives a *Hn_Ack* message from the new AR. The handoff duration on uplink is defined as the time duration required to notify the new AR of the incoming MN. From this figure it can be seen that the network topology has a noticeable impact on the handoff duration on downlink and uplink for MIFA in predictive mode. Because the path between the old and new AR in the applied mesh topology is shorter than in the applied hierarchical topology, the mesh topology improves the performance. According to the simulation results, the mesh topology reduces the average handoff duration by 23.3 % and 17.8 % compared to the hierarchical topology on uplink and downlink, respectively.

6.4.2. Impact of Network Topology on the Performance of MIP

Figure 6.12 illustrates the range and the average handoff latency experienced when employing MIP in the hierarchical and the mesh network topology. Notice that MIP performs comparably in both topologies with respect to the average handoff latency. The difference between the handoff latency in the two topologies is only 6.8 %. A similar result can be seen in figure 6.13, which shows the range and the average number of dropped packets per handoff on downlink and uplink when employing MIP in both topologies. According to the simulation results, the average number of dropped packets per handoff resulting from the use of the hierarchical topology is only 1 % better than that resulting from the deployment of the mesh topology. The main result obtained from the both figures is that there is no remarkable impact of network topology on the performance of MIP.

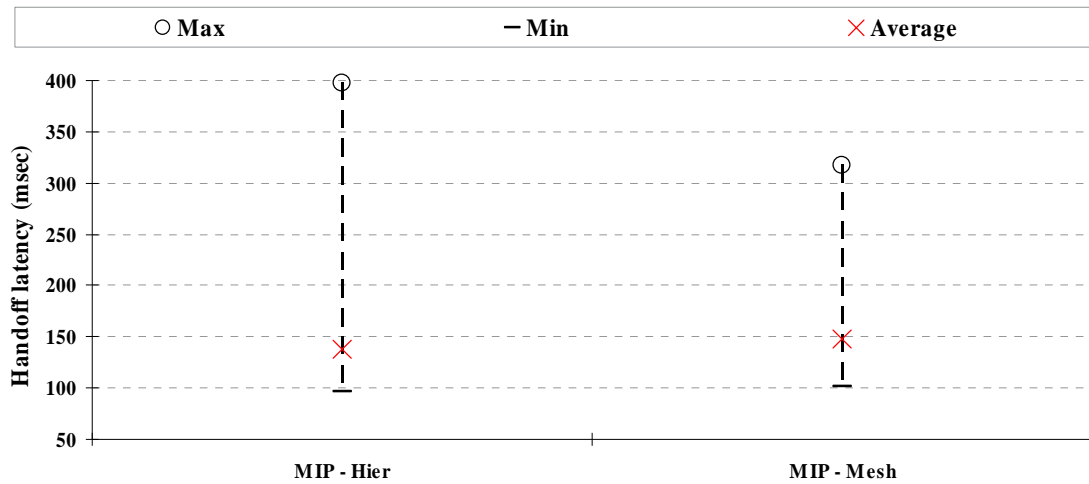


Fig 6.12: Impact of network topology on the handoff latency resulting from employing MIP under dynamically changing network conditions

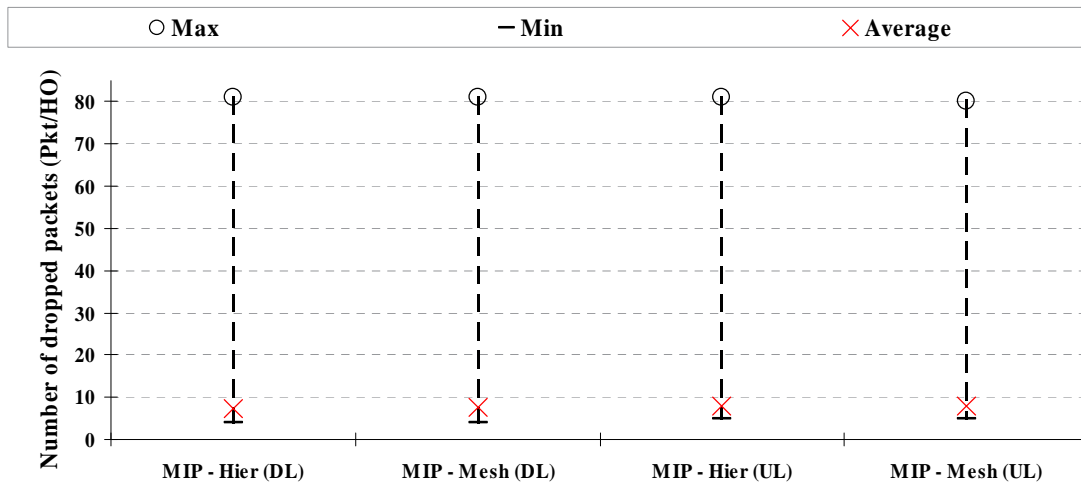


Fig 6.13: Impact of network topology on the number of dropped packets per handoff on downlink and uplink resulting from employing MIP under dynamically changing network conditions

6.4.3. Impact of Network Topology on the Performance of HAWAII

Figure 6.14 expresses the range and the average handoff latency experienced employing HAWAII in the hierarchical and the mesh network topology.

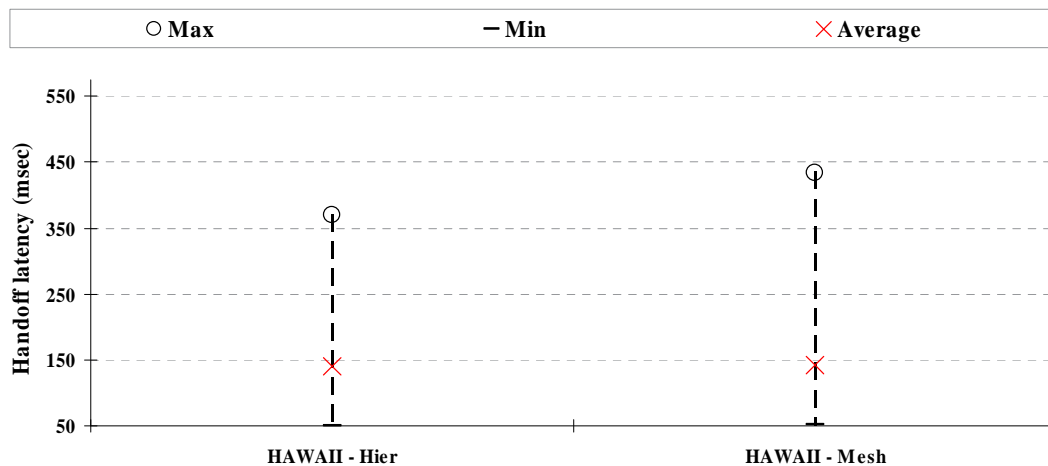


Fig 6.14: Impact of network topology on the handoff latency resulting from employing HAWAII under dynamically changing network conditions

This figure shows that there is no remarkable difference in performance deploying both topologies with respect to the average handoff latency. According to the simulation results, there is a difference of less than 1 %. Of course, the difference in the range of the handoff latency is due to the dynamically changing network conditions.

Figure 6.15 shows the range and the average number of dropped packets per handoff on downlink and uplink when employing HAWAII in both topologies. From this figure it can be seen that the mesh topology delivers a small improvement over the hierarchical topology with respect to the range as well as the average number of dropped packets per handoff on downlink. This is because the crossover router is the node, which stops forwarding on the old path. As known, the path between the new AR and the crossover router is shorter in the mesh topology than in the hierarchical one. According to the simulation results, the mesh topology produces 8.3 % fewer dropped packets per handoff with respect to the average number of dropped packets per handoff on downlink. Regarding the number of dropped packets per handoff on uplink, HAWAII performs in the mesh topology slightly better than in the hierarchical topology with respect to the average number of dropped packets. According to the simulation results, HAWAII drops 6.2 % fewer in the mesh topology than in the hierarchical topology.

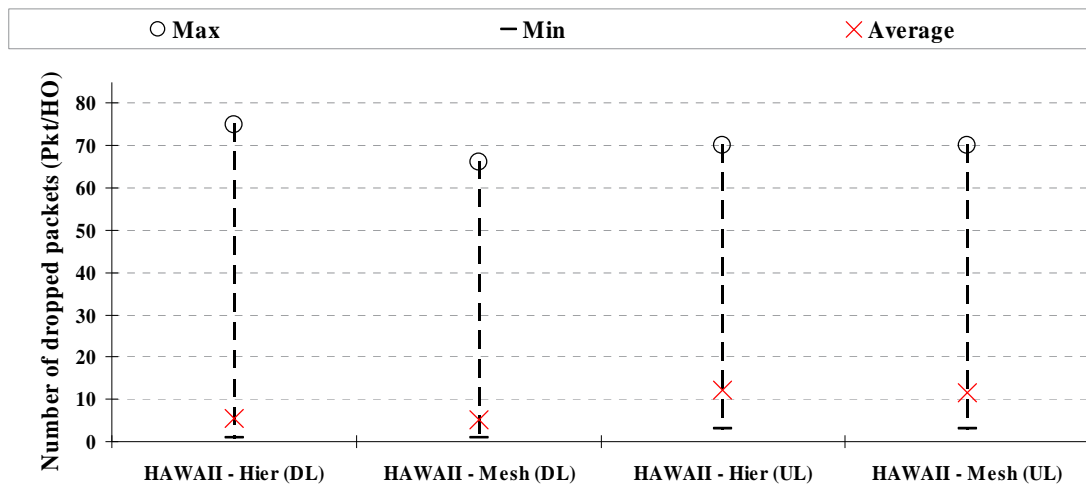


Fig 6.15: Impact of network topology on the number of dropped packets per handoff on downlink and uplink resulting from employing HAWAII under dynamically changing network conditions

The main results obtained from the above discussion are the following: there is no remarkable impact of the topology on the performance of HAWAII with respect to the handoff latency. Considering the number of dropped packets per handoff on downlink as well as uplink, the mesh topology produces a slight performance improvement over the hierarchical one.

6.4.4. Summary

The main obtained results can be summarized as follows: there is no impact of the network topology on the performance of MIFA in predictive mode in the studied scenarios. Of course, good network planning is essential. Considering MIFA in reactive mode, there is no impact of the network topology regarding the performance of MIFA on uplink. However, there is a clear improvement in the performance on downlink deploying a mesh topology. Regarding MIP and HAWAII, there is no remarkable impact of the network topology.

6.5. Impact of Network Load

This section studies the impact of network load on the performance of HAWAII, MIP and MIFA. The hierarchical topology presented in figure 6.1 is applied. UDP as well as TCP traffic is used in this study. For UDP, a downlink and an uplink UDP stream between CN0 and the observed MN are used. The packet length of both UDP streams is 500 bytes and the packet arrival rate is 50 packets per second. Regarding TCP, a TCP connection between CN0 and the observed MN is established. The duration of this TCP connection is 4.5 minutes. The number of MNs is 160. The observed MN moves at a speed of 40 km/h from AR0 to AR15 as well. The number of active MNs is changed from 1 to 60. The active MNs exchange UDP as well as TCP traffic with the CNs present in the scenario. The remaining MNs stay inactive and produce only signaling traffic. The load remains constant during the simulation. Although the idle MNs produce only signaling traffic and, therefore, produce negligible load compared to that resulting from active MNs, they should not be withdrawn from the study. The aim is to let the scenario be as realistic as possible. As known, there are always active and idle MNs in each radio access network. The performance metrics are the handoff latency, the number of dropped packets per handoff for UDP traffic and the average congestion window size for TCP traffic.

6.5.1. Impact of Network Load on the Performance of MIFA

Figure 6.16 illustrates the distribution function of the handoff latency on downlink experienced when employing MIFA in reactive mode inside the studied scenario under different loads. It can clearly be seen that MIFA in this mode is a fast solution for mobility management even in high-loaded networks.

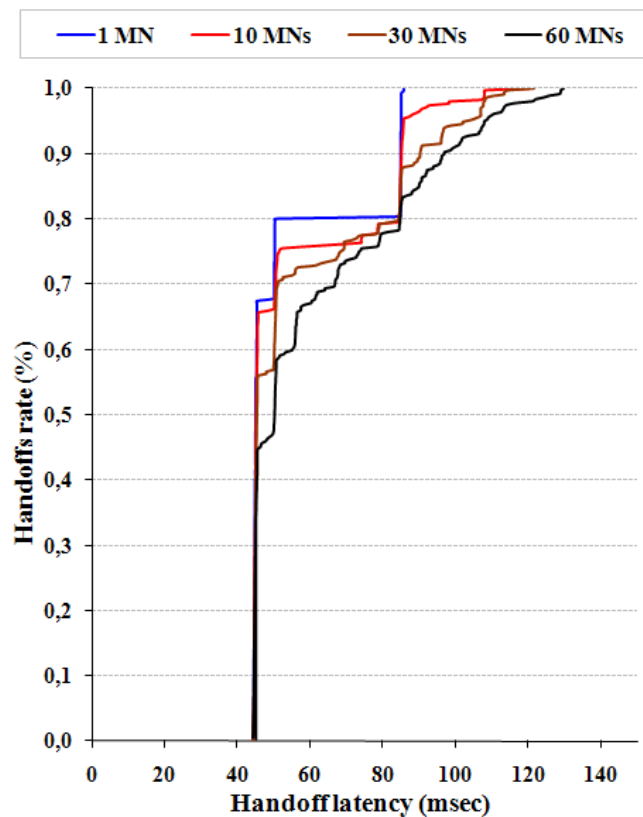


Fig 6.16: Distribution function of the handoff latency on downlink resulting from employing MIFA in reactive mode in a hierarchical topology under different loads

For low-loaded networks (1 and 10 active MNs), approximately 80 % of all handoffs can be completed in less than 80 msec. Clearly, increasing the number of active MNs increases the handoff latency. 94.3 % and 91 % of all handoffs have a latency of less than 100 msec if the network contains 30 and 60 active MNs, respectively. An additional result can be derived from this figure, namely that most handoffs experienced on downlink are completed in less than 50 msec, 67.4 %, 66 %, 56.6 % and 47.4 % of all handoffs for networks with 1, 10, 30 and 60 active MNs, respectively. The reason why the load does not have a significant impact on the downlink handoff latency employing MIFA in reactive mode is that MIFA control messages must travel from the new to the old AR. Therefore, only the load on the new AR, old AR and the hops in between as well as the load on the wireless link play a significant roll.

Figure 6.17 shows the handoff latency on uplink employing MIFA in reactive mode in the studied scenario. This figure proves that MIFA achieves very fast handoffs on uplink under all load situations. According to the obtained results, the handoff latency on uplink will not exceed 50 msec regardless of the applied load. For low-loaded networks, 47.1 % and 35.5 % of all executed handoffs are completed in less than 5 msec if the network contains only 1 and 10 active MNs, respectively. Additionally, 77.5 % and 70.3 % of all handoffs have a latency of less than 10 msec, while 92 % and 89.7 % of the handoffs have been finished in less than 20 msec under these assumptions. When the number of active MNs is increased to 30, it can be seen that 30.1 %, 54.5 % and 78 % of all handoffs are completed in less than 5, 10 and 20 msec, respectively. For high-loaded networks, in which the network contains 60 active MNs, 16.9 %, 47.8 % and 72 % of all handoffs are finished in less than 5, 10 and 20 msec, respectively. The reason behind the minimal impact of the load on the handoff latency on uplink is that the MN contacts only its new AR to resume its uplink communication. Therefore, only the current load of this AR as well as the load on the wireless link is of interest for MIFA. The load in the core network does not play any role.

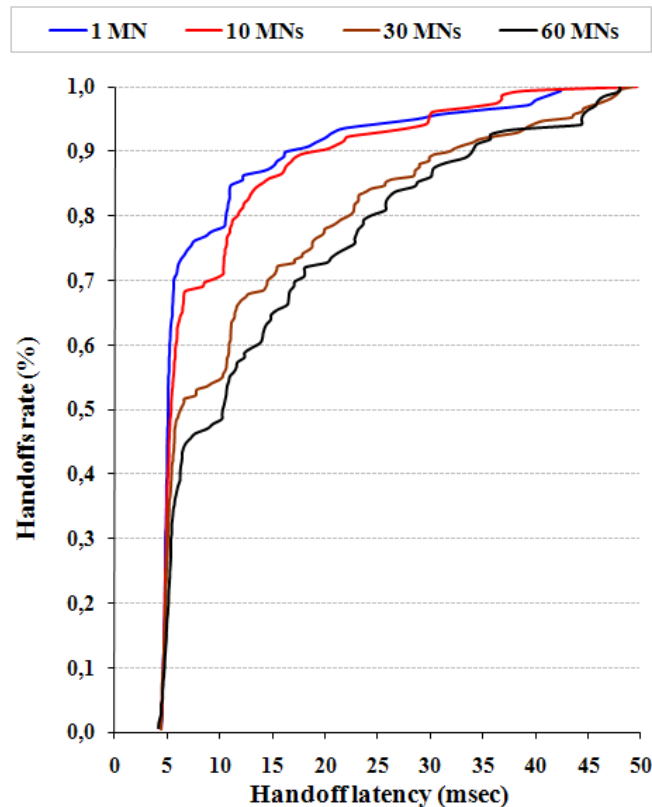


Fig 6.17: Distribution function of the handoff latency on uplink resulting from employing MIFA in reactive mode in a hierarchical topology under different loads

Let us now discuss the performance of MIFA in predictive mode. As currently known, MIFA detects the new AR before the handoff occurs and so begins the layer 3 handoff in advance. If the L2-trigger is raised at sufficient time before breaking the radio link with the old AR, the layer 3 handoff latency on downlink can be eliminated. The dominant factor is the size of the overlapping area, which should be designed carefully to guarantee the predictive handoff under most situations. As mentioned in section 6.3.1, the observed MN spends at a speed of 40 km/h 1.35 sec in the overlapping area. This time is sufficient to guarantee employing MIFA in predictive mode. The layer 3 handoff latency on uplink is negligible. As soon as the MN finishes the layer 2 handoff, a **BA** message indicating a successful completion of the layer 3 handoff will be sent to the MN.

Figure 6.18 shows the distribution function of the number of dropped packets per handoff on downlink experienced when employing MIFA in reactive mode in the studied scenario under different loads. The results that can be derived from this figure are similar to those derived from figure 6.16. MIFA achieves smooth handoffs in reactive mode even in high-loaded networks. Regardless of network load, the number of dropped packets per handoff does not exceed 2 in approximately 56.5 % of all handoffs. Increasing the load increases the number of dropped packets accordingly. MIFA performs in low-loaded networks with 1 and 10 active MNs very well. The number of dropped packets per handoff does not exceed 2 in 86.7 % of all handoffs. For networks with 30 active MNs, the number of dropped packets per handoff in 68.5 % of all handoffs does not exceed 2 lost packets, while in 85.6 % of the handoffs there are no more than 3 lost packet per handoff. Even in high-loaded networks that contain 60 active MNs, MIFA drops no more than 2 and 3 packets in 56.5 % and 71.7 % of all handoffs, respectively. The reason behind the strong performance of MIFA has been highlighted while discussing figure 6.16.

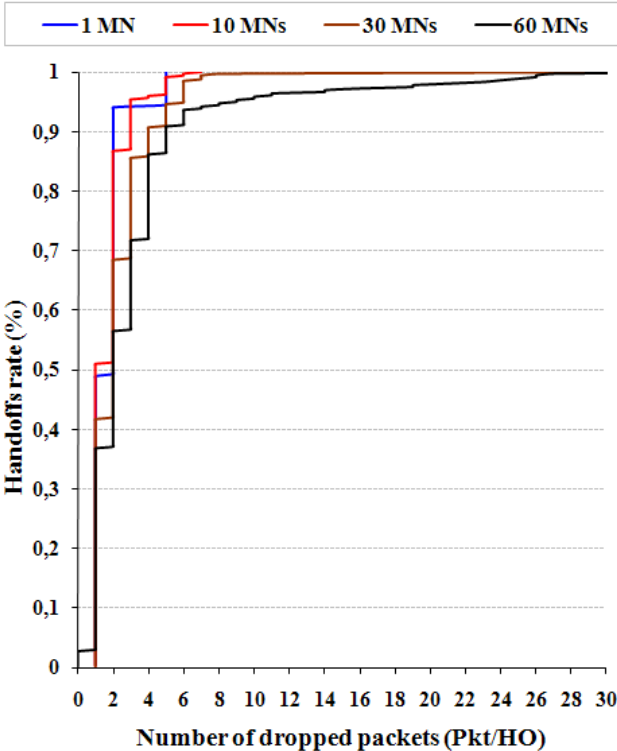


Fig 6.18: Distribution function of the number of dropped packets per handoff on downlink resulting from employing MIFA in reactive mode in a hierarchical topology under different loads

Let us now consider the number of dropped packets on uplink employing MIFA in reactive mode, see figure 6.19. This figure shows that MIFA is able to achieve very fast handoffs

regardless of network load. 41.2 % of all handoffs under all studied loads are even lossless. Obviously, increasing the load decreases the number of lossless handoffs. According to the simulation results, 41.2 %, 46.4 %, 64.1 % and 70.5 % of all handoffs are lossless if the network contains 60, 30, 10 and 1 active MNs, respectively. Regarding MIFA in predictive mode, lossless layer 3 handoffs on downlink as well as on uplink are achieved. The reason for this has been discussed previously.

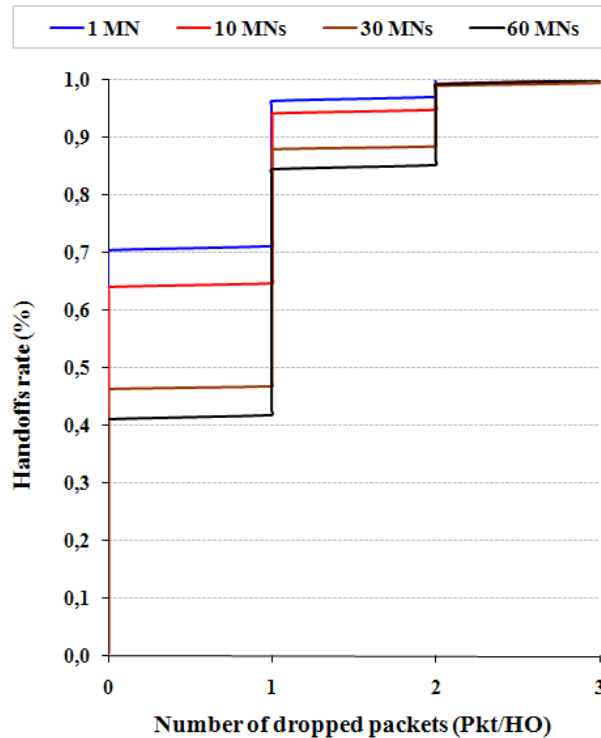


Fig 6.19: Distribution function of the number of dropped packets per handoff on uplink resulting from employing MIFA in reactive mode in a hierarchical topology under different loads

6.5.2. Impact of Network Load on the Performance of MIP

The distribution function of the handoff latency experienced when employing MIP in the studied scenario under different loads is presented in figure 6.20. It can be seen that network load has a significant impact on the performance of MIP with respect to the handoff latency. According to our simulation results, MIP completes approximately 72 % of all handoffs in less than 125 msec and approximately 90 % in less than 130 msec if the network contains only 1 and 10 active MNs, respectively. As soon as network load increases, the handoff latency increases accordingly. For networks with 30 active MNs, only 56.4 % of the handoffs can be finished in a latency of less than 125 msec, while 20 % of the handoffs require more than 150 msec to be completed. If the number of active MNs is increased to 60, the number of handoffs that can be completed in less than 125 msec decreases to only 28.8 % of all handoffs, while the number of handoffs requiring more than 150 msec increases to 46.8 %.

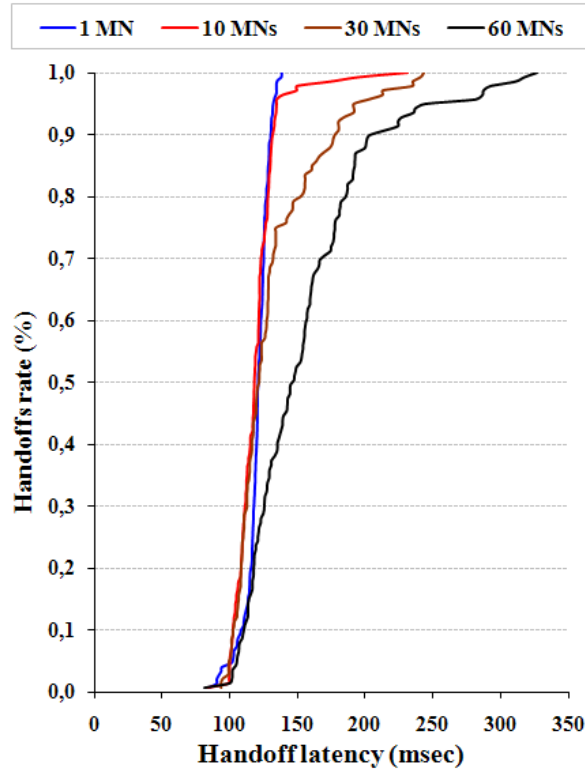


Fig 6.20: Distribution function of the handoff latency resulting from employing MIP in a hierarchical network topology under different loads

Let us now analyze the impact of network load on the performance of MIP with respect to the number of dropped packets per handoff. Figure 6.21 presents the distribution function of the number of dropped packets per handoff on downlink under different loads.

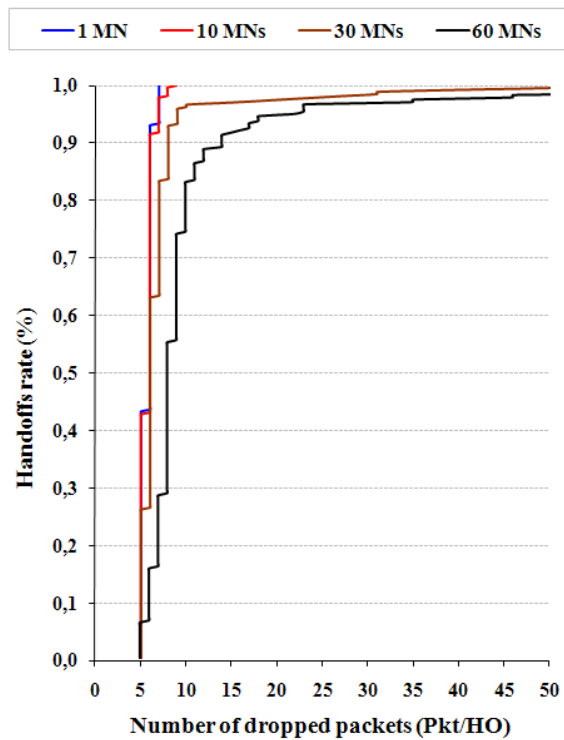


Fig 6.21: Distribution function of the number of dropped packets per handoff on downlink resulting from employing MIP in a hierarchical topology under different loads

With respect to the number of dropped packets per handoff on downlink, it can be seen that increasing the number of active MNs from 1 to 10 does not have a remarkable impact on MIP. This is, of course, because the network is still low-loaded and data packets are not required to wait in routers queues. Under these situations, approximately 92 % of all handoffs are executed with no more than 6 dropped packets per handoff. Increasing the number of active MNs to 30 produces a remarkable impact on MIP. Only 63 % of achieved handoffs are completed with less than 7 dropped packets per handoff. For networks with 60 active MNs, a weak performance can clearly be seen. According to our results, only 16 % of all handoffs produce less than 7 dropped packets per handoff.

Similar results can be derived with respect to the dropped packets per handoff on uplink, see figure 6.22. MIP performs comparable in networks containing 1 and 10 active MNs. When the network contains only 1 active MN, there are no more than 6 dropped packets per handoff in 93.8 % of all handoffs, while MIP drops no more than 6 packets per handoff in 89.2 % of the handoffs when there are 10 active MNs. Increasing the number of active MNs to 30 decreases the handoffs that can be executed with less than 7 lost packets to 36.9 %. For networks with 60 active MNs, the handoffs that can be executed with less than 7 dropped packets decreases to 5.4 %.

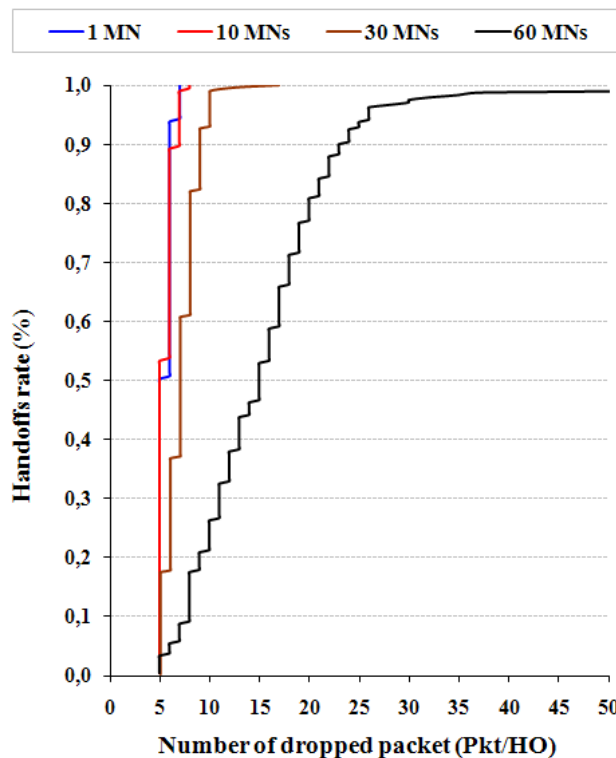


Fig 6.22: Distribution function of the number of dropped packets per handoff on uplink resulting from employing MIP in a hierarchical topology under different loads

6.5.3. Impact of Network Load on the Performance of HAWAII

Figure 6.23 presents the distribution function of the handoff latency experienced when employing HAWAII under different loads. This figure shows that HAWAII is highly affected by network load. It works well for low-loaded networks. If there is only one active MN in the network, 9.3 % and 45.3 % of all handoffs have a latency of less than 72 and 100 msec, respectively. Only 11.3 % of the handoffs require more than 150 msec. If the number of active MNs is increased to 10, HAWAII continues to work well. 11.4 %, 43.6 % and 85 % of the

handoffs are completed in less than 72, 100 and 150 msec. Increasing the load produces a significant increase in the handoff latency. Only 24.3 % and 13.6 % of all handoffs are completed in less than 100 msec if the network contains 30 and 60 active MNs, respectively. If there are only 30 active MNs in the network, 20.7 % of the handoffs require more than 150 msec. The number of handoffs requiring a latency of more than 150 msec increases to 57.9 % for networks with 60 active MNs. The results can be interpreted as follows: HAWAII control messages must travel from the new to the old AR. In addition to the processing required in the new and the old ARs, these control messages should be processed by each hop supporting HAWAII in between. This processing takes a considerable time if the network is high-loaded.

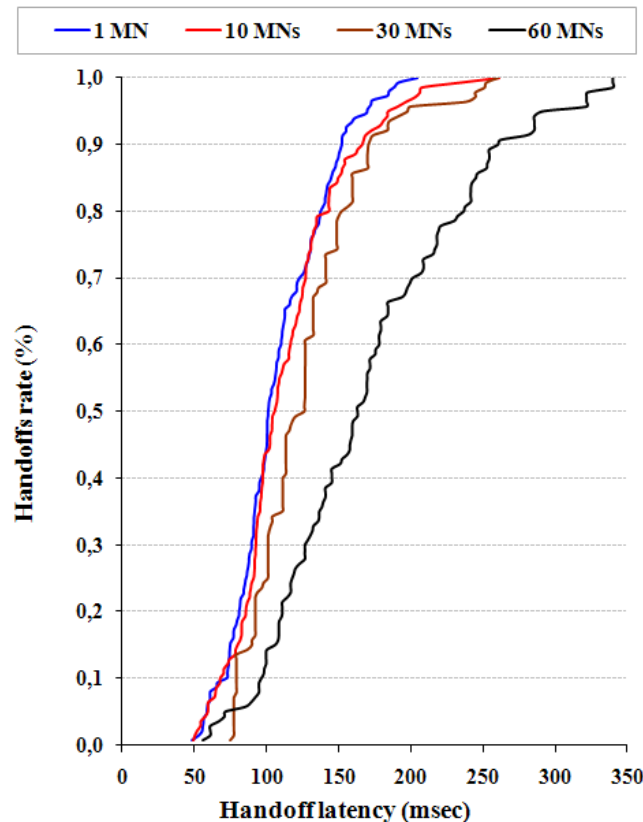


Fig 6.23: Distribution function of the handoff latency resulting from employing HAWAII in a hierarchical topology under different loads

Figure 6.24 shows the distribution function of the number of dropped packets per handoff on downlink. Again, this figure shows that HAWAII is highly affected by network load. According to the simulation results, HAWAII drops only 2 packets per handoff in 55.3 %, 38 % and 32.7 % of the handoffs for networks with 1, 10 and 30 active MNs, respectively. Less than 4 packets are dropped in 74.7 %, 66.7 % and 56.7 % of all handoffs for the previous three cases. Under high load situations, only 14.9 % of all handoffs can be executed with no more than 2 dropped packets per handoff, while no more than 3 packets per handoff are lost in 19.6 % of the handoffs. The reason behind the strong dependency on network load has been discussed above.

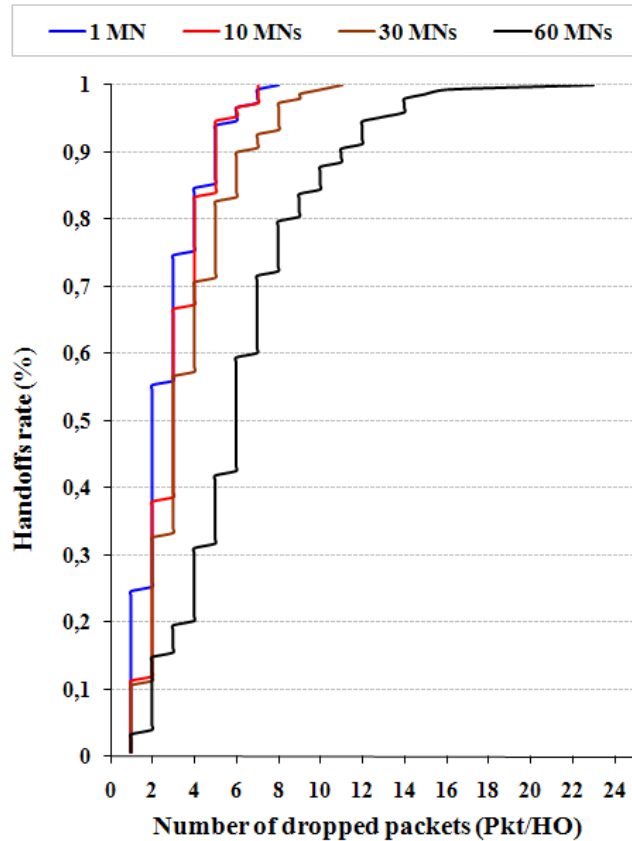


Fig 6.24: Distribution function of the number of dropped packets per handoff on downlink resulting from employing HAWAII in a hierarchical topology under different loads

Similar results are derived when studying the number of dropped packets per handoff on uplink, see figure 6.25. For low-loaded networks, 45.5 % and 39.8 % of the handoffs can be executed with no more the 6 dropped packets per handoff when the network contains 1 and 10 active MNs, respectively. Approximately 96 % of all handoffs are executed with less than 20 dropped packets per handoff as well. The impact of the load becomes noticeable when increasing the number of active MNs to 30. Only 30.5 % of the handoffs are achieved with less than 7 dropped packets per handoff, while 94.3 % of all handoffs are executed with less than 20 dropped packets per handoff. Considering a high-loaded network, it can be observed that the performance of HAWAII is highly degraded. Only 11.8 % of the handoffs produce no more than 6 dropped packets per handoff, while 80.3 % of all handoffs are achieved with less than 20 dropped packets per handoff. Notice that HAWAII drops on uplink significantly more than on downlink. This is, of course, because the number of dropped packets per handoff on downlink depends on the delay between the new AR and the crossover node, while the number of dropped packets per handoff on uplink depends on the complete handoff latency.

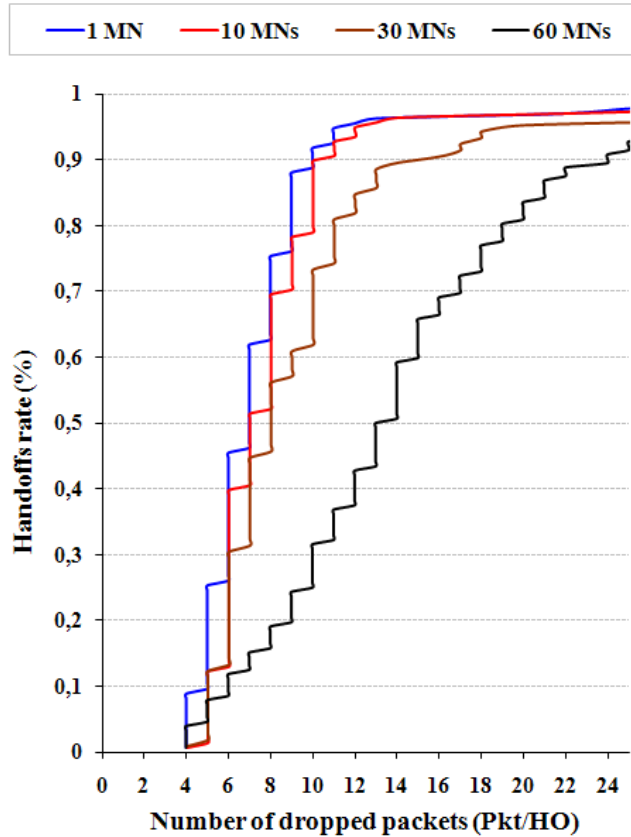


Fig 6.25: Distribution function of the number of dropped packets per handoff on uplink resulting from employing HAWAII in a hierarchical topology under different loads

6.5.4. Comparative Analysis

6.5.4.1. Average Handoff Latency

Let us now compare the average handoff latency resulting from employing HAWAII, MIP and MIFA in the studied scenario under different loads, see figure 6.26. Notice that the best performance is achieved by MIFA in predictive mode under all studied network loads. The reason for this was discussed in section 6.5.1. A fast handoff is guaranteed under low as well as high load even if MIFA should be employed in reactive mode. MIP and HAWAII are significantly outperformed by MIFA under all studied network loads. Increasing the load slightly increases the handoff latency resulting from MIFA in reactive mode, while there is a significant increase in the handoff latency employing MIP and HAWAII. The worst impact of the load is seen with HAWAII. For networks with 1 active MN, MIFA in reactive mode performs on downlink 55.3 % and 50.2 % better than MIP and HAWAII, respectively. On uplink, MIFA in reactive mode is 92.9 % and 92.1 % faster than MIP and HAWAII, respectively. MIFA itself performs in this mode on uplink 84 % better than on downlink. For networks with 10 active MNs, MIFA in reactive mode is 53.3 % and 51.2 % faster on downlink than MIP and HAWAII, respectively. On uplink, MIFA performs approximately 92 % better in this mode than both MIP and HAWAII. For MIFA itself, it is 83.3 % faster on uplink than on downlink. For networks with 30 active MNs, the simulation results have shown that MIP is 56 % slower on downlink than MIFA in reactive mode, while HAWAII is 54.3 % slower. On uplink, MIFA in reactive mode is approximately 90 % better than MIP and HAWAII. Furthermore, it is 77.1 % faster on uplink than on downlink. Considering networks with 60 active MNs, MIFA performs in reactive mode 60.3 % better on downlink and 64.5 %

better than MIP and HAWAII, respectively. On uplink, it is 90.5 % and 91.4 % better than MIP and HAWAII, respectively. MIFA itself performs in reactive mode 75.8 % better on uplink than on downlink.

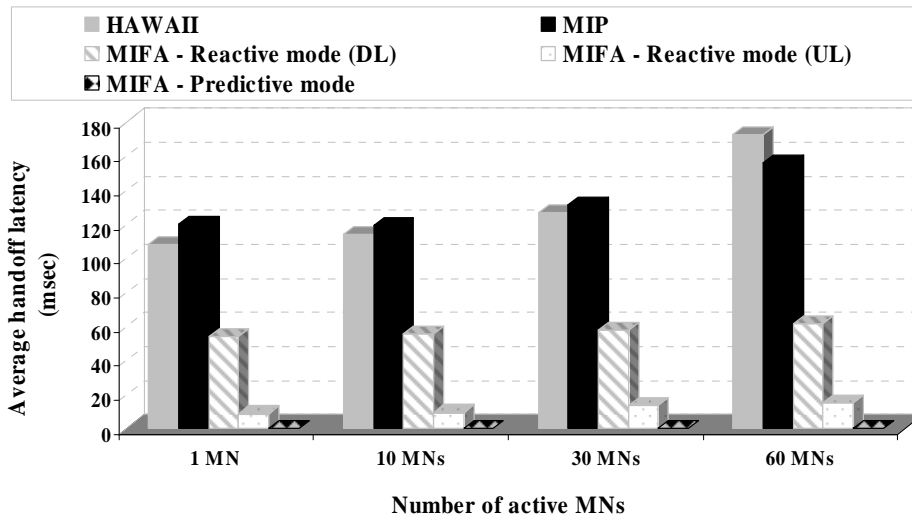


Fig 6.26: Average handoff latency resulting from employing HAWAII, MIP and MIFA in a hierarchical topology under different loads

Let us now analyze the impact of network load on the average handoff latency in more detail, see figure 6.27. This figure assumes that the reference value is the average handoff latency resulting from employing each protocol in a network containing only one active MN. An increase in the average handoff latency due to increasing the load is measured, accordingly, as a ratio to this reference value.

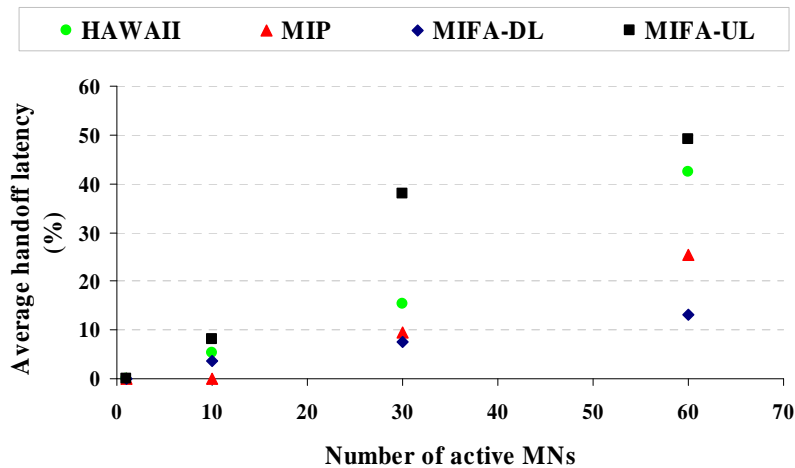


Fig 6.27: Impact of network load on the average handoff latency resulting from employing HAWAII, MIP and MIFA in reactive mode in a hierarchical topology under different loads

This figure shows that increasing the number of active MNs from 1 to 10 increases the handoff latency experienced by MIFA in reactive mode by 3.5 % on downlink and 8 % on uplink. HAWAII experiences an increase of 5.4 %, while MIP remains unchanged. For 30 active MNs, the average handoff latency experienced by MIP increases by 9.4 %, while HAWAII experiences an increase of 15.4 %. Considering MIFA in reactive mode, the average handoff latency increases by 7.5 % on downlink and 38 % on uplink. Increasing the number of active MNs to 60 produces an increase of 25 %, 42 %, 13 % and 49 % in the average handoff latency experienced by MIP, HAWAII, MIFA in reactive mode on downlink and MIFA in reactive mode on uplink, respectively. Notice that the biggest increase is experienced by MIFA in reactive mode on uplink. The reason behind this is the small handoff

latency resulting from MIFA in this mode, which makes any small increase due to network load remarkable.

6.5.4.2. Average Number of Dropped Packets Per Handoff

Figure 6.28 shows the average number of dropped packets per handoff on downlink and uplink employing HAWAII, MIP and MIFA in the studied scenario under different loads. Again, MIFA performs best in predictive mode and guarantees seamless handoffs in reactive mode, especially on uplink. MIP and HAWAII are outperformed by MIFA under all studied loads. Notice that HAWAII drops significantly more on uplink than on downlink. In contrast, MIP drops on uplink comparable to what it drops on downlink in low- and middle-loaded networks. In high-loaded networks, MIP drops on uplink more than on downlink. The long handoff latency under this load and the dropping of control messages due to high network load are the reasons behind this behavior. MIFA drops significantly less on uplink than on downlink. The reason for this has been discussed previously.

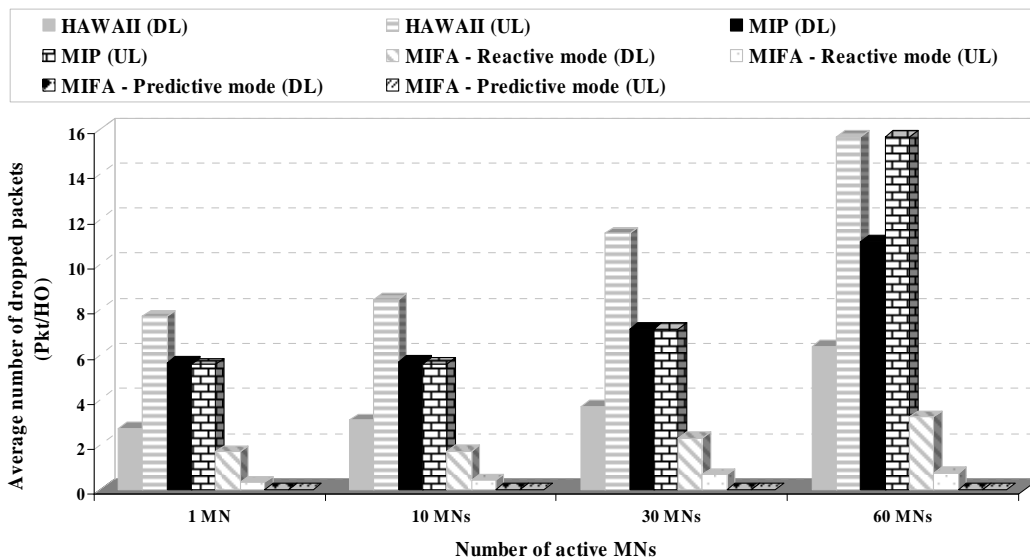


Fig 6.28: Average number of dropped packets per handoff resulting from employing HAWAII, MIP and MIFA in a hierarchical topology under different loads

Let us consider networks with 1 active MN. Regarding the average number of dropped packets per handoff on downlink, employing MIFA in reactive mode results in 37.7 % and 70 % fewer dropped packets than HAWAII and MIP, respectively. On uplink, MIP is outperformed by 94 %, while HAWAII is outperformed by 95.7 %. Considering MIFA itself, it drops in this mode 80.4 % less on uplink than on downlink. For networks with 10 active MNs, the simulation results have shown that MIFA in reactive mode drops on downlink 54.3 % and 69.3 % less than HAWAII and MIP, respectively. On uplink, HAWAII drops 95 % more than MIFA in reactive mode, while MIP drops 92.4 % more. MIFA itself drops in this mode 75.4 % less on uplink than on downlink. HAWAII drops in networks containing 30 active MNs 38.2 % more on downlink and 94.2 % more on uplink than MIFA in reactive mode, while MIP results in 68 % and 90.7 % more dropped packets per handoff than MIFA in reactive mode on downlink and uplink, respectively. MIFA drops in this mode 70.8 % more on downlink than on uplink. Considering networks with 60 active MNs, MIFA in reactive mode drops 49 % less on downlink than HAWAII and 70.7 % less than MIP. With respect to the average number of dropped packets per handoff on uplink, MIFA drops 95.2 % less in reactive mode than HAWAII and MIP. Moreover, it drops 76.8 % more on downlink than on uplink.

Let us now analyze in more detail the impact of increasing the network load on the average number of dropped packets per handoff experienced when employing the three studied protocols, see figure 6.29. Similar to figure 6.27, the reference value is the average number of dropped packets per handoff resulting from employing each protocol in a network containing only one active MN. The increase in the average number of dropped packets is measured, accordingly, as a ratio to this reference value. This figure shows that the impact of the load on the performance of MIFA in reactive mode and MIP is comparable with respect to the average number of dropped packets per handoff on downlink. Regarding the average number of dropped packets per handoff on uplink, MIFA in reactive mode is more affected by network load than MIP for low- and middle-loaded networks. This not because MIFA in reactive mode is more load-sensitive than MIP. The small value of the average number of dropped packets per handoff resulting from MIFA, however, makes any small increase due to network load more remarkable than by MIP. In high-loaded networks, MIP is more affected by the load than MIFA in reactive mode. The reason for this has been discussed previously. Compared to HAWAII, MIFA in reactive mode is in most cases less affected by network load than HAWAII with respect to the average number of dropped packets per handoff on downlink. The reason for this is that only the ARs should support MIFA, while all domain nodes should be HAWAII-aware. Taking the average number of dropped packets per handoff on uplink into account, MIFA in reactive mode is more affected by the load than HAWAII. Again, this does not mean that MIFA is more load-sensitive than HAWAII. However, the small average number of dropped packets per handoff resulting from MIFA in this mode makes any small increase due to the load remarkable.

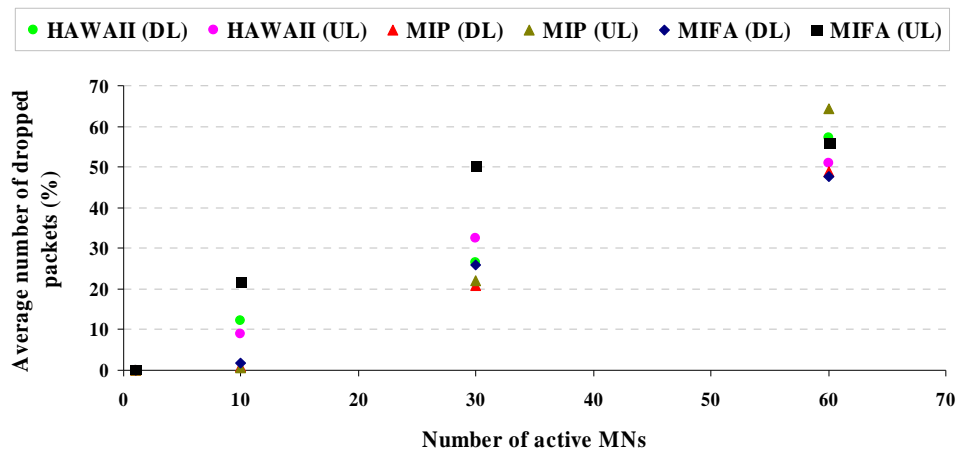


Fig 6.29: Impact of network load on the average number of dropped packets per handoff on downlink and uplink resulting from employing HAWAII, MIP and MIFA in reactive mode in a hierarchical topology under different loads

6.5.4.3. Average Congestion Window Size

As known, packet dropping highly affects the performance of TCP, which assumes a network congestion and reacts by executing the slow start mechanism. Increasing the load produces more packet dropping. This forces TCP to employ the slow start mechanism more often resulting in a reduction of the average congestion window size of the TCP connection. This is shown in figure 6.30, which shows the average congestion window size for the studied TCP connection employing MIP and MIFA in the studied scenario under different loads. Notice that the best performance in a network containing only one active MN is achieved by MIFA in predictive mode. The average congestion window size resulting from MIFA in predictive mode is more than that resulting from MIFA in reactive mode, which in turn clearly outperforms MIP. According to the simulation results, the average congestion window size resulting from employing MIFA in predictive mode is 26.8 % and 55.3 % more than that

resulting from MIFA in reactive mode and MIP, respectively. The average congestion window size resulting from MIP is 22.5 % less than that resulting from MIFA in reactive mode.

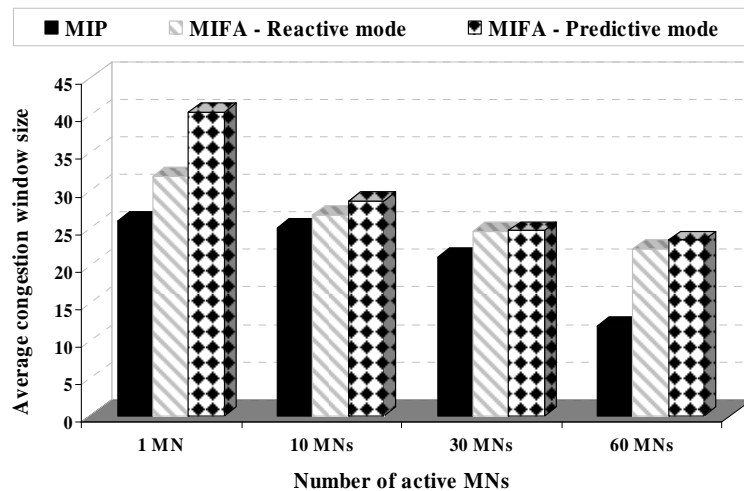


Fig 6.30: Average congestion window size resulting from employing MIP and MIFA in a hierarchical topology under different loads

Considering a network with 10 active MNs, it can be observed that the average congestion window size decreases with both MIFA and MIP. MIFA in predictive mode still performs better than MIFA in reactive mode, which continues to perform better than MIP. However, the difference between the average congestion window size resulting from employing MIFA in predictive and reactive mode grows smaller. According to the simulation results, the average congestion window size resulting from MIFA in predictive mode is only 7.1 % and 14.4 % better than that resulting from MIFA in reactive mode and MIP, respectively. MIFA performs in reactive mode 6.8 % better than MIP. For networks with 30 active MNs, MIFA performs in predictive mode same as in reactive mode. Compared to MIP, MIFA performs 16 % better. In networks with 60 active MNs, MIFA is only 5.1 % better in predictive mode than in reactive mode. Compared to MIP, MIFA in predictive mode is 96 % better, while MIFA in reactive mode is 86.4 % better. Notice that increasing the load causes the performance of MIFA in predictive mode to be comparable to its performance in reactive mode. The reason for this is that most executed slow start mechanisms employing MIFA are executed because of packet dropping due to the load and not due to the handoffs themselves.

Let us now analyze the impact of network load on the average congestion window size in more detail, see figure 6.31.

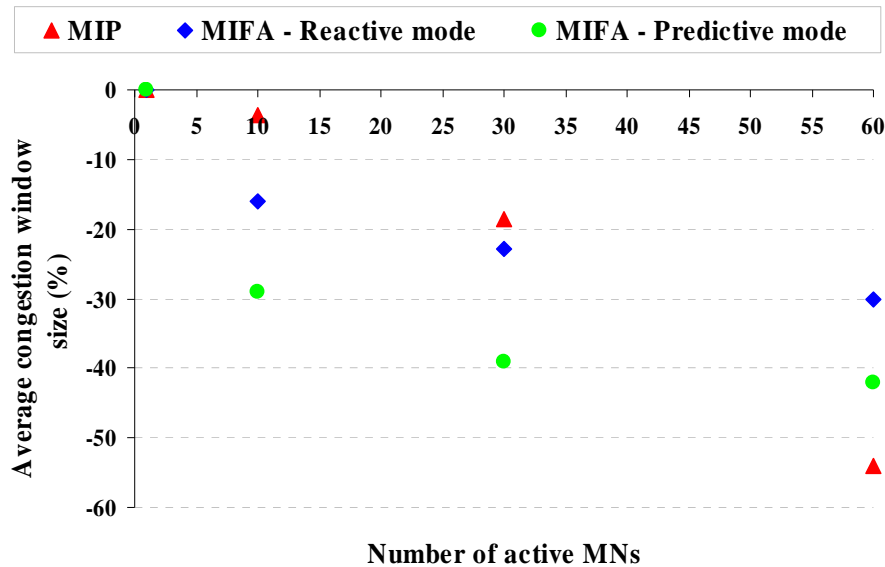


Fig 6.31: Impact of network load on the congestion window size resulting from employing MIP and MIFA in reactive mode in a hierarchical topology under different loads

We assume that the reference value is the average congestion window size resulting from employing each protocol in a network containing only one active MN. Any decrease in the average congestion window size due to increasing network load is calculated as a ratio to the reference value. This figure shows that increasing the number of active MNs from 1 to 10 produces a small decrease in the average congestion window size when employing MIP, 3.7 % according to our simulation. However, a remarkable decrease, 16 %, is seen by MIFA in reactive mode. The biggest decrease results when employing MIFA in predictive mode, 29 % in our simulation. These behaviors can be interpreted as follows: in a network containing only one active MN, there is no packet dropping due to nodes congestion. The dropping results only from the handoffs themselves. For MIP, there are dropped packets in most handoffs during the studied TCP connection due to the resulting long handoff latency. This results in frequent execution of the slow start mechanism and, therefore, performance degradation. MIFA, on the other hand, achieves lossless handoffs in predictive mode and, thus, no slow start mechanism is executed due to handoffs themselves. Packets' dropping employing MIFA in reactive mode is sometimes, however, possible. Therefore, MIFA performs less well in this mode than in predictive mode, yet still better than MIP. Although the load in a network containing 10 active MNs is still relatively low, it is possible that some packets may get lost due to the load, which results in executing the slow start mechanism sometimes. This degrades, of course, the performance of MIP as well as MIFA. This degradation, however, is clearly remarkable by MIFA in predictive mode. The reason for this is because MIFA has not suffered from the slow start mechanism in this mode as the network contained one active MN, which makes any small change remarkable. In a similar way, the degradation in performance experienced when employing MIFA in reactive mode and MIP can be interpreted. Under high load situations, 60 active MNs in this study, MIP drops many packets. This produces long communication disruptions and forces TCP to employ the slow start mechanism often. Therefore, the performance of MIP is degraded clearly more than that of MIFA in both modes with respect to the average congestion window size.

6.5.5. Summary

The main obtained results can be summarized as follows: considering real-time traffic, increasing the network load results in a remarkable increase in the handoff latency and so in the number of dropped packets per handoff. For non-real-time traffic, increasing the load

produces more dropped packets and so forces TCP to employ the slow start mechanism more often, which reduces the average congestion window size. Regarding the real-time traffic, the best performance is achieved by MIFA in predictive mode under all studied network loads. The layer 3 handoff latency is eliminated and so lossless handoffs are achieved. This is, of course, due to the sufficient time the MN spends inside the overlapping area. Therefore, a good network planning is essential. MIFA in reactive mode performs significantly better than MIP and HAWAII on downlink as well as on uplink. Considering non-real-time traffic, MIFA also performs better than MIP. However, increasing the load makes the performance of MIFA in predictive mode comparable to its performance in reactive mode. The reason behind this behavior is that most executed slow start mechanisms employing MIFA are executed because of packet dropping due to the load and not due to the handoffs themselves.

6.6. Impact of MN Speed

In order to study the impact of MN speed, the hierarchical topology presented in figure 6.1 is used. 60 MNs are made active, while the other 100 remain in idle mode. The active MNs exchange UDP traffic with the CNs. An active MN that moves from AR0 to AR15 is observed. Again, a downlink and an uplink constant bit rate UDP stream with a packet arrival rate of 50 packets per second and a packet length of 500 bytes are exchanged between CN0 and the observed MN. Throughout the simulation, the load is not changed and the other MNs do not move. The speed of the observed MN is changed among a speed of a pedestrian (3 km/h), a cyclist (20 km/h), a car driver inside a city (50 km/h), a car driver on an autobahn (120 km/h) and a high speed device (300 km/h).

6.6.1. Impact of MN Speed on the Performance of MIFA

Figure 6.32 presents the distribution function of the handoff latency on downlink resulting from employing MIFA in reactive mode in the studied scenario. This figure shows that MN speed has a significant impact on the performance of MIFA with respect to the handoff latency. MIFA performs comparably at a speed of 20 and 50 km/h. Approximately 86 % of all executed handoffs can be completed in less than 90 msec. However, MIFA shows slightly better performance at these speeds than at 3 km/h. This is mainly due to the ping pong effect that arises clearly at low speeds. Handoffs at a speed of 120 km/h take almost more time to be finished than at 3, 20 and 50 km/h. This is because the probability that the MN will leave the overlapping area before receiving any advertisement from the new AR is relatively high, which in turn increases the movement detection time. However, even at this speed, MIFA is able to finish 69 % of the handoffs in less than 90 msec. The handoff latency increases significantly when the MN moves at a speed of 300 km/h. This results mainly from the long movement detection time that increases significantly at such speeds.

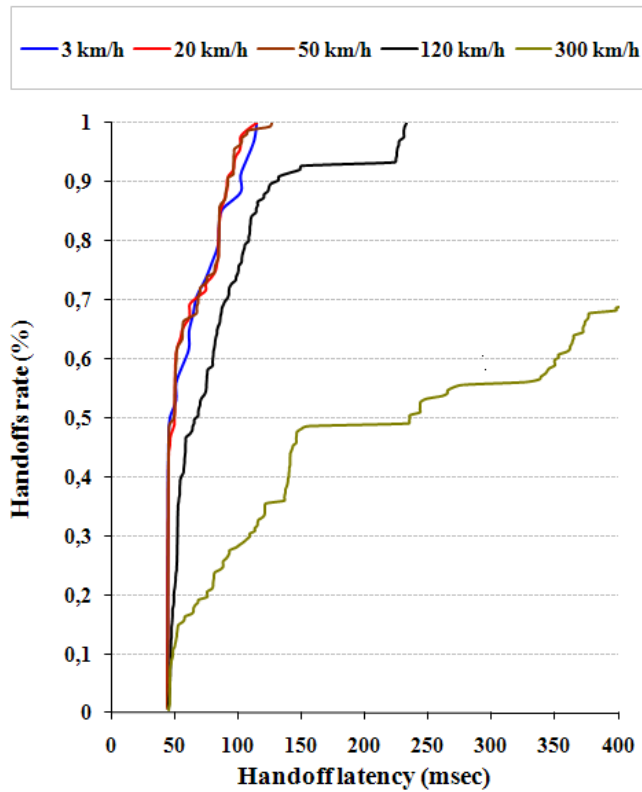


Fig 6.32: Distribution function of the handoff latency on downlink resulting from employing MIFA in reactive mode in a hierarchical topology while moving at different speeds

Let us now study the handoff latency on uplink resulting from employing MIFA in reactive mode at the mentioned speeds, see figure 6.33.

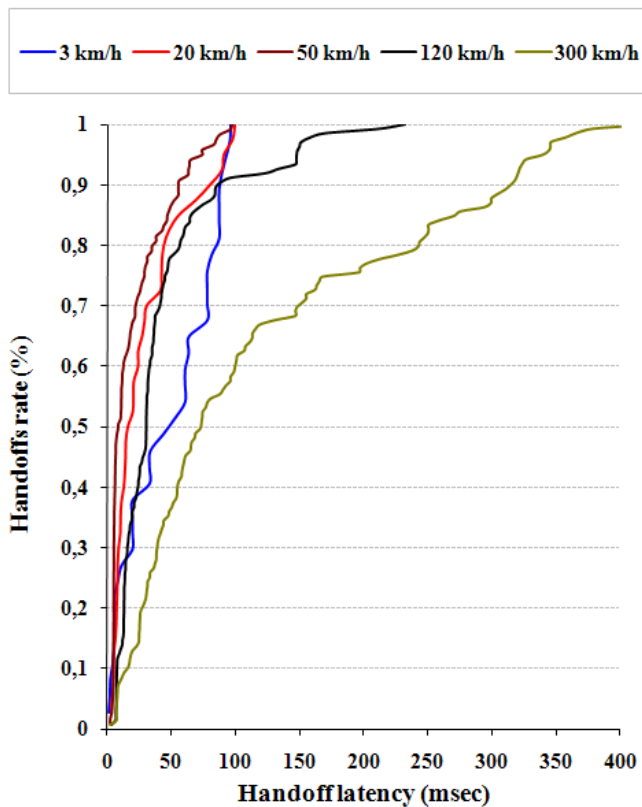


Fig 6.33: Distribution function of the handoff latency on uplink resulting from employing MIFA in reactive mode in a hierarchical topology while moving at different speeds

This figure shows that the impact of the ping pong effect is significantly stronger on uplink than on downlink. As the MN moves away from the current AR, the wireless link quality degrades more and more. Therefore, and due the used movement detection method (ECS), long delays will be experienced when the MN attempts to register with this AR after registering with a new one. Therefore, MIFA does not perform very well at a speed of 3 km/h. Only 37.8 % of the handoffs can be finished in less than 20 msec. The impact of the ping pong effect decreases with increasing MN speed. This is why MIFA performs better at a speed of 20 km/h than at 3 km/h. The handoff latency does not exceed 20 msec in 52.5 % of all handoffs. Further performance improvement can be achieved by increasing the MN speed to 50 km/h. At this speed, 67.2 % of all handoffs need less than 20 msec to be completed. At a speed of 120 km/h, the ping pong effect is no longer visible. Although the performance of MIFA in reactive mode at this speed begins to deteriorate somewhat, MIFA still performs better than at 3 km/h for 90.2 % of all handoffs. At a speed of 300 km/h, the performance of MIFA deteriorates significantly for the same reason presented while discussing the handoff latency on downlink, see figure 6.32.

For the handoff latency on downlink and uplink resulting from employing MIFA in predictive mode, see figures 6.34 and 6.35, we see that MIFA can eliminate the layer 3 handoff latency when moving at speeds of 3, 20 and 50 km/h. This is because the MN remains inside the overlapping area at these speeds long enough to trigger the predicted mode even in the case of control messages dropping on the wireless link. Even if the MN moves at higher speeds (120 and 300 km/h), MIFA can still be employed in predictive mode in most cases. At a speed of 120 km/h, MIFA can be employed in predictive mode in 72.2 % and 77.5 % of the handoffs on downlink and uplink, respectively, while at a speed of 300 km/h, MIFA can trigger the predictive mode in approximately 58 % of all handoffs.

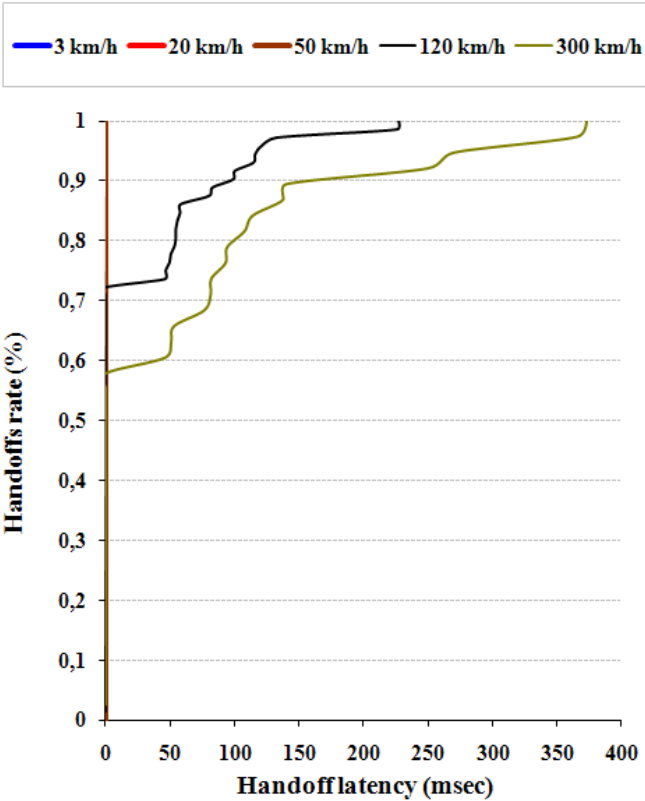


Fig 6.34: Distribution function of the handoff latency on downlink resulting from employing MIFA in predictive mode in a hierarchical topology while moving at different speeds

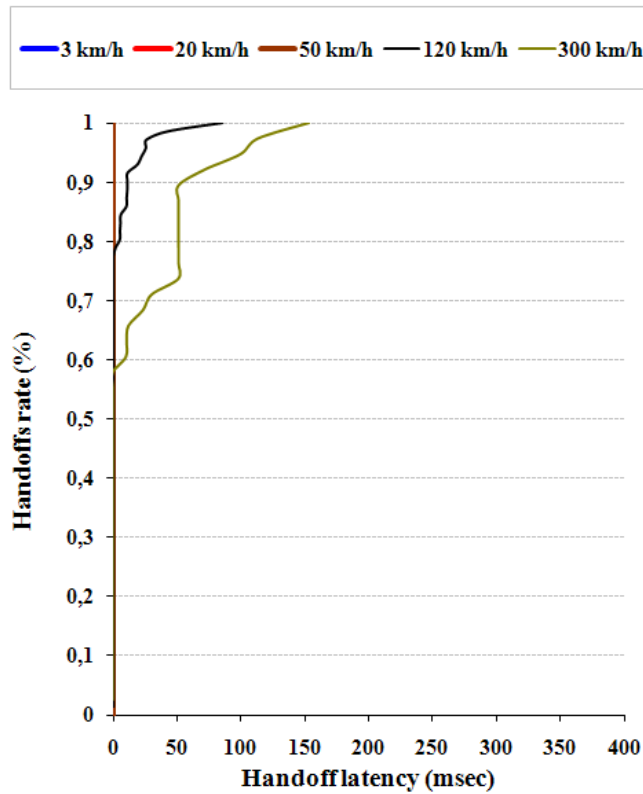


Fig 6.35: Distribution function of the handoff latency on uplink resulting from employing MIFA in predictive mode in a hierarchical topology while moving at different speeds

The number of dropped packets per handoff on downlink and uplink when employing MIFA in reactive mode is presented in figure 6.36 and figure 6.37, respectively.

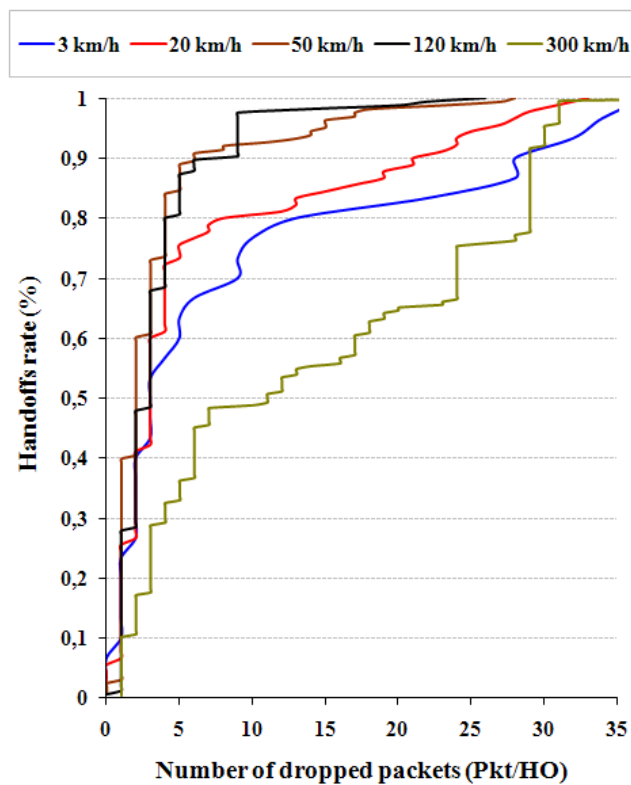


Fig 6.36: Distribution function of the number of dropped packets per handoff on downlink resulting from employing MIFA in reactive mode in a hierarchical topology while moving at different speeds

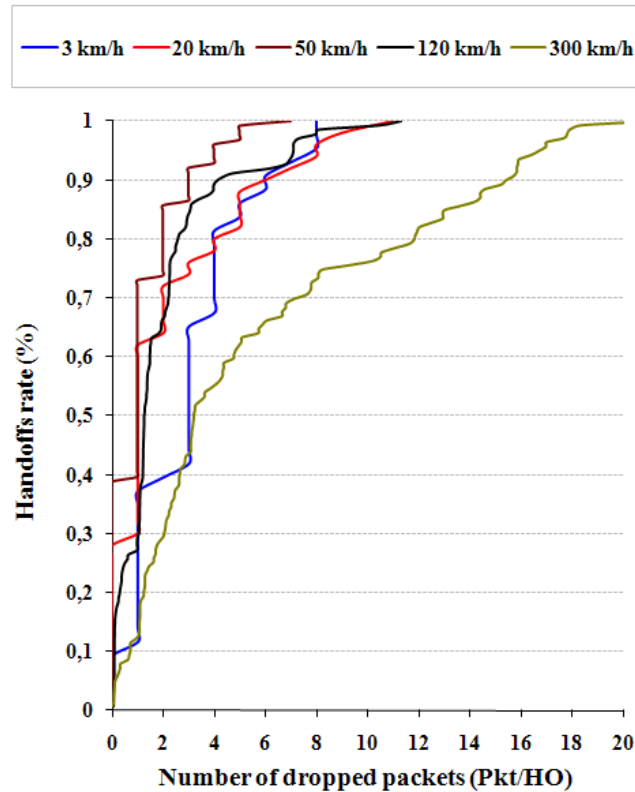


Fig 6.37: Distribution function of the number of dropped packets per handoff on uplink resulting from employing MIFA in reactive mode in a hierarchical topology while moving at different speeds

Let us first analyze dropped packets per handoff on the downlink. MIFA performs comparable at speeds of 50 and 120 km/h. Decreasing the speed degrades performance. MIFA is better at speeds of 120 and 50 km/h than at a speed of 20 km/h, which in turns is even better than when moving at a speed of 3 km/h. The reason for this is the ping pong effect, which appears clearly at slow speeds and produces more dropped packets during the handoff. Considering a speed of 300 km/h, MIFA produces significantly more dropped packets per handoff than at the other studied speeds. This is because of the long movement detection time resulting from moving very fast through the cell and the overlapping area. Similar results can be derived regarding the dropped packets per handoff on uplink.

Let us now study the number of dropped packets per handoff on downlink and uplink when attempting to use MIFA in predictive mode before employing the reactive mode, see figure 6.38 and figure 6.39. Regarding the dropped packets per handoff on downlink, MIFA in predictive mode can avoid packet dropping due to the layer 3 handoff latency when moving at a speed of 3, 20 and 50 km/h. Even at a higher speed, 120 km/h in this study, MIFA is able to operate in predictive mode in most handoffs, 70.8 % of all handoffs according to the simulation results. At a speed of 300 km/h, however, the number of dropped packets per handoff experiences a clear increase. Similar results can be seen with respect to dropped packets per handoff on uplink when the MN moves at a speed of 3, 20 and 50 km/h. At 120 and 300 km/h, however, MIFA produces lossless handoffs in most cases. According to the simulation results, 84.7 % and 63.6 % of the achieved handoffs are lossless when moving at a speed of 120 and 300 km/h, respectively. The reason is the very fast handoff on uplink that can be achieved by MIFA regardless of whether the predictive mode could be operated or not.

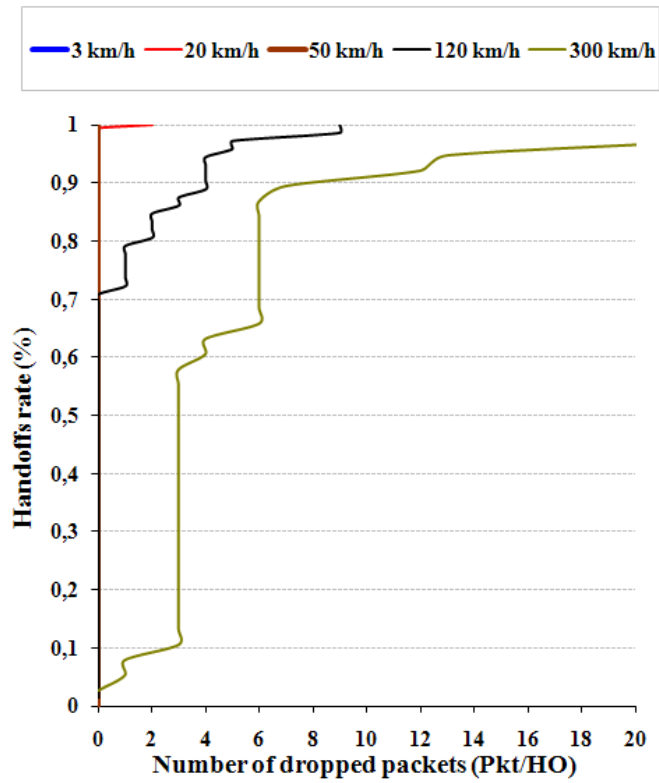


Fig 6.38: Distribution function of the number of dropped packets per handoff on downlink resulting from employing MIFA in predictive mode in a hierarchical topology while moving at different speeds

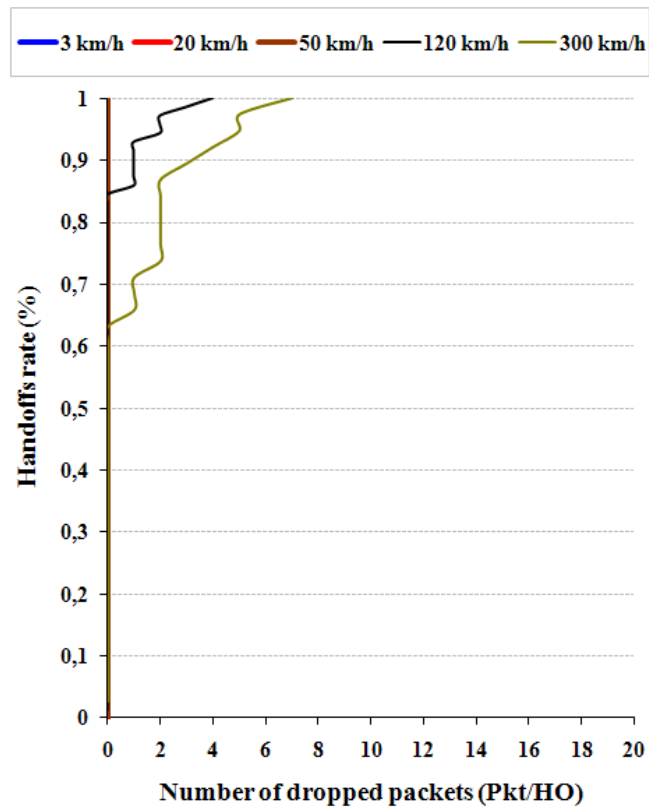


Fig 6.39: Distribution function of the number of dropped packets per handoff on uplink resulting from employing MIFA in predictive mode in a hierarchical topology while moving at different speeds

6.6.2. Impact of MN Speed on the Performance of MIP

Figure 6.40 shows the handoff latency resulting from employing MIP in the studied scenario. Notice that MIP performs comparably at speeds of 3, 20 and 50 km/h. Compared to MIFA, approximately 86 % of all handoffs can be achieved in less than 90 msec on downlink when employing MIFA in reactive mode, while the handoff latency in less than 1 % of the handoffs does not exceed 90 msec when employing MIP. MIFA in reactive mode is even better on uplink than on downlink, 91.2 %, 92.5 % and 98.3 % of the handoffs can be finished in a latency of less than 90 msec at speeds of 3, 20 and 50 km/h, respectively. Employing MIFA in predictive mode eliminates the layer 3 handoff latency at these speeds. Similar to MIFA, increasing the speed results in a significant increase in the handoff latency resulting from MIP. The reason for this is the long movement detection time resulting when moving at high speeds through the cells.

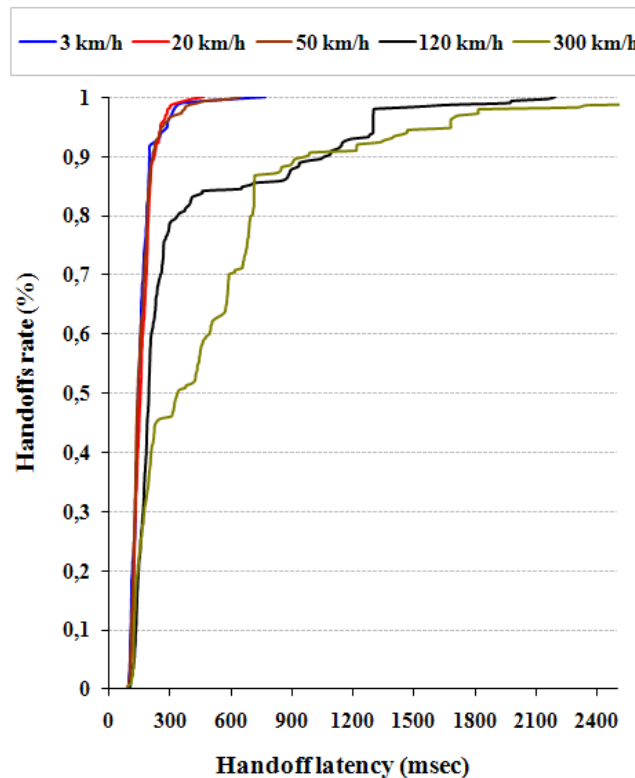


Fig 6.40: Distribution function of the handoff latency resulting from employing MIP in a hierarchical topology while moving at different speeds

Figure 6.41 and figure 6.42 present the distribution function of the number of dropped packets per handoff using MIP on downlink and uplink, respectively.

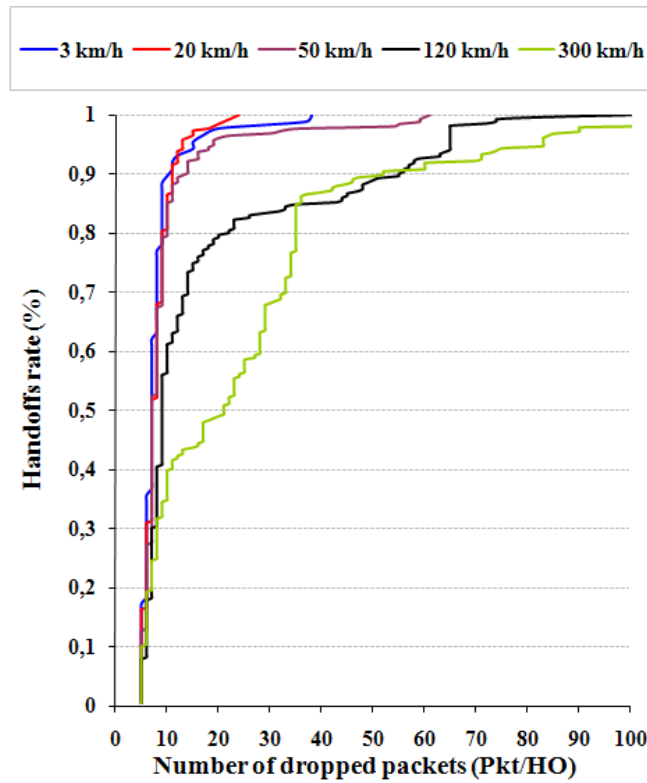


Fig 6.41: Distribution function of the number of dropped packets per handoff on downlink resulting from employing MIP in a hierarchical topology while moving at different speeds

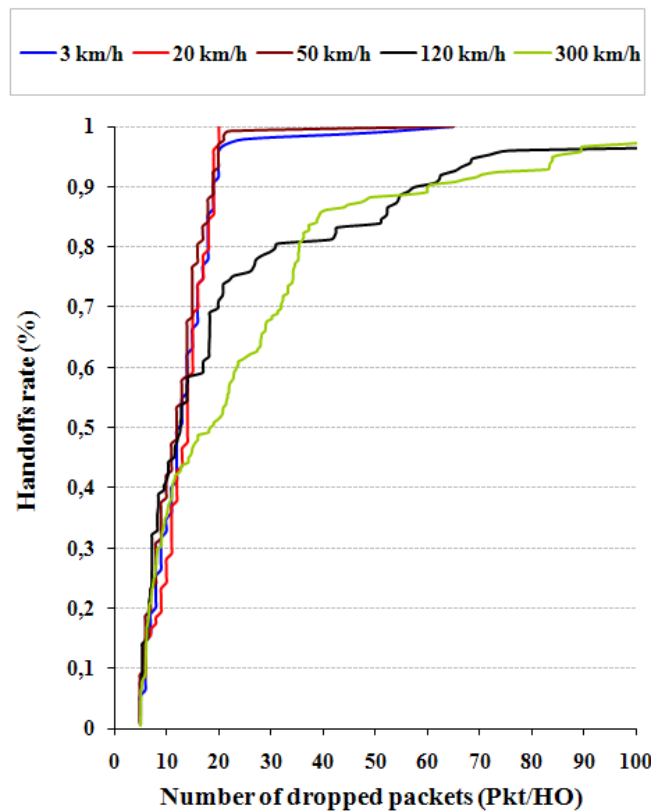


Fig 6.42: Distribution function of the number of dropped packets per handoff on uplink resulting from employing MIP in a hierarchical topology while moving at different speeds

Again, MIP functions comparably at speeds of 3, 20 and 50 km/h with respect to the number of dropped packets per handoff. In contrast to MIFA, the impact of the ping pong effect is

very small and even unnoticeable when employing MIP. This is because MIFA achieves almost faster handoffs than MIP. Therefore, data packets will be quickly redirected between the old and the new AR many times during the ping pong handoff, which degrades the performance and makes the ping pong effect noticeable, especially at low speeds. Due to the long handoff latency experienced when employing MIP, the redirecting of packets takes relatively longer times than with MIFA, which reduces the impact of the ping pong effect. Increasing the speed further to 120 km/h degrades the performance of MIP significantly, while MIFA performs very well at this speed. Similar to MIFA, the performance of MIP is degraded significantly at a very high speed (300 km/h in this study) due to the long movement detection time.

6.6.3. Impact of MN Speed on the Performance of HAWAII

Figure 6.43 presents the distribution function of the handoff latency resulting from employing HAWAII in the studied scenario.

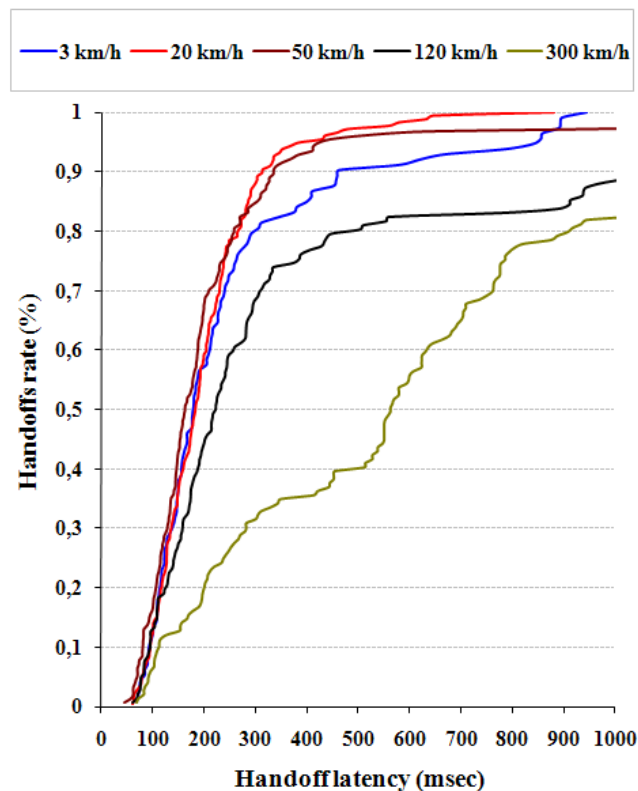


Fig 6.43: Distribution function of the handoff latency resulting from employing HAWAII in a hierarchical topology while moving at different speeds

Notice that HAWAII performs comparably at a speed of 20 and 50 km/h. In addition, its performance at these speeds is comparable to its performance at a speed of 3 km/h in about 60 % of all handoffs. In the remaining handoffs, HAWAII performs at a speed of 20 and 50 km/h better than at 3 km/h. Similar results were observed with MIFA and are due to the ping pong effect. Let us compare the handoff latencies at these speeds to their counterparts employing MIFA. MIFA in predictive mode eliminates the layer 3 handoff latency at these speeds. MIFA in reactive mode can complete all handoffs on uplink in less than 100 msec at the mentioned speeds. On downlink, 85.3 %, 94.3 % and 95.6 % of the handoffs can be completed in less than 100 msec when moving at 3, 20 and 50 km/h, respectively. In contrast, the handoff latency in 11.5 %, 11.9 % and 17.5 % of the handoffs on uplink and downlink does not

exceed 100 msec when employing HAWAII while moving at 3, 20 and 50 km/h, respectively. Increasing MN speed produces a clear increase in the handoff latency and is, as mentioned before, due to the resulting long movement detection time. At a speed of 120 km/h, MIFA finishes 90.3 % of the downlink handoffs in less than 100 msec when it attempts to use the predictive mode before triggering the reactive mode in case of failures. On uplink, all handoffs in this case do not exceed 100 msec. Even if the MN can operate only in the reactive mode, MIFA does not require more than 100 msec to complete 73.4 % and 91.2 % of the handoffs on downlink and uplink, respectively. Regarding HAWAII, only 12.7 % of the handoffs can be finished in less than 100 msec on uplink as well as on downlink. At a speed of 300 km/h, 78.9 % of the downlink handoffs and 94.7 % of the uplink handoffs can be finished in less than 100 msec when employing MIFA in predictive mode. Regarding MIFA in reactive mode, 28 % of the downlink and 60.4 % of the uplink handoffs do not require more than 100 msec for completion. With HAWAII, only 6 % of the handoffs can be completed in less than 100 msec.

Figure 6.44 and figure 6.45 show the distribution function of the number of dropped packets per handoff on downlink and uplink when employing HAWAII in the studied scenario.

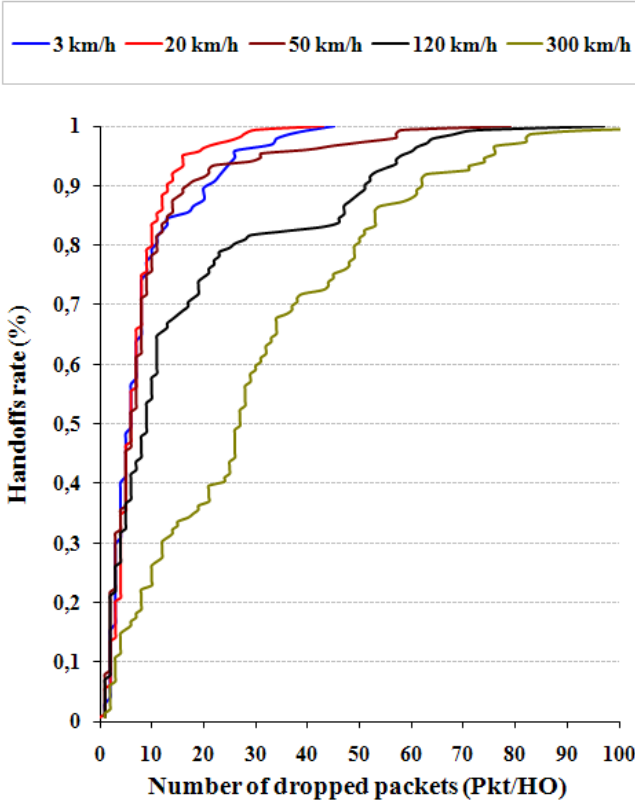


Fig 6.44: Distribution function of the number of dropped packets per handoff on downlink resulting from employing HAWAII in a hierarchical topology while moving at different speeds

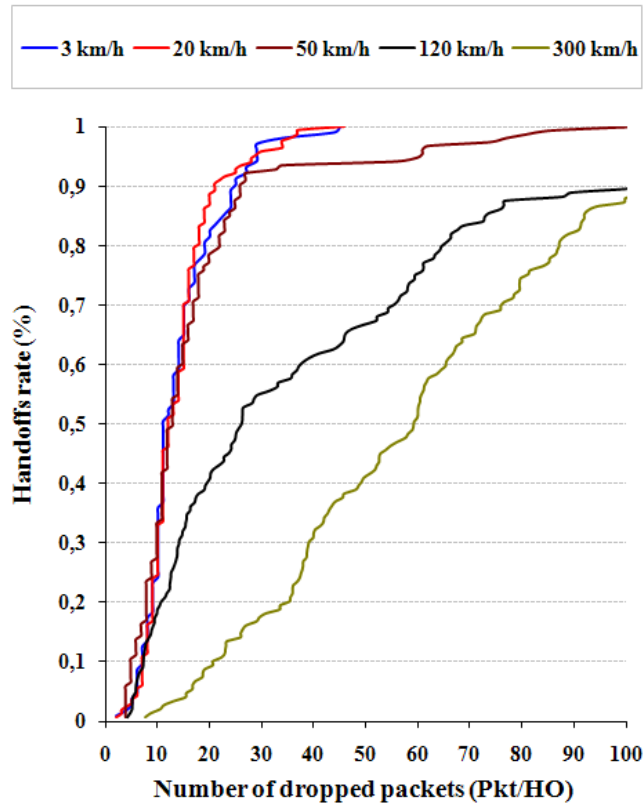


Fig 6.45: Distribution function of the number of dropped packets per handoff on uplink resulting from employing HAWAII in a hierarchical topology while moving at different speeds

The results that can be derived from these both figures are similar to the results of figure 6.43. Regarding the number of dropped packets per handoff on downlink, HAWAII performs at 3, 20 and 50 km/h comparably in approximately 80 % of the handoffs. In the remaining handoffs, HAWAII performs at 20 km/h better than at 50 km/h, which in turn performs sometimes better than at 3 km/h. This is, of course, due to the ping pong effect. Similar to the handoff latency, the number of dropped packets increases significantly with increasing MN speed. Similar results can be observed regarding the number of dropped packets per handoff on uplink.

6.6.4. Comparative Analysis

6.6.4.1. Average Handoff Latency

Figure 6.46 illustrates the average handoff latency resulting from employing HAWAII, MIP and MIFA while moving at the mentioned speeds.

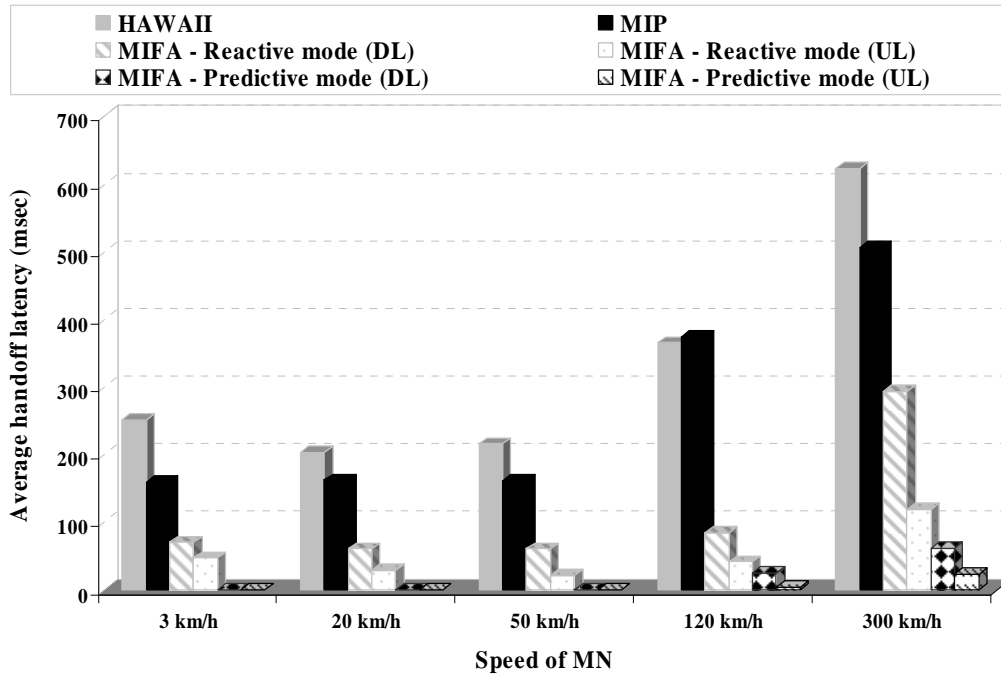


Fig 6.46: Average handoff latency resulting from employing HAWAII, MIP and MIFA in a hierarchical topology while moving at different speeds

As can be seen in this figure, the best performance is achieved by MIFA in predictive mode at all studied MN speeds. The time the MN spends inside the overlapping area while moving at 3, 20 and 50 km/h is sufficient to trigger the layer 3 handoff in advance even in case of dropping of the *BU* message on the old wireless link. When the MN moves at a speed of 120 and 300 km/h, MIFA remains able to operate in predictive mode in most cases. According to our simulation results, the probability that MIFA can trigger the layer 3 handoff in advance is 77.5 % and 57.9 % for 120 and 300 km/h, respectively. If MIFA can operate only in reactive mode, better performance than MIP and HAWAII is still achieved at all studied speeds. Compared to MIP, MIFA in reactive mode performs on downlink 56.1 %, 62.6 %, 62.6 %, 77.4 % and 42 % better when the MN moves at 3, 20, 50, 120 and 300 km/h, respectively. On uplink, MIFA in reactive mode outperforms MIP by 70.5 %, 82 %, 87.2 %, 89 % and 76.5 % when moving at the above mentioned speeds, respectively. Comparing to HAWAII, it is outperformed by MIFA in reactive mode by 72.1 %, 70.2 %, 72.1 %, 77 % and 52.8 % with respect to the average handoff latency on downlink when moving at 3, 20, 50, 120 and 300 km/h, respectively. Regarding the uplink handoff latency, MIFA in reactive mode is 81.3 %, 85.6 %, 90.5 %, 89.7 % and 80.9 % better than HAWAII when moving at the mentioned speeds, respectively.

The ping pong effect can be observed clearly when using MIFA and HAWAII, e.g. increasing the speed from 3 to 20 km/h reduces the handoff latency by 19.1 % when employing HAWAII. For MIFA in reactive mode, this increase results in a decrease in the handoff latency by 13.4 % on downlink and 37.7 % on uplink.

6.6.4.2. Average Number of Dropped Packets Per Handoff

Let us now compare the average number of dropped packets per handoff resulting from the three protocols, see figure 6.47.

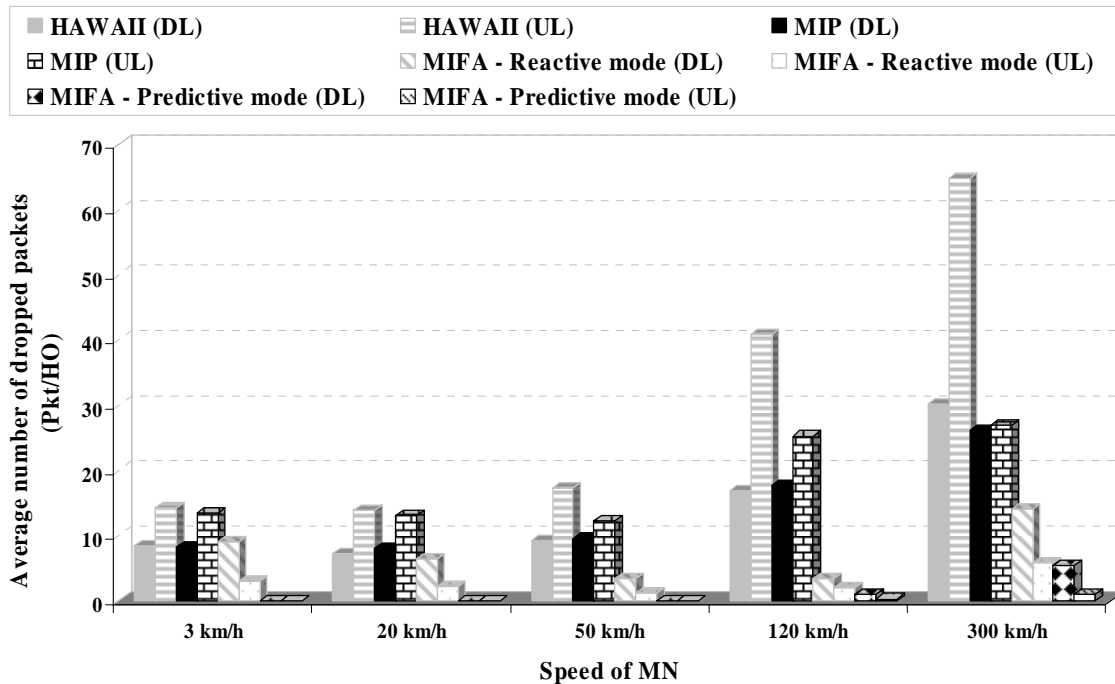


Fig 6.47: Average number of dropped packets per handoff resulting from employing HAWAII, MIP and MIFA in a hierarchical topology while moving at different speeds

Let us first consider MIFA in reactive mode. Due to the ping pong effect and the fast handoff achieved by MIFA, which results in a fast redirecting of the MN's data packets between the old and the new AR, MIP and HAWAII outperform MIFA regarding the average number of dropped packets per handoff on downlink when moving at 3 km/h by 9.6 % and 5.6 %, respectively. On uplink, however, due to the very fast handoff achieved by MIFA that requires only contacting the new AR to resume the uplink communication, MIFA still performs significantly better than MIP and HAWAII, 78 % and 79.2 % better according to the simulation results. If MIFA attempts to employ the predictive mode before the reactive one, packet dropping due to the layer 3 handoff can be eliminated at this speed. Increasing the speed of the MN to 20 km/h reduces the ping pong effect. MIFA in reactive mode, therefore, outperforms both other protocols. With respect to the average number of dropped packets per handoff on downlink, MIFA drops in reactive mode 20.3 % and 11.3 % less than MIP and HAWAII, respectively. On uplink, smooth handoffs are achieved by MIFA, which performs significantly better than MIP and HAWAII. If the MN operating MIFA can receive signals from more than one AP/BS, and, thus, can attempt to operate MIFA in predictive mode first, the dropping due to the layer 3 handoff can be eliminated.

Increasing the speed to 50 km/h further decreases the ping pong effect by MIFA. Clearly, this improves the performance. If the MN operating MIFA can utilize only the reactive mode, MIP and HAWAII are outperformed on downlink by 63.7 % and 62.4 % compared to MIFA. On uplink, MIFA performs 90.6 % and 93.3 % better than MIP and HAWAII, respectively. If the predictive mode can be utilized, packet dropping due to the layer 3 handoff can be eliminated. Increasing the speed to 120 km/h deteriorates the performance of MIP and HAWAII significantly, while MIFA remains almost unaffected. Let us first analyze the performance of MIFA in reactive mode at this speed. On downlink, MIFA performs approximately 80 % better than MIP and HAWAII, while on uplink, MIP and HAWAII are outperformed by MIFA by 92.2 % and 95.2 %, respectively. Considering the performance of MIFA in predictive mode, the dropping resulting from the layer 3 handoff at 120 km/h is not eliminated. However, as mentioned before, there is a high probability that MIFA can still be operated in predictive mode. According to our simulation results, MIFA in this mode results

in a performance improvement of approximately 95 % over MIP and HAWAII on downlink. On uplink, MIFA performs approximately 99 % better than both MIP and HAWAII. Increasing the speed to 300 km/h negatively affects the performance of the three protocols. However, MIFA remains significantly better than MIP and HAWAII.

6.6.5. Summary

The main results obtained from this study can be summarized as follows: MIFA outperforms MIP and HAWAII under all studied speeds with respect to the handoff latency. A fast handoff can be achieved even at high speeds. Moreover, there is a high probability to operate MIFA in predictive mode even at high speeds. The ping pong effect exists, however, with MIFA as well as with HAWAII causing higher average handoff latency at low speeds than at higher ones. Considering the number of dropped packets per handoff, MIFA in reactive mode outperforms the other two protocols at all studied speeds with respect to the number of dropped packets per handoff on uplink. However, MIFA is slightly outperformed by MIP and HAWAII at very low speeds (a speed of a pedestrian in this study) with respect to the number of dropped packets per handoff on downlink. The reason behind this behavior is, of course, the ping pong effect. At all other studied speeds, MIFA drops on downlink significantly less than the other two protocols. At a speed of 3, 20 and 50 km/h, MIFA in predictive mode achieves lossless handoffs. At higher speeds (120 and 300 km/h), MIFA is unable to employ the predictive mode in all handoffs, which results in some losses.

It should be mentioned that the movement detection method plays a dominant roll regarding the ping pong effect. In the achieved simulation, the ECS method has been used. This method is not appropriate, however, for low speeds. Therefore, it is expected that the LCS method will improve the performance at low speeds. ECS should be used, however, for high speeds. Thus, it may make sense to adapt the movement detection algorithm to the speed of the MN. The analysis of this assumption is a topic for future research.

6.7. Conclusion

This chapter has presented a detailed performance evaluation of MIFA compared to MIP and HAWAII. In addition to studying the behavior of these protocols under dynamically changing network conditions, the impact of the network topology, network load and MN speed have been analyzed as well. The performance metrics are the handoff latency, number of dropped packets per handoff deploying real-time traffic and average congestion window size deploying non-real-time traffic. The main obtained results can be summarized as follows:

1. Performance under dynamically changing network conditions:

- a. The best performance is achieved by MIFA in predictive mode if the time the MN spends inside the overlapping area is sufficient to trigger the layer 3 handoff in advance. Therefore, good network planning helps to significantly improve the performance of MIFA in this mode.
- b. MIFA offers very fast and seamless handoffs on uplink as well as on downlink even if it can be operated in reactive mode only.
- c. MIFA clearly outperforms MIP and HAWAII with respect to the handoff latency and the number of dropped packets per handoff on downlink and uplink.
- d. MIFA drops on uplink less than on downlink. In contrast, HAWAII drops on downlink less than on uplink, while MIP drops comparably on downlink and uplink.

2. **Impact of network topology:**

- a. There is no significant impact of network topology on the performance of MIFA in predictive mode. A well planned network topology is, as mentioned before, essential.
- b. There is no significant impact of network topology on the performance of MIFA in reactive mode regarding the handoff latency and the number of dropped packets per handoff on uplink. With respect to the handoff latency and the number of dropped packets per handoff on downlink, there is a clear improvement in the performance deploying a mesh topology.
- c. There is no remarkable impact of network topology on the performance of MIP
- d. There is no remarkable impact of network topology on the performance of HAWAII with respect to the handoff latency. Considering the number of dropped packets per handoff on downlink as well as on uplink, the mesh topology produces a slight performance improvement over the hierarchical topology.

3. **Impact of network load:**

- a. Increasing network load increases the handoff latency and the number of dropped packets per handoff accordingly in different ratios for the studied protocols. In contrast to this, the congestion window size decreases while increasing the load.
- b. MIFA performs better than MIP and HAWAII under all studied loads.
- c. With respect to the average congestion window size, increasing the load makes the performance of MIFA in predictive mode comparable to its performance in reactive mode.
- d. Network load has a significant impact on the performance of MIP and HAWAII. HAWAII is even more affected by the network load than MIP.

4. **Impact of MN speed:**

- a. Increasing MN speed results in an increase in the handoff latency and the number of dropped packets per handoff in different ratios for the studied protocols.
- b. MIFA outperforms MIP and HAWAII under all studied speeds with respect to the average handoff latency and the number of dropped packets per handoff on uplink.
- c. The ping pong effect is seen clearly by MIFA as well as by HAWAII causing higher average handoff latency and so more dropped packets per handoff at slow speeds than at faster ones. This effect causes MIFA at slow speeds (the speed of a pedestrian in this study) to be slightly outperformed by MIP and HAWAII regarding the number of dropped packets per handoff on downlink. The movement detection method has a major roll regarding the ping pong effect. Therefore, it may make sense to adapt the movement detection algorithm to the speed of the MN, e.g. LCS for slow speeds and ECS for high speeds.

After the evaluation of MIFA, the dissertation focuses on providing an adaptive eLearning environment to support studying and dissemination of mobility management issues covered in the previous chapters. The adaptive eLearning environment must be capable of personalizing eLearning contents based on learners' characteristics, so that a wide range of learners can benefit from this environment. Our main goal is to enable researchers to quickly become involved in current research trends in the field of mobility management, while other learners (e.g. students, lecturers, etc.) are provided with courses containing the topics required to eliminate gaps in their knowledge. To achieve this goal, the problematic of providing

adaptivity in eLearning environments should be carefully analyzed. Following that, the capabilities of existing adaptive eLearning environments of satisfying our goal should be investigated. If there is no adaptive eLearning environment capable of achieving the mentioned goal, a new one has to be developed. These issues are addressed in the next chapter.

7. Adaptive eLearning: New Opportunities for Learning

The growing importance of eLearning and its role in the modernization of teaching and learning methods can be seen from the tremendous number of newly eLearning courses being offered as well as research projects aiming at developing new concepts for eLearning. Although new technologies, in particular the Internet, have created many new opportunities of interest for eLearning environments, these opportunities have certainly not yet been exhausted. For example, the management of large amounts of information, provision of eLearning contents and interaction with learners show that many demands have yet to be met. One of these demands is the design and the implementation of adaptive eLearning environments capable of personalizing eLearning contents. Current eLearning environments offer little or no support for adaptivity. Therefore, the development of adaptive eLearning environments capable of satisfying learners' preferences is currently a main challenge. This chapter deals with this challenge and describes a new adaptive eLearning environment named MAeLE. Our main aim behind the development of MAeLE is to provide personalized courses on mobility management issues, so that researchers can quickly become involved in research, while other learners, e.g. students, are able to obtain information on the topics without excessive detail.

This chapter is organized as follows: section [7.1](#) provides a brief introduction. eLearning platforms are shortly described in section [7.2](#). Section [7.3](#) discusses known standards of eLearning. The meaning of adaptivity is discussed in section [7.4](#) followed by a brief description of well-known adaptive eLearning environments and their drawbacks. MAeLE is described in detail in section [7.5](#). Finally, section [7.6](#) concludes the chapter.

7.1. Introduction

The fast-paced economical and societal developments form a great challenge for new education organizations. Tremendous developments in communication technologies, speedy globalization of knowledge and increased Internet use require continuous development of new knowledge systems for qualification and education [[SBH01](#)]. The requirements of these systems can be satisfied through the development and deployment of eLearning concepts capable of improving educational and learning quality. eLearning itself is the unified term used to describe the fields of Computer-Based Training (CBT) and Web-Based Training (WBT). It describes methods to deliver instructions and information using new technologies [[BHH02](#)]. The shift from teaching to learning is one of the characteristics of new eLearning contents. It implies self-controlled learning and defines learners as the users who require this kind of learning [[Ber03](#)].

Three types of education are described in [[SBH01](#)], namely face-to-face, distance and online education. Face-to-face education is the conventional education type in which learners can discuss and communicate with each other and thus assist one another in acquiring knowledge. The main advantage of this education type is the strong interactivity between learners and with lecturers. The drawback of face-to-face education, however, lies in its dependency on location and time. Distance education is the well-known remote-controlled education type used to present learning materials to learners at different locations. This education type is instructor-oriented and can be independent of location and time. Online education represents

the most widely used type. It delivers and requires interactivity as well as various methods to discuss and exchange knowledge through integration of new communication technologies, especially the World Wide Web (WWW). This type of education is location- and time-independent and often learner-oriented. Education types are tightly bound to learning strategies, which present one of the most important success factors for CBT and WBT systems. One can distinguish between directed, self-directed and collaborative strategies [SBH01]. Directed learning, also termed as learning by telling, is simple and instructor-centered. The knowledge is offered to learners, which play a passive role. Self-directed learning, referred to as learning by doing as well, is more complex and is learner-centered. Learners specify their learning scenarios and tempos. Knowledge is acquired through experiments and practice. Learners play an active role in this strategy. The role of the instructor in self-directed learning is solely restricted to facilitating the learning process. Collaborative learning, also called learning by reflection and discussion, is the most complex strategy and is team-centered. Knowledge is acquired through discussion between learners, on one side, and with lecturers, on the other side. Such discussions result in knowledge exchange between team members, who play active as well as reflective roles. The instructor acts in collaborative learning only as a moderator.

Based on the discussion above, eLearning courses have to contain learning materials constructed according to an adequate learning strategy to actively provide learners with knowledge. Moreover, these courses should be supported by other components (e.g. communication possibilities, management tools, etc.) that, together with the learning materials, form an eLearning environment [DSt02], sometimes also referred to as an eLearning platform. eLearning environments must have an oriented processing of knowledge and be interactive [The04], [Haa02]. Moreover, each eLearning environment has to be flexible and extendable in terms of simple and dynamic integration of new knowledge as well as the use of new technologies, also referred to as an open eLearning environment [KNo00]. The main characteristic of such eLearning environments is that learning materials are developed in a form that can be simply updated and integrated into other eLearning scenarios. [Sch04] defines the open eLearning environment as an environment in which learning is organized by the learner. Interactivity and feedback are the most important properties of such environments. Interactivity between learners and Learning Objects (LOs) aims at motivating the learners and individualizing learning events. A detailed description of interactivity functions in open eLearning environments is provided in [NHH04].

7.2. eLearning Platforms

In the recent past, a great number of eLearning platforms have been developed. These platforms can be broadly classified into Content Management Systems (CMSs), Learning Management Systems (LMSs) and Learning Content Management Systems (LCMSs). The following briefly introduces these kinds of platforms.

7.2.1. Content Management System (CMS)

CMSs focus mainly on the management of online contents. They are not specialized for eLearning courses. However, they can be used for them. CMSs offer functions for the management of contents, conversion of contents into different formats, construction of navigations, searches, derivation of metadata¹ for content elements, access control, etc. They separate between the contents themselves, the design and the logic. Normally, contents are

¹ Metadata are data about data. For example, in case of learning materials, metadata can be the author, the language of the material, etc.

stored in a neutral format, e.g. XML. The outcome of CMSs (e.g. HTML or PDF) is generated at run-time [Fle08]. The structure of CMSs is divided into three levels [Lud05]. The first level describes the creation of contents. In this level, individual elements and contents (e.g. text, video, etc.) are created and assigned certain user access rights. The second level takes care of proving content elements and allowing them to be online. The third level is responsible for creating and presenting the required output. An important component of the CMS is the Web CMS (WCMS) [Fle08], which is responsible for presenting contents in the WWW. The presentation in the WWW often depends on logic stored on the server side. The client used to manage the content, configure content presentation, etc. is usually a normal web browser. A well-known CMS is typo 3 [AFH06], [Typo3].

7.2.2. Learning Management System (LMS)

LMS is software for the organization and supervision of web-based learning and teaching [MHa02]. It contains Databases to store learning contents as well as metadata describing them. Additionally, individual learning processes are recorded in the databases [MHa02]. Furthermore, LMS provides possibilities enabling synchronic and asynchronous communication between learners, on one side, and with tutors, on the other side [Fis06]. Tools supporting the learning, such as whiteboards, calendars, etc., are usually provided as well.

7.2.3. Learning Content Management System (LCMS)

This platform combines the properties of CMS and LMS [Lud05]. In addition to the functions known from CMSs (e.g. contents creation, processing, reusing, etc.), LCMSs also support functions of LMSs (e.g. creation of learning courses, management of users, etc.). They focus on the creation and management of LOs. Normally, LCMSs provide author-tools to help authors in creating LOs. These tools enable the storage of metadata for the created LOs in suitable databases. The metadata are used to describe and search for LOs according to certain criteria [MHa02]. Clearly, this simplifies the reuse of LOs in other courses, also called Reusable LOs (RLOs). Metacoocn [Metac], ILIAS [Ilias], Moodle [Ger07], [Moodl] and DotLRN [Dotlr] are well-known LCMSs. A detailed comparison between these LCMSs is provided in [Fle08]. The following describes one of these LCMS in more detail.

7.2.3.1. Moodle

Moodle is an acronym for Modular Object-Oriented Dynamic Learning Environment. It is a free open-source web application developed to help educators in creating eLearning courses. A screenshot of the Moodle main page is shown in figure 7.1. It uses PHP as a basic programming language and a database engine, e.g. MYSQL. Moodle groups LOs in courses accessed by different users with different access rights. The LOs are created as text, HTML pages or uploaded files, e.g. photos, PDF, etc. Many models are integrated in Moodle to support creation of eLearning courses. The most important models are the following, see [Fle08].

1. **Learning models:** Moodle supports the following learning models
 - a. **Lesson:** this model consists of many websites and navigation as well as many questions constructed mostly in multiple choices format.
 - b. **Document:** this model can be a file, folder, external link or website.
 - c. **Task:** this model is responsible for creating assignments to be completed by learners within a certain time period.

- d. **Database:** this model is responsible for the database used for the course, e.g. to store metadata for the course, results of tests, etc.
 - e. **Glossary:** this model enables the creation of abbreviation lists.
 - f. **Test:** this model is responsible for creating tests to control the progress and development of learners.
 - g. **Workshop:** this model enables the creation of assignments to be completed during a certain time slot. The solutions of each learner can be seen by all learners and even evaluated.
2. **Communication models:** this model enables Moodle users to communicate with each other. The following communication possibilities exist in Moodle.
- a. **Chat:** this model is java-based and uses HTTP protocol. It enables chatting between users.
 - b. **Questionnaire:** this model is responsible for the creation of questionnaires. However, the questionnaire is anonym for learners only. The course administrator knows who has written what?
 - c. **Forum:** Moodle enables the creation of forums to collaboratively interact and exchange information between Moodle users.
 - d. **WiKi:** this model enables users to upload documents and entries to be shared with other users.



Fig 7.1: Screenshot of Moodle main page

Let us now discuss the technical realization of Moodle. Moodle uses templates for the main page as well as for many other models, which enables a large number of configurations and changes to its Frontend. Since version 1.8, Moodle provides a flexible user management model, which allows seven roles to be assigned, namely administrator, course creator, teacher,

non-editing teacher, student, guest and authenticated user¹. Regarding the user characteristics that Moodle records, the user model is limited and focuses mainly on contact data. Moodle has a flexible plugin interface that enables a simple extension of the system through new plugins. New models can be easily integrated as well.

A drawback of Moodle is that it does not separate between logic and presentation. The libraries that provide basic functions for Moodle are not written in an object-oriented way. They are, however, well documented. The courses created are static for all registered users. There is no personalization of eLearning contents.

7.3. eLearning Standards

As mentioned previously, a large number of eLearning platforms have been developed in the recent past. In order to guarantee an acceptable level of quality and interoperability between these platforms, standardization of eLearning platforms and contents is necessary [Paw04]. First, many specifications and standards have been developed by separate national and international organizations. After that, a consortium of several organizations was established to develop standards for eLearning. The following briefly describes this consortium, its key contributions to eLearning and the well-known eLearning standards.

7.3.1. Cooperation Network of the Standardization Consortium

The cooperation network of the standardization consortium according to [BHH02] is presented in figure 7.2.

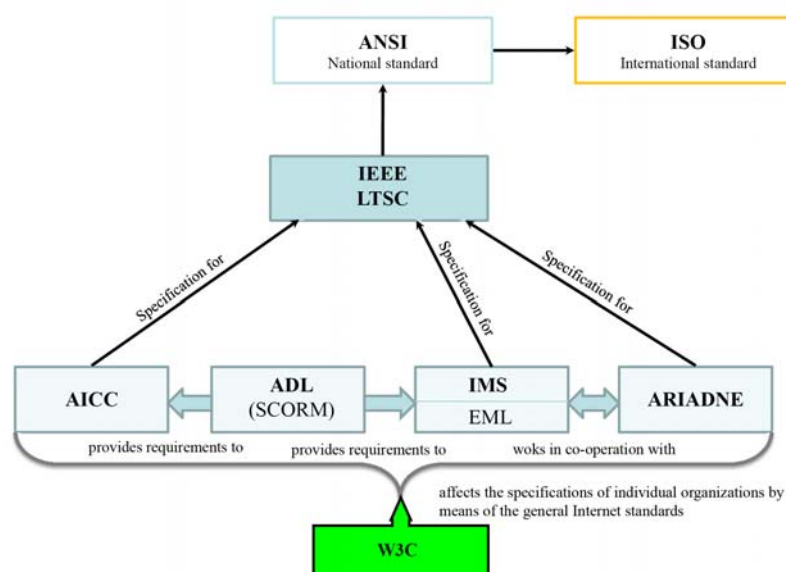


Fig 7.2: Cooperation network of the standardization consortium

The organizations of this consortium are as follows:

1. **Aviation Industry Computer-based training Committee (AICC)**: a group of professionals that creates CBT specifications and guidelines for the aviation industry of the USA [Aicc]. Its main contribution for eLearning is the Computer Managed Instruction (CMI) guidelines aiming at supporting the aviation industry with efficient CBT methods that enable interoperability between different LMSs. This group submits

¹ A default role for all logged-in users

its specifications and recommendations to the Learning Technology Standards Committee (LTSC) for standardization.

2. **Advanced Distributed Learning (ADL)**: a government-sponsored organization aiming at developing new guidelines to make current and future LMSs interoperable in terms of platforms and contents [[Adl](#)]. The most widely accepted ADL specification is the ADL Shareable Content Object Reference Model (SCORM). SCORM combines the best elements of other organizations into one document that can be easily implemented. ADL cooperates tightly with AICC and the Instructional Management System (IMS) group.
3. **Instructional Management System (IMS)**: a global learning consortium consisting of members from the industry, governments and scientific institutes and organizations [[Ims](#)]. The main goals of IMS include the development of guidelines for the reusability of eLearning contents, interoperability between different eLearning systems and platform-independent eLearning systems. This group offers a disciplined approach for describing various resources through metadata. Key contributions of IMS include the Content Packaging (CP) and the Learner Information Package (LIP), which enables describing of didactical aspects. Similar to AICC, this group submits its specifications and recommendations to LTSC for standardization.
4. **Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE)**: established to further apply and develop the results of the EU projects ARIADNE and ARIADNE II [[Ariad](#)]. This group develops methods for the creation, management and reuse of eLearning materials with a main focus on pedagogical aspects. It focuses on providing eLearning contents with suitable metadata. It works, therefore, in tight cooperation with IMS. Similar to AICC and IMS, this group also submits specifications to LTSC for standardization.
5. **Institute for Electrical and Electronic Engineers Learning Technology Standards Committee (IEEE LTSC)**: represents the committee within IEEE responsible for accrediting recommendations of other organizations as well as providing standards that will be submitted to the American National Standards Institute (ANSI) and International Standards Organization (ISO) for further processing [[Ltsc](#)]. Currently, there are four well-known standards of the LTSC [[Ltsc](#)].
 - a. **Learning Technology System Architecture (LTSA)**: describes abstract systematic learning technologies [[FTo99](#)].
 - b. **Learning Object Metadata (LOM)**: used to describe LOs using metadata [[Ltsc-a](#)].
 - c. **Public And Private Information (PAPI)**: defines user characteristics that should be stored in the user model of LMSs [[Far01](#)].
 - d. **Computer Managed Instruction (CMI)**: provides guidelines for enabling interoperability between different LMSs [[Ltsc](#)].

7.3.2. *Shareable Content Object Reference Model (SCORM)*

SCORM standard [[Scorm](#)] contains many specifications from different organizations, i.e. the content structure guidelines of AICC, LOM and the IMS content packaging and sequencing. This standard aims at enabling interoperability, accessibility and reusability of eLearning contents for industry, government, and academia. Main components of the SCORM reference model include a Content Aggregation Model (CAM), Run-Time Environment (RTE) and Sequencing and Navigation (SN).

CAM consists of a content model, metadata and content packaging. The content model defines learning resources as assets and Sharable Content Objects (SCOs). An asset is the smallest learning resource, e.g. text, photo, music file, etc. Many assets can be combined to form a more meaningful asset. The SCO is the smallest learning resource able to communicate with a LMS. Many SCOs can be combined to form a content package that can be shared easily between SCORM-enabled LMSs. The description of learning resources themselves is done according to the LOM standard. RTE describes the behavior of the LMS at run-time, e.g. how the content should be accessed, the communication between the content and the LMS realized, etc. RTE contains elements able to be linked to SCOs. This allows for the observation and tracking of SCOs behavior at run-time, e.g. to track how advanced user is. Communication between the SCOs and the LMS occurs according to the IEEE ECMAScript, see [Lud05]. The SN describes how the navigation should change depending on user activities and visited resources. Activity trees are defined, which change according to user behaviors. The status of SCOs¹ can be stored and passed on to the LMS.

7.3.3. *Learning Object Metadata (LOM)*

LOM contains semantic and syntax of metadata used to describe LOs. The aim is to simplify the identification, search, reuse, importation and exportation of LOs. The basic structure of LOM contains 9 categories.

1. **General:** contains general information about the LO, e.g. keywords, language, etc.
2. **Lifecycle:** contains information about the story and the actual status of the LO, e.g. version, status, etc.
3. **Meta-Metadata:** contains information about the metadata themselves, e.g. metadata scheme, etc.
4. **Technical:** contains information about technical requirements for the LO, e.g. format, size, etc.
5. **Educational:** contains pedagogical information about the LO, e.g. interactivity type, difficulty, etc.
6. **Rights:** describes copyright and usage conditions, e.g. copyright, cost, etc.
7. **Relation:** defines the relationship between the LO and other LOs, e.g. kind, resource, etc.
8. **Annotation:** enables editing comments related to the LO, e.g. entity, date, etc.
9. **Classification:** contains information about the classification of the LO in a particular classification system, e.g. purpose of the LO, description, etc.

7.4. **Adaptivity in eLearning Standards and Systems**

Personalization of eLearning processes and scenarios was defined as a dominant success factor for learning. Achieving adaptivity can be seen as a prospective change from computer-based learning, in which the computer is seen as a teaching medium, to learner-oriented learning, in which the computer is a learning medium [MGr05]. It has been shown that learners react positively if their personal preferences are taken into account [HJM06]. This section defines the term adaptivity and discusses adaptivity support within the standards

¹ The status of SCO can be active or not. In other words, visited or not.

introduced in the above section. Moreover, well-known adaptive eLearning systems will be briefly described.

7.4.1. *Adaptivity and Adaptive Hypermedia Systems (AHSs)*

Adaptivity is defined as the ability of a learning system to personalize itself based on learner characteristics. Three types of adaptivity are defined in [Sch06], namely interface adaptivity, static learner adaptivity and dynamic learner adaptivity. Interface adaptivity refers to the ability of the system to adapt its interfaces, windows, dialogs, etc. to user preferences. Static learner adaptivity stands for the personalization of learning processes according to user characteristics upon initiating these processes. There are no adaptivity actions performed during the processes themselves. Dynamic learner adaptivity provides a run-time adaptivity. It is assumed that the system tracks user activities and reacts by adapting learning processes accordingly.

According to [Bru96], AHSs are all hypertext and hypermedia systems that reflect some features of the user in a user model and apply this model to adapt certain aspects. The user model is a main component of each AHS and should contain the user features to which the system has to adapt. The model should be actualized depending on user activities and behavior. AHSs gather information on user features either by observations and behavior tracking or by asking users to enter personal information, e.g. via questionnaires, tests, etc. Figure 7.3 shows the interaction between the AHS and the user model according to [Bru96].

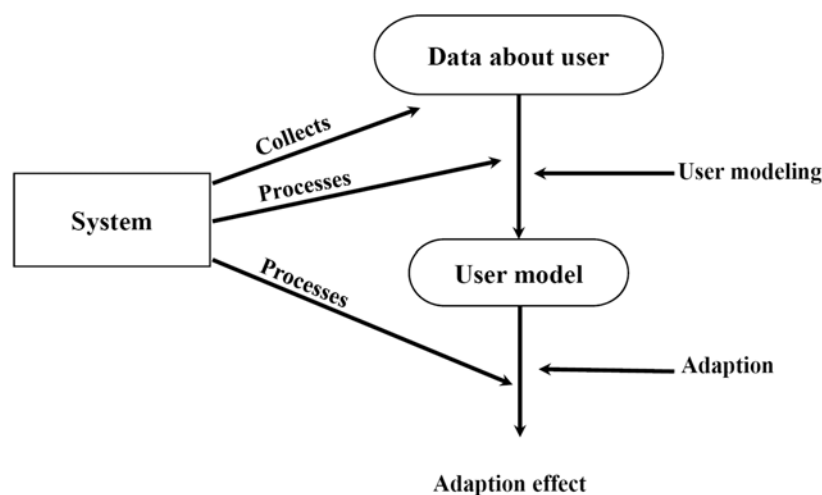


Fig 7.3: Interaction between the AHS and the user model

7.4.2. *Adaptivity Support in Existing eLearning Standards*

The eLearning standards presented in section 7.3 lack of adaptivity support. SCORM is a well-known model for eLearning environments. It has obtained a great acceptance in different fields of eLearning. However, the adaptation abilities of SCORM are restricted and focus only on allowing the definition of several organizations for the same course as well as the definition of sequencing information, which allows for the determination of a set of rules to select the next activity that should be shown. A proposal to enhance SCORM with adaptivity is presented in [RFD06]. Adaptivity on two levels has been proposed. The first is adaptivity at the activity level, while the second is adaptivity at the SCOs level. Adaptivity at the activity level depends mainly on a set of sub-activities related to each other according to pre-defined adaptation rules. Adaptivity at the SCOs level depends on self-adaptive SCOs capable of showing different behaviors according to learner characteristics. However, due to a lack of

focus on adaptivity during the development of SCORM, the proposals to enhance it with adaptivity support remain limited. The lack of adaptivity in current standards has resulted in the development of adaptive eLearning platforms, which form the focus of the following sections.

7.4.3. Adaptive eLearning Systems

There are many projects attempting to support adaptive eLearning. An eLearning environment named Adaptive Learning Environment (ALE), for example, is presented in [TRM03]. This environment integrates an intelligent tutoring system, a computer instruction management system and a set of cooperative tools. This environment can produce individualized courseware for students based on their current state of knowledge, their preferences and learning styles according to a chosen learning strategy. Authors can create contents using pre-defined templates. These templates combine several content elements with various pedagogical functions, e.g. introduction, definition, etc. Metadata are added to each content element. The user model records the interaction history, tested knowledge and user readiness, which results from comparing the learner object pre-requests, i.e. the content pre-requested by the user as he registered for the course, with user interaction history.

Another adaptive eLearning environment named L3 (Life-Long Learning) is introduced in [Lei01]. L3 is an eLearning platform combining the functionality of a traditional LMS with the power of a CMS. L3 structures its learning materials in four types of containers, namely learning networks, LOs, instructional elements and tests. Instructional elements represent actual learning contents. There are different types for instructional elements, e.g. example, action, introduction, etc. Many instructional elements are combined in a LO. The test element is used to create tests or questionnaires. The learning network is the topmost level, which contains other elements as well as possibly other learning networks. Learning paths are defined in each course depending on the relations between the instructional elements and their content type. L3 defines two categories of strategies, macro and micro. Macro strategies are responsible for defining the order of higher-level elements, while micro strategies cover the order of instructional elements.

In order to increase the clarity of adaptive eLearning environments and their functioning, the following sections describe several well-known eLearning environments in more detail.

7.4.3.1. Knowledge Tree

Knowledge Tree [Bru04] is a distributed architecture for adaptive eLearning, see figure 7.4. It aims at replacing the normal LMS with distributed communication servers. The architecture of knowledge tree assumes that four types of servers and services are present, namely activity servers, value-adding services, learning portals, and student model servers. The servers and services represent the interest of content and service providers, course providers and learners. The learning portal stands for a course provider, e.g. a university, company, etc. The portal is similar to LMSs in two regards. First, both provide a centralized login for students. Second, they allow for teachers to be responsible for courses that, in practice, are built from many parts obtained from many distributed activity servers, which provide learning contents and learning support services. In other words, activity servers stand for the content and service providers. Contents reusing aspects represent the most important aspects for these servers. However, reusing does not refer simply to the copying of contents and their insertion into a course. Rather, the contents may be interactive. Therefore, activity servers must provide a complete support for students working with the activities residing on those servers. Such support encourages content providers to build adaptive contents that can be personalized and adapted to students' requirements. Up-to-date information about each student can be obtained

from the student model server. Activity servers have to monitor learner progress and update some features of the students, e.g. goals, interests, etc. The value-adding server is used to add extra contents or services that add some valuable functionalities, e.g. adaptive sequencing, visualization, etc. This server combines features of portal and activity servers. Similar to the portal, the value-adding server can query activity servers and access to activities. Like an activity server, it can be queried and accessed by a portal.

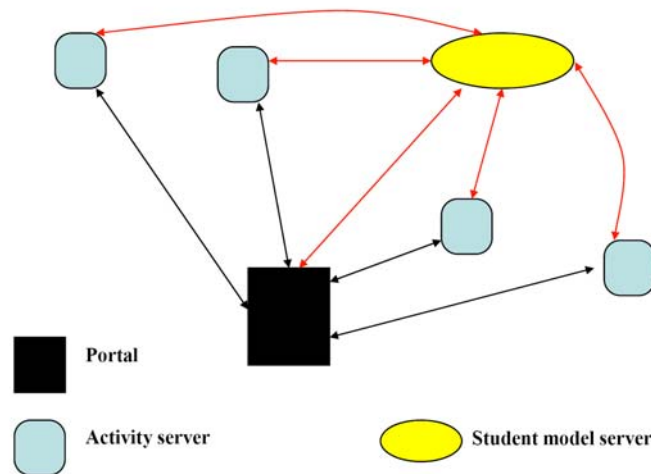


Fig 7.4: Knowledge tree architecture

Notice that this architecture enables teachers and students to work via one portal with courses hosted on distributed servers. It is a flexible and open architecture. Many portals, activity servers and student model servers can exist in one system. A competition between content providers is even encouraged. The knowledge tree architecture relies on several clearly-defined communication protocols between various servers and services. The architecture seems to be well designed. However, this architecture does not accurately define the way in which adaptivity is to be achieved, nor does it specify which user features are more important. These issues are the challenges of adaptive eLearning systems.

7.4.3.2. *Adaptive Hypermedia Architecture (AHA)*

AHA is an open source platform for adaptive hypermedia applications. Adaptive presentation, i.e. adaptation of eLearning contents, as well as adaptive navigation, i.e. selecting the adequate navigation method such as direct guidance, link annotation, etc., can be achieved with AHA [Ruh07]. The structure of AHA is shown in figure 7.5. **Authoring Tools** aim at supporting authors in creating adaptive learning contents. The **WWW Server** is the core of the architecture. It stores the **Domain**, **User** and **Adaptation Model**. In addition, all servlets of the AHA engine are executed on this server. The **Domain Model** contains learning contents, while the **User Model** contains user features. The **Adaptation Engine** stores the rules that should be executed to achieve adaptivity. The **AHA Engine** is responsible for processing user queries. Furthermore, it determines and executes the adaptation rules that should be executed according to these queries. After that, contents are selected from the **Domain Model** and adapted to users. If the execution of the adaptation rules results in actualization of user properties, the **AHA Engine** should perform this actualization.

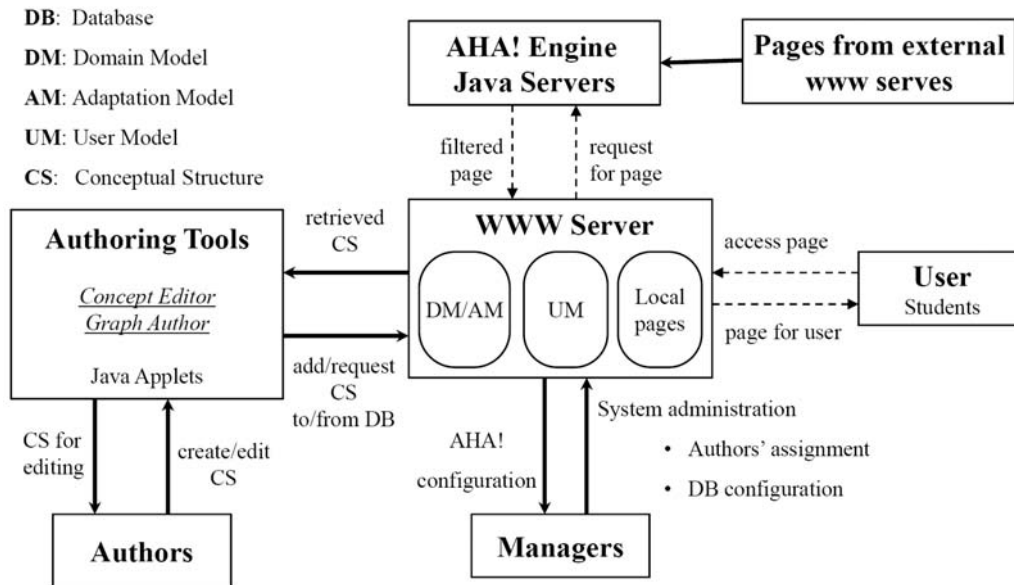


Fig 7.5: Structure of AHA

The adaptation process occurs as follows: as the user requests a concept¹ from the AHA system, the **Adaptation Engine** searches for the concept in the **Domain Model** and uploads the websites assigned to the requested concept for further processing. Following this, the **user model** is accessed. Afterwards, the adaptation rules stored in the **Adaptation Model** and related to the requested concept will be executed, which may result in updating the **User Model**. The changes in the **User Model** may trigger the execution of other adaptivity rules. The rules will, then, be further executed and the **User Model** will be updated once more and so on until no further adaptivity rules must be executed. The websites assigned to the concept will be presented after processing according to adaptivity rules. If these websites contain links to other concepts, it will be tested whether the user is able to access these concepts. Depending on this check, the links will be adapted to the user as well.

7.4.3.3. Adaptive Personalized eLearning Service (APeLS)

APeLS [Con04] is a well-structured AHS based on metadata-based adaptivity. APeLS requires metadata describing not only contents, but also adaptivity rules, navigation structures, etc. Two new terms are introduced in APeLS, namely a narrative and a pagelet. The narrative contains a rule set and metadata describing this rule set, which defines how the personalized course should be constructed. The pagelet is composed of both contents and metadata describing them.

The structure of APeLS is presented in figure 7.6. As seen from this figure, APeLS relies on metadata and information repositories to store the models of the system. The **Learner Metadata Repository** stores metadata representing individual learners. These metadata should conform to the applied learner model, **Learner Modeller** in the figure. The **Pagelet Metadata Repository** stores metadata representing each piece of learning content or pagelet. The **Pagelet Content Repository** contains all pagelets referred to by the **Pagelet Metadata Repository**. The **Narrative Metadata Repository** contains the metadata describing LOs, while pedagogical aspects for each narrative are stored in the **Narrative Repository**. **Course Model Repository** is responsible for storing the personalized courses previously created by

¹ The concept refers to a chapter of a course or the course itself. A concept is normally assigned many websites.

the system. The **Candidate Selector Metadata Repository** contains metadata describing each candidate selector stored in the **Candidate Selector Repository** that contains the rules, which should be applied to select a candidate from a candidate group. There are two types of candidate groups, namely **Candidate Pagelet Groups** and **Candidate Narrative Groups**. **Candidate Pagelet Groups** contain references for metadata stored in the **Pagelet Metadata Repository** and fulfill the same learning goal. **Candidate Narrative Groups** contain groups of narratives encapsulating the same knowledge. The **Learner Modeller** contains learner features. The **Rule Engine** is responsible for interpreting narratives and producing personalized course models. The **Candidate Selector** is executed by the **Rule Engine** when there is more than one candidate group. The **Candidate Selector** has to select the candidate best matching learner features.

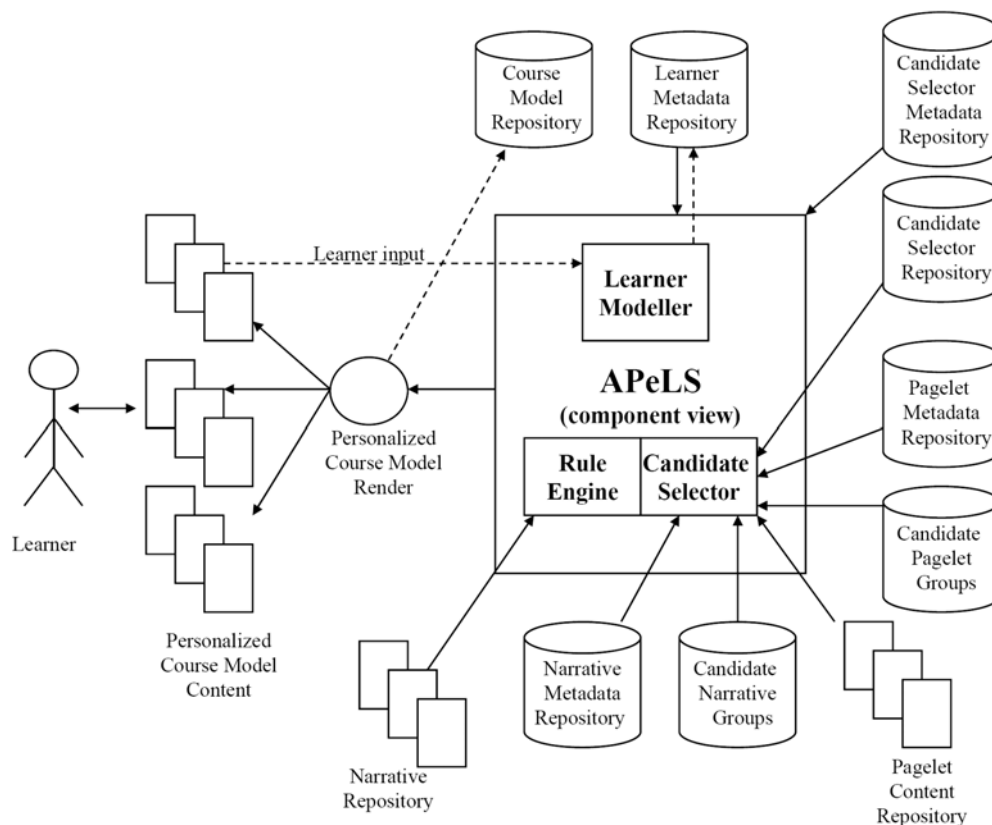


Fig 7.6: Structure of APeLS

The adaptivity procedure in APeLS can be briefly described as follows: as the learner logs in and access to a certain course, he is given a multiple choice test from which the system can gather the information required to achieve adaptivity. The learner model is actualized as a result and the **Rule Engine** is executed. The engine accesses the required narratives and repositories and generates the required personalized course. This environment is a powerful one. It can achieve a presentation as well as navigation adaptivity. This depends, of course, on the narratives generated. APeLS is, however, a very complex AHS. A great number of metadata are necessary. Courses authors have to provide contents with suitable metadata. In addition, adaptivity narratives must be defined, which requires a thorough understanding of the system.

7.4.4. Summary

The analysis of existing adaptive eLearning environments has resulted in the following results.

1. Most eLearning environments are specialized for certain fields, e.g. mathematics, physic, etc. The structure of courses and pedagogical aspects are derived in conformation to the field of focus.
2. Most eLearning environments focus on the structure of the environment more than on the adaptivity process. Questions such as, how adaptivity should be achieved, which features have to exist in the learner model, what is the relation between the metadata of eLearning contents and learner characteristics, etc., are not thoroughly investigated.
3. Adaptivity is achieved depending on one or several learner features, mostly depending on the knowledge level the learner has. This implies neglecting many of features relevant for adaptivity. Even APels can support adaptivity depending on an unlimited number of learner features by allowing narratives to be freely written. This forces course authors to have a thorough understanding of adaptivity rules and semantics, which is not desired.

All in all, to satisfy our goals in personalizing courses in the field of mobility management in IP-based mobile communication networks for a wide range of learners who span from students to engineers and researchers, a new eLearning environment should be developed. Moreover, the adaptivity process has to consider most learners features. This has led to the development of MAeLE. This new eLearning environment will be detailed in the next section.

7.5. Metadata-Driven Adaptive eLearning Environment (MAeLE)

7.5.1. Requirements of MAeLE

Similar to [Con04], in order to overcome the shortcomings of other adaptive eLearning environments, the new environment has to satisfy the following requirements:

1. Delivering of adequate eLearning experience to learners in consideration of pedagogical aspects.
2. Provision of different adaptive effects dependent on different sets of models. More specifically, presentation and navigation adaptivity should be provided simultaneously. Moreover, the adaptivity process must consider most features of learners.
3. Provision of an extensible architecture, so that new adaptive rules, navigation methods and learning strategies are easily integrated.
4. Allowance for the creation of reusable eLearning contents that can be exported to other eLearning environments supporting the well-known eLearning standards. Importation of eLearning contents from such environments to MAeLE should be possible as well.
5. Provision of tools to enable authors to create contents along with their metadata in a simple manner without requiring a thorough understanding of the environment and its specifications.
6. Allowance of run-time adaptivity.
7. Establishment of interfaces enabling a simple configuration of adaptive eLearning environment behavior, e.g. updates to adaptivity rules, integration of new navigation methods, etc.
8. Guarantee of compatibility with SCORM model.

7.5.2. Architecture of MAeLE

MAeLE extends the functionality of a traditional LMS by developing an adaptivity framework capable of generating eLearning courses at run-time. The following provides an insight into the structure of MAeLE along with the main tasks of each component, see figure 7.7.

The **User Model** is used to characterize the user, mainly dependent on the current state of knowledge, preferences, learning styles, learning goals, etc. MAeLE categorizes user characteristics into two categories. The first contains the characteristics that are not important for the adaptivity process (e.g. contact information, name, etc.), while the second includes adaptivity-important characteristics. Section 7.5.4 illustrates the **User Model** in more detail.

It is highly recommended that the content should be reusable. Therefore, eLearning contents should not contain any sequence logic or metadata. They are saved in **Content** database as small meaningful LOs, referred to as assets in SCORM. The metadata describing each asset are stored in a separate database (called **Metadata** in the figure). The metadata do not only describe the LOs, but contain also some information indicating for which learner characteristics they are adequate. This helps significantly in the personalization of eLearning contents. Furthermore, a reference to the adequate learning strategy should also exist in these metadata, see section 7.5.5 for details. Metadata describing the supported eLearning strategies and navigation methods are stored in separate databases (called **Didactical Strategies** and **Navigation** in the figure). Keeping these metadata in separate databases enables updates as well as integration of new strategies or navigation methods in a very simple way.

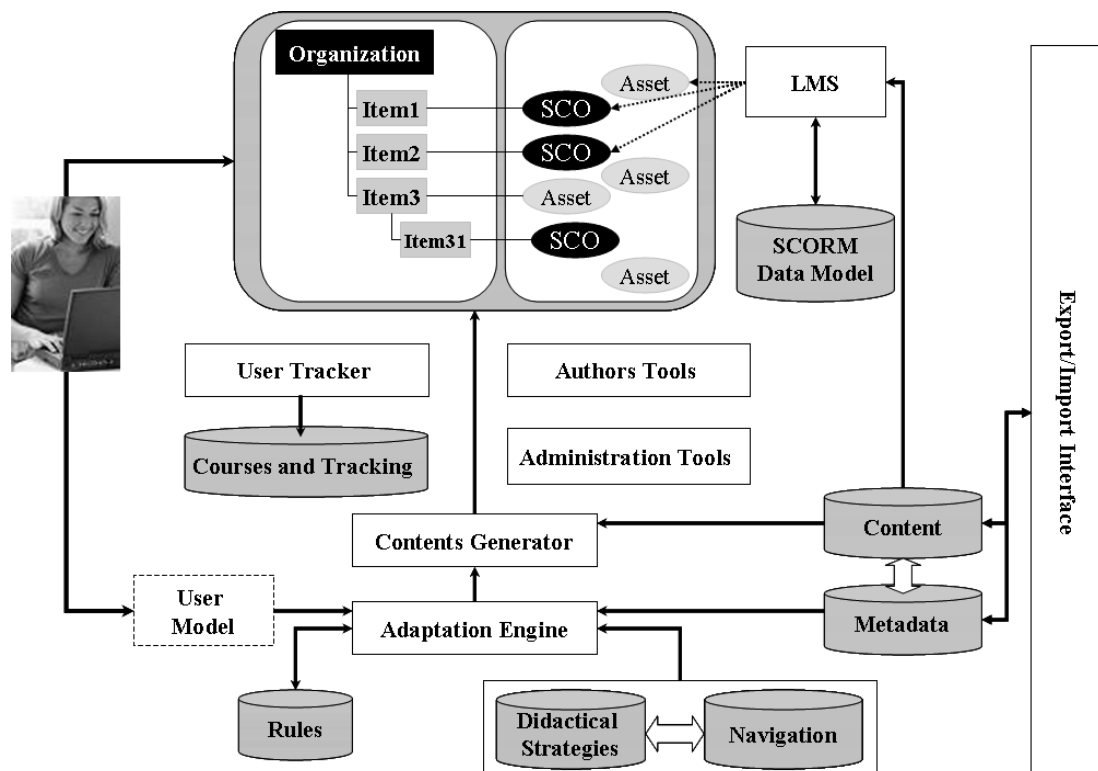


Fig 7.7: Structure of MAeLE

The **Adaptation Engine** is the main component of MAeLE. It constructs, according to pre-defined adaptivity rules, the structure of the course taking into account learner features as well as the metadata describing eLearning contents. In other words, it has to define the adequate eLearning contents as well as adequate strategy and navigation method for the user, see section 7.5.3 for details. In order to make MAeLE flexible and extensible in terms of

integrating new adaptivity rules, these rules are stored in a separate database (called **Rules** in the figure). The output of the **Adaptation Engine** is a set of parameters that define accurately desired contents, the adequate strategy and the suitable navigation method. These parameters are fed into the **Contents Generator** that builds the course along with the defined navigation.

Authors Tools are used to help authors in creating their eLearning contents and integrating them into MAeLE. **Administration Tools** are tools developed mainly to help the administrators of MAeLE in administration tasks such as reconfiguring the system, updating the **User Model**, etc. A very important component is the **Export/Import Interface** that enables the import and export of eLearning contents between MAeLE and other eLearning systems and even according to various eLearning standards.

The system records the behavior of the user at run-time, e.g. which links has been visited, which kind of eLearning content is preferred, etc. This task is achieved by the **User Tracker**. This component can be realized by means of various technologies, e.g. eye tracking [BGP04]. The tracking of user behavior may change some user features at run-time. MAeLE reacts dynamically to such changes and re-generates the non-visited parts of the course. The main motivation behind the re-generation of the non-visited parts of the course instead of the course as a whole is that learners will be confused if they move to a pre-accessed part of the course and find contents different from what they had worked with. The gathered tracking information as well as structures of the courses currently handled by the user are stored in a database named **Courses and Tracking**.

7.5.3. *Adaptivity Process in MAeLE*

As shown in figure 7.7, the adaptation process of MAeLE is rules-based. The adaptation engine contains two main components:

1. **Static Adaptation Engine**: this component is responsible for creating the structure of the course as the user logs in to the system. This is achieved as follows: the static adaptation engine looks for a previous course structure for the current user in the **Courses and Tracking** database. If such a structure is found, the **Static Adaptation Engine** checks the tracking information gathered from the last login of the user. Based on this information, if an adaptation should be performed, this component executes the related adaptation rules. Otherwise, the same structure is used and fed into the **Contents Generator**. Even if some adaptation rules must be executed, they will be executed only for the part of the course the user has not accessed yet. The motivation for this is, as mentioned previously, that the user will be confused if he has to deal with a different structure for the course each time he logs in.
2. **Run-Time Adaptation Engine**: this component is responsible for achieving the adaptivity at run-time, i.e. while the user handles the course. As mentioned previously, the user actions are tracked by the **User Tracker**. This tracking may result in updating the user model, which may trigger the execution of some adaptation rules. These rules are executed by the run-time adaptation engine, which may result in changes to some parts of the course. The new structure is built and fed into the **Contents Generator**, which in turn performs the required updates. For the same reason presented above, adaptivity rules are only applied to the parts of the course that have not yet been handled by the user.

Notice that each adaptivity action requires the execution of one or more adaptation rules. So as to enable maximum system extensibility, adaptation rules are stored in a separate database, which simplifies the integration of new rules. The structure of the rules is rather simple and has the form of nested if-statements as illustrated in figure 7.8. **Parameter 1** and **parameter 2**

are either characteristics of a learner or a LO. In most cases, **parameter 1** represents metadata for a LO, while **parameter 2** represents a characteristic of a learner, who should obtain the LO characterized by this metadata. The **Next rule ID** specifies the next rule that should be executed, i.e. the adaptation rule requires executing another rule, which may require executing another rule and so on, also nested if-statements.

```

If < parameter 1 > < comparison operator > < value 1 > then
{
    < parameter 2 > < comparison operator > < value 2 >
    Next rule ID = < value 3 >
}

```

Fig 7.8: Structure of adaptation rules

7.5.4. *Modeling of the User*

The most important part of any adaptive eLearning environment is the user model. It should contain all user characteristics. In addition to the typical user data, such as login and contact data, the user model should also contain adaptivity-relevant characteristics. In the following, such characteristics will be presented briefly, see [DWM08] and [Bal08].

1. **Learning goal:** an accurate definition of user learning goals is one of the main success factors of any adaptive eLearning environment. A learning goal is defined as a certain behavior to be acquired by the user [Cla95]. There are many works done to formulate learning goals. For example, user learning goals of studying courses in the field of theoretical informatics include: acquisition of knowledge, being capable of using the acquired knowledge, understanding the content, being able to critical analyze the knowledge, developing new knowledge through synthesis, communication of possibilities, presentation of capabilities, team work, evaluation of knowledge, capabilities for self-organizing and self-control and spontaneity [Gri00]. Teaching and learning goals are not the same according to [Ker01]. Learning goals are defined in this research as factors that are hard to predict. The study presented in [BGa99] determines seven phases for learning, which are: knowledge acquisition, practice, test/control, repetition, looking up, application, simulation and communication. Each phase presents a task oriented at a certain learning goal and satisfied by certain learning content. Based on previous research results, learning goals are defined for MAeLE as follows:
 - a. **Generation:** the generation and acquisition of new knowledge.
 - b. **Applying:** capability of applying the acquired knowledge in practice.
 - c. **Looking up:** research something forgotten in previously acquired knowledge.
 - d. **Repeat:** repeating the course in order to improve understanding.
2. **Pre-knowledge:** information about the pre-knowledge of the user is a very important factor for achieving personalization of eLearning contents. Moreover, the adequate navigation method relates to the pre-knowledge of the user¹. Learning and knowledge

¹ For example, users with low pre-knowledge require more support from the system. Thus, direct guidance is suitable for them. In contrast, users with high pre-knowledge prefer to move freely in the course. Therefore, link annotation is adequate for them.

are two definitions that are tightly related to each other [Bal08]. Learning can be seen as an active process, while knowledge is the goal and the result of this process. In [DDr87], a learning model consisting of five levels from newcomer up to expert is presented. Depending on this study and further relevant studies [Bal08], the levels of user pre-knowledge are selected as follows:

- a. **Level 1:** newcomers who begin learning the basics. The offered course for such users should contain basic facts.
 - b. **Level 2:** beginners who start understanding different cases and situations presented in the handled course. They begin to apply acquired facts in a context-oriented way. However, they cannot navigate and handle eLearning materials independently.
 - c. **Level 3:** professionals who know most facts and rules in handled materials. However, they can somehow hardly make a decision in given case studies.
 - d. **Level 4:** skilled persons who do not need to divide given case studies and problems into small parts and handle them separately. Instead, they begin analyzing the problem as a whole and develop a solution for it.
 - e. **Level 5:** experts who handle given situations and make decisions intuitively.
3. **Cognitive style:** this describes mainly the way in which learners collect and organize information. Information collection describes the way of presenting the information, so that it can be cached simply by learners. Depending on this, one can distinguish between different learning types [Apo02], which are: visual, auditory, haptic, verbal and conversational. A visual learning type prefers visual information presentation, e.g. photos, drawings, etc. An auditory learning type prefers forming the learning materials as audio elements. A haptic learner prefers presentations with interactive simulations, where he can change and try. Information should be formed as definitions and abstract facts for a verbal learning type, while a conversational learning type prefers communicating with others.
 4. **Learning style:** learning style is a very important personal property. According to [Bal08], one can distinguish between four learning styles: activist, reflector, theorist and pragmatist. An activist is open and motivated for everything new. He works actively and reacts quickly to opportunities, problems, etc. A reflector normally takes enough time to think about a given problematic. A theorist prefers to work with theories and concepts, while a pragmatist prefers a relation between theory and practice. He learns better if he can apply the acquired knowledge in practice.
 5. **Motivation:** motivation is seen as a very necessary requirement for a successful learning process and is defined in [Pae93] as the readiness of a learner to apply his capabilities, knowledge and competence to achieve a certain learning goal. One can distinguish between two types of motivations, namely intrinsic and extrinsic. Intrinsic motivation is something personal coming from the learner himself and reflecting perhaps his requirements, preferences, etc. Extrinsic motivation is normally related to eLearning environments. Both motivation types are tightly bound to each other, e.g. an intrinsic motivation can emerge from an extrinsic motivation [Bal08].
 6. **Learning platform:** learning platforms are mainly the technical requirements needed to achieve a complete eLearning environment. According to [Ger07], there are three types of learning platforms that can be differentiated according to the priorities of learners. These three types are: platforms for learning by distributing, learning by interacting and learning by collaborating. From the learner's point of view, learning by

distributing (also referred to as learning by telling) provides knowledge that should be handled by the learner, who plays a passive role. Learning by interacting refers to learning by doing something. It depends on the feedback from the learner, who plays an active role. The learner defines his contents and navigation path inside materials by himself. Learning by collaborating stands for learning in groups, where learners work together and solve a given task collaboratively.

7. **Role of the learner:** the role of a learner in a certain eLearning environment determines his rights to access to materials. One can distinguish between guest, member and an active member. The guest does not register his personal information in the system and has access only to public data and courses. The member is a learner who registers himself in the eLearning environment and has access to certain courses. An active member is a member who belongs to a certain member group.

7.5.5. Learning Contents Metadata and their Relation to User Characteristics

MAeLE uses LOM standard to describe eLearning contents. The main motivation behind the use of LOM is the wide acceptance and usage of this standard in existing eLearning environments. In order to achieve a successful adaptation of contents to users, it is necessary to study the relation between the metadata describing eLearning contents and the metadata describing learners. Considering LOM standard, the category **education**, which groups the educational and pedagogic characteristics of LOs, is the most adequate category where the relation to learners can be defined. LOM defines 11 parameters in this category, see [[Ltsc-a](#)]. Adaptivity-relevant parameters in the **education** category and their relation to learner characteristics are briefly described in the following.

1. **Interactivity type:** this parameter describes the supported predominant mode of learning. There are three values for this parameter, namely active, expositive and mixed. An active LO is the object that produces an action by the learner. It relates to learning by doing and prompts the learner for semantically meaningful inputs. An active LO may be a simulation, questionnaire, etc. An expositive LO is the object that displays information without prompting the learner for semantically meaningful input. It relates to the passive learning, where the user absorbs the content exposed to him. Expositive documents include, for example, essays, graphics, texts, etc. A mixed LO is an object that blends both the active and expositive interactivity types. It represents learning by collaborating. An example is a hypermedia document with an embedded simulation applet.

Taking the metadata describing the learner into account, we can see a relation between the interactivity type of a LO and the learning style of a learner. An activist learner works actively and reacts quickly to opportunities, problems, etc. Thus, active LOs are the most adequate in this case. Mixed LOs may be appropriate as well, while expositive LOs are not. In a similar way, we can match the LO interactivity types to learner learning styles, see table 7.1. The table can be interpreted as follows: for an activist, the most adequate LO interactivity type is active. In case there is no LO from this interactivity type, the mixed LO is selected. If there is no mixed LO, an expositive one is chosen.

Learner learning style	LO interactivity type (ordered according to priority)		
	Priority 1	Priority 2	Priority 3
Activist	Active	Mixed	Expositive
Reflector	Expositive	Mixed	Active
Theorist	Expositive	Mixed	Active
Pragmatist	Active	Mixed	Expositive

Tab 7.1: Mapping between LO interactivity types and learner learning styles

2. **Learning resource type:** this parameter specifies the type of a LO. LOM defines a list of values that can be assigned to this parameter, which are: exercise, simulation, questionnaire, diagram, figure, graph, index, slide, table, narrative text, exam, experiment, problem statement, self assessment and lecture. Taking the metadata describing the learner into account, we can find a relation between the learning resource type of a LO and the cognitive style of a learner, e.g. a visual learner prefers presenting the information optically. Thus, a LO with the learning resource type “figure” is adequate for this learner, while an exercise is less suitable. Table 7.2 presents the learning resource types of a LO and the corresponding cognitive styles ordered according to their priority.

LO learning resource type	Learner cognitive style (ordered according to priority)				
	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5
Exercise, questionnaire, exam, experiment and self assessment	Haptic	Conversational	Verbal	Visual	Auditory
Simulation	Haptic	Conversational	Visual	Verbal	Auditory
Problem statement	Verbal	Visual	Auditory	Haptic	Conversational
Diagram, figure, graph, index and slide	Visual	Verbal	Haptic	Conversational	Auditory
Table, narrative text and lecture	Auditory (if they are presented as audio files)	Verbal	Visual	Conversational	Haptic

Tab 7.2: Mapping between LO learning resource types and learner cognitive styles

3. **Interactivity level:** this parameter determines the interactivity degree of a LO. Interactivity in this context refers to the degree to which the learner can influence the behavior of the LO. The values that can be assigned to this parameter are: very low, low, medium, high and very high. It is obvious that there is a relationship between the interactivity level and the interactivity type of a LO. An active LO for example normally has a medium to very high interactivity level, while an expositive LO typically has a very low interactivity level.

4. **Semantic density:** this parameter is defined as the degree of LO conciseness. The values that can be assigned to this parameter are very low, low, medium, high and very high. A LO's semantic density is independent of its difficulty. It is best illustrated with expositive LOs. However, it can be used with active ones as well.
5. **Intended end user role:** this parameter determines the user for which the LO was designed. The values defined in LOM for this parameter are teacher, author, learner and manager. Notice that the learner is the user who works with the LO, while the author is the user who creates the LO. The manager is the person who manages the delivery of this LO, e.g. for a university or college. Clearly, a relationship can be observed between the intended end user role of a LO and the role of a learner. As the focus is on the user who works with LOs, only the learner is of interest for the adaptivity process. To cover the user roles, i.e. guest, member and active member, the values of the intended end user role should be extended to include these values. We propose to exchange the learner intended end user role with three values, namely guest, member and active member. This simplifies the matching between the intended end user role of a LO and the role of a learner.
6. **Difficulty:** this parameter determines how difficult it is to work with a LO. LOM differentiates between five levels of difficulty, namely very easy, easy, medium, difficult and very difficult. Considering the characteristics of learners, we can see a tight relationship between the difficulty of a LO and the pre-knowledge of a learner. Clearly, for a better understanding of contents, the level of the contents should match the level of the learner. The relation between the difficulty of a LO and the pre-knowledge of a learner are illustrated in table 7.3.

LO difficulty	Learner pre-knowledge
Very easy	Level 1
Easy	Level 2
Medium	Level 3
Difficult	Level 4
Very difficult	Level 5

Tab 7.3: Mapping between the difficulty of a LO and the pre-knowledge of a learner

7. **Description:** this parameter contains comments that determine how a LO should be used. There are no proposed values for this parameter in the LOM standard. It may contain hints, however, that help the adaptivity process in selecting the best LOs for the learner. We propose the use of this parameter to provide hints indicating which goal the LO is suitable for. Thus, the comments should be generation, application, looking up and repetition. Notice that these comments correspond to the learning goals defined for the learner, see section [7.5.4](#).

7.5.6. *Learning Contents Structure and Personalization*

This section describes the structure of the contents created by MAeLE and provides an insight into how the metadata of such contents are organized as well as how they can be personalized. As mentioned previously, the smallest pieces of learning contents are called assets. They are described by means of metadata, structured according to the LOM standard, and combined

with each other to form the whole course. The main advantage of the assets is their reusability, which in turn is highly affected by the size and the granularity of the assets [Alf06]. A well-known reusability concept is presented in [Hor02]. This concept defines five levels, which are media, pages, lessons, courses and curricula. The media present the assets that can be combined with each other to form meaningful pages. The lesson is a set of pages describing a certain topic of the course. A curriculum includes many courses covering several issues. Similar reusability concept is used by MAeLE as well. The difference is that the media are combined with each other to form meaningful LOs instead of pages, see figure 7.9. LOs form in turn the lessons of a course.

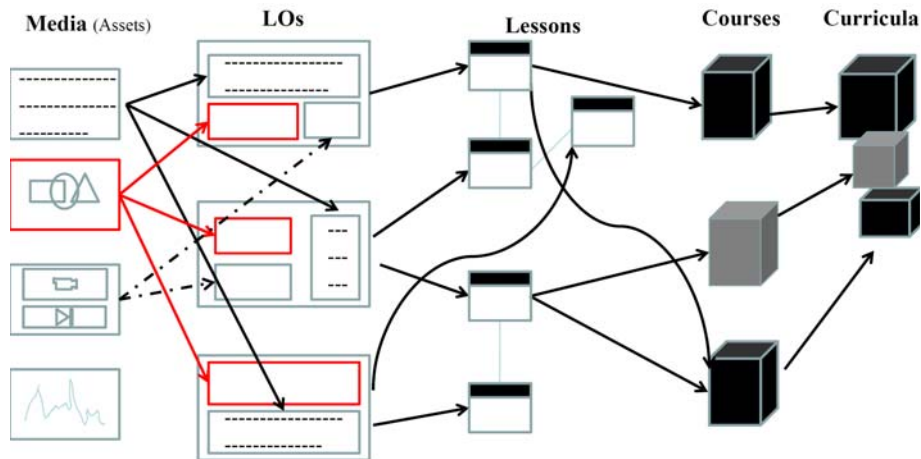


Fig 7.9: Reusability concept applied to MAeLE

So as to structure eLearning contents according to the above described concept while taking into account the personalization of these contents, a 3D up-to-down model starting from the curricula and moving down to the media has been developed. As an organization begins defining its curricula, it has to define the set of users for which courses should be created as well as with which languages. The structure of the curricula is visualized using a 3D coordinates. The Z axis represents the language. The Y axis represents the intended end user role, while the X axis denotes the courses themselves. The structure of curricula and an example curriculum are shown in figure 7.10. Notice that the curriculum of a learner is in principle a learning path in this structure, see the example curriculum in the figure.

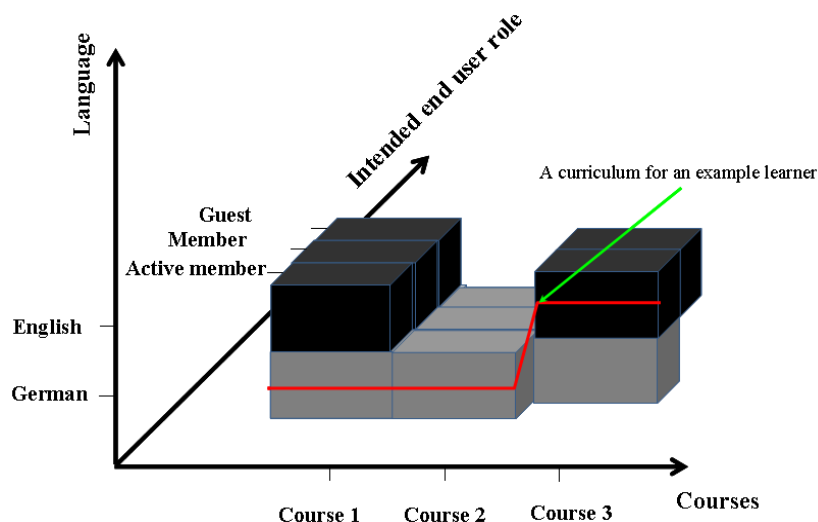


Fig 7.10: Structure of curricula

The example learner first studies course 1 and course 2 in German. He is an active member in the both courses. Afterwards, the example learner studies the third course in English and takes on another roll (a member in this example). The personalization of curricula can be done simply by defining the path of each learner in the proposed structure.

After defining the whole curricula, the courses themselves should be created. Each course author has to determine the goals learners may have. Subsequently, difficulty levels of the course are chosen. The structure of the course can also be visualized by means of a 3D coordinates. The difficulty is presented on the Z axis, while learning goals of learners are shown on the Y axis. The X axis stands for the course lessons, see figure 7.11. As mentioned previously, learning goals of learners are included in the LOM parameter named **description**. Notice that the chosen difficulty levels strongly depend on the learner group for which the course is created. Clearly, it is not necessary to have all difficulty levels defined by the LOM standard for each learning group.

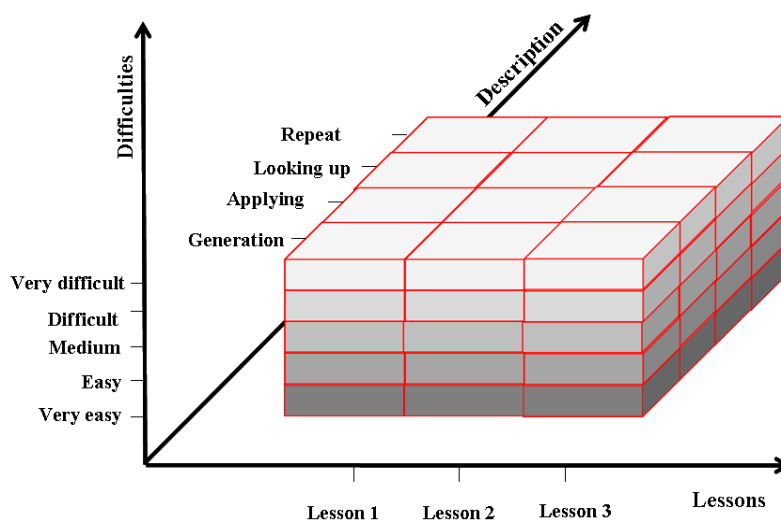


Fig 7.11: Structure of an example course

The next step is to define the structure of LOs for each lesson. The most important parameters describing a LO are the semantic density and the context of the LO. The context denotes the environment, within learning of the LO takes place. Using the context to describe a LO of a lesson requires extending the list of values defined by the LOM standard (e.g. for a university, values such as laboratory, lecture, etc. can be used). The structure of each lesson is then described by means of 3D coordinates as depicted in figure 7.12.

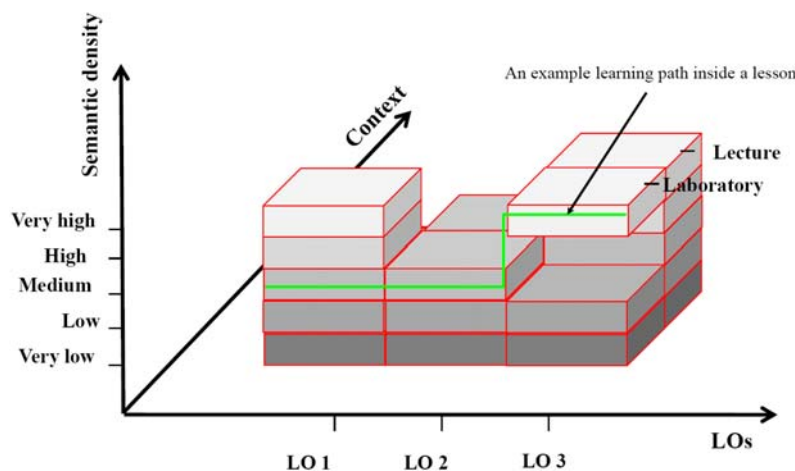


Fig 7.12: Structure of an example lesson

The Z axis stands for the semantic density, while the Y axis denotes the context. The axis X refers to the lesson's LOs. Notice that the selected levels of the semantic density and the context highly depend on the group of learners for which the course is created. The personalization of each lesson is achieved mainly by defining the path between the LOs that form the lesson, see the example illustrated in the figure above.

The last step is to structure the LOs themselves. As known, each LO consists of many assets that can be simply structured according to their interactivity and resource type. Therefore, we use 3D coordinates to structure each LO. The Z axis represents the interactivity type, while the Y axis denotes the resource type of assets. The X axis refers to the assets of the LO, see figure 7.13. Notice that the proposed structure of the LO simplifies the personalization of it according to learner characteristics. A personalized LO is only a path between the assets included in this LO.

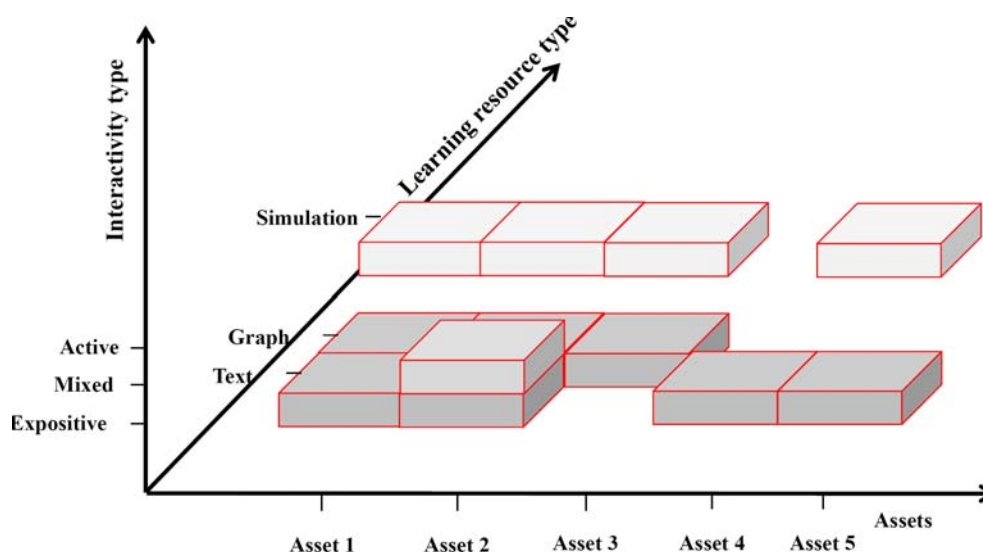


Fig 7.13: Structure of an example LO

The personalization process of MAeLE works in a way opposite to structuring the course. First, notice that all properties used to structure learning contents in the different levels of the applied reusability concept are, in principle, metadata specified according to the LOM standard with a minimal extension. Furthermore, they are used to describe the assets and LOs forming the courses. In order to simplify the personalization process, MAeLE considers each lesson and course as a single LO and refers to them as a combined LO. Each combined LO is described using the metadata the LOM standard uses to describe assets and LOs. The metadata describing combined LOs are not entered by the course author as is the case for assets. In contrary, MAeLE derives these metadata automatically depending on the metadata of the assets and LOs forming them.

Based on the discussion above, a bottom-up process is used to generate personalized courses. More specifically, LOs are personalized first depending on the interactivity and resource type of assets forming them. As mentioned previously, the adequate interactivity and resource type for a specific learner can be derived from his learning and cognitive style. Subsequently, each LO is assigned the appropriate metadata, which in turn are derived from the metadata describing the assets included in the LO. Following this, lessons are personalized depending on the semantic density and the context of the LOs forming them. Each lesson is then considered as a combined LO and assigned the appropriate metadata. The next step is the personalization of courses depending on the difficulty and the description of the lessons included. Personalization of curricula is then achieved based on the language of courses and

the intended end user role. Notice that learners may choose their curriculum independently as well.

7.5.7. Case Studies

Let us now clarify the personalization of eLearning contents by means of two example learners of the course **Mobile Communication**. This course is offered for Bachelor students in the faculty of Computer Science and Automation at the Ilmenau University of Technology. The assumed characteristics of both learners are listed in table 7.4.

Characteristic	Learner 1	Learner 2
Learning goal	Generation	Applying
Pre-knowledge	Level 1	Level 2
Cognitive style	Visual	Haptic
Learning style	Theorist	Activist
Motivation	Intrinsic	Intrinsic
Learning platform	Learning by distributing	Learning by interacting
Role of the learner	Active member	Active member

Tab 7.4: Characteristics of the two example learners

Let us now personalize an example LO for both learners. The example LO explains MIPv4. It contains 4 assets. The first asset contains an introduction to the protocol, while the second explains the movement detection procedure. The third and fourth assets present the handoff procedure and the drawbacks of MIPv4, respectively. The four assets are shown in figure 7.14.

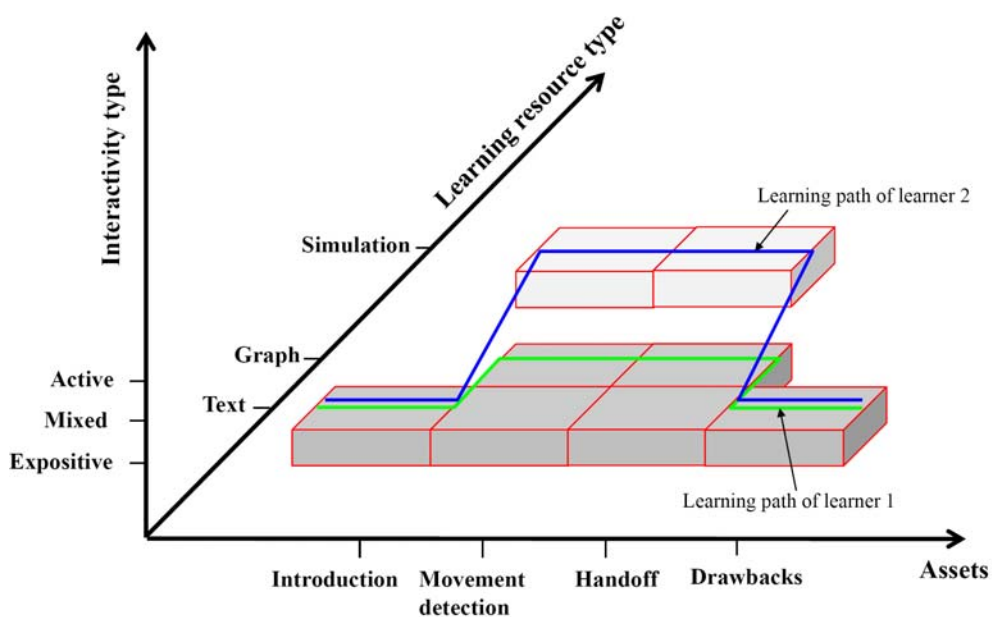


Fig 7.14: Personalization of an example page

As can be seen from this figure, the first and fourth assets exist only as a text with an expositive interactivity type. The second as well as the third asset exists as a text and a graph with an expositive interactivity type and as a simulation with an active interactivity type. Thus, the first and the fourth assets are chosen as texts for both learners. For the second and third assets, the first learner obtains the graph, while the second learner obtains the simulation. Learning paths between the four assets are shown in the figure for both learners.

7.5.8. *Prototype Implementation*

MAeLE is implemented prototypically as an adaptive framework for Moodle. The following briefly introduces the implemented prototype. The installation of our framework is rather simple. First, Moodle should be extracted to the folder “htdocs” of the web server used. MAeLE must then be extracted to the folder “htdocs/moodle”. Afterwards, the installation of Moodle is performed without any changes. New adaptive courses can be created in Moodle in the same way as normal courses, see figure 7.15.

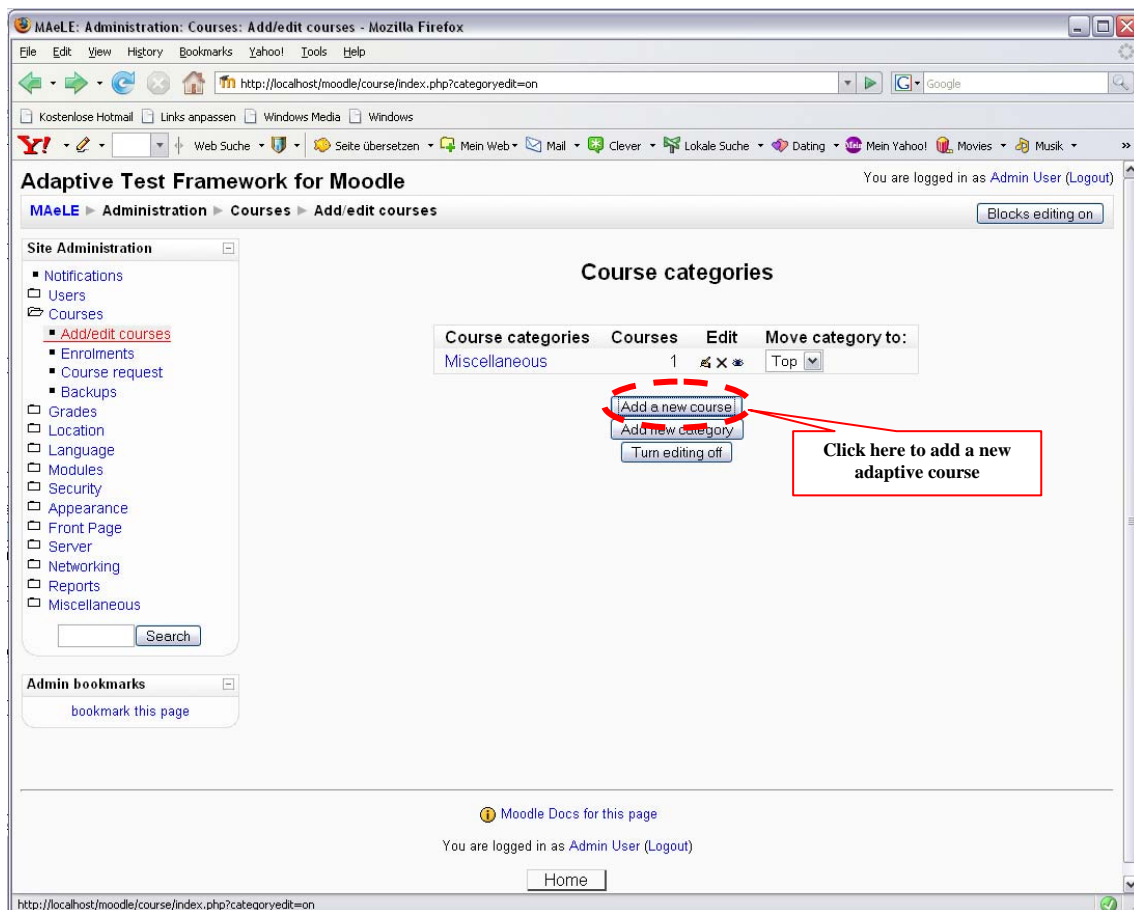


Fig 7.15: Creation of a new adaptive course in Moodle

After the adaptive course has been created, course contents should be added. This is done simply and in the standard way used by Moodle. The course administrator has to select **Adaptive Course** to add the course specified for the first topic, see figure 7.16. Notice that each topic is a separate course from MAeLE point of view, while all topics are one course from the user’s point of view. Keeping in mind that users may characterize themselves incorrectly when they register for the course, separating the whole course into small courses is aimed at optimizing the adaptivity process by utilizing the experiences gathered from visited topics to increase the probability of a correct personalization of unvisited topics. Moreover,

this enables the export of one topic as an eLearning course to other eLearning systems. The item **Adaptive Course** does not exist in the standard Moodle. It is integrated explicitly for MAeLE. An example course for mobility management is shown in figure 7.17. Notice that the adaptivity process is triggered by user access to the course.

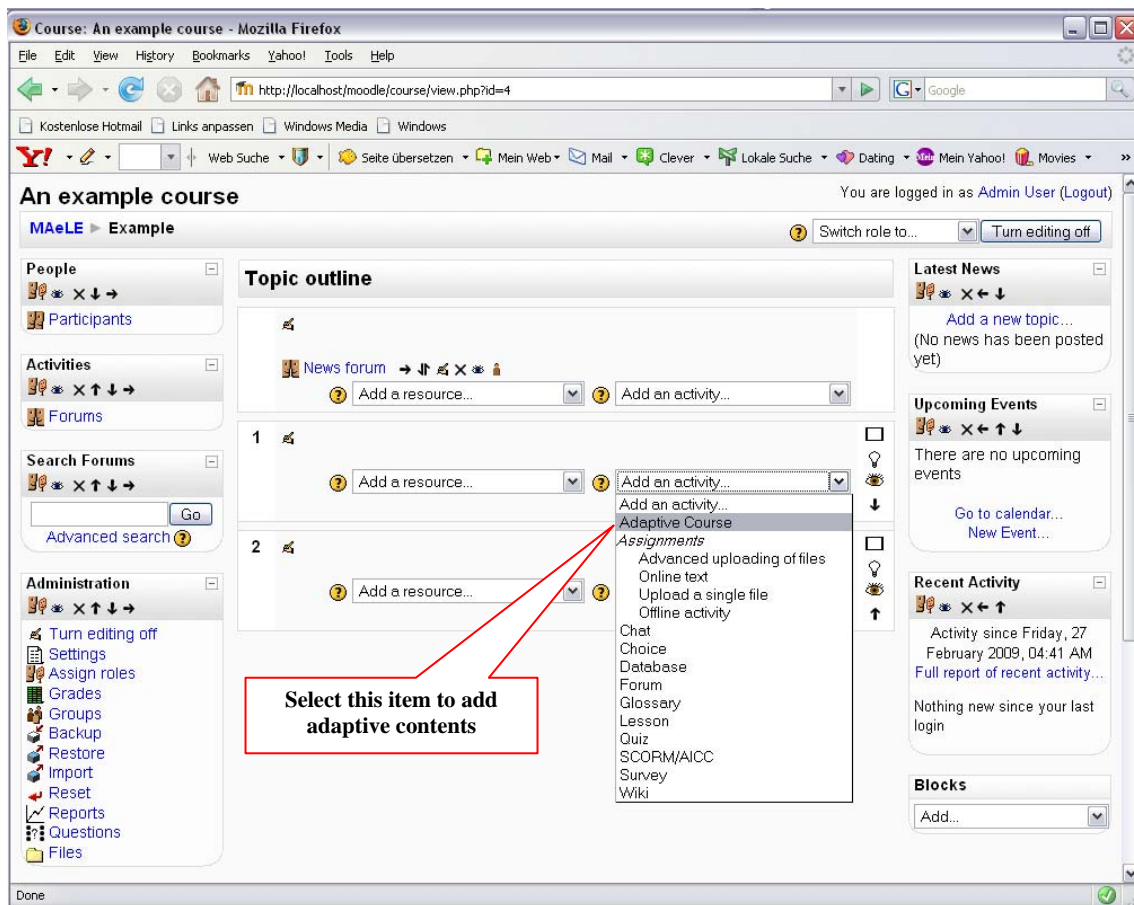


Fig 7.16: Adding of contents to the created adaptive course

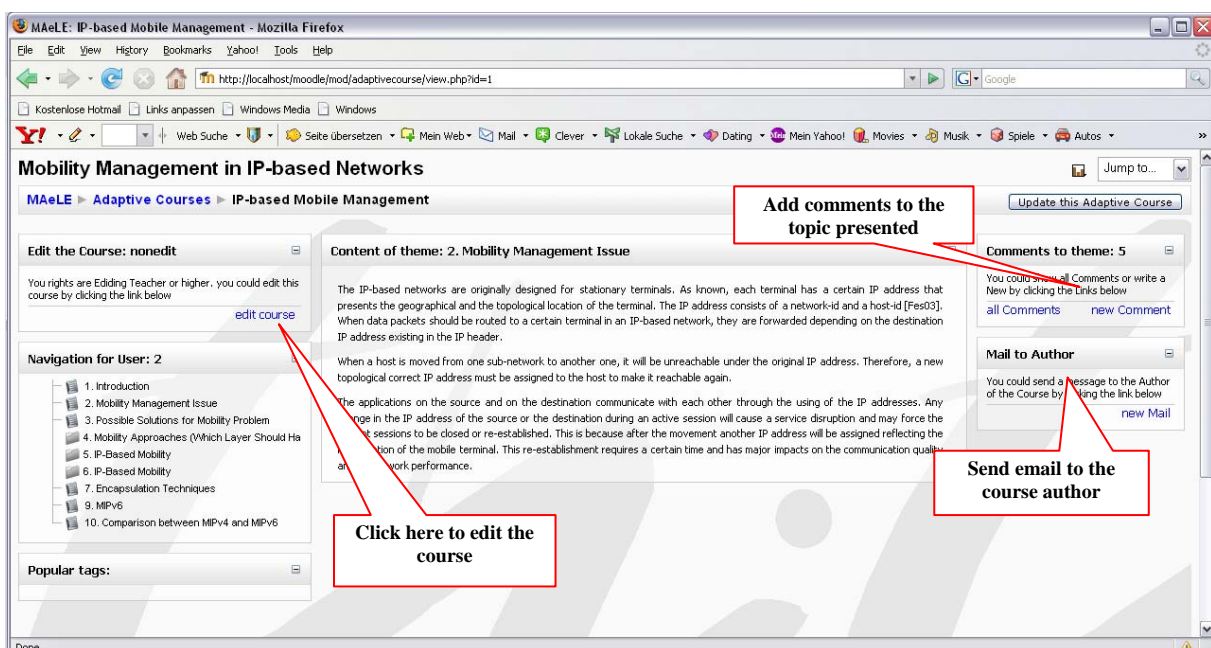


Fig 7.17: An example course for mobility management

The frontend of MAeLE enables users to comment any LO of the course or to write an email to the course author. The frontend is the same for all users with all rights levels Moodle assigns. Clearly, users with administration rights see more options than normal users. They can edit the course, create assets and LOs and assign metadata, see figure 7.18. As this figure shows, MAeLE provides many toolbars to simplify the work of authors. The course navigation can be edited in a simple way as well. The author must click on “edit tree” to edit the whole course navigation, see figure 7.19.

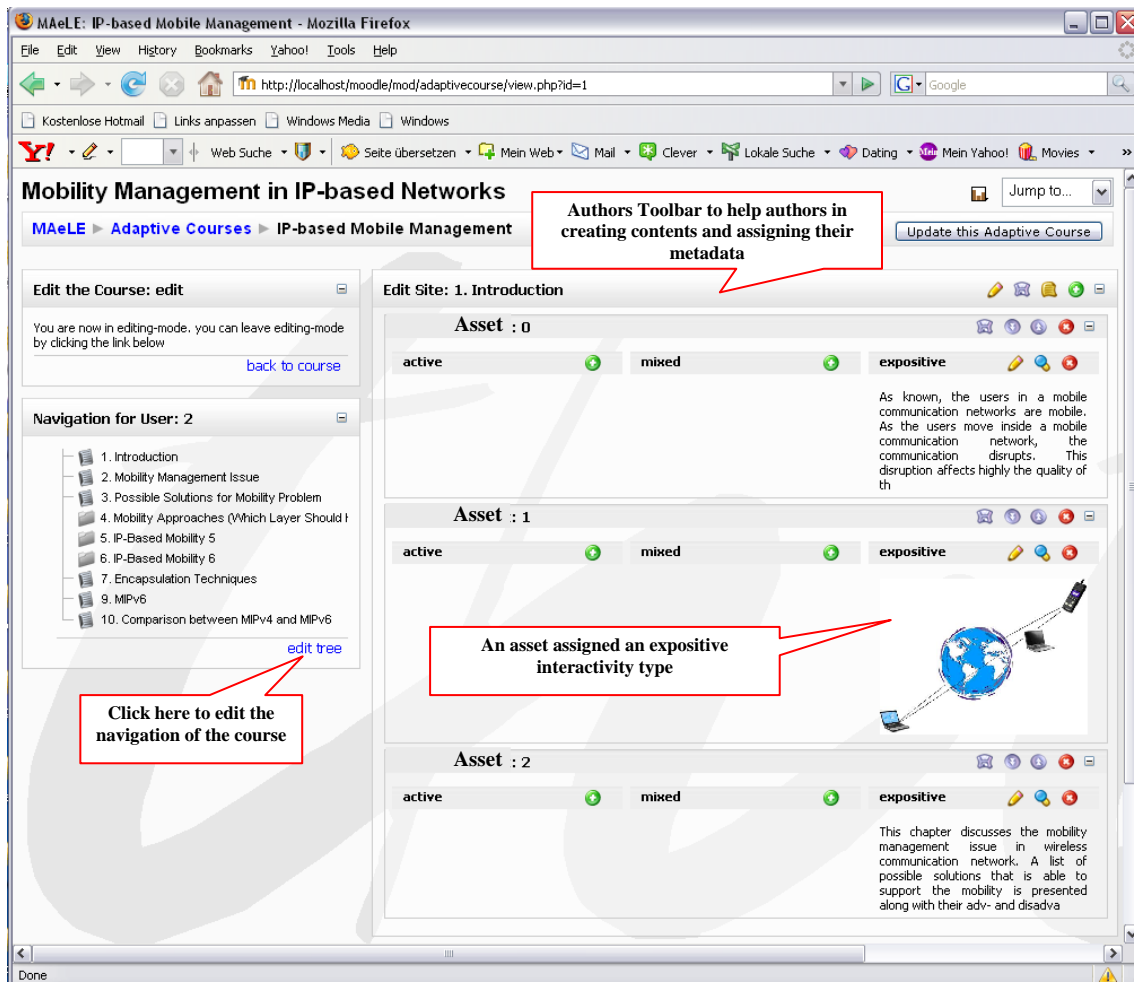


Fig 7.18: Authors frontend of MAeLE – editing of LOs

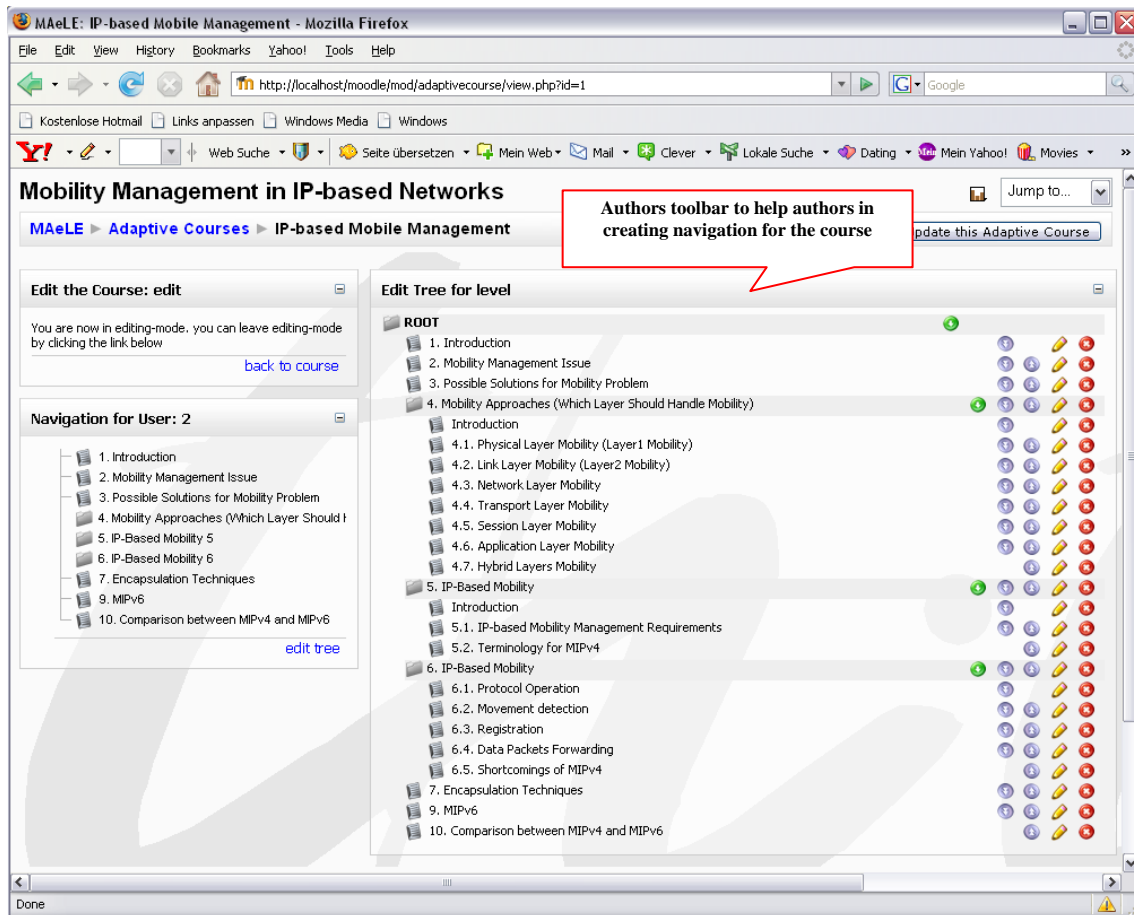


Fig 7.19: Authors frontend of MAeLE – editing of course navigation

7.6. Conclusion

This chapter has discussed the building of an adaptive eLearning environment to support the study and dissemination of mobility management issues covered in this dissertation. The chapter has briefly introduced the benefits of eLearning and has reviewed eLearning platforms, standards and previous efforts to support adaptivity. The review has concluded that there is a lack of adaptivity support in existing environments, which focus mainly on achieving either navigation or presentation adaptivity based on one or more selected user characteristics. Many important user characteristics are not taken into account in the adaptivity process. Moreover, there is a lack of providing both navigation and presentation adaptivity simultaneously. Finally, most existing environments do not separate between contents and the sequence logic, which prohibits the reusability of eLearning contents.

So as to eliminate the drawbacks of previous eLearning environments, a new adaptive eLearning environment named MAeLE has been developed. The new environment extends the functionality of a traditional LMS by means of an adaptivity framework. It generates personalized eLearning courses with adequate navigation. The main idea is to provide contents with adequate metadata describing them and stored in a separate database, which in turn enhances contents reusability. The chapter has structured the environment and has defined the metadata describing the contents and their relation to user adaptivity-relevant characteristics. To be compatible with existing standards, LOM has been used to describe eLearning contents. The separation between contents, sequence logic and metadata enables the reusability of eLearning contents created by MAeLE. The proposed framework is flexible

and extensible. New rules, strategies, navigation methods and contents can be simply integrated into the environment. The proposed environment has been implemented as a prototypical framework extending Moodle functionalities. The prototype has been briefly presented in this chapter as well.

8. Conclusions and Outlook

This chapter concludes the dissertation with the main results and provides an outlook for further research on mobility management issues. The chapter is organized as follows: section [8.1](#) summarizes the main results, while section [8.2](#) delivers an outlook for further research on the topics presented in the dissertation.

8.1. Conclusions

This dissertation has addressed the mobility management issue in IP-based mobile communication networks. As known, wireless communication networks experience tremendous development. Furthermore, a wide variety of new applications far beyond classical phone services arise continually. The tremendous success of the Internet along with the demand for always-on connectivity regardless of users' locations has triggered the development of 4G networks. This new generation is intended to complement and replace 2G and 3G systems. Ubiquitous access to information and use of applications anywhere and anytime, supporting large data volumes and guarantee of minimum delays are the key features of 4G networks. Different from previous networks, 4G networks comprise a set of heterogeneous networks integrating different existing and future systems (e.g. GSM, UMTS, WLAN, WIMAX, LTE/SAE systems, etc.) by means of a common IP core. 4G is termed, therefore, All-IP and is expected to be widely deployed in the near future, serve fixed as well as mobile subscribers and offer any type of service, anytime, anywhere and anyhow under dynamic network conditions. However, in order to reach the ambitious goals of 4G, several challenges must be solved, e.g. to efficiently manage mobility between cells connected by an IP core, to guarantee a certain QoS, to secure communication links and content, etc.

The mobility problem in IP-based mobile communication networks lies in the dichotomy of IP addresses. An IP address of a certain MN represents the point of attachment. This IP address is required to establish a session between the MN and other nodes in the network. In case the MN changes its point of attachment, it should be assigned a new topologically correct IP address. Changing the IP address during an ongoing session may result in dropping this session, typically resulting in a disruption of the application. IP-based mobility management aims mainly at minimizing or even eliminating this disruption. Other goals, however, should also be satisfied, e.g. minimizing of costs, interworking properly with IP routing and features, security, scalability, deployability, robustness, etc.

The classical solution used to support mobility in IP-based mobile communication networks is MIP. Each MN is assigned two IP addresses. The first is a fixed address, referred to as a home address, acting as an identity for the MN. The second IP address is, however, a temporary one and defines the current point of attachment. Data packets destined to the MN are tunneled normally from the HA to the current point of attachment. Performance studies have shown that MIP suffers from many drawbacks including triangular routing, encapsulation overhead, poor handoff performance, etc. Therefore, MIP is not adequate for delay-sensitive applications. It is only suitable for supporting global mobility, referred to as macro mobility as well. The thresholds of MIP have triggered the development of new mobility management solutions able to satisfy requirements of future All-IP networks.

Based on the motivation presented above and on an in-depth analysis of mobility management problem in IP-based mobile communication networks, the goals of this dissertation were defined as follows: analysis of existing mobility management solutions implemented in different layers of the TCP/IP protocol suite focusing on their ability to satisfy the requirements of future All-IP networks. If there is a need for further developments, a new mobility management solution should be developed. This solution has to minimize or even restrict the handoff latency to the latency resulting from changing the radio between two points of attachment. The developed solution should be evaluated by means of mathematical models and simulation studies. At last, an adaptive eLearning environment has to be developed to support studying and dissemination of mobility management issues covered in this dissertation. By means of such an environment, researchers can quickly get involved in current research areas and other users (e.g. students, lecturers, etc.) will obtain courses introducing the offered topics and covering deficiencies in their knowledge.

Considering the objectives of this dissertation, a classification of mobility management approaches regarding their implementation in the layers of the TCP/IP reference model was introduced in chapter 2. Subsequently, well-known solutions implemented in each layer were briefly described and classified. The main obtained result says that network layer mobility management solutions emerge to be the most suitable to provide mobility support in future All-IP mobile communication networks. Therefore, mobility management solutions implemented in the network layer were studied in more detail in chapter 3. Well-known layer 3 mobility management solutions were presented according to the classification presented previously in chapter 2. Furthermore, the described solutions were compared to each other with respect to the handover management, paging, new nodes that should be introduced to the network, nodes that should be updated, used network topology, dependency on layer 2 information, usage of a tunnel, expected handover performance and load balancing. This comparison has shown that each layer 3 mobility management protocol attempts to somehow improve the performance. However, this improvement is done either by making some constraints on the network or on the MN itself. Therefore, there is an inevitable need to develop a layer 3 mobility management solution that can satisfy the requirements of future All-IP networks and avoid the drawbacks of existing solutions.

The aim to satisfy the requirements of future All-IP networks has led to the development of a layer 3 mobility management solution named MIFA. It advances the state of the art and supports a continuous communication between the CN and the MN while registration is in progress. This is achieved by delegating the authentication to the FAs/ARs, so that the MN requires only contacting its new FA/AR to be able to resume communication. Local authentication relies on groups of neighbor FAs/ARs, to which the MN may move in the future. Each FA/AR establishes such a group, termed L3-FHR, and provides its members with information related to the MN in advance. Provided that the MN has moved to one of these members, the distributed information enables fast re-authentication of the MN after the handoff. The specification of MIFA was defined for both IPv4 and IPv6 networks. It can be operated in predictive as well as reactive mode. MIFA localizes the mobility management without introducing any new intermediate nodes more than currently known from MIP. Moreover, no restrictions on the topology are made.

After describing the new proposal, this dissertation has focused on the evaluation of the new solution compared to other existing ones. The evaluation was first made using mathematical model and has focused on the evaluation of the performance and the estimation of the cost resulting from employing the studied solutions. The motivation behind the mathematical analysis lies in the quick development of such models that are able to provide a good estimation of performance. Although the implementation as well as simulation delivers detailed and accurate results, it normally takes a long time. This dissertation addressed the

problem that there is no generic mathematical model that can be used to analyze a wide range of mobility management solutions. Therefore, such a generic mathematical model was developed and described in chapter 5. The developed model analyzes the performance with respect to the average handoff latency and expected average number of dropped packets per handoff taking control messages dropping into account. It can be applied to break-before-make as well as make-before-break mobility management protocols. The cost estimation focuses on the estimation of the location update and packet delivery cost. The total cost is calculated, after that, as the sum of these two costs with an adequate weighting factor for each. The parameters of the generic model must be selected according to the characteristics of the studied protocols, mobility scenarios and network topologies. This has simplified studying the impact of mobility scenarios and network topologies on performance and cost. After describing the generic mathematical model, it was applied to compare MIFA to well-known mobility management protocols, namely MIPv4, MIPv6, MIPRR, HAWAII, Proxy MIPv6, pre-registration method for MIPv4 and FMIPv6. The same mobility scenario is used, while two symmetric network topologies (a hierarchical and a mesh one) were used in this analysis. Each protocol uses the topology best matching its requirements. After that, a comprehensive analysis of the impact of mobility scenarios was achieved as well. For this purpose, MIFAv6, MIPv6, Proxy MIPv6 and HAWAII were analyzed in a hierarchical topology under different mobility scenarios. The analysis has shown that MIFA presents a very fast mobility management scheme. It clearly outperforms the other studied mobility management solutions with respect to the average handoff latency and expected average number of dropped packets per handoff. Employing MIFA in predictive mode minimizes the handoff latency on downlink and uplink to the latency resulting from the layer 2 handoff. The lost in downlink data packets can be even eliminated. MIFA in this mode is sufficient for high speeds as well. According to the achieved analysis, the MN can cross an overlapping area between two neighbor cells at a maximum speed of 118 km/h without suffering from any latency more than the layer 2 handoff latency if the length of the MN's path inside the overlapping area is 10 meters. Moving faster than 118 km/h forces MIFA to be operated in reactive mode. This has, however, a negligible impact on the performance on uplink. Although more latency is suffered on downlink, MIFA remains adequate for delay-sensitive applications. Considering the cost resulting from employing MIFA compared to that resulting from employing other studied solutions, MIFA does not produce additional packet delivery cost than that resulting from MIP. It outperforms, therefore, most other studied protocols. However, the significant improvement in the performance is achieved at the cost of adding additional signaling messages, which has made the location update cost resulting from MIFA more than that resulting from most studied protocols.

Chapter 5 has addressed the fact that performance and cost are usually analyzed separately, although they relate to each other. Improving performance is achieved mostly by introducing some extra cost. Network providers wanting to update their infrastructures to support a new mobility management solution are interested in knowledge of the performance they will gain as well as the cost they must consider. Such a discussion was addressed in chapter 5, which provided methods to answer questions such as which performance gain can be achieved at which cost? In order to validate the mathematical model, its results were compared to results of simulation as well as real testbeds. MIFAv4 and MIPv4 were evaluated using the same network topology and mobility scenario under the same conditions by means of the generic mathematical model and simulation studies modeled in NS2. The validation has proven that the developed generic mathematical model delivers a sound evaluation of the performance of mobility management protocols under low-loaded networks. The accuracy of the generic model lies in a range of ± 23 % for different loads. To validate the generic mathematical model compared to results of real testbeds, our mathematical model was applied to the testbeds implemented in [Fes03] to validate MIPv4 and MIPRR. Clearly, the same

assumptions of the testbeds and the same mobility pattern are used. The validation has shown that the accuracy of the generic mathematical model lies in a range of $\pm 30\%$ of the testbeds results.

After the mathematical analysis has shown that MIFA presents a promising solution, a detailed evaluation by means of simulation studies was required. For this purpose, MIFA was modeled in NS2 and compared to a macro mobility solution (MIP) and a micro mobility one (HAWAII). The selected simulation scenarios were oriented mainly to study the impact of network topology, network load and MN speed. The simulation results showed that MIFA performs better than both selected solutions in approximately all studied cases. There is no significant impact of network topology on the performance of MIFA in predictive mode. Of course, a good network planning is essential. More concrete, the cells should overlap, so that the MN can trigger the predictive mode in most cases. For MIFA in reactive mode, a mesh network topology improves the performance as compared to a hierarchical one with respect to the number of dropped packets per handoff on downlink. For the handoff latency and the number of dropped packets per handoff on uplink, there is no significant impact of network topology. Considering the load impact, increasing the load increases the handoff latency and the number of dropped packets per handoff for all studied protocols. However, MIFA remains better under all loads. Regarding the impact of the MN speed, a ping pong effect is clearly seen with MIFA causing higher average handoff latencies and so more dropped packets per handoff at slow speeds (the speed of a pedestrian in this study) than at faster ones. Increasing MN speed decreases this effect with MIFA, which performs very well even at high speeds. Increasing the speed to a very high value (300 km/h in this study) clearly degrades the performance. To overcome the ping pong effect at slow speeds, the dissertation has proposed adapting the movement detection algorithm to the MN speed, e.g. LCS for slow speeds and ECS for high speeds.

Following a detailed analysis of MIFA by means of simulation studies, this dissertation focused on building an adaptive eLearning environment that should be used to support the study and dissemination of mobility management issues covered in this dissertation. After introducing the term eLearning and well-known standards for eLearning, adaptivity basics and currently existing adaptive eLearning environments were briefly discussed. This dissertation determined that there is a lack of adaptivity in existing environments, which focus mainly on adapting either the navigation or the presentation of eLearning contents depending on one or more selected user characteristics. This dissertation highlighted the fact that many important user characteristics are not taken into account. There is a lack of providing both navigation and presentation adaptivity at the same time. Most existing environments do not separate between contents and sequence logic, which prevents the reusability of eLearning contents. The analysis of existing adaptive eLearning environments has shown that there is a need to develop a new adaptive eLearning environment able to overcome the drawbacks of previous ones. This has led to the structuring of a new adaptive eLearning environment named MAeLE, which extends the functionality of a traditional LMS through an adaptivity framework. It generates personalized eLearning courses with adequate navigation at run-time. The adaptivity in MAeLE depends on users' characteristics and adequate metadata related to eLearning contents. After defining the structure of MAeLE, the user modeling in the proposed environment has been discussed. User characteristics were mapped to metadata describing the contents. For this purpose, LOM was investigated and extended. The separation between contents and sequence logic or metadata enables the reusability of eLearning contents. The proposed framework is flexible and extensible. New rules, strategies, navigation methods and contents can simply be integrated into the environment. MAeLE is implemented as a prototypical framework extending Moodle functionalities. A brief description of the prototypical implementation was introduced in chapter 7.

8.2. Outlook

Depending on the topics handled in this dissertation, many new interesting research questions arise:

1. **Paging extension for MIFA:** as mentioned in this dissertation, the significant improvement of the performance of MIFA is related to increased location update cost little bit. This cost can, however, be reduced through paging support. Until now, there are no paging functions implemented in MIFA. It will be useful to do some research in this direction. The main arising questions here are: how to define the optimal size of paging areas? Should this size be fixed or dynamic depending on certain factors, e.g. network load, number of MNs, etc.? Would the paging area size be the same for all MNs or optimized for each one? How paging procedure should be designed?

Answering the above questions requires an in-depth analysis of paging issue, studying the impact of the network load and MN mobility patterns on the size of paging areas, comparing between the cost resulting from using fixed and dynamically changing paging area sizes, etc. An efficient paging extension will enhance MIFA and enable it to serve a larger number of MNs.

2. **Network based mobility management:** as mentioned in chapters [2](#) and [3](#), network-based mobility management solutions aim at managing mobility without interactions with MNs. In such a way, the nodes with legacy IP stack can be mobile and benefit from mobility support. This research field seems to be promising. Therefore, it would be interesting to further develop MIFA to be able to support mobility without interactions with MNs. This will enable the MNs that do not support MIFA or even the nodes that do not support mobility at all to benefit from the fast and smooth handoffs offered by MIFA.
3. **Enhancing MIFA to support QoS:** this dissertation has shown that MIFA represents a very good solution for mobility management. However, no QoS is supported. It would be very useful to enhance MIFA with functions able to offer QoS. There are currently some efforts being done to further develop MIFA to be able to offer QoS. These efforts focus mainly on integrating MIFA with other QoS protocols, such as RSVP [[BZB97](#)] and QoS NSIS Signaling Layer Protocol (QoS-NSLP) [[MKM08](#)]. There are currently two solutions being developed, namely QoS-aware MIFA (QoMIFA) [[AMB06](#)] and Mobility management aware next step In Signaling for All-IP Mobile communication networks (MaISAM) [[AMD08](#)]. QoMIFA and MaISAM can offer seamless mobility simultaneously with fast reservation of resources during and after handoffs. These two solutions can at the moment be operated in reactive mode only. Further development of these solutions to be operable in predictive mode is of a great interest. In addition, comprehensive simulation studies are required to evaluate both proposals.
4. **Mobility management in the networks employing the Multi-Protocol Label Switching (MPLS) [[RVC01](#)]:** this dissertation has specified MIFA for IP networks. There are, however, many network providers implementing MPLS in their backbones. With MPLS, packets forwarding depends on short labels of fixed length. The header of an IP packet is only processed once at the boundary of the MPLS network. Within the MPLS cloud, forwarding is based on labels placed in front of the IP header (layer 2.5). The path a packet takes is based on its label and is called a Label Switched Path (LSP). Each packet assigned to the same Forwarding Equivalence Class (FEC) is marked with the same label and uses exactly the same path through the network. By employing signaling protocols, like RSVP Traffic Engineering (RSVP-TE) [[ABG01](#)],

a LSP can be set up and managed dynamically by using dynamic or static (pre-computed) routes. Additionally, the LSP can be configured to provide QoS guarantees and to follow automatic reconfiguration when failures occur or network states change. Nowadays, MPLS is being employed widely and there is a trend to enhance it with mobility functions. Therefore, it will be useful to integrate the traffic engineering and QoS capabilities of MPLS with MIFA as a mobility management protocol. There are currently some research efforts being done to achieve such integration. An initial proposal attempting to integrate MIFA control messages with MPLS signaling traffic has been developed [DBM06]. The main advantages of the proposed approach are the transparency to MNs that must support MIP only, reducing the encapsulation overhead for IP-in-IP tunneling, fast handoff (uplink and downlink), fast re-reservation, fast release of resources, fast recovery of failures (message dropping, no support of MIFA, etc.), enhancing of security and allowing for a stepwise integration in existing networks. The specification of the proposed approach, however, is not yet complete. In addition, the new proposal does not attempt to utilize the layer 2 triggers to optimize the performance and does not support paging. There is need for further research to address these issues. Additionally, it is very important to define how the MPLS and IP domains supporting MIFA can communicate with each other.

5. **Improving the accuracy of the generic mathematical model:** this dissertation has presented in chapter 5 a generic mathematical model capable of evaluating a wide range of mobility management protocols. The developed model significantly simplifies the evaluation of mobility management protocols and enables a comparison between them under the same conditions (network topology and mobility scenario). The validation results say that the accuracy of the generic model lies in a range of ± 23 % for different loads. Further research studies on the load impact are, however, necessary. More concrete, the load should be considered when calculating the handoff latency, number of dropped packets per handoff, location update cost and packet delivery cost. The main aim is to increase the accuracy of the model under different network loads. Further simulations to validate the load impact are necessary as well.
6. **Further development of the generic mathematical model:** in addition to improving the accuracy, there are many other research trends that can be addressed in this direction. Until now, the model evaluates the performance using UDP traffic only. Developing the model to evaluate the protocols using TCP traffic is very important. In addition to this, developing the model to evaluate the protocols that support QoS simultaneously with mobility is of a great interest, e.g. to calculate the time required to reserve network resources, expected number of data packets sent as best effort until the resources are reserved, call blocking probability, etc.
7. **Further development of the adaptive eLearning environment:** this dissertation presented a new adaptive eLearning environment (MAeLE). This environment has been structured carefully. Modeling of users and matching their features to the content metadata has been discussed as well. However, the developed environment is rule-based. Further development of this environment to be able to use artificial intelligence principles may significantly improve the adaptation efficiency. In addition to this, statistical evaluation studies are necessary to evaluate MAeLE and to assess whether the used adaptation technique is suitable for learners or not.

A. MIFA Control Messages

This appendix provides a complete list of MIFA control messages including a brief description of them.

MIFAv4	
Message name	Description
<i>Agnt_Sol</i>	This message is broadcasted from the MN. Each FA that receives this message replies a unicast <i>Agnt_Adv</i> message. This message is built according to the standard specification of MIPv4.
<i>Agnt_Adv</i>	This message is broadcasted from FAs and is used to advertise their existence and properties. The message is built according to the standard specification of MIPv4 with one of the reserved bits as <i>MI</i> flag. This flag is set to 1.
<i>Pr_Rt_Sol</i>	This message is transmitted from the MN to the old FA after the appearance of the L2-trigger at the MN. It asks the old FA to unicast a <i>Pr_Rt_Adv</i> message. <i>Pr_Rt_Sol</i> message is used in predictive mode only.
<i>Pr_Rt_Adv</i>	This message is sent from the old FA on behalf of the new one as an answer of the <i>Pr_Rt_Sol</i> . The message is constructed similar to the <i>Agnt_Adv</i> message. However, the information included in this message relates to the new FA. <i>Pr_Rt_Adv</i> message is used in predictive mode only.
<i>Reg_Rqst</i>	<p>This message is used in the following two cases:</p> <ol style="list-style-type: none"> 1. During the initial registration procedure: <i>Reg_Rqst</i> is sent from the MN to the HA and is constructed according to the standard specification of MIPv4 with one of the reserved bits set as <i>MI</i> flag. <ol style="list-style-type: none"> a. This message is transmitted from the MN to the HA via the current FA. It is possible that the current FA adds some information required to distribute shared secrets to the current FA as well as the MN, see [PJA00] and [PCa01]. b. If a certain security mechanism is used to distribute the required shared secrets, e.g. AAA, the current FA does not add anything to this message. 2. During a normal registration using MIFAv4: <i>Reg_Rqst</i> is constructed according to the specification of MIFAv4 and is sent from the MN towards the new FA. It should contain: <ol style="list-style-type: none"> a. A MIFA authentication extension containing <i>Auth₁</i>. b. The <i>MI</i> flag set to 1. c. A Replay protection extension. d. A MN-FA authentication extension.
<i>Reg_Rply</i>	<p>This message is used in the following two cases:</p> <ol style="list-style-type: none"> 1. During the initial registration procedure: <i>Reg_Rply</i> is sent from the HA towards the MN as a response to the <i>Reg_Rqst</i> message. <ol style="list-style-type: none"> a. This message may contain information, from which the current FA and

	<p>the MN can derive the required shared secrets</p> <ol style="list-style-type: none"> i. between the MN and the current FA and ii. between the current FA and the HA. <p>b. If a certain security mechanism is used to distribute the shared secrets, e.g. AAA, no information related to the shared secrets is included.</p> <p>2. During a normal registration using MIFAv4 in both reactive and predictive mode: Reg_Rply is built according to specification of MIFAv4 and is transmitted from the new FA to the MN. It should contain:</p> <ol style="list-style-type: none"> a. A MIFA authentication extension containing $Auth_2$. b. The MI flag set to 1. c. A Replay protection extension. d. Two random variables and the shared secret for the next registration with the next new FA. e. A MN-FA authentication extension. <p>If a certain security mechanism is used, it will take care of distributing the shared secret and, thus, no security information is required in the Reg_Rply message.</p> <p>3. During the registration using MIFAv4 in predictive mode: Reg_Rply message is sent from the new FA to the old one, which uses this message as an indicator for the success/failure of the handoff.</p>
M_P_Not	<p>This message is used in the following two cases:</p> <ol style="list-style-type: none"> 1. During the initial authentication exchange procedure: The M_P_Not message is sent from the current FA towards the HA. A FA-HA authentication extension should authenticate this message. The message is used to ask the HA to send the MN-specific data. 2. During the information distribution procedure: M_P_Not messages are sent from the current FA to all members of the current L3-FHR. This message contains the MN-specific data, which include: <ol style="list-style-type: none"> a. A HA features extension containing features of the HA that can be offered to the MN, e.g. simultaneous binding, GRE, etc. b. The shared secrets between the MN and the members of the current L3-FHR and between these members and the HA. If these shared secrets are distributed by means of a key distribution mechanism or they can be derived by the members of the current L3-FHR themselves, no shared secrets will be sent with the M_P_Not message. c. The replay protection extension: <ol style="list-style-type: none"> i. For timestamp-based replay protection: This extension contains the HA's timestamp and the last received MN's timestamp, see section 4.2.9.5. ii. For nonce-based replay protection: This message contains two nonces. The first is the HA's nonce used in the current registration. The second nonce is the HA's nonce that should be used in the next registration with the next new FA. For details, see section 4.2.9.5.
M_P_Ack	<p>This message is used in the following two cases:</p> <ol style="list-style-type: none"> 1. During the initial authentication exchange procedure: The M_P_Ack message is sent from the HA towards the current FA. This message contains: <ol style="list-style-type: none"> 1. A HA features extension containing features supported by the HA and can be offered to the MN, e.g. simultaneous binding, GRE, etc.

	<ol style="list-style-type: none"> 2. MIFA authentication values ($Auth_1$ and $Auth_2$). 3. A Replay protection extension. 4. A FA-HA authentication extension. <p>2. During the information distribution procedure: <i>M_P_Ack</i> messages are used to acknowledge the receiving of <i>M_P_Not</i> messages. They are sent from the members of the current L3-FHR to the current FA. Each <i>M_P_Ack</i> message should contain a FA-FA authentication extension.</p>
<i>PFA_Not</i>	This message is sent from the new FA to the old one to notify it of the new CoA. It should contain the MN's home address as well as the new CoA. A FA-FA authentication extension should exist at the end of this message.
<i>PFA_Ack</i>	This message is sent from the old FA to the new one as a response to the <i>PFA_Not</i> message. It should contain a FA-FA authentication extension.
<i>HA_Not</i>	<p>This message is sent from the new FA to the HA to update the MN's new CoA at the HA. This message should contain:</p> <ol style="list-style-type: none"> 1. The shared secret between the HA and current L3-FHR members. This shared secret will be used to secure the messages that will be exchanged between the HA and the next new FA. If a security mechanism is used to distribute the shared secret, e.g. AAA, or the HA can derive this shared secret somehow, no information related to the shared secret should be included in the <i>HA_Not</i> message. 2. Two new random variables generated by the new FA. 3. A FA-HA authentication extension.
<i>HA_Ack</i>	<p>This message is sent from the HA to the new FA as a response to the <i>HA_Not</i> message. This message contains:</p> <ol style="list-style-type: none"> 1. A HA features extension. 2. MIFA authentication values ($Auth_1$ and $Auth_2$). 3. A Replay protection extension. 4. A FA-HA authentication extension
<i>Int_Ack</i>	This message is sent from the old FA to the MN to indicate that the <i>Reg_Rqst</i> message sent from the MN to the new FA via the old one has not been dropped on the wireless link. This message contains a handoff possibilities extension that indicates if a handoff to the new FA is possible employing MIFAv4 or not. This message is used in predictive mode only.

Tab A.1: Control messages of MIFAv4

MIFAv6	
Message name	Description
RS	This message is broadcasted from the MN. Each AR that receives this message responds by a unicast RA message. The RS message is built according to the standard specification of MIPv6.
RA	This message is sent from ARs to advertise their existence and properties. RA message is built according to the standard specification of MIPv6 with one of the reserved bits as a MI flag.

<i>Pr_Rt_Sol</i>	This message is sent from the MN to the old AR after appearance of the L2-trigger to ask for a <i>Pr_Rt_Adv</i> message.
<i>Pr_Rt_Adv</i>	This message is sent from the old AR on behalf of the new one as a response to the <i>Pr_Rt_Sol</i> message. This message is built similar to the <i>RA</i> message. However, the information included in this message is related to the new AR.
<i>BU</i>	<p>This message is used in the following cases:</p> <ol style="list-style-type: none"> 1. During the initial registration: <i>BU</i> is sent from the MN to the HA and is built according to MIPv6 specification with on of the reserved bits as a <i>MI</i> flag. <ol style="list-style-type: none"> a. As mentioned in chapter 4, this message is sent from the MN to the HA via the current AR. It is possible that the current AR adds some information required to distribute the shared secrets to the current AR and the MN. This depends mainly on the used security mechanism, see [PJA00] and [PCa01]. b. If a certain mechanism is used to distribute the shared secrets, e.g. AAA, the current AR does not modify this message. 2. During the initial registration: <i>BU</i> is sent from the MN to the CN and is built according to standard specification of MIPv6. 3. During a normal registration using MIFAv6: <i>BU</i> is sent from the MN to the new AR. The message is authenticated using the IPSec-SA established between the MN and the new AR. The message should contain: <ol style="list-style-type: none"> a. A MIFA authentication extension containing $Auth_1$. b. The <i>MI</i> flag set to 1. 4. During a normal registration employing MIFAv6: <i>BU</i> is sent from the new AR to the HA to update the mobility binding. The message is authenticated using the IPSec-SA established between the HA and the new AR. This message should contain: <ol style="list-style-type: none"> a. The shared secrets that should be used to establish an IPSec-SA between the HA and current L3-FHR members. The IPSec-SA will be used to secure the control messages that will be exchanged between the HA and the next new AR. If a certain mechanism is used to distribute the shared secret, e.g. AAA, no information related to the shared secret should be included in the <i>BU</i> message. b. The new random variables generated by the new AR

<i>BA</i>	<p>This message is used in the following cases:</p> <ol style="list-style-type: none"> 1. During the initial registration: <i>BA</i> is sent from the HA to the MN as a response to the <i>BU</i> message. This message may contain information, from which the current AR and the MN can derive the shared secret between the MN and the current AR on one side and between the current AR and the HA on the other side. If a security infrastructure exists, it takes care of distributing the shared secrets and, thus, no security information is required in the <i>BA</i> message. 2. During the initial registration: <i>BA</i> is sent from the CN to the MN as a response to the <i>BU</i> message. It follows the standard specification of MIPv6. 3. During a normal registration using MIFAv6: <i>BA</i> is sent from the new AR to the MN as a response to the <i>BU</i> message. It should contain: <ol style="list-style-type: none"> a. A MIFA authentication extension containing $Auth_2$. b. The <i>MI</i> flag set to 1. c. Two new random variables and the shared secret required for the next registration with the next new AR. 4. During the registration using MIFAv6 in predictive mode: <i>BA</i> is sent from the new AR to the old one, which uses this message as an indicator for the success/failure of the registration.
<i>M_P_Not</i>	<p>This message is used in the following two cases:</p> <ol style="list-style-type: none"> 1. During the initial authentication exchange procedure: The <i>M_P_Not</i> message is transmitted from the current AR to the HA. It is used to ask for the information required to construct the MN-specific data. 2. During the information distribution procedure: <i>M_P_Not</i> messages are distributed from the current AR to the members of the current L3-FHR. These messages contain the MN-specific data.
<i>M_P_Ack</i>	<p>This message is used in the following two cases:</p> <ol style="list-style-type: none"> 1. During the initial authentication exchange procedure: The <i>M_P_Ack</i> message is sent from the HA to the current AR. It contains mainly the information required to authenticate the MN during the next registration. More concrete, this message should contain: <ol style="list-style-type: none"> a. A HA features extension that determines what are the services and the features the HA is able to offer to the MN. b. MIFA authentication values ($Auth_1$ and $Auth_2$) 2. During the information distribution procedure: The <i>M_P_Ack</i> message is used to acknowledge the receipt of the <i>M_P_Not</i> message. It is sent from the members of the current L3-FHR to the current AR.
<i>Hn_Not</i>	<p>This message is sent from the new AR to the old one to notify it of the new CoA. This message is used in reactive mode only.</p>
<i>Hn_Ack</i>	<p>This message is sent from the old AR to the new one as a response to the <i>Hn_Not</i> message. This message is used in reactive mode only.</p>
<i>Int_Ack</i>	<p>This message is sent from the old AR to the MN to indicate that the <i>BU</i> message sent from the MN to the new AR via the old one has not been dropped on the wireless link. This message contains the handoff possibilities extension that indicates if a handoff to the new AR employing MIFAv6 is possible or not. This message is used in predictive mode only.</p>

Tab A.2: Control messages of MIFAv6

B. Establishment of L3-FHRs

As presented in section 4.1, the establishment of L3-FHRs aims at accelerating the layer 3 handoff procedure. A L3-FHR of a certain FA is the set of adjacent FAs, to which movements from this FA are possible. Determining the members of a L3-FHR depends on many factors, e.g. FAs geographical location, users' movement patterns, applied traffic, etc. A L3-FHR does not necessitate containing all adjacent FAs, e.g. in case of obstacles that prevent movements between these FAs. Therefore, how to build these L3-FHRs is a critical issue in MIFAv4. In the following, three algorithms for the establishment of these L3-FHRs will be provided along with a discussion about how L3-FHRs can be established in heterogenous networks. Notice that the methods presented in this appendix are discussed for MIFAv4. They can be, however, used for MIFAv6 without any modifications.

B.1. L3-FHRs Optimized Dynamic Selection Algorithm

A FHR selection algorithm to accelerate the layer 2 handoff between neighbor APs is presented in [PCh02] and [PCh02a]. FHR is a set of neighbor APs, to which the MN may move. The proposed algorithm optimizes the FHR for each MN depending on its traffic and mobility characteristics. This method is extended to dynamically build and alter the L3-FHRs needed by MIFAv4, so that each MN obtains its optimized L3-FHR based on its mobility and traffic characteristics, e.g. the service class, movement pattern, etc. This will help in accelerating the layer 3 handoff even if the MN moves at high speeds.

The user's service class and movement patterns are important factors. Some users may be satisfied with their services in spite of a session disconnection during handoffs. Other users may want a seamless connectivity without data loss. Some users may move from one FA to another one more frequent than others. Furthermore, some users move around faster than others. It is obvious that the number of members in the L3-FHRs associated with the above presented user types are different, i.e. to support those users, whose movement frequency and their velocity are high, a higher number of neighboring FAs should be included in the L3-FHR.

The L3-FHR optimized dynamic selection algorithm consists of two phases. The first phase constructs a weighted bi-directional graph that includes all FAs. The second phase builds the L3-FHRs according to the weighted graph and according to the mobility and traffic characteristics of each MN.

Movements between two FAs (FA_i and FA_j) in the weighted bi-directional graph are weighted through a weight $w_{(i,j)}$, which can be calculated from the equation below.

$$w_{(i,j)} = \begin{cases} 0 & i = j \\ A & i \neq j \quad FA_i \quad \text{and} \quad FA_j \quad \text{are} \quad \text{neighbors} \\ \infty & i \neq j \quad FA_i \quad \text{and} \quad FA_j \quad \text{are} \quad \text{not} \quad \text{neighbors} \end{cases} \quad (43)$$

Notice that traffic patterns are asymmetric in real scenarios. Thus, values of $w_{(i,j)}$ and $w_{(j,i)}$ may not be the same. The weight value A can be calculated from the following equation.

$$A = \frac{1}{H_{(i,j)}} \quad (44)$$

Where $H_{(i,j)}$ is the handoff ratio between FA_i and FA_j and given by equation (45), where $Nh_{(i,j)}$ is the number of handoff events between FA_i and FA_j . $Rh_{(i,j)}$ is the time duration, within these $Nh_{(i,j)}$ handoffs have been executed.

$$H_{(i,j)} = \frac{Nh_{(i,j)}}{Rh_{(i,j)}} \quad (45)$$

Based on the equations presented above, we notice that the term $w_{(i,j)}$ will be smaller, if the number of handoff events increases. The first phase of this algorithm results in a z by z weighted matrix (WM), where z is the number of FAs in a certain domain. Figure B.1 shows the weighted bi-directional graph and the WM matrix for an example domain.

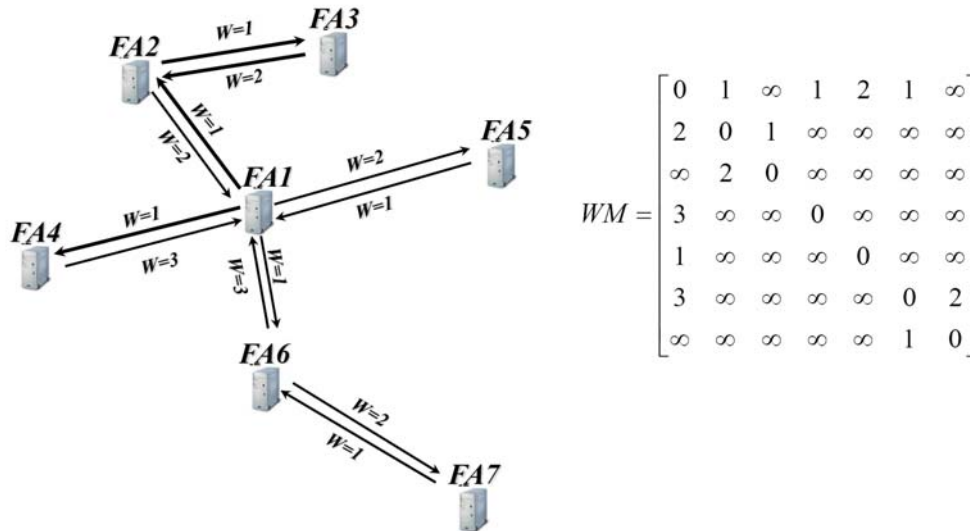


Fig B.1: Weighted graph and weighted matrix for an example domain

To select the FAs, whose must be in a L3-FHR for a certain MN, the user's service class level is denoted as Cl . The high value of Cl the better service the user wants to use. In other words, a better performance and more seamless handoffs should be guaranteed. This implies that more FAs should be associated with this L3-FHR.

The result of this algorithm is a vector G_{FA-i} . The number of elements in this vector is z . If FA_j is included in the L3-FHR, the element presenting this FA in this vector is set to true or one and false or zero in the other case. Let us consider the example presented in figure B.1 and assume that FA_1 is the current FA, where the MN locates. The selection of the FAs belonging to the corresponding L3-FHR depends on a simple rule that says: FA_j is a member of the L3-FHR of FA_i if the sum of the weights from FA_i to FA_j is smaller than a certain

value wB and the handoff scope between FA_i and FA_j is smaller than a threshold (h). wB is the weight bound and is related to the required service class the MN wants to guarantee. The handoff scope (h) defines if the handoff occurs only between two adjacent FAs or between two non-adjacent FAs. This is necessary to take the MNs moving at very high speeds inside access networks with small cells into account. Depending on this analysis, G_{FA-1} will be [0101010], [0111110] and [0111111] for $h = 2$ and $wB = 1, 2$ and 3 , respectively.

The main advantage of this algorithm is the optimization of each L3-FHR for each MN according to the MNs parameters described above. Due to the dynamic changing of users' characteristics, A L3-FHR associated with a certain MN may change from time to time. Establishing the L3-FHRs using such a dynamic method should improve the performance and deliver better services to MNs. However, instead of one L3-FHR that includes all FAs in the vicinity, where movements are possible, each FA should have many L3-FHRs according to the users it serves and should change them dynamically. This will complicate the establishment of such L3-FHRs.

B.2. L3-FHRs Selection Using Neighbor Graph

Neighbor Graph [MSA04a] is a data structure that determines the candidates FAs a MN may move to from a certain FA. Notice that two FAs (FA_i and FA_j) have a relationship with each other if it is possible for a MN to perform a layer 3 handoff through a path of motion between the physical locations of the APs belonging to FA_i to the APs controlled by FA_j . This relationship corresponds in most cases to the physical distance (vicinity) between the FAs. Their may be, however, FAs that are geographically neighbors, but they have not any relationship between each other, e.g. in case of obstacles that prevent movements between these FAs.

A neighbor graph of a FA defines a undirected graph (ψ, ζ) , where Ψ is the set of all FAs including the current one. ζ is the set of edges between these FAs. It is defined that there is an edge between FA_i and FA_j if they have a relationship. Depending of this, FA_k will be a neighbor of FA_i if $FA_k \in \psi$ and $(FA_i, FA_k) \in \zeta$. Let us define the association pattern $\Gamma(c)$ for a client c as $\{(FA_i, t_i), (FA_j, t_j), \dots\}$. This means that the client c will associate with FA_i at time t_i and, after that, will associate with FA_j at time t_j and so on. The client maintains continuous logical network connectivity from time t_i to t_j . It is obvious that the association pattern can be captured by the neighbor graph.

Figure B.2 presents an example for a physical topology of a wireless network. There are five FAs with the placement shown in the figure below. The dashed lines show potential paths of motion for a client between the FAs. If the client is associated to FA_1 , the new FA will be either FA_2 , FA_3 or FA_5 . There is no edge or relationship between FA_1 and FA_4 . This is because no paths of motion are allowed by the physical topology. In other words, a client can not directly move from FA_1 to FA_4 without at least temporarily going through FA_2 , FA_3 or FA_5 . The neighbor graph presented in figure B.3 captures this locality information in the form of a data structure. Thus, the graph is sufficient to obtain this relationship from the given network topology.

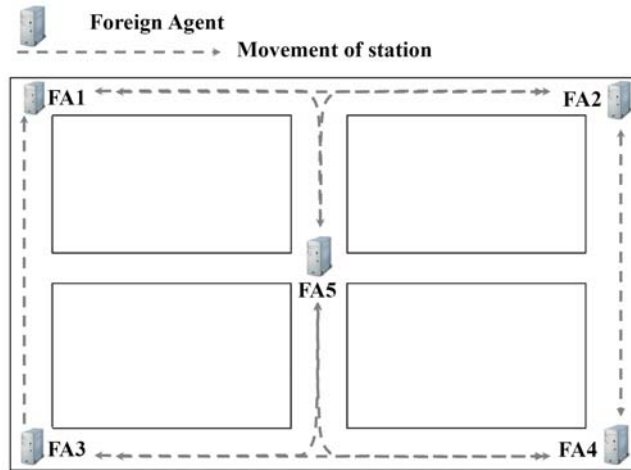


Fig B.2: Physical locations of the FAs in an example domain

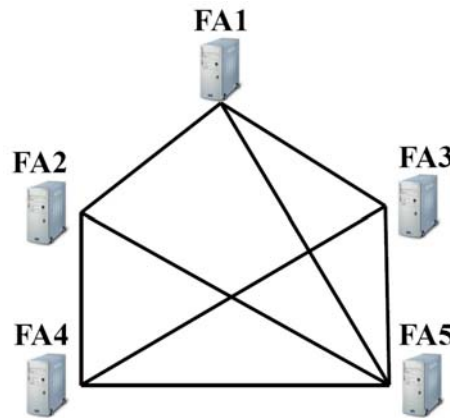


Fig B.3: Neighbor graph of the example presented in figure B.2

B.3. L3-FHRs Dynamic Selection Algorithm

This algorithm is the default one used to build L3-FHRs when employing MIFAv4. The network is responsible for determining the L3-FHRs. This can be simply achieved by observing MNs' movements inside the network. When a MN moves to a new FA, it includes the address of its old FA in the *Reg_Rqst* message. If the new FA is not a member of the L3-FHR of the old FA, it has to join this L3-FHR, see chapter 4. After a certain time, the L3-FHRs will be constant and all FAs the MN may move to from a certain FA will be covered.

A main advantage of this method is that there is no need to take care of any changes in the network topology. The network adapts its L3-FHRs dynamically and after a certain time all FAs in the network will be covered again. After establishing the L3-FHRs, they will be used for all MNs regardless of their movement patterns.

B.4. Establishment of L3-FHRs in Heterogeneous Networks

While the above written sections discuss the establishment of L3-FHRs in homogenous networks and even in the same administrative domain, it is more interesting to discuss how these L3-FHRs can be established in heterogeneous networks and between FAs of different administrative domains. Figure B.4 shows an example scenario where MNs move among two different IP-based domains as well as a UMTS access network.

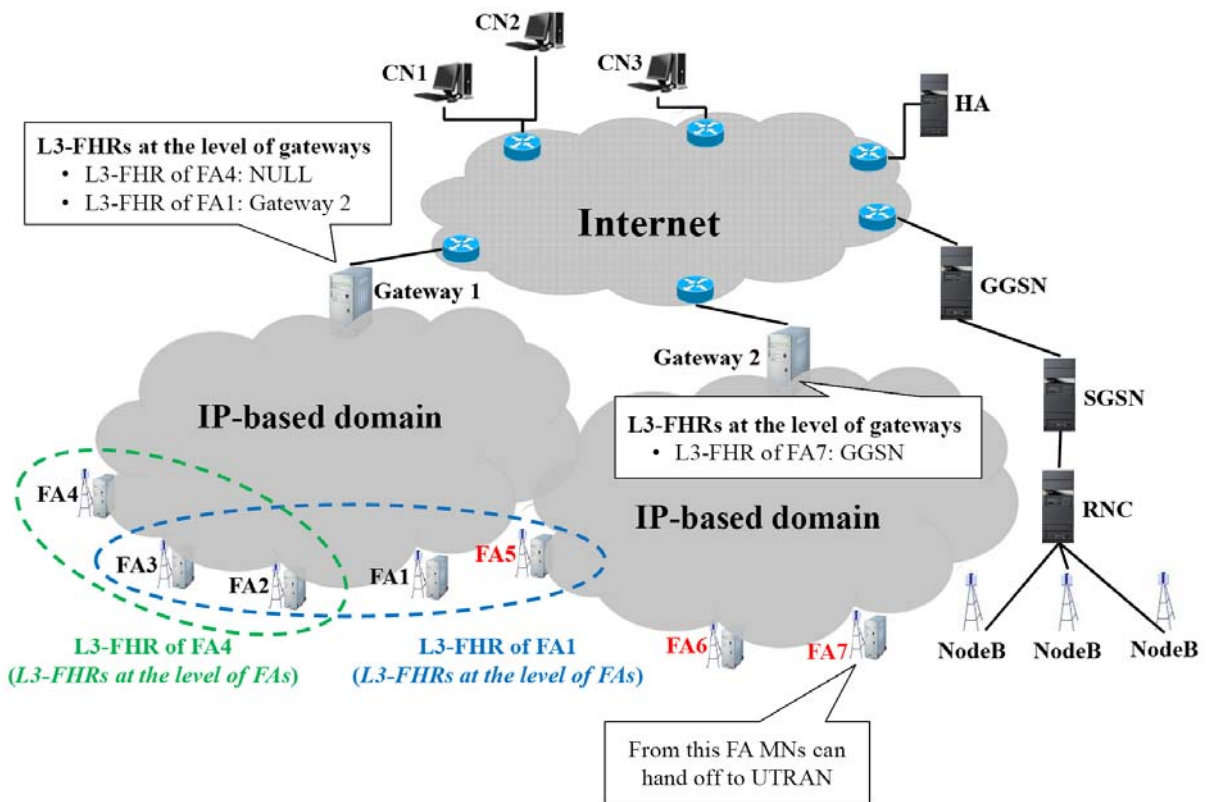


Fig B.4: An example scenario for the establishment of L3-FHRs in heterogeneous networks

Let us first consider the two different IP-based domains, where each is controlled by a gateway. L3-FHRs can be constructed either at the level of FAs or gateways. At the level of FAs, L3-FHRs can be established using any method of the methods presented in this appendix. Figure B.4 shows an example L3-FHR containing FAs from the two domains (the L3-FHR of FA1). The main issue in this context is that there should be SAs between the FAs of different administrative domains, which can be a problem if the operators of the two domains do not accept establishing such security. Of course, this will prohibit constructing the L3-FHRs at the FAs' level. In case there is a roaming agreement between the two operators, mechanisms for generating the required SAs should be determined to enable constructing the L3-FHRs.

If the building of L3-FHRs at the level of FAs is not possible, L3-FHRs can be constructed at the level of gateways. In this case, each gateway has to maintain a L3-FHR for each FA it serves. A L3-FHR of a FA contains then the gateways controlling the new domains MNs may move into from the FA. If the L3-FHR of a FA is NULL, this means that MNs move from this FA only to FAs locating in the same domain. Let us take a look at the example provided in figure B.4. The example shows that MNs can move from FA4 only to FAs belonging to the same domain, while MNs may move from FA1 to FAs in the same domain as well as to a FA in a different domain. So, the L3-FHR of FA4 is NULL, while the L3-FHR of FA1 contains the gateway controlling the new domain. Notice that these L3-FHRs should be stored in the gateway that controls the domain. Again, SAs should be established between the gateways belonging to the same L3-FHR (using AAA architecture for example, see [DMB06]). Constructing the L3-FHRs at the level of gateways is used to consider different radio access networks MNs may move to as well since they are simply considered as domains controlled by different gateways (the gateway can be a GGSN for UMTS and GSM/GPRS access networks for example)

C. SDL Specification of MIFAv4

This appendix presents the important parts of MIFAv4 specification with SDL.

C.1. MIFA System

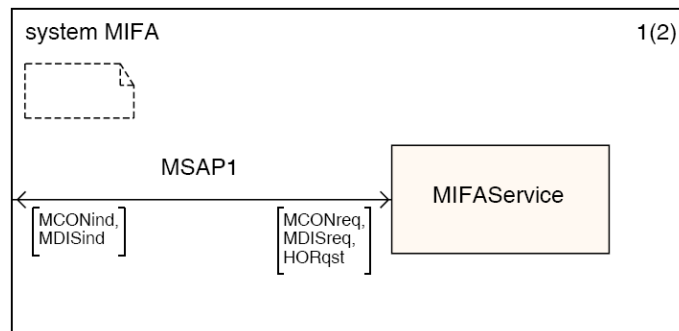


Fig C.1: MIFAv4 system

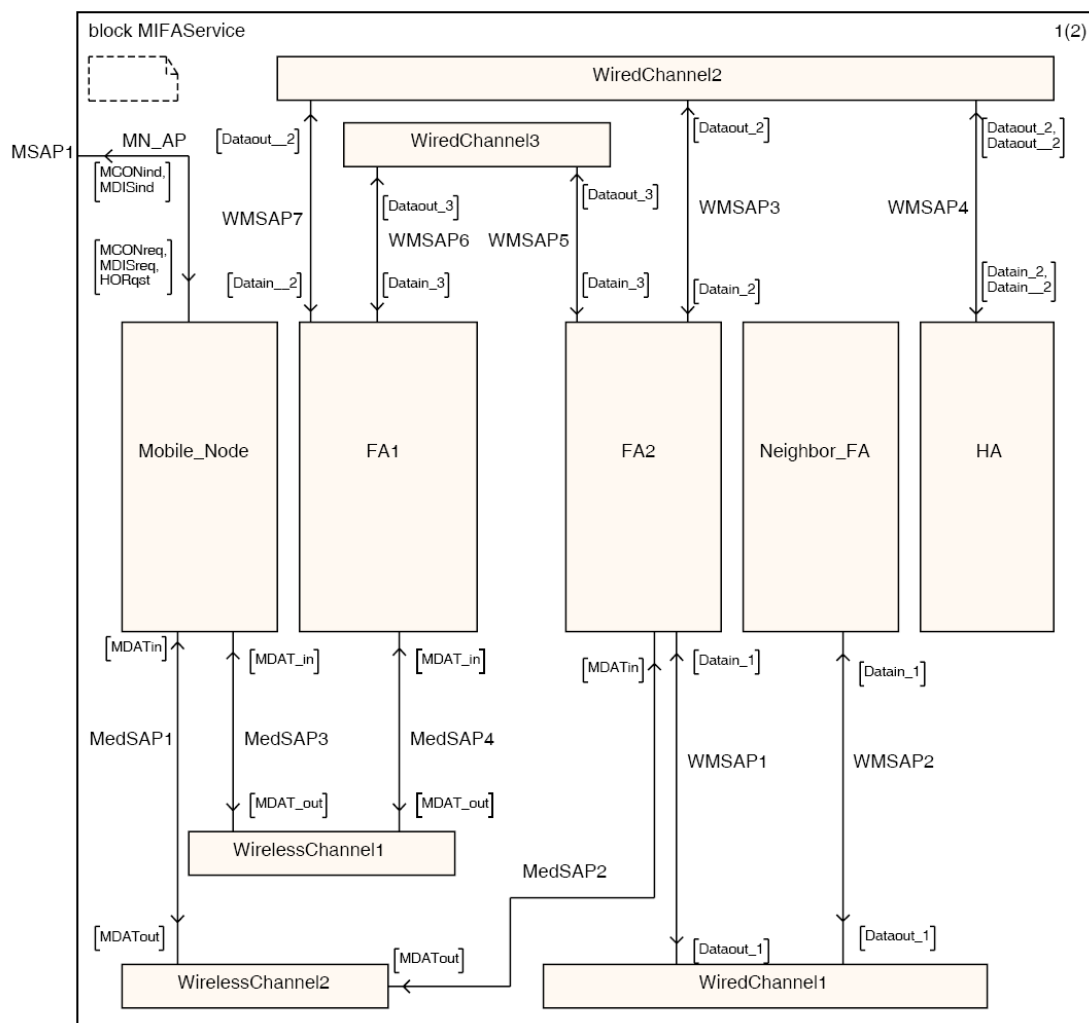


Fig C.2: Components of MIFAv4 system - MIFAService block

Signal
MCONind,MDISind,MCONreq,MDISreq,MOVreq,M_P_Not,HA_Not,M_P_Ack,HA_Ack,HORqst,
MPNNot(Ack),MPAck,AGENTadv(Advertisement) , Reg_Rqst(RegistrationRequest),Reg_Rply;

```

newtype MIFA_PDUType
  literals CR,CC,DR,DC,AAAdv,ASol,RegRqst,RegRply,
  MPNNot,MPAck, M_P_Not,M_P_Ack,L2_LU,L2_LD,AReq,
  ARes,HANot,HAAck,Int_Ack,MPNNot_i,MPAck_i,RegRqst_f,RegRply_f,
  PrRtAdv,PrRtSol;
endnewtype MIFA_PDUType;

newtype WCH_PDUType
  literals AReq,ARes,DR,DC,AAAdv,ASol,RegRqst,RegRply,Int_Ack,
  L2_Trigger,L2_LU_Ch1,L2_LU_Ch2,L2_LD,AR_CH1,AR_CH2,
  MCONreq,MDISind,PrRtSol,PrRtAdv,MCONind,RegRqst_f,L2_LU;
endnewtype;

newtype WiredCH1_PDUType
  literals MPNNot,MPAck;
endnewtype WiredCH1_PDUType;

newtype WiredCH2_PDUType
  literals M_P_Not,M_P_Ack,RegRqst,RegRply,HANot,HAAck,
  MPAck_i,RegRply_f,RegRqst_f,MPNNot_i;
endnewtype WiredCH2_PDUType;

newtype WiredCH3_PDUType
  literals RegRqst_f,RegRply_f,MPNNot_i,MPAck_i;
endnewtype WiredCH3_PDUType;

newtype WCH_MIFASDU struct
  Mtype WCH_PDUType;
  CH CR;
  ADVStruct Advertisement;
  REGStruct RegistrationRequest ;
endnewtype WCH_MIFASDU;

newtype WirelessMedium_SDUType struct
  payload WCH_MIFASDU;
endnewtype WirelessMedium_SDUType;

newtype WiredMedium1_SDUType struct
  payload WiredCH1_MIFASDU;
endnewtype WiredMedium1_SDUType;

newtype WiredMedium2_SDUType struct
  payload WiredCH2_MIFASDU;
endnewtype WiredMedium2_SDUType;

newtype WiredMedium3_SDUType struct
  payload WiredCH3_MIFASDU;
endnewtype WiredMedium3_SDUType;

newtype MessageType
  literals CR,CC,DR,DC,ASol,
  AAdv,RegRqst,RegRply;
endnewtype MessageType;

newtype WiredCH1_MIFASDU struct
  Mtype WiredCH1_PDUType;
  AckStruct Ack;
endnewtype WiredCH1_MIFASDU;

newtype WiredCH2_MIFASDU struct
  Mtype WiredCH2_PDUType;
  REGStruct RegistrationRequest;
  CH CR;
endnewtype WiredCH2_MIFASDU;

synonym Home_Address charstring='101001';
synonym HA_Address charstring='101000';

newtype Binding struct
  HAAddress charstring;HomeAddress charstring;
  FAAddress CoA;seq integer;lifetime real;MI MIFAFlag;
endnewtype;

newtype VisitorList struct
  HAAddress charstring;HomeAddress charstring;
  FAAddress CoA;seq integer;lifetime real;MI MIFAFlag;
  MIFAextension boolean;initialReg boolean;
  Acknowledged boolean;
endnewtype;

newtype Ack struct
  Notificated boolean;AckWanted boolean;
endnewtype;

newtype CoA
  literals 12012021,13013021,0000;
endnewtype;

newtype MIFAFlag
  literals 0,1;
endnewtype;

newtype MIFAMode
  literals Predictive,Reactive,MIP,NotDefined;
endnewtype;

newtype Advertisement struct
  Address CoA;Sequencenummber integer;
  MI MIFAFlag;
endnewtype Advertisement;

newtype RegestrationRequest struct
  HAAddress charstring;
  HomeAddress charstring;
  Address CoA;
  Sequencenummber integer;
  MI MIFAFlag;lifetime real;
  mode MIFAMode;
endnewtype RegestrationRequest;

newtype CR
  literals WirelessCH1,WirelessCh2;
endnewtype;

newtype WiredCH3_MIFASDU struct
  Mtype WiredCH3_PDUType;
  REGStruct RegistrationRequest;
  AckStruct Ack;
endnewtype WiredCH3_MIFASDU;

newtype FA_MIFASDU struct
  Mtype MIFA_PDUType;
  ADVStruct Advertisement;
  REGStruct RegistrationRequest ;
  AckStruct Ack;
endnewtype FA_MIFASDU;

```

Fig C.3: Defined types, signals and parameters - 1

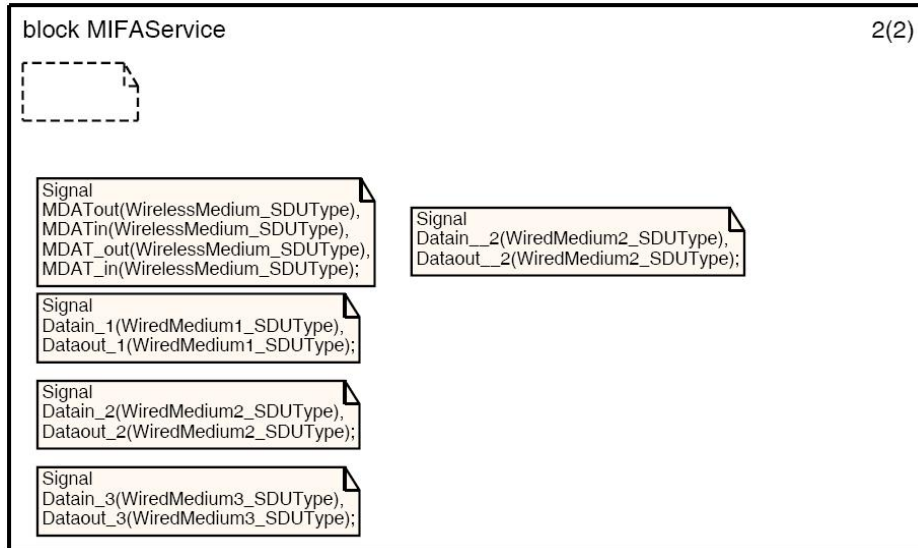


Fig C.4: Defined types, signals and parameters - 2

C.2. MN

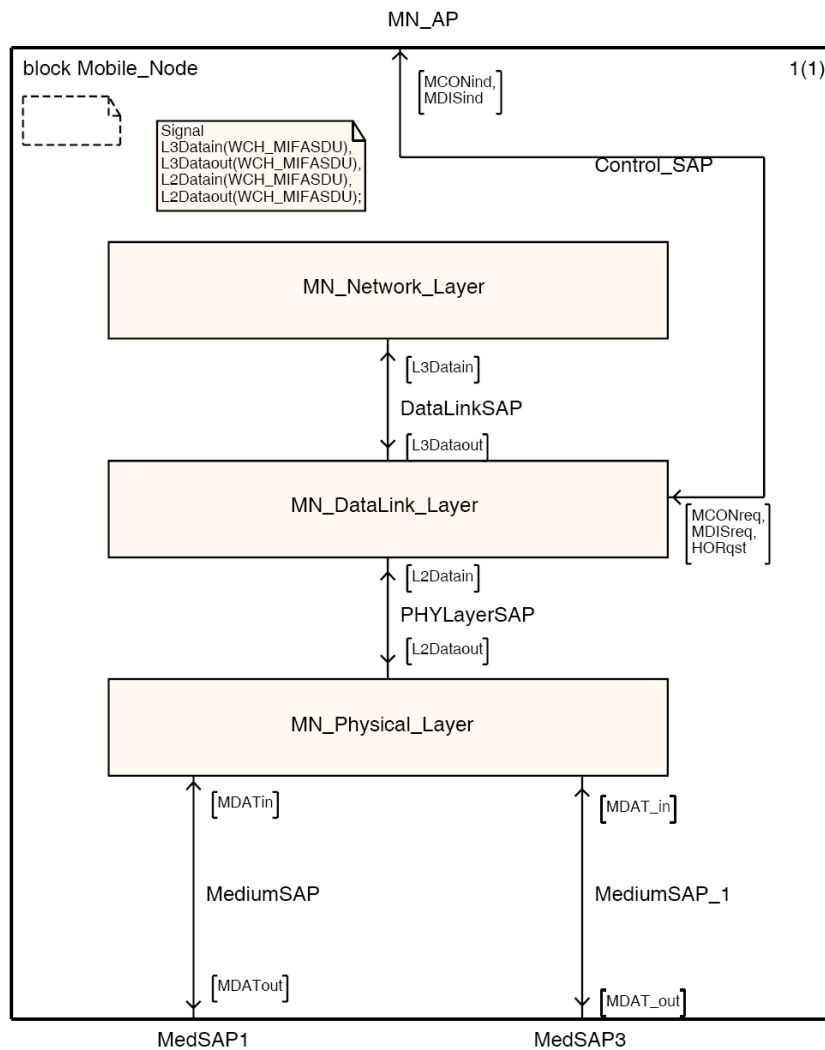


Fig C.5: Structure of the MN

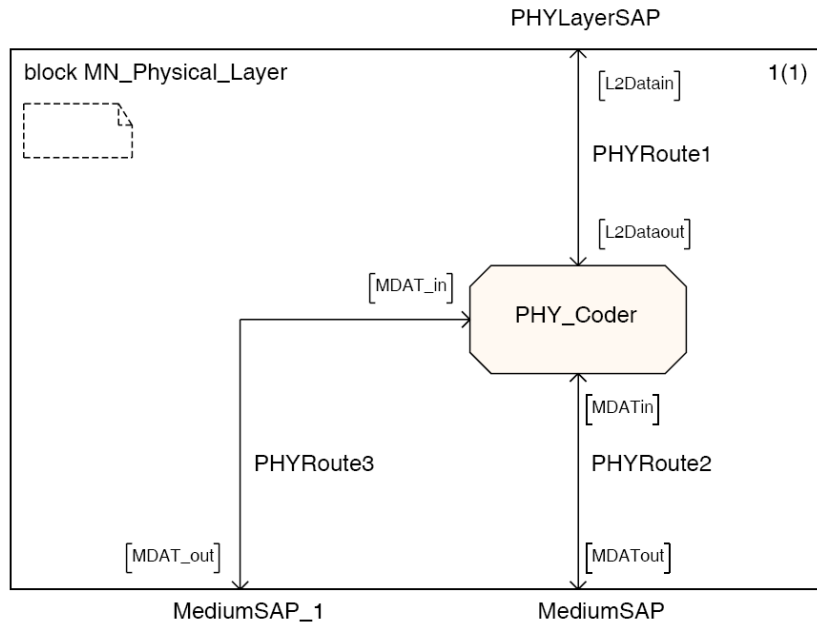


Fig C.6: Physical layer of the MN, **MN_Physical_Layer** block

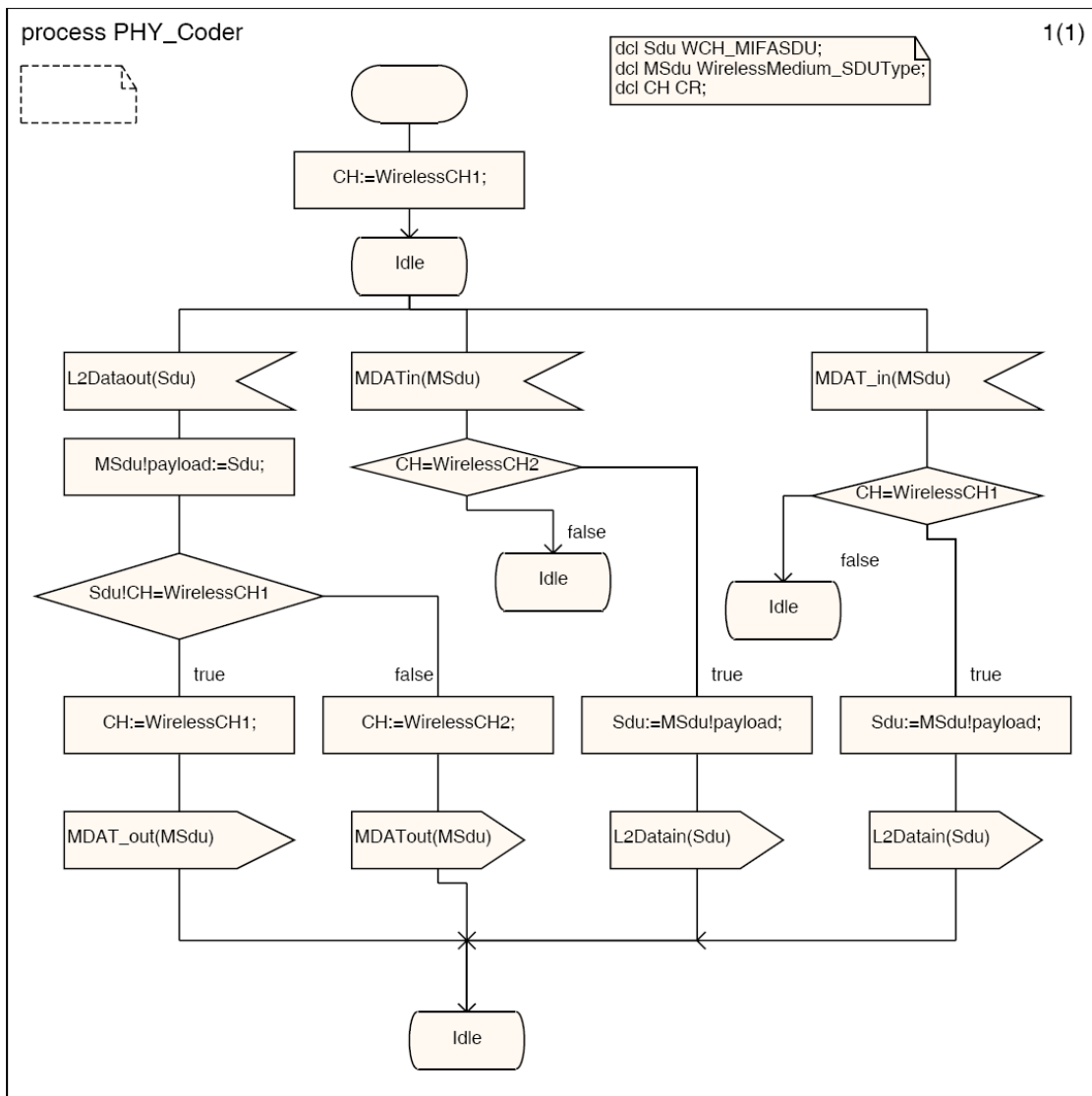


Fig C.7: **PHY_Coder** process existing in the **MN_Physical_Layer** block

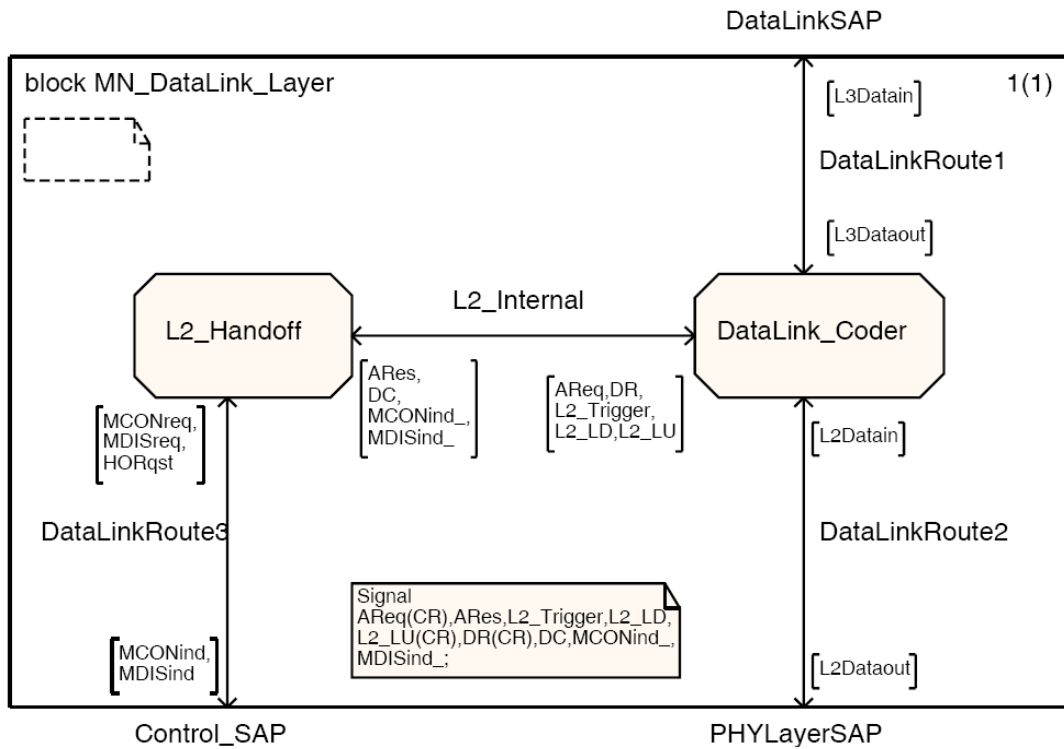


Fig C.8: Data link layer of the MN, **MN_DataLink_Layer** block

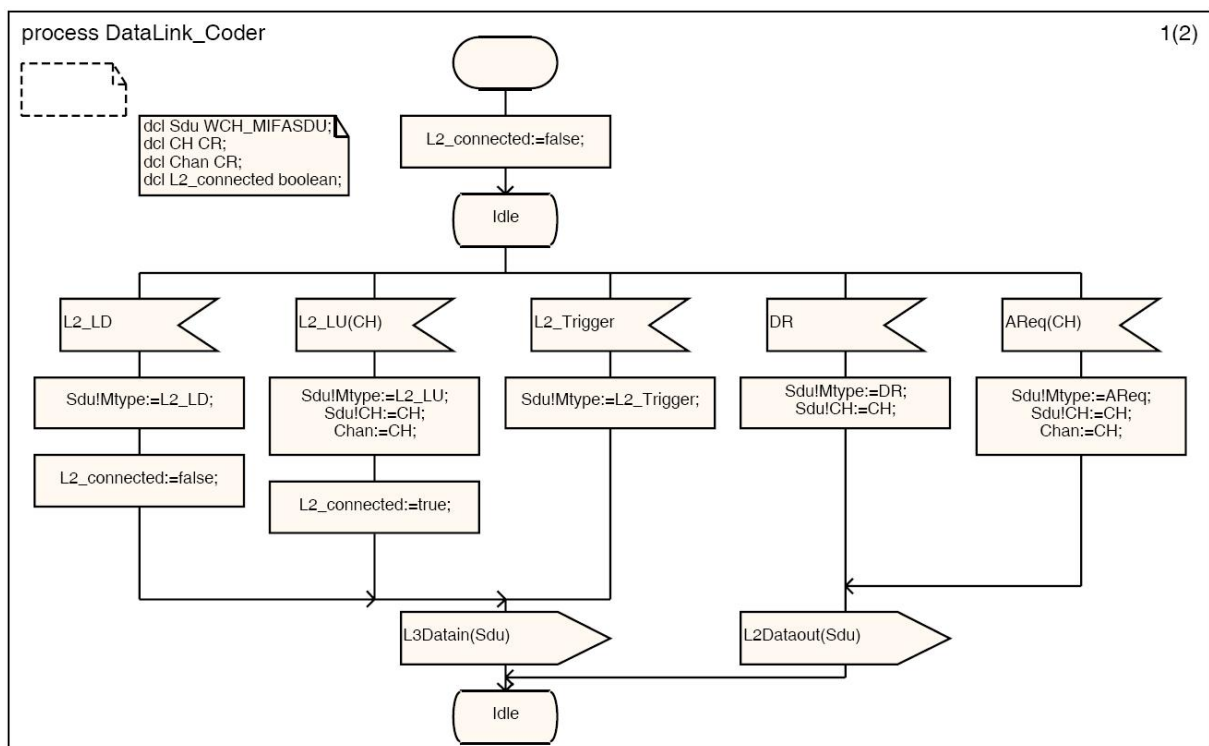


Fig C.9: **DataLink_Coder** process existing in the **MN_DataLink_Layer** block - (1)

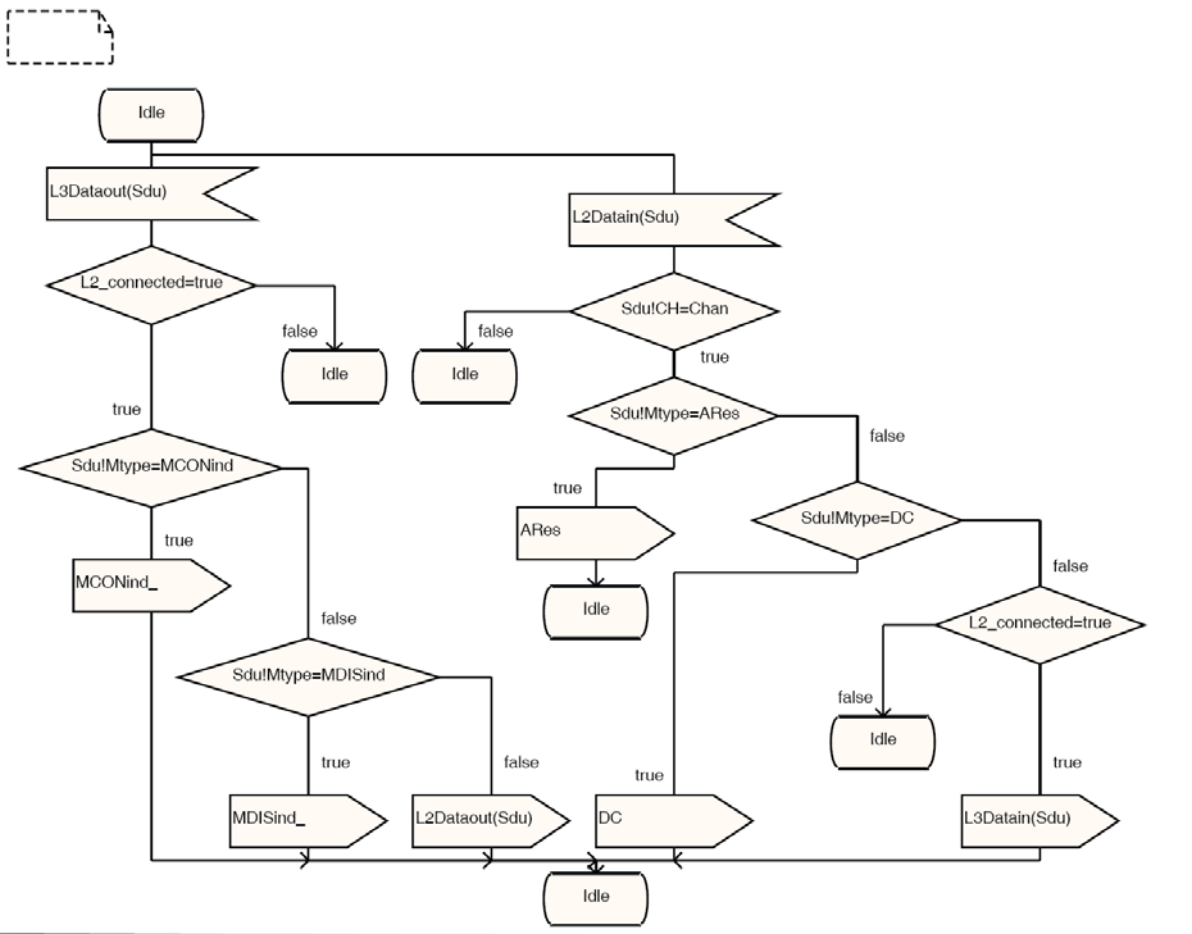


Fig C.10: DataLink_Coder process existing in the MN_DataLink_Layer block - (2)

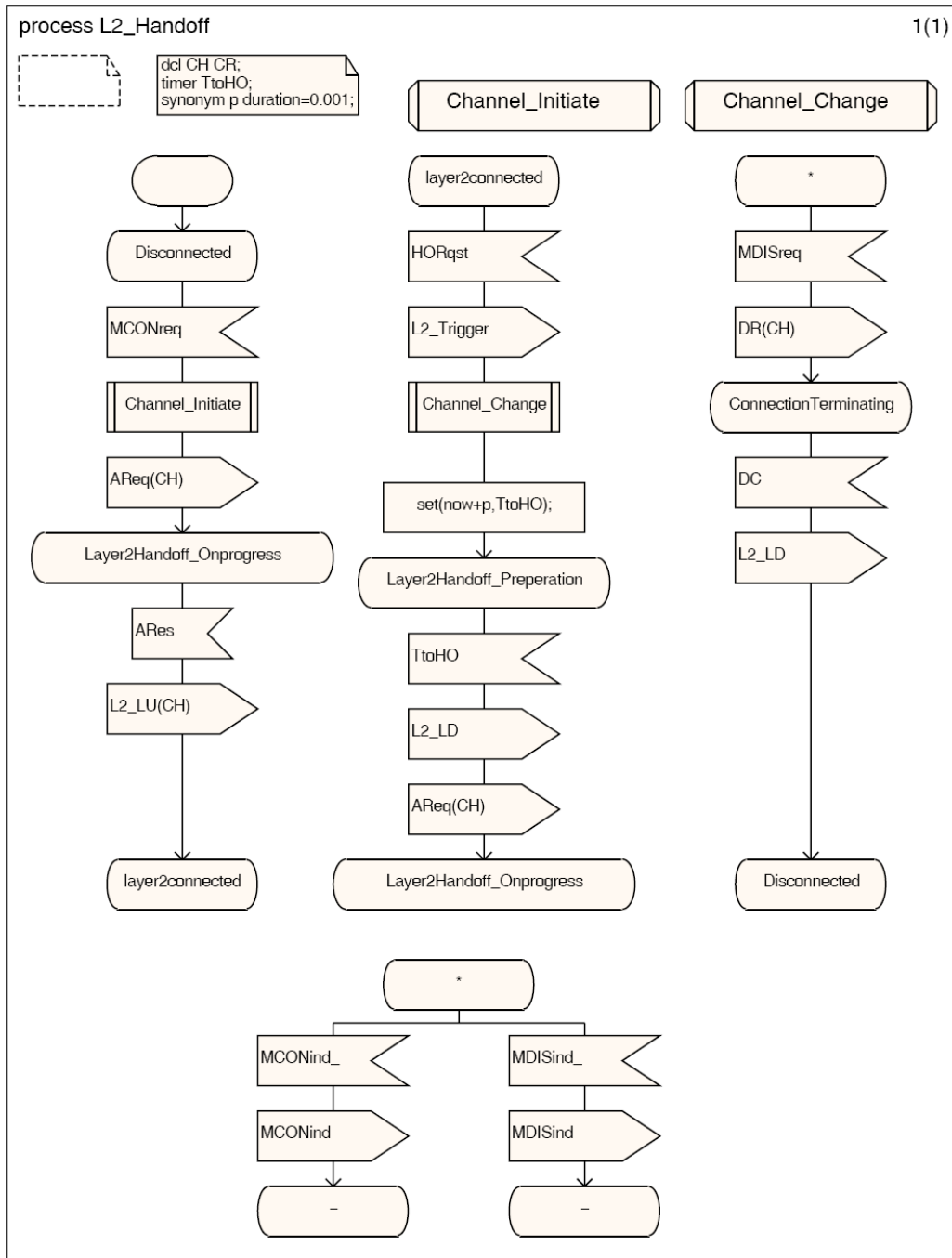


Fig C.11: L2_Handoff process existing in the MN_DataLink_Layer block

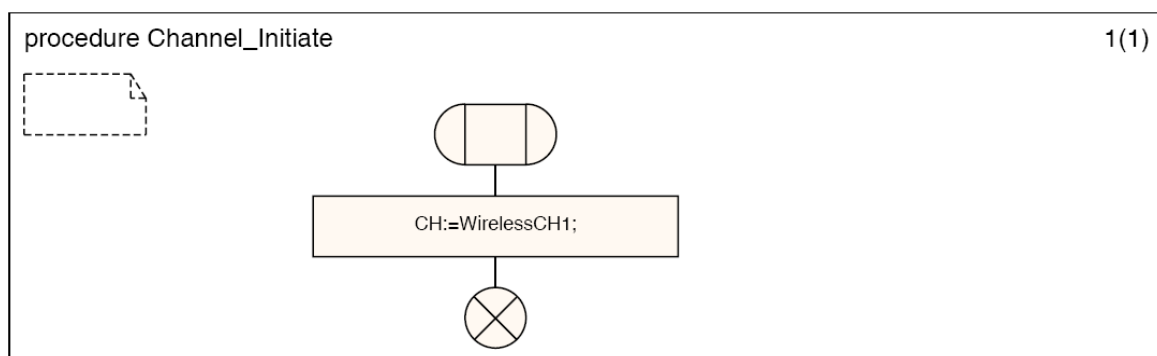


Fig C.12: Channel_Initiate procedure

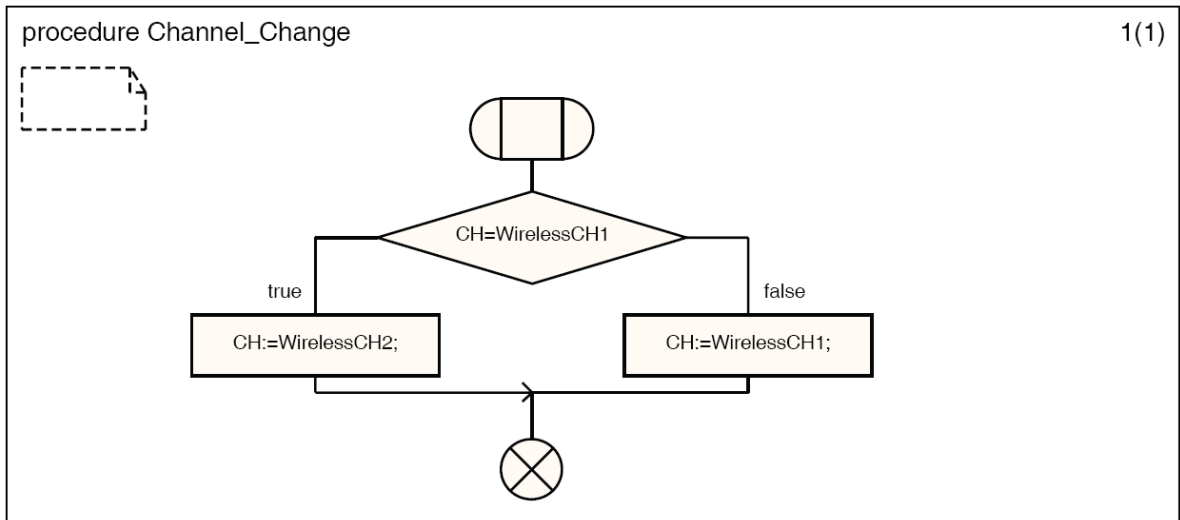


Fig C.13: **Channel_Change** procedure

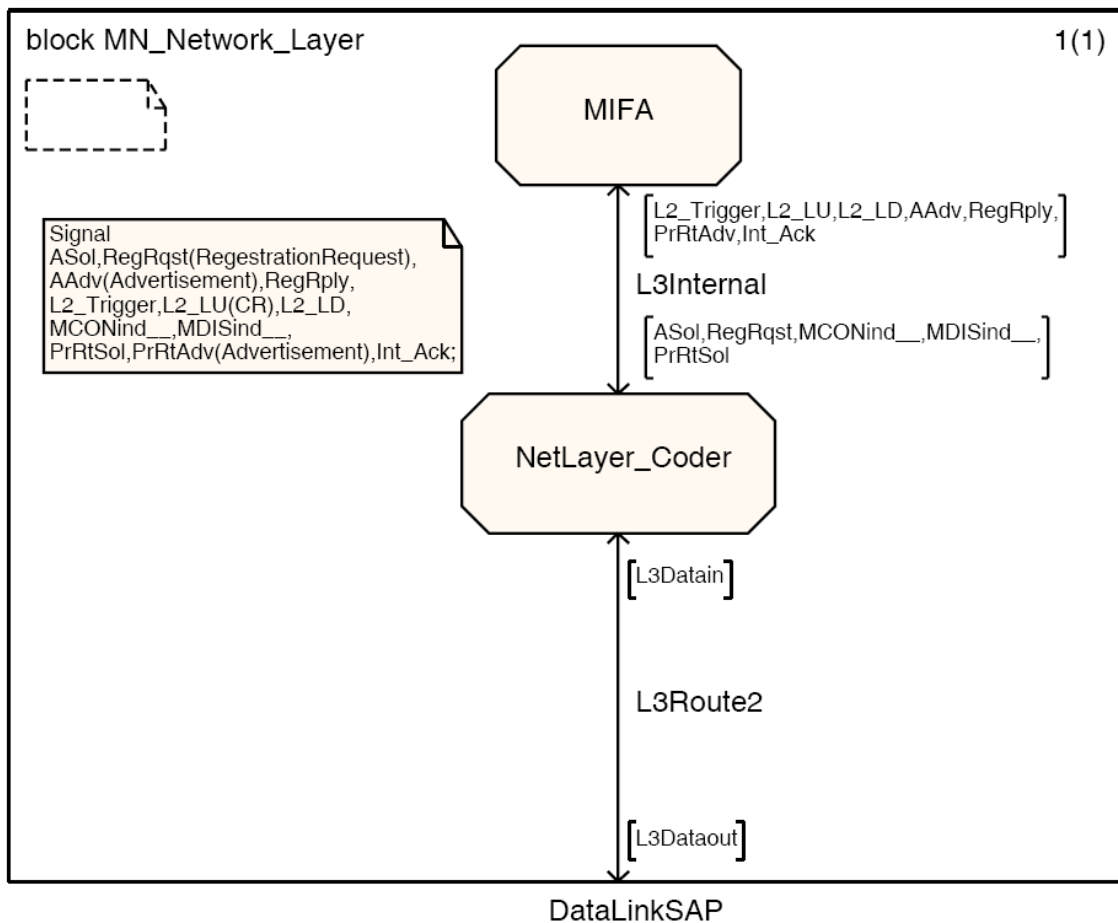


Fig C.14: Network layer of the MN, **MN_Network_Layer** block

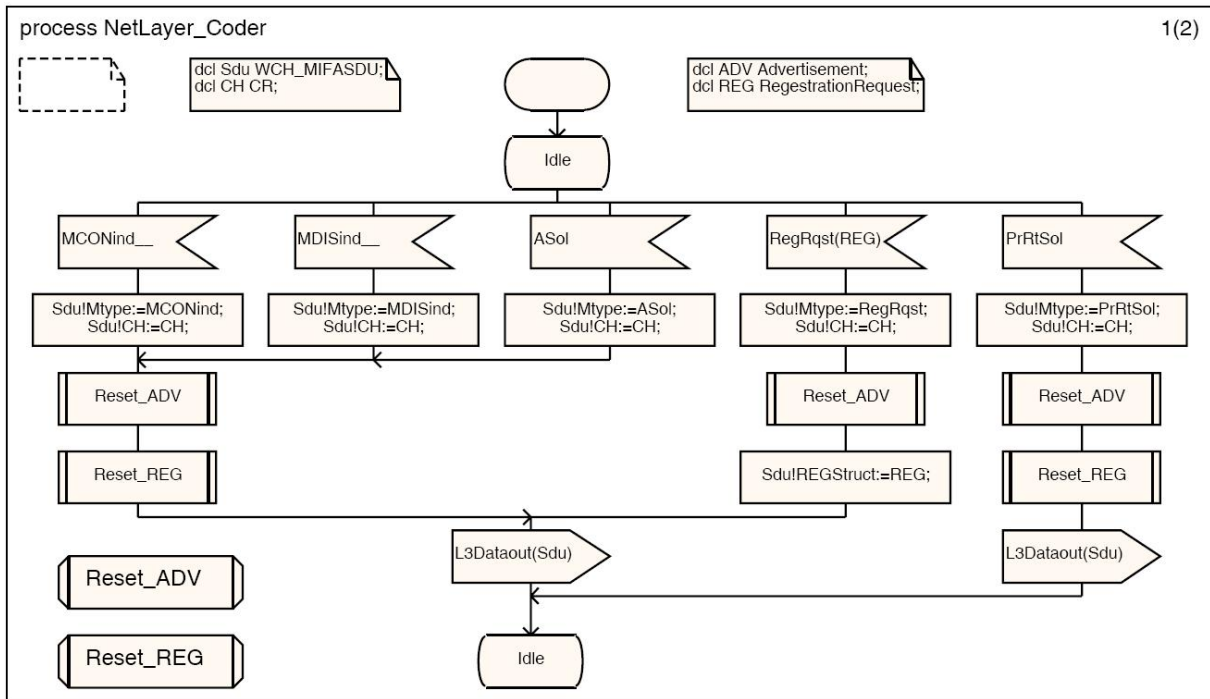


Fig C.15: **NetLayer_Coder** process existing in the **MN_Network_Layer** block - (1)

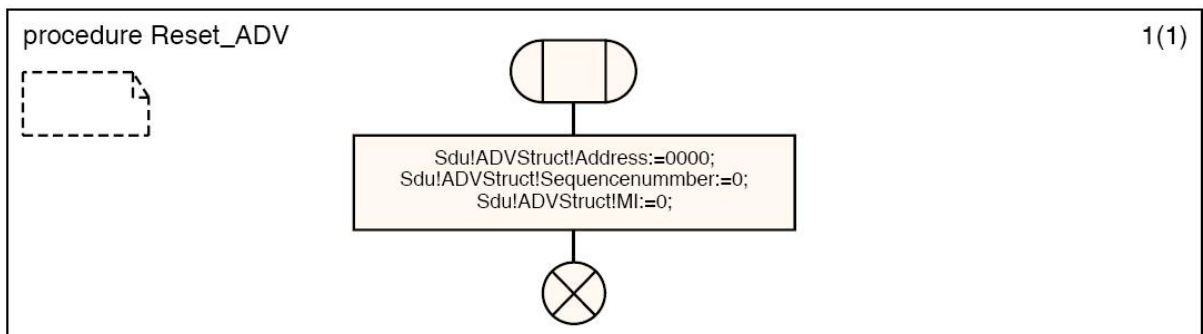


Fig C.16: **Reset_ADV** procedure

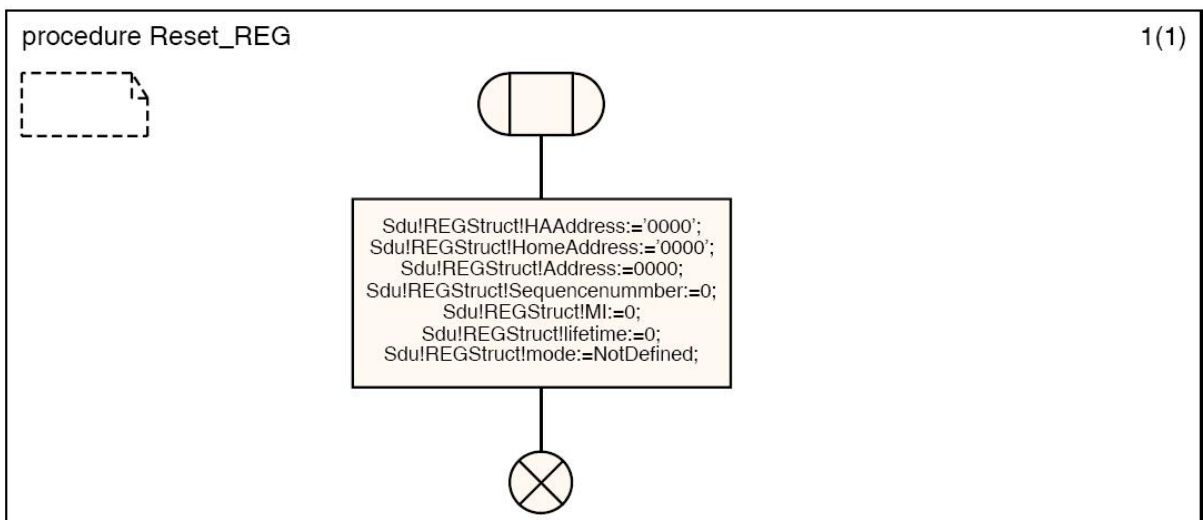


Fig C.17: **Reset_REG** procedure

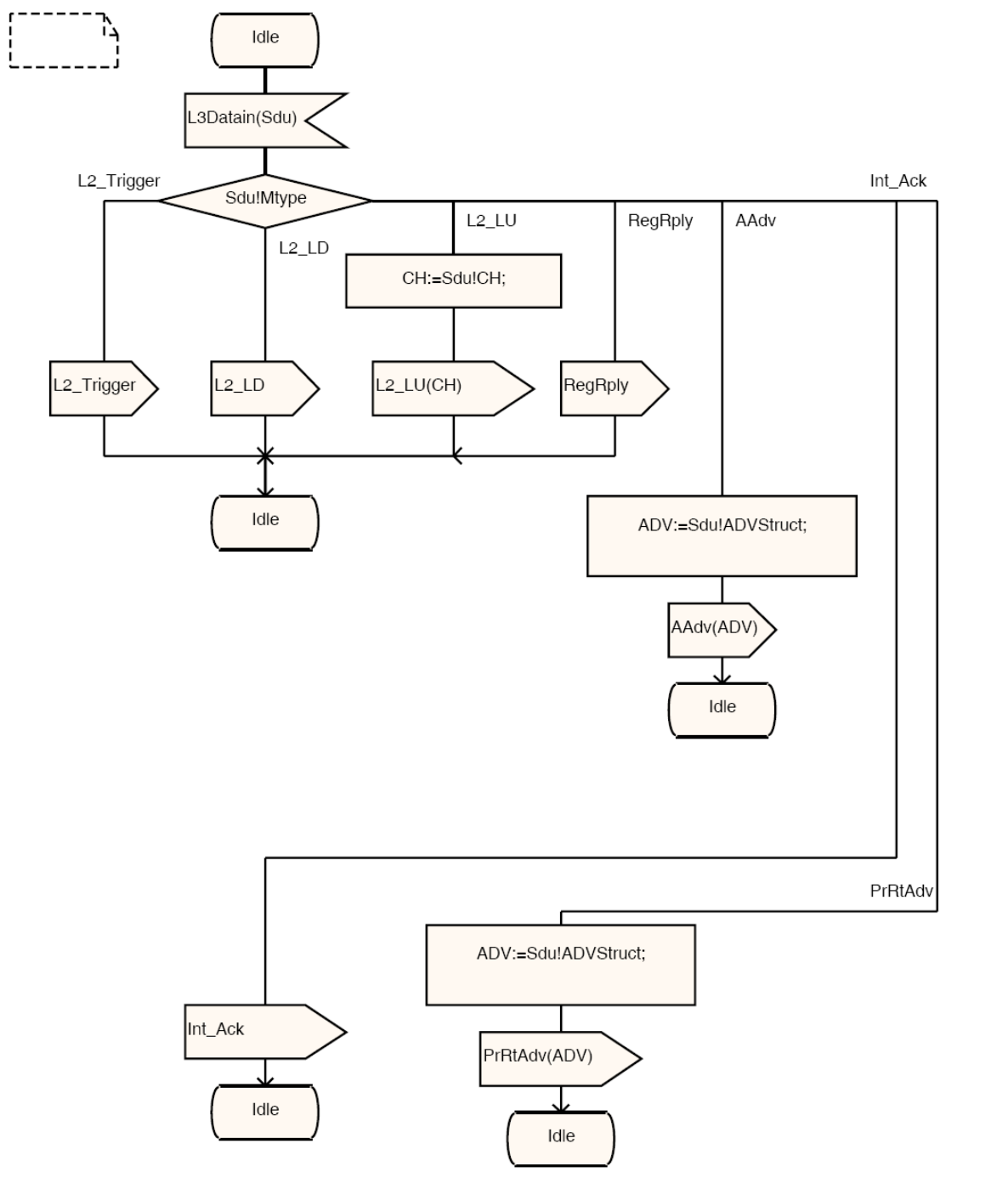


Fig C.18: NetLayer_Coder process existing in the MN_Network_Layer block - (2)

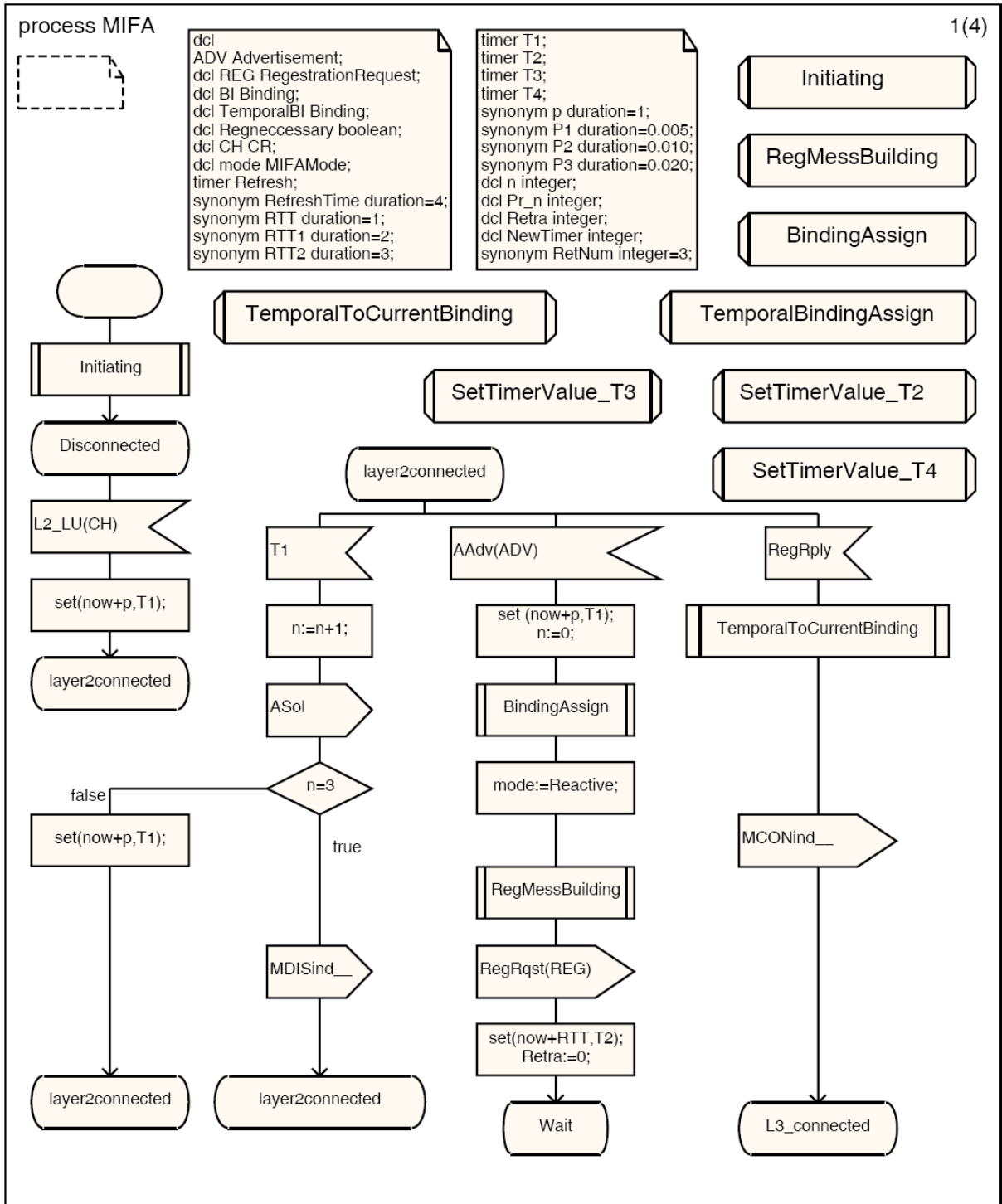


Fig C.19: MIFA process existing in the MN_Network_Layer block - (1)

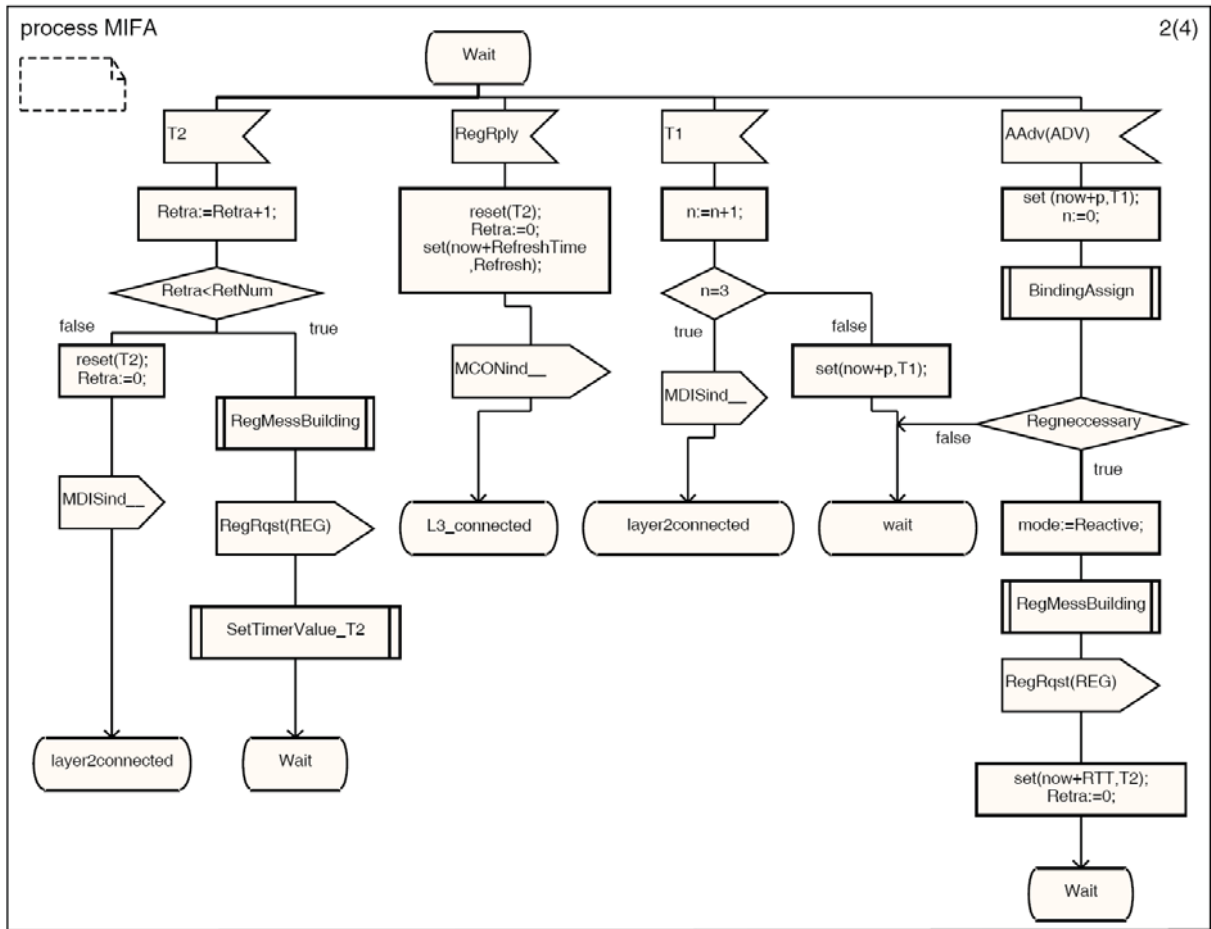


Fig C.20: **MIFA** process existing in the **MN_Network_Layer** block - (2)

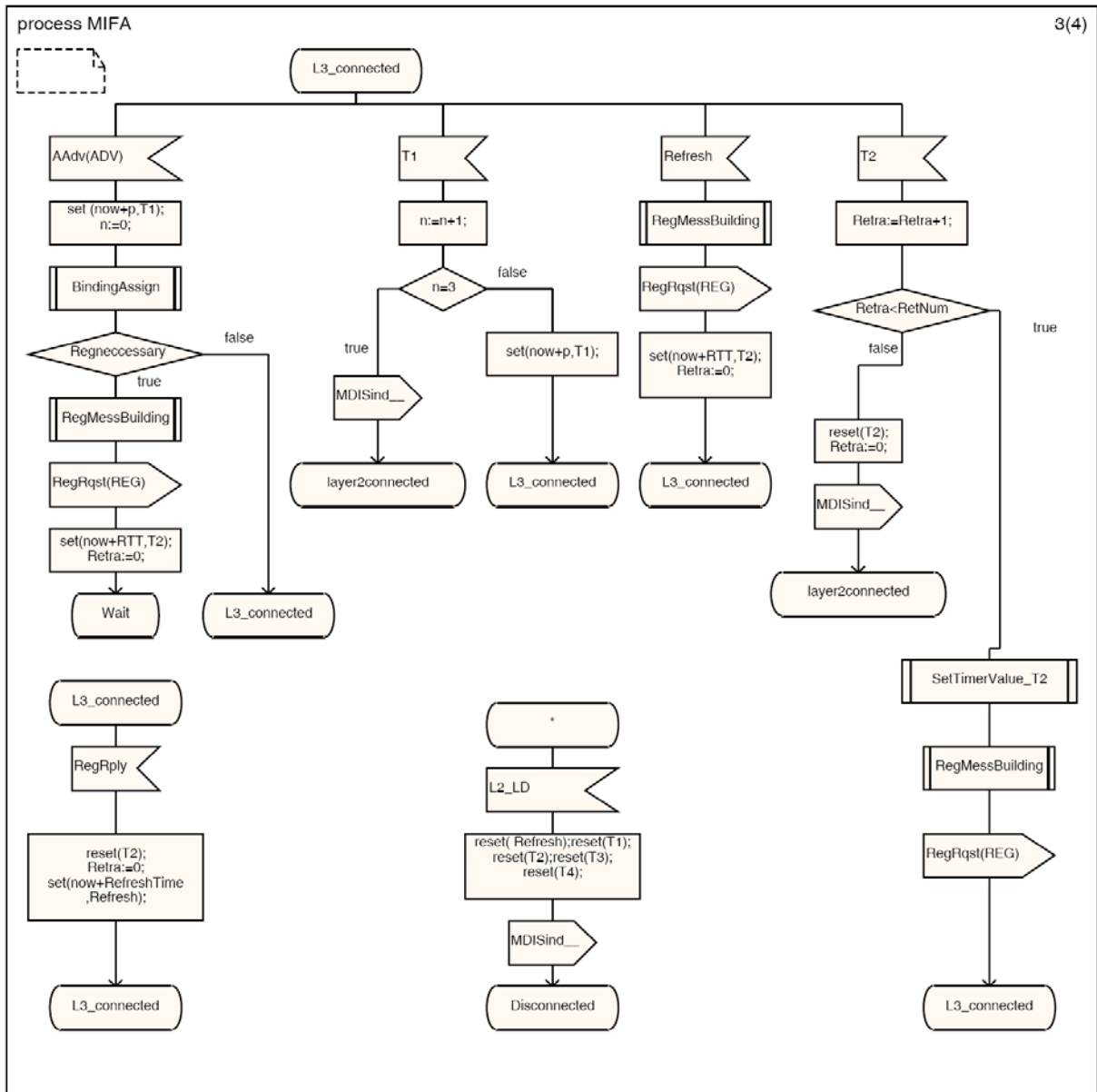


Fig C.21: MIFA process existing in the MN_Network_Layer block - (3)

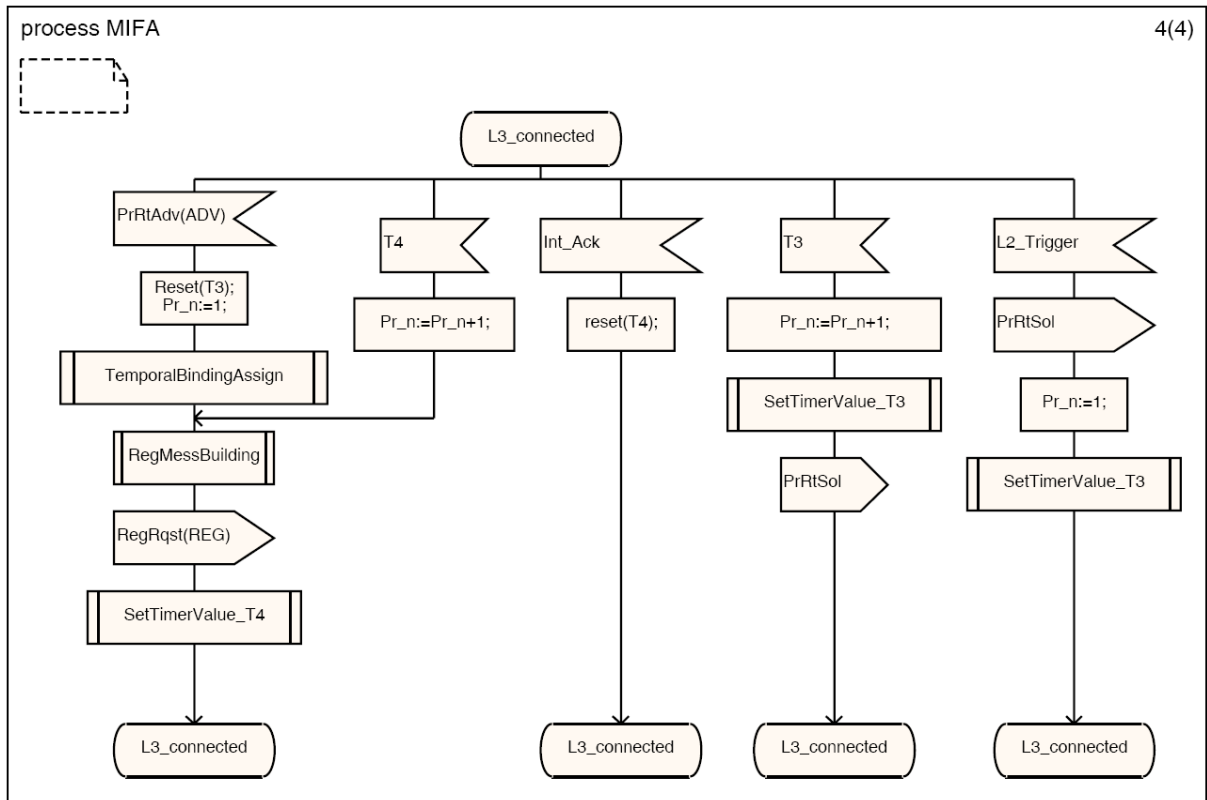


Fig C.22: MIFA process existing in the MN_Network_Layer block - (4)

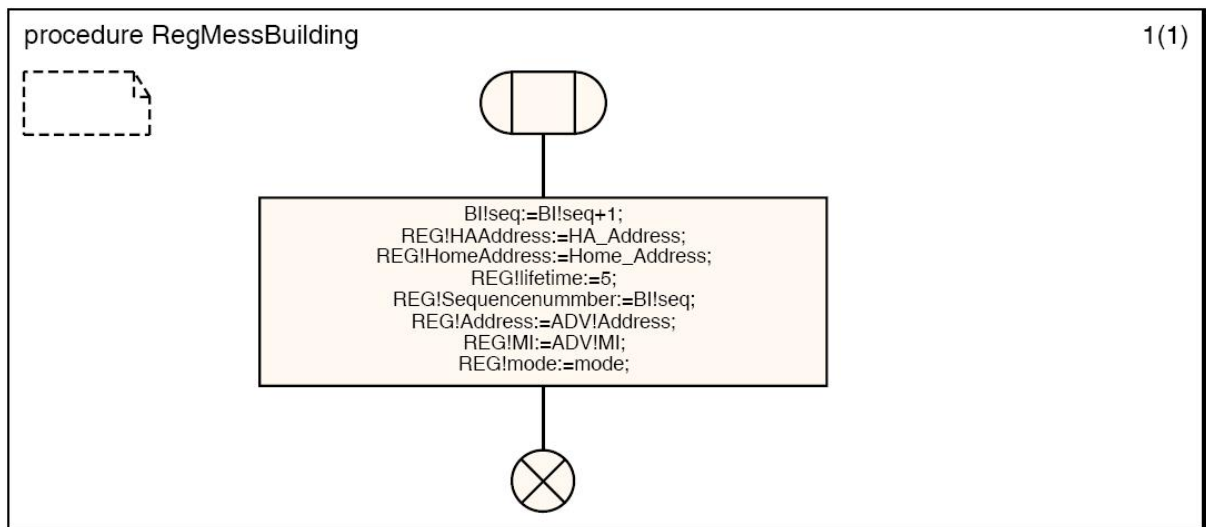


Fig C.23: RegMessBuilding procedure

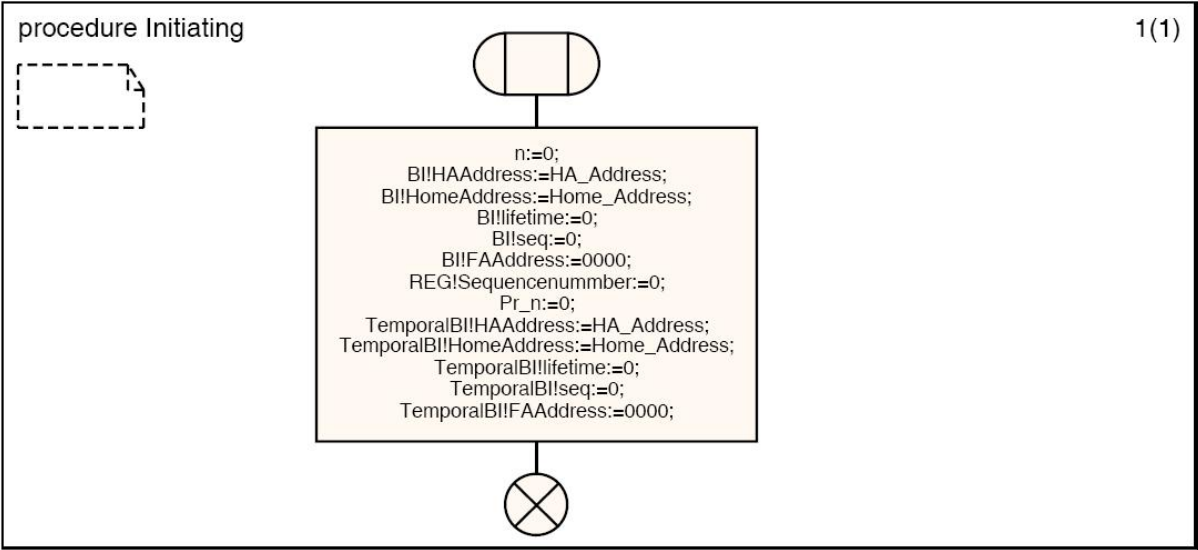


Fig C.24: **Initiating** procedure

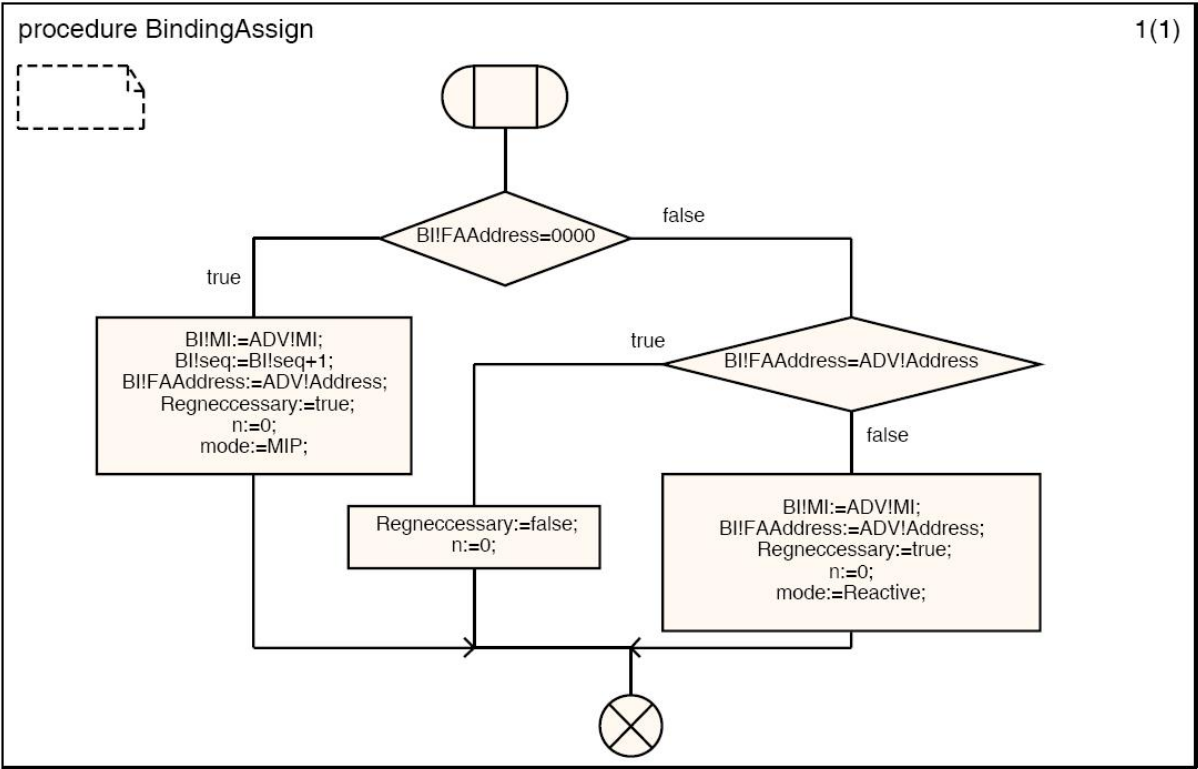


Fig C.25: **BindingAssign** procedure

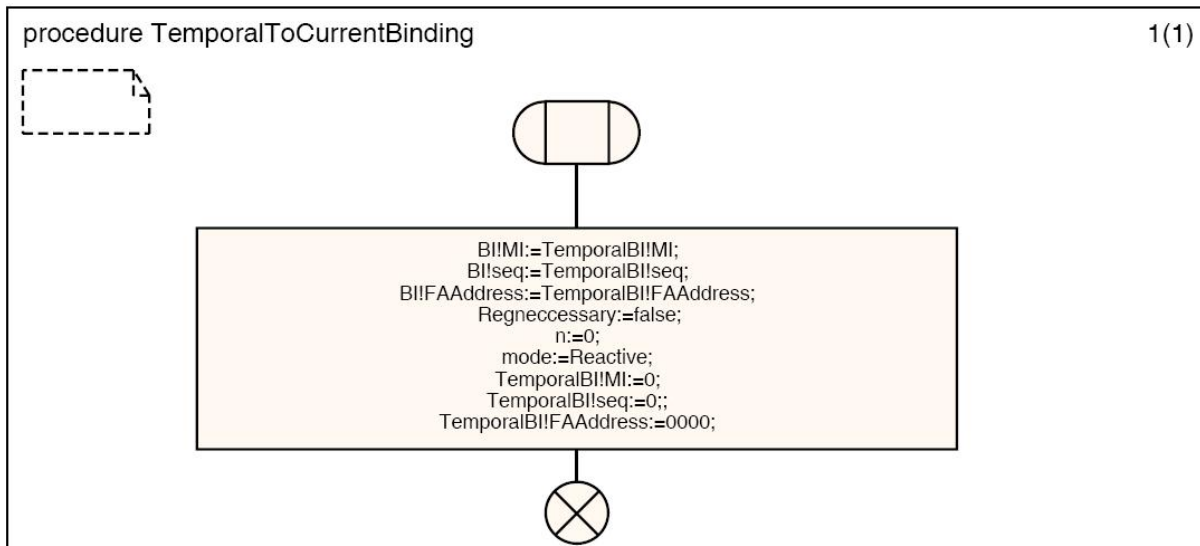


Fig C.26: **TemporalToCurrentBinding** procedure

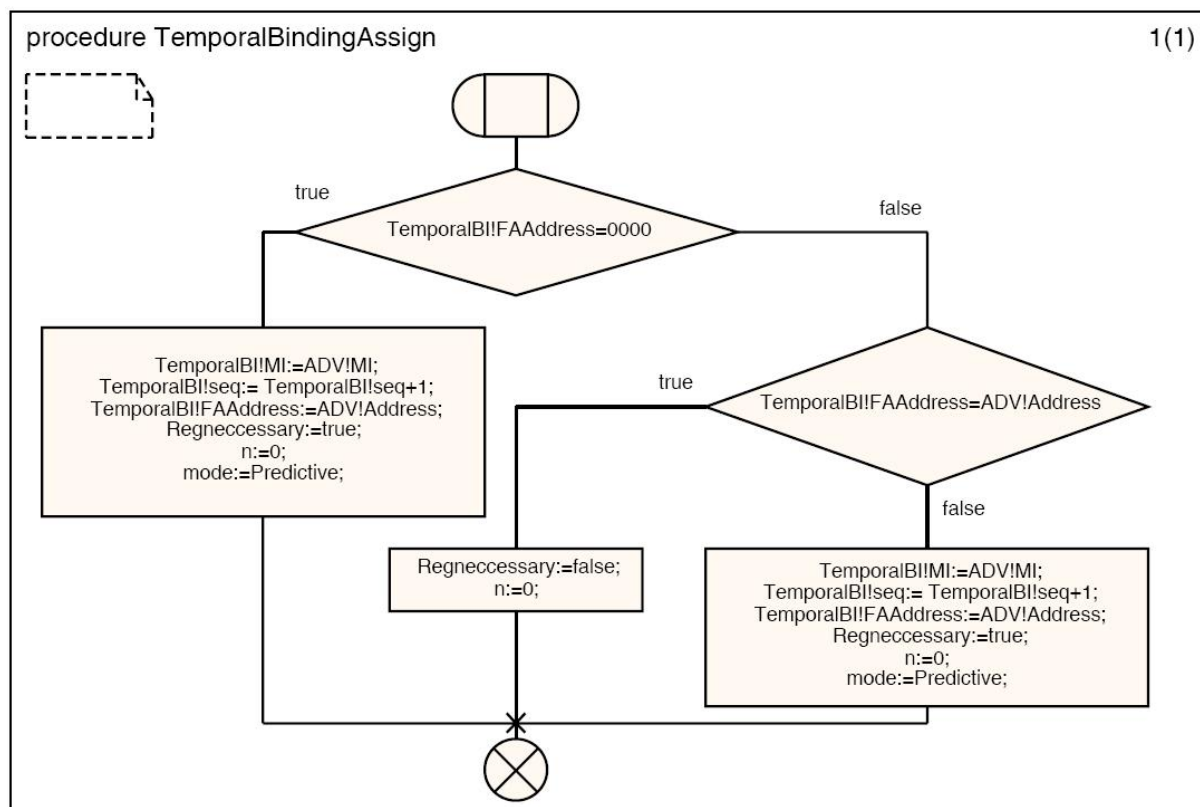


Fig C.27: **TemporalBindingAssign** procedure

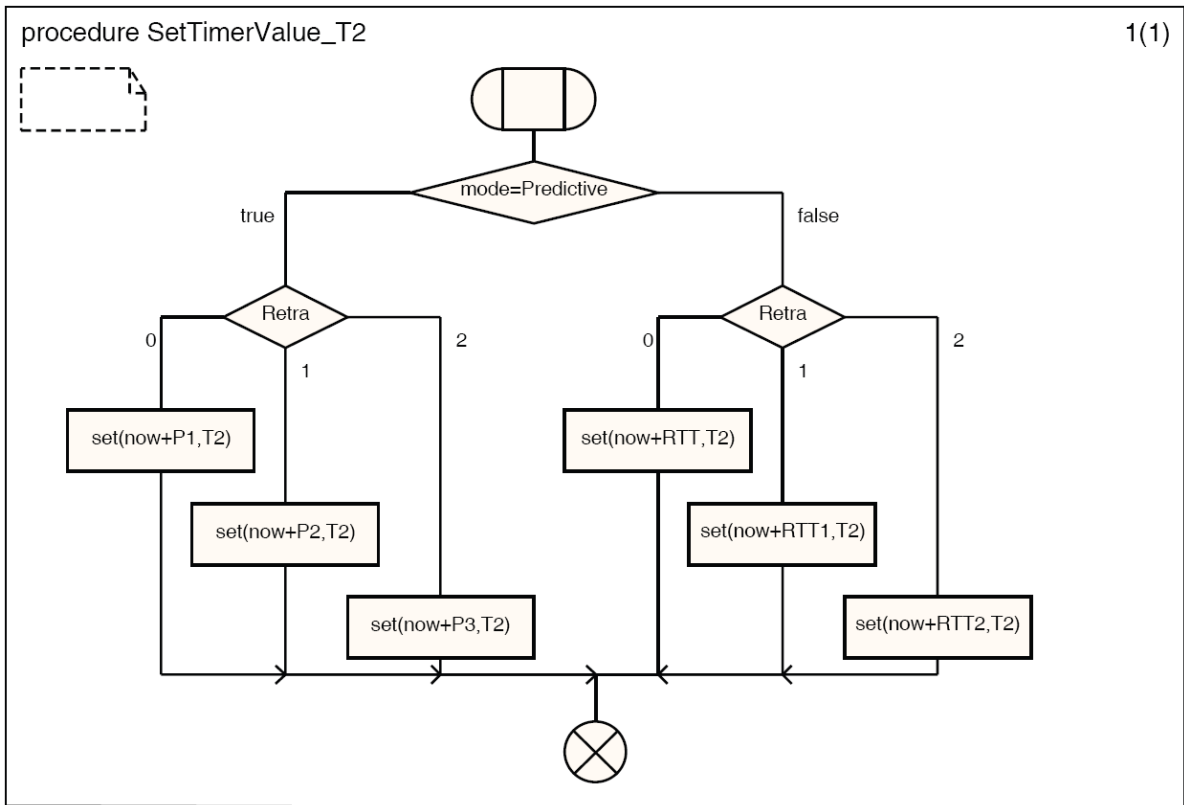


Fig C.28: SetTimerValue_T2 procedure

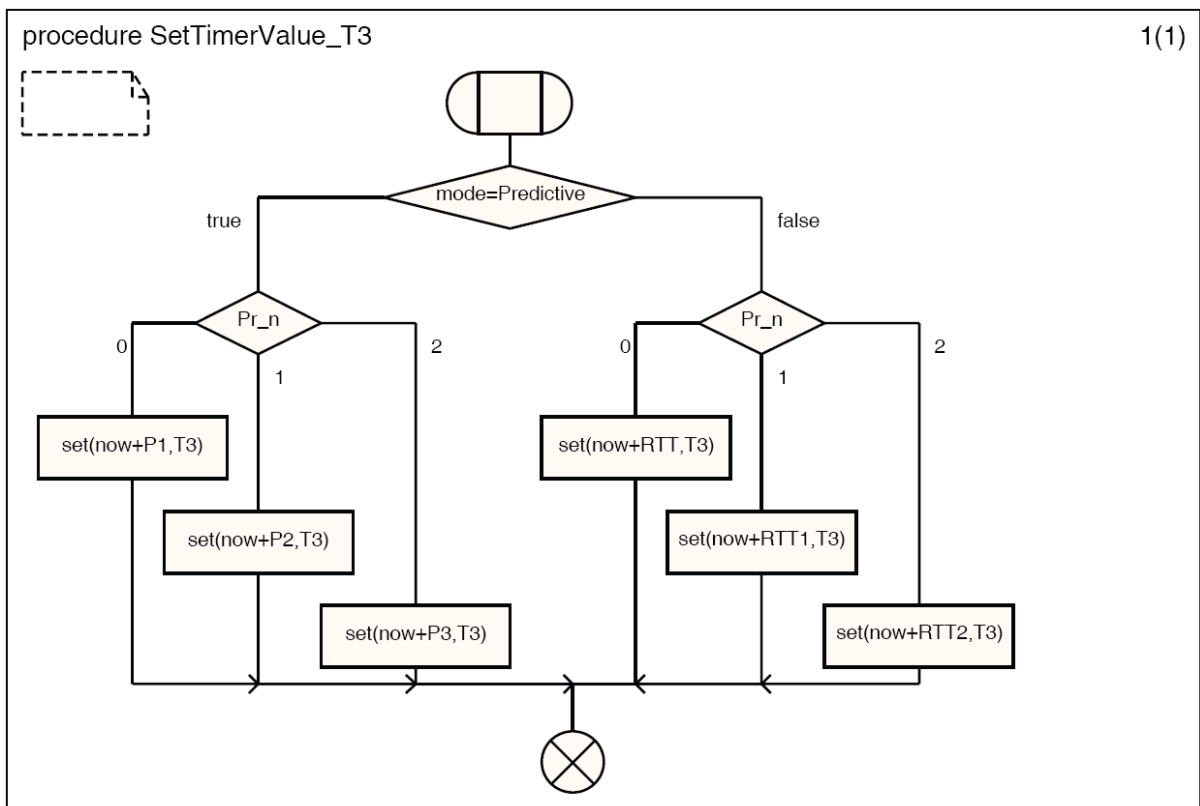


Fig C.29: SetTimerValue_T3 procedure

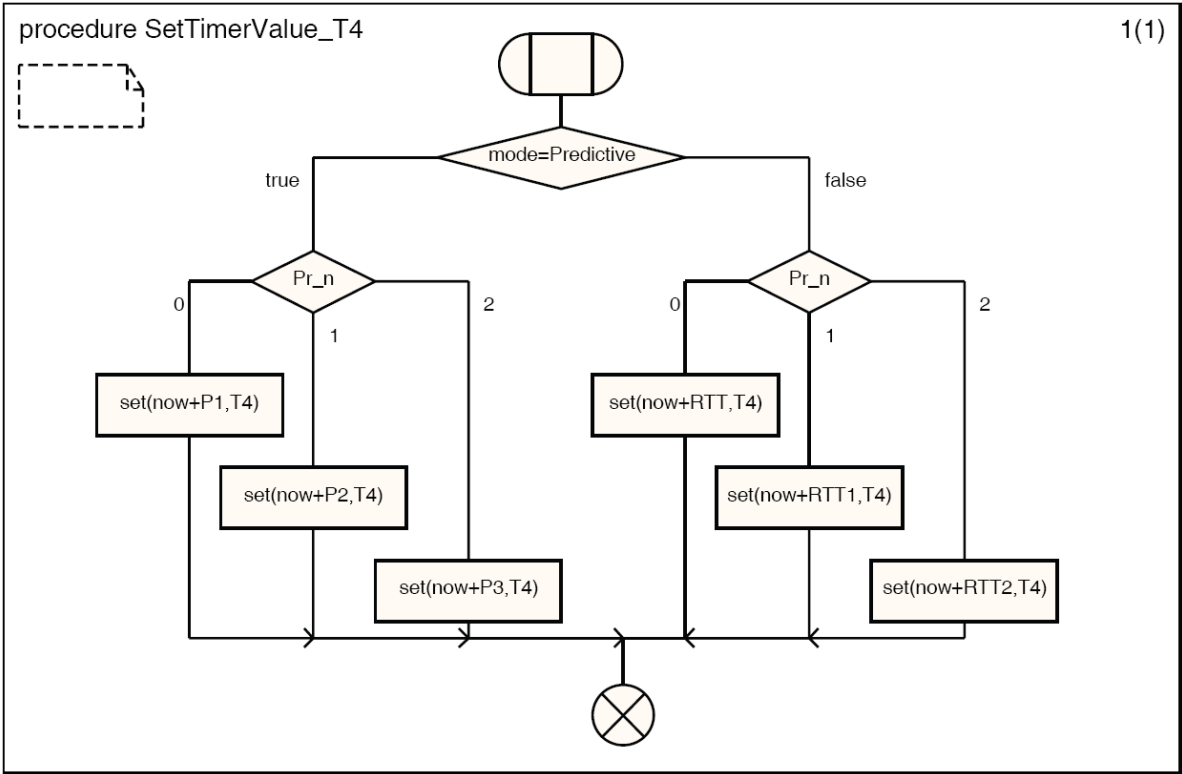


Fig C.30: SetTimerValue_T4 procedure

C.3. FA2

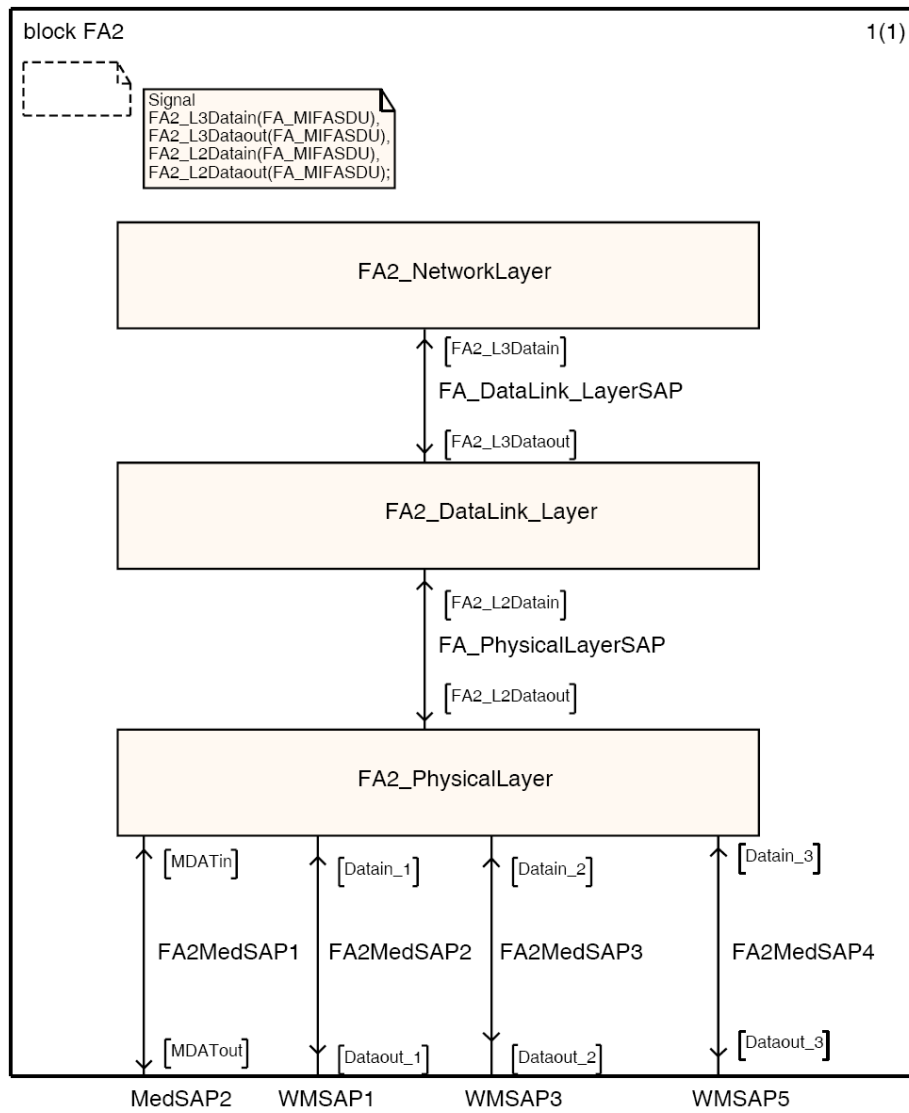


Fig C.31: Structure of FA2, FA2 block

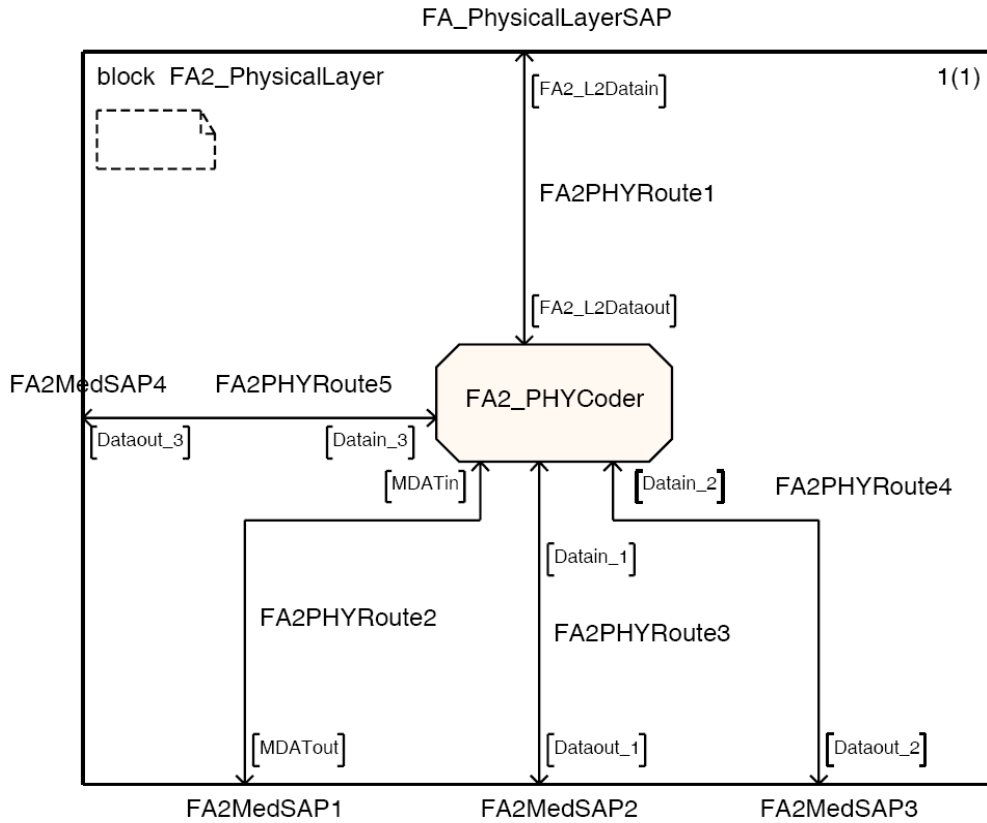


Fig C.32: Physical layer of FA2, **FA2_PhysicalLayer** block

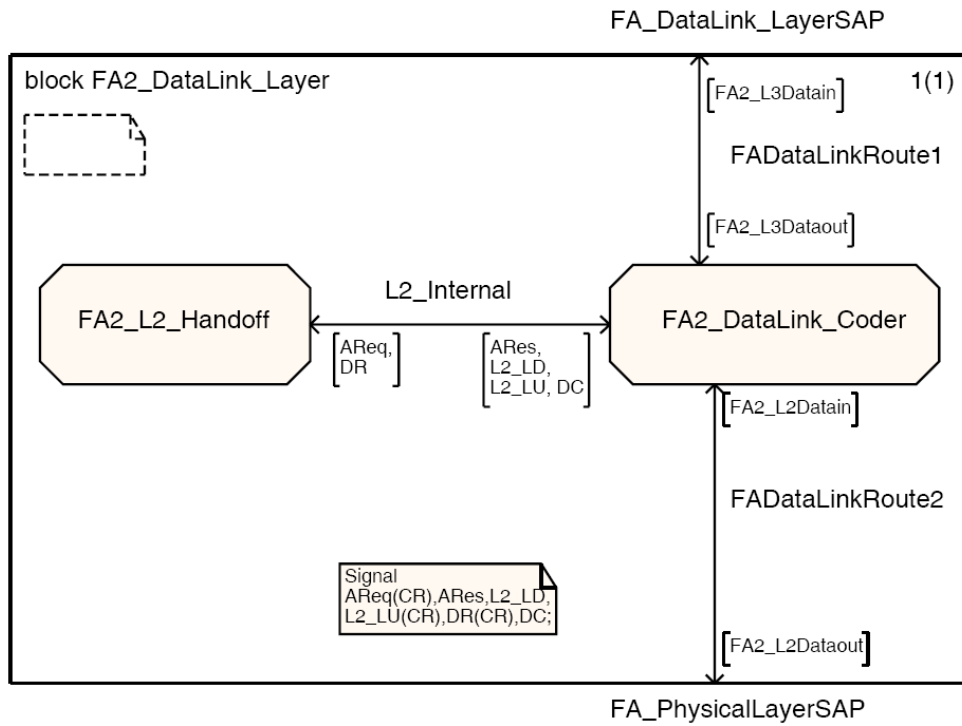


Fig C.33: Data link layer of FA2, **FA2_DataLink_Layer** block

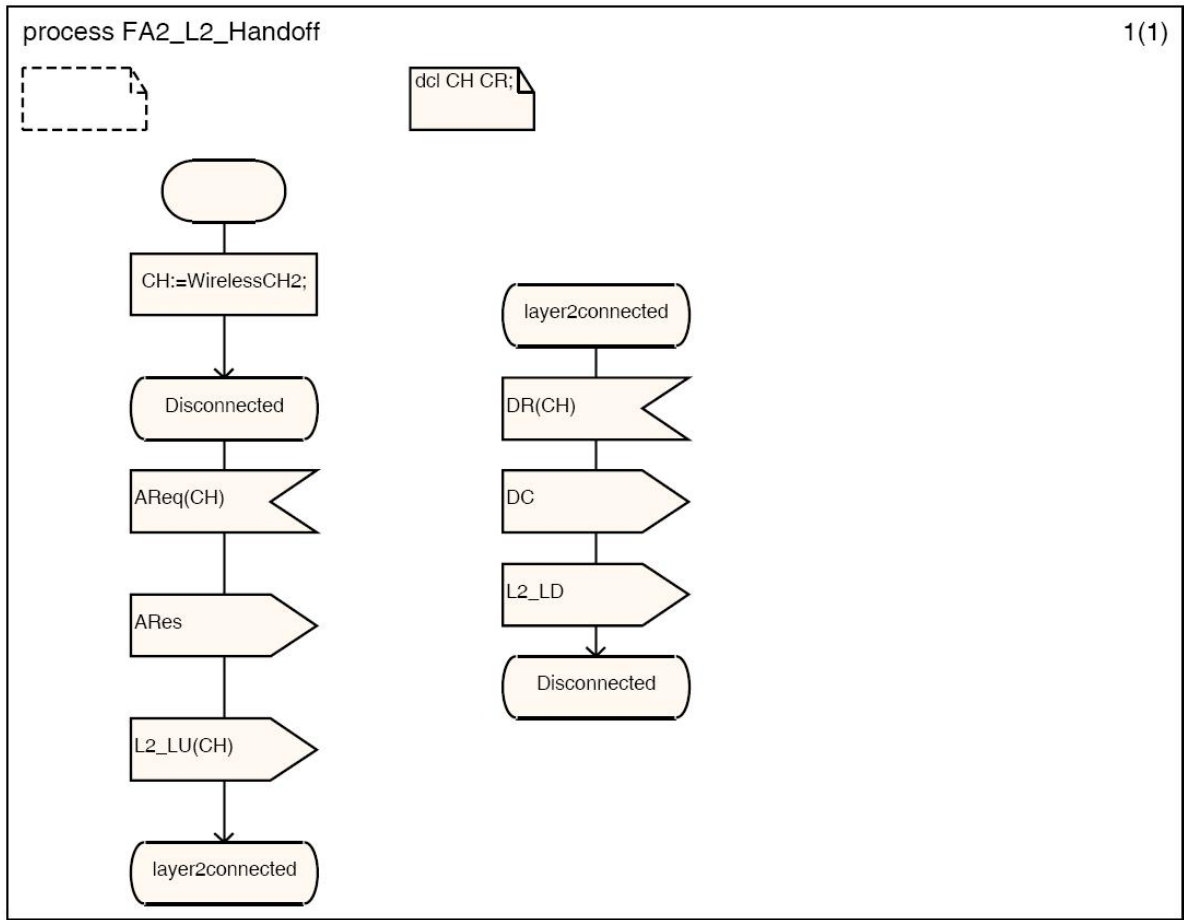


Fig C.34: **FA2_L2_Handoff** process

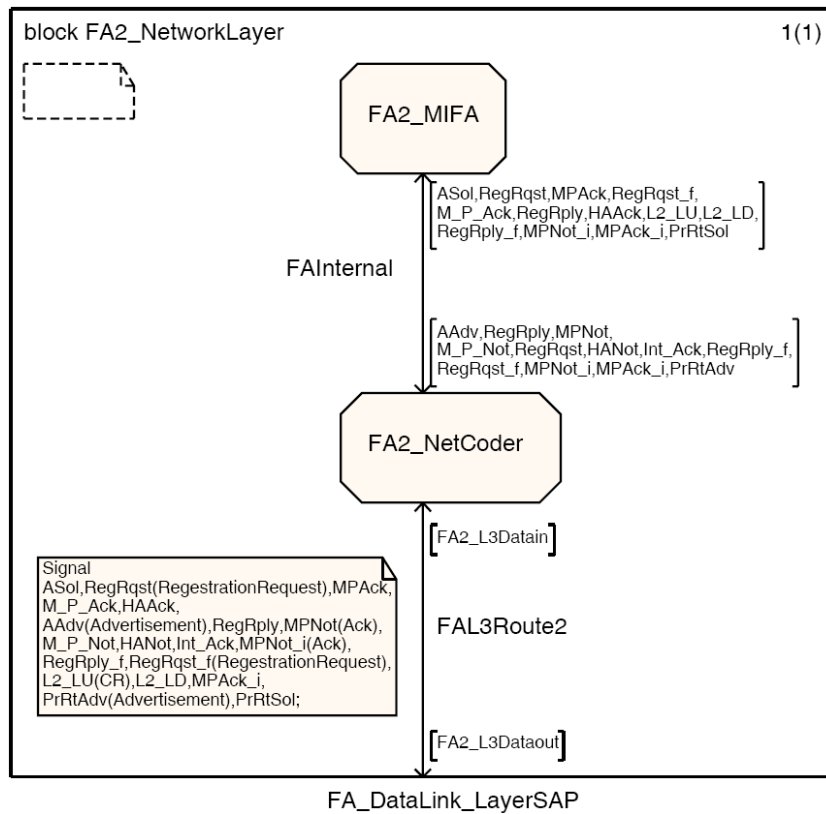


Fig C.35: Network layer of FA2, **FA2_NetworkLayer** block

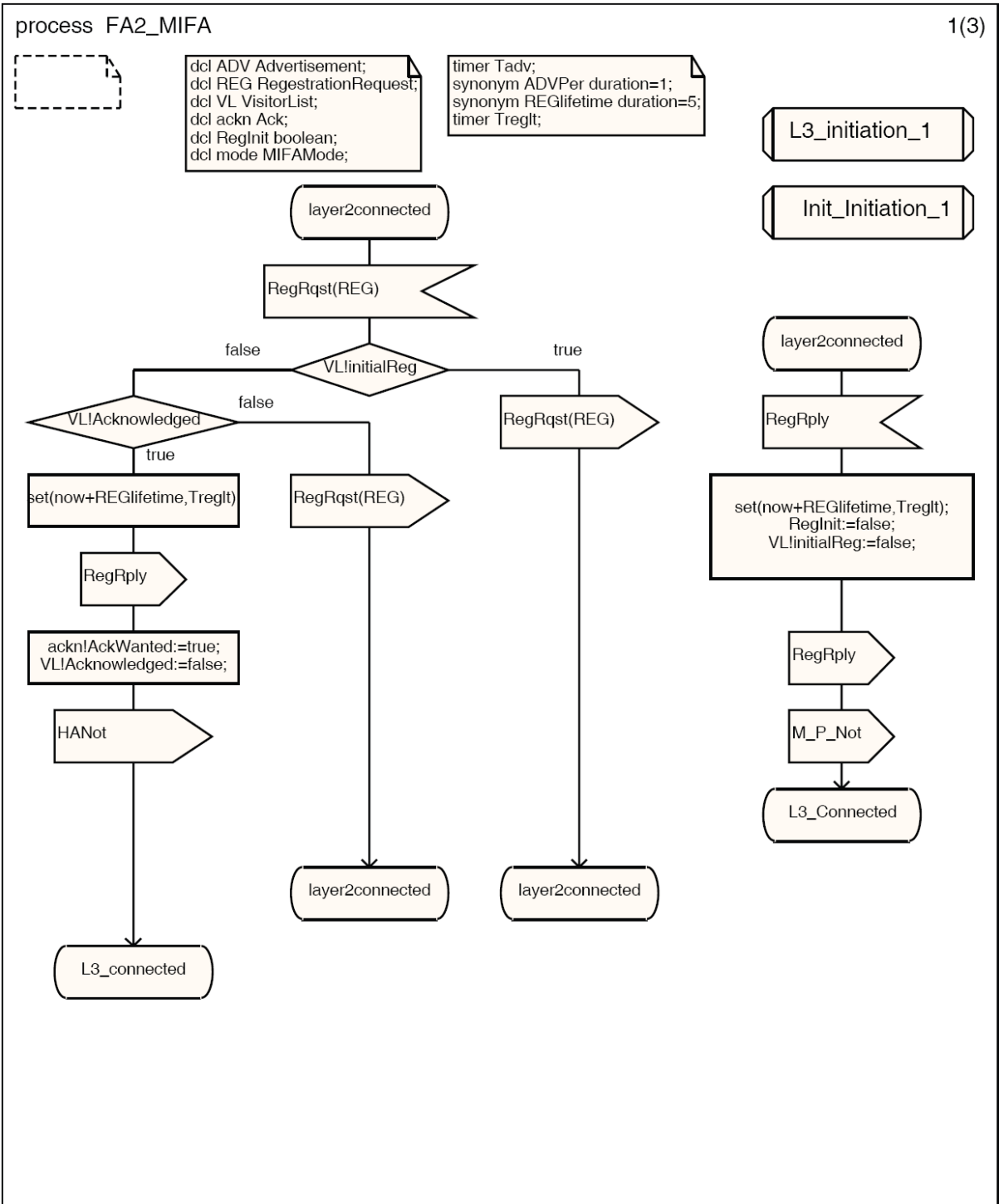


Fig C.36: FA2_MIFA process - (1)

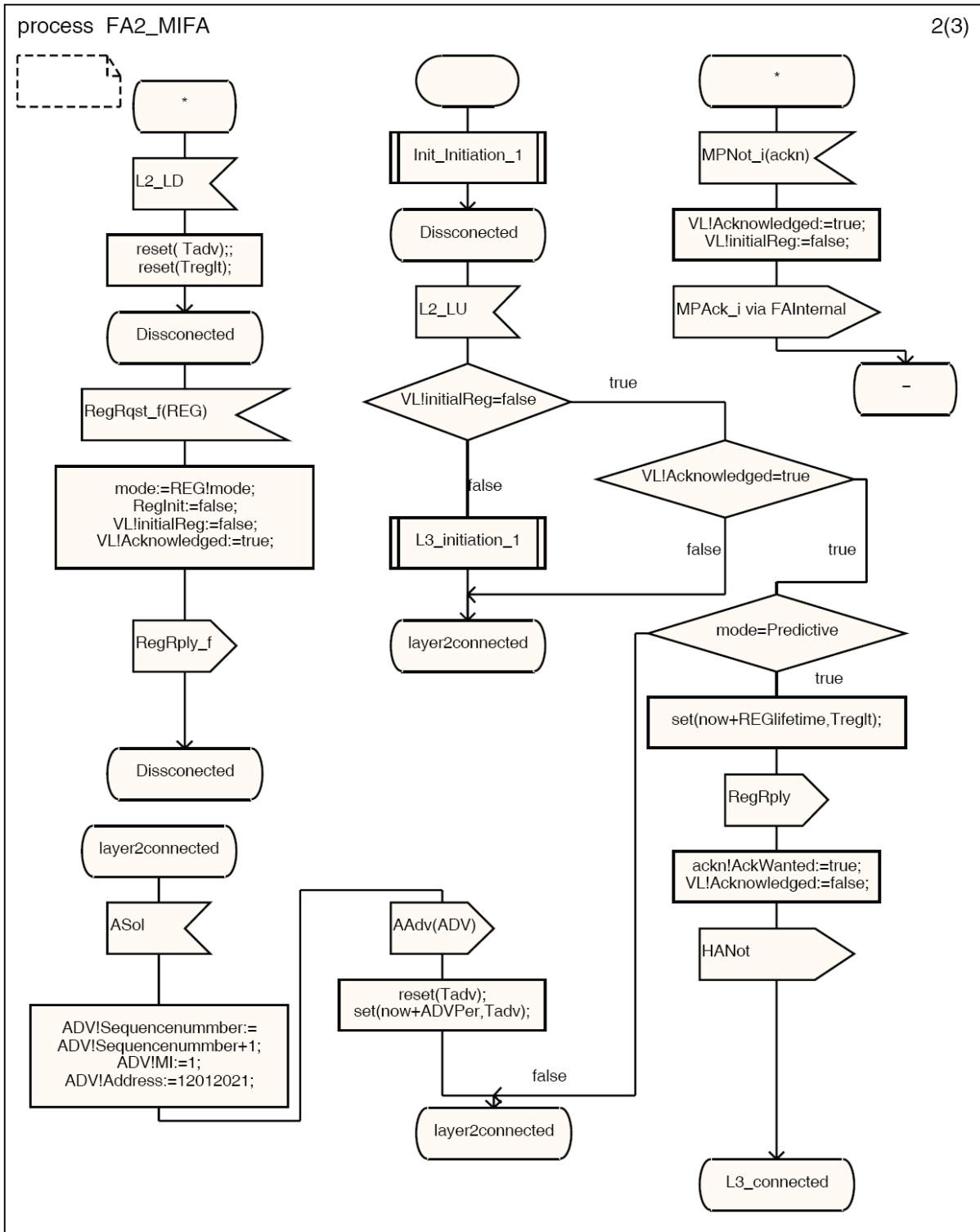


Fig C.37: FA2_MIFA process - (2)

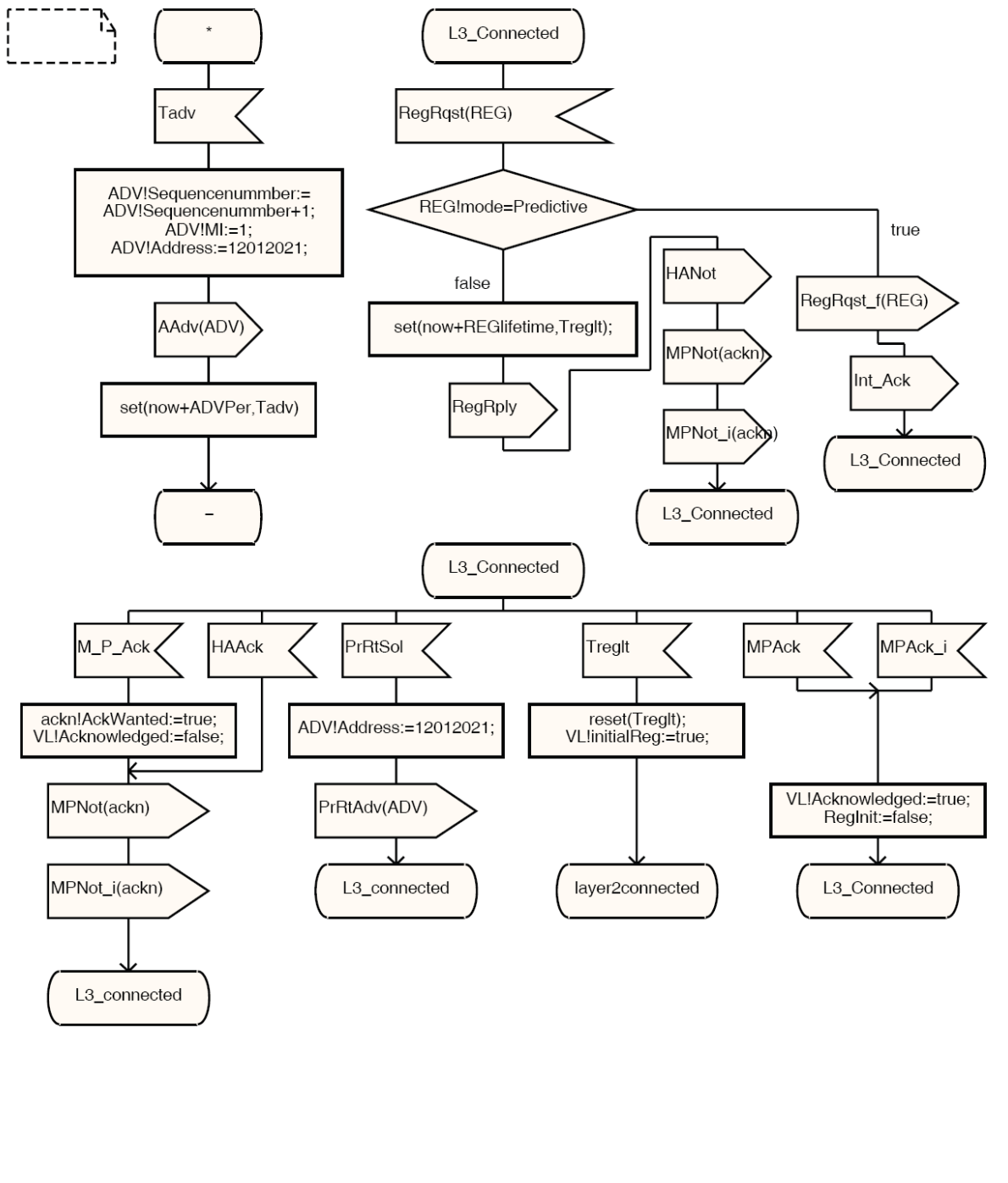


Fig C.38: FA2_MIFA process - (3)

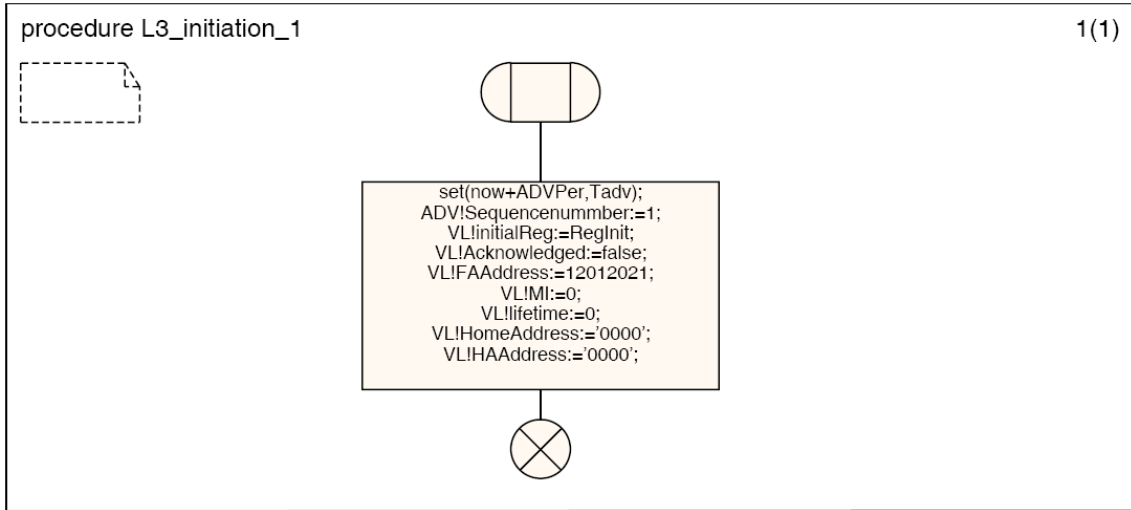


Fig C.39: **L3_initiation_1** procedure

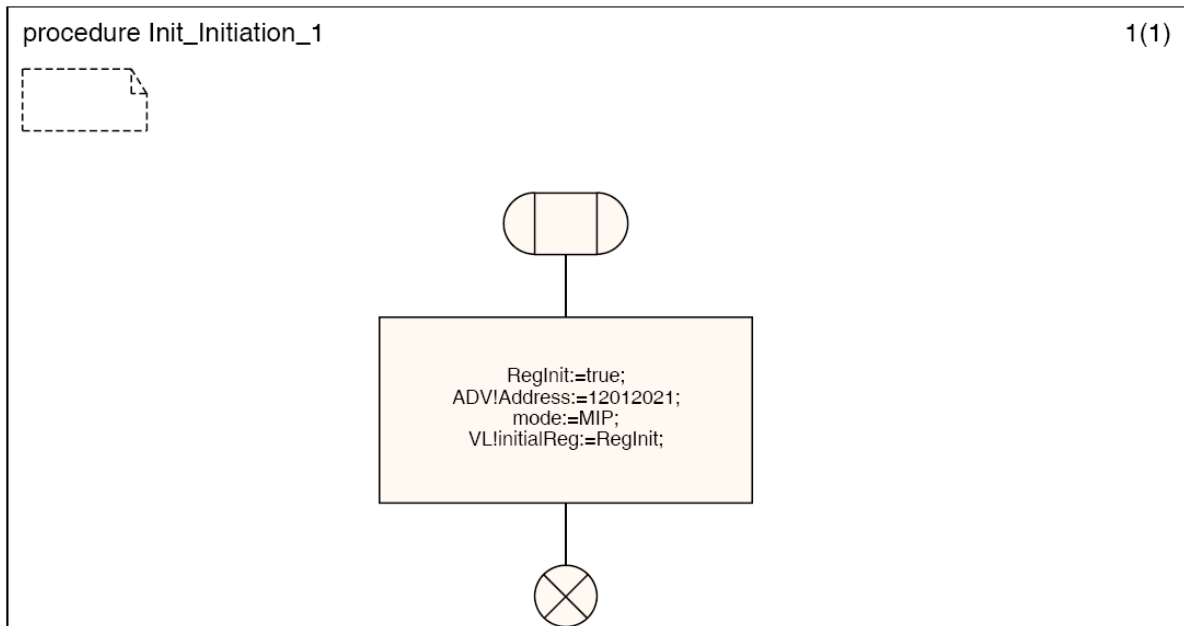


Fig C.40: **Init_Initiation_1** procedure

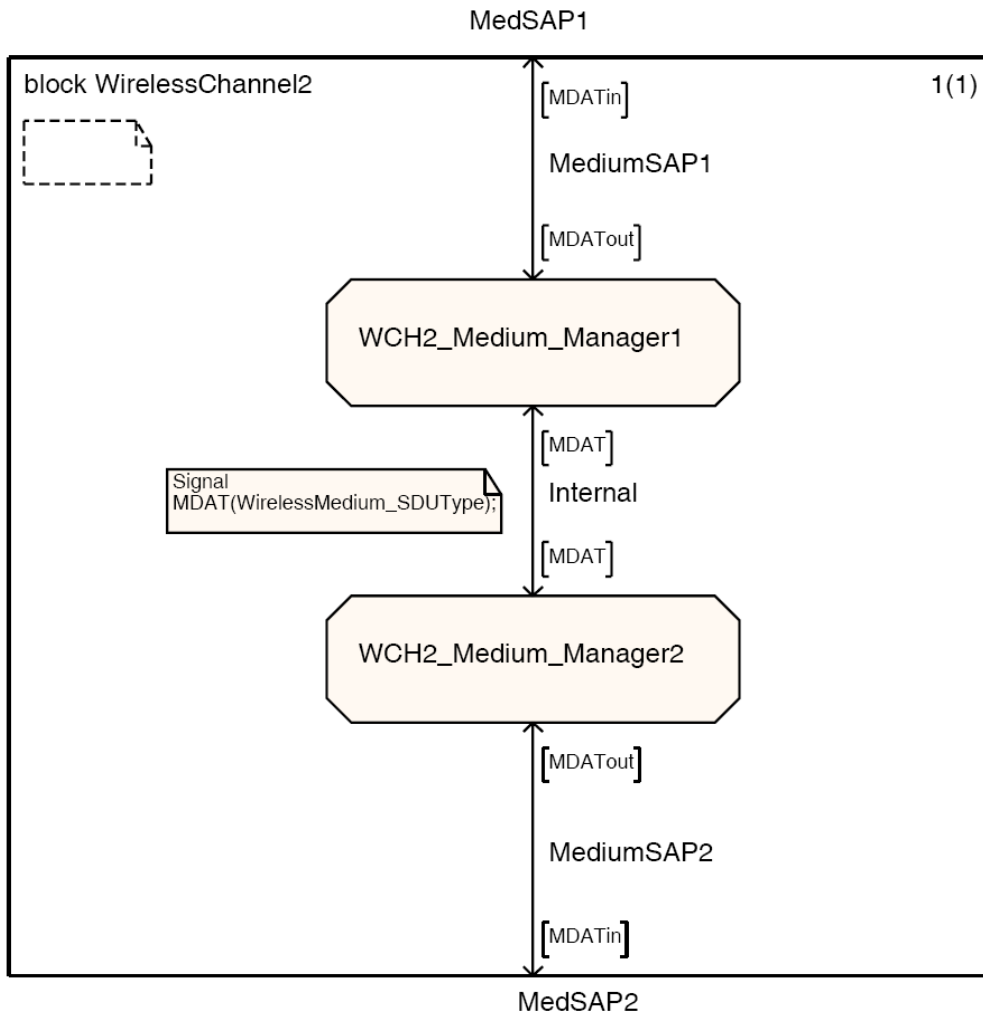


Fig C.41: Structure of wireless channel 2, **WirelessChannel2** block

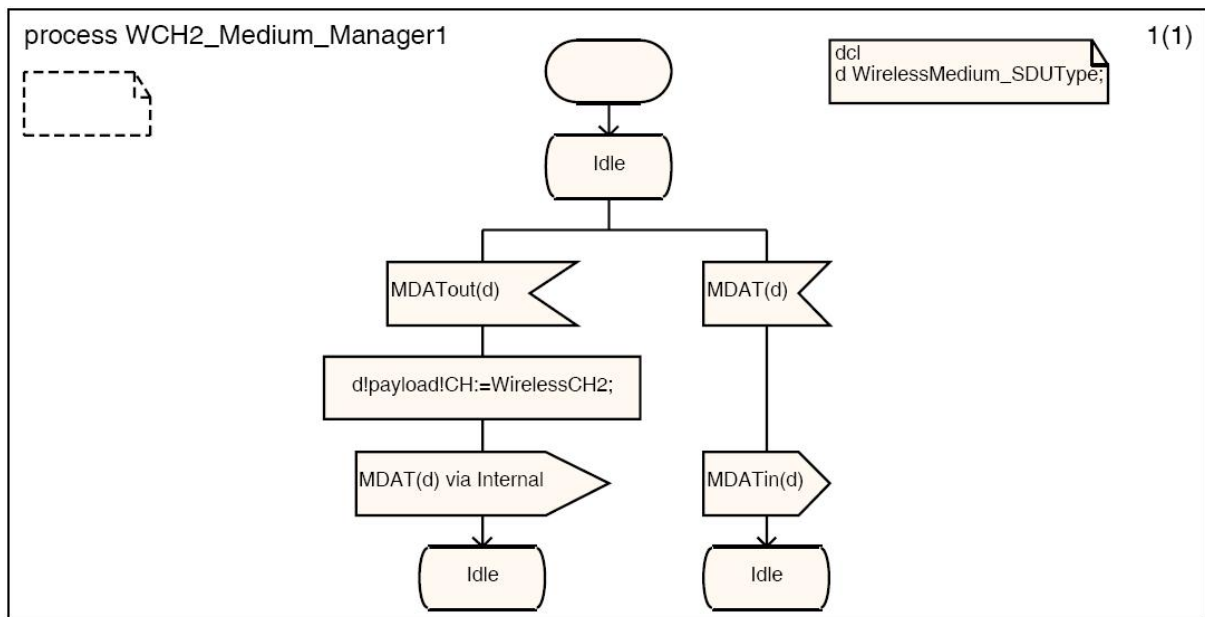


Fig C.42: **WCH2_Medium_Manager1** process

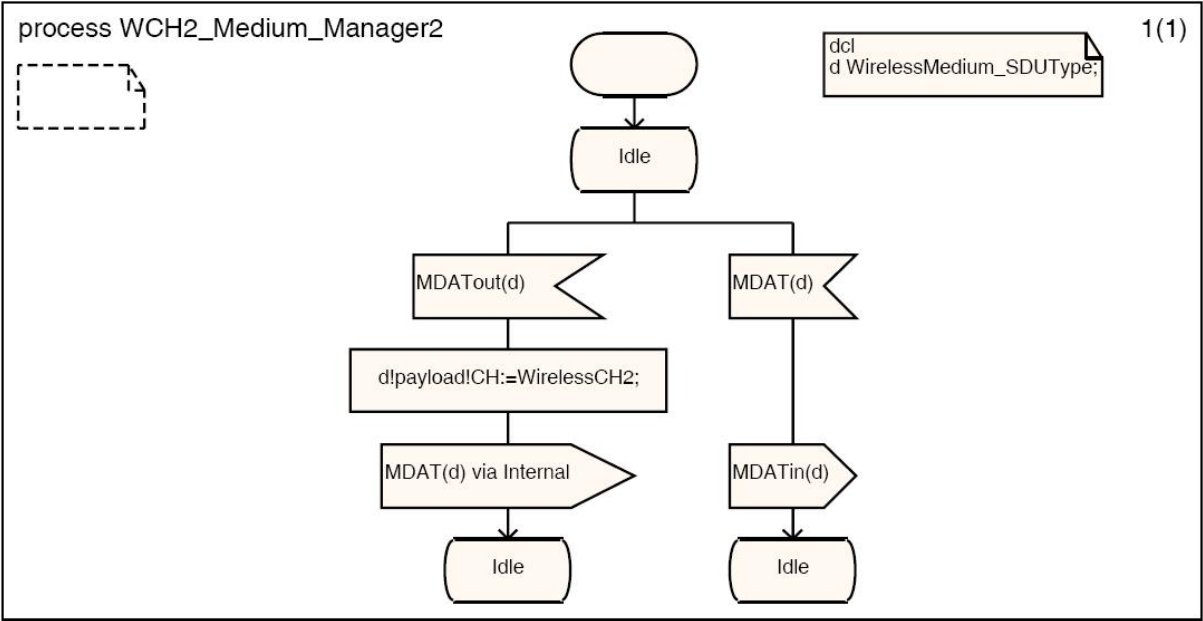


Fig C.43: **WCH2_Medium_Manager2** process

D. Parameters Used in the Generic Mathematical Model

This appendix provides a complete list of all parameters used in the generic model to analyze the mobility protocols studied in section 5.6. After presenting the parameters of the studied protocols, an example describing how these parameters can be derived is provided.

D.1. Network Topology

D.1.1. Hierarchical Topology

The parameters of the applied hierarchical network topology are presented in table D.1.

$D_{MA,MR}$	$D_{MR,GW}$	$D_{GW,HA}$	$D_{GW,CN}$	$D_{CN,HA}$	δ_D	δ_s	ρ	τ_2
2 hops	2 hops	5 hops	4 hops	3 hops	0,05	0,5	10	5 msec

Tab D.1: Parameters of the applied hierarchical network topology

The values of τ_1 depends on the used radio access technology. In the analysis provided in this dissertation, a fast radio access technology is assumed. Therefore, τ_1 is supposed to be 2 msec.

D.1.2. Mesh Topology

The parameters of the applied mesh network topology are provided in table D.2.

$\bar{D}_{MA,GW}$	$\bar{D}_{newMA,oldMA}$	$\bar{D}_{MA,MR}$	$D_{GW,HA}$	$D_{GW,CN}$	$D_{CN,HA}$	δ_D	δ_s	ρ	τ_2
3 hops	2 hops	1.5 hops	5 hops	4 hops	3 hops	0,05	0,5	10	5 msec

Tab D.2: Parameters of the applied mesh network topology

The same assumptions are assumed regarding the value of τ_1 .

D.2. Movement Model

The values of R and G are listed in table D.3 below.

Hierarchical topology		Mesh topology	
R	G	R	G
0,75	0,25	1	0

Tab D.3: Values of R and G

D.3. Application of the Model to Break-Before-Make Protocols

D.3.1. MIPv4

The parameters and assumptions listed in table D.4 are used.

Used network topology	Mesh
BUnode	HA
InNodes	No
Current MA	New MA
B	$[0 \ 0 \ R \ 0 \ 0 \ 0]$
T	$[0 \ 0 \ T_{HA} \ 0 \ 0 \ 0]$, T_{HA} is calculated from equations (6) and (7)
LP	$[0 \ 0 \ LP_{HA} \ 0 \ 0 \ 0]$, LP_{HA} is calculated from equations (12) and (14)
LUC	$[0 \ 0 \ luc_{HA} \ 0 \ 0 \ 0]$, luc_{HA} is calculated from equation (25)
d	$[0 \ 0 \ d_{HA} \ 0 \ d_{nMA} \ 0]$ for $pd_{CN,MN}$, d_{HA} and d_{nMA} are calculated from equations (34) and (35)
T_{timer}	$2 * RTT$, RTT is the round trip time between the MN and the HA
Triangular packet routing	CN→HA→MA→MN
Route optimization	No
η_1	1
η_2	1
λ	50 packets per second
Forwarding during the handoff	No
Layer 2 handoff latency	50 msec
$Fc_{handoff}$	0

Tab D.4: Parameters and assumptions for MIPv4

The parameters required to calculate T_{HA} are listed in table D.5.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	$\bar{D}_{currentMA,BUnode}$
2	2	2	1	0	0.1 msec	0.5 msec	0	0	4 msec	8 hops

Tab D.5: Parameters required to calculate T_{HA} employing MIPv4

The parameters required to calculate t_{HA} are listed in table D.6.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ'	$\bar{D}_{currentMA,BUnode}$
1	1	1	1	0	0.1 msec	0.5 msec	0	0	4 msec	8 hops

Tab D.6: Parameters required to calculate t_{HA} employing MIPv4

The parameters required to calculate luc_{HA} are listed in table D.7.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
2	2	2	1	0	10	25	0	0	$2*\rho*\delta_s$	8 hops

Tab D.7: Parameters required to calculate luc_{HA} employing MIPv4

D.3.2. MIPv6

The parameters and assumptions listed in table D.8 are used.

Used network topology	Mesh
BUnode	<ul style="list-style-type: none"> HA for triangular routing CN for route optimization.
InNodes	No
Current MA	New MA
B	<ul style="list-style-type: none"> For performance evaluation: <ul style="list-style-type: none"> $[0 \ 0 \ R \ 0 \ 0 \ 0]$ for triangular routing $[0 \ 0 \ 0 \ 0 \ 0 \ R]$ for route optimization For cost estimation: $[0 \ 0 \ R \ 0 \ 0 \ R]$
T	<ul style="list-style-type: none"> $[0 \ 0 \ T_{HA} \ 0 \ 0 \ 0]$ for triangular routing $[0 \ 0 \ 0 \ 0 \ 0 \ T_{ANP}]$ for route optimization T_{HA} and T_{ANP} are calculated from equations (6) and (7)
LP	<ul style="list-style-type: none"> $[0 \ 0 \ LP_{HA} \ 0 \ 0 \ 0]$ for triangular routing $[0 \ 0 \ 0 \ 0 \ 0 \ LP_{ANP}]$ for route optimization LP_{HA} and LP_{ANP} are calculated from equations (12) and (14)
LUC	$[0 \ 0 \ luc_{HA} \ 0 \ 0 \ luc_{ANP}]$, luc_{HA} and luc_{ANP} are calculated from equation (25)
T_{timer}	$2 * RTT$ <ul style="list-style-type: none"> RTT is the round trip time between the MN and the HA for the triangular routing RTT is the round trip time between the MN and the CN for the route optimization
Triangular packet routing	CN→HA→MN
Route optimization	CN→MN
d	<ul style="list-style-type: none"> For $pd_{CN, MN}$

	<ul style="list-style-type: none"> ▪ $\begin{bmatrix} 0 & 0 & d_{HA} & 0 & 0 & 0 \end{bmatrix}$ for triangular routing ▪ $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ for route optimization d_{HA} is calculated from equation (34)
η_1	1
η_2	1
λ	50 packets per second
Forwarding during the handoff	No
Layer 2 handoff latency	50 msec
$Fc_{handoff}$	0
$\Delta_{Auto-Conf} + \Delta_{DAD}$	50 msec
$\Delta'_{Auto-Conf} + \Delta'_{DAD}$	25

Tab D.8: Parameters and assumptions for MIPv6

The parameters required to calculate T_{HA} and T_{ANP} are listed in table D.9.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	$\bar{D}_{currentMA,BUnode}$
2	2	2	1	0	0	0.5 msec	0	0	$4 + \Delta_{Auto-Conf} + \Delta_{DAD}$	8 hops (BUnode is HA) 7 hops (BUnode is CN)

Tab D.9: Parameters required to calculate T_{HA} and T_{ANP} employing MIPv6

The parameters required to calculate t_{HA} and t_{ANP} are listed in table D.10.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ'	$\bar{D}_{currentMA,BUnode}$
1	1	1	1	0	0	0.5 msec	0	0	$4 + \Delta_{Auto-Conf} + \Delta_{DAD}$	8 hops (BUnode is HA) 7 hops (BUnode is CN)

Tab D.10: Parameters required to calculate t_{HA} and t_{ANP} employing MIPv6

The parameters required to calculate luc_{HA} are listed in table D.11.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
2	2	2	1	0	0	25	0	0	$2 * \rho * \delta_s + \Delta'_{Auto-Conf} + \Delta'_{DAD}$	8 hops

Tab D.11: Parameters required to calculate luc_{HA} employing MIPv6

The parameters required to calculate luc_{ANP} are listed in table D.12.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
2	2	2	1	0	0	25	0	0	0	7 hops

Tab D.12: Parameters required to calculate luc_{ANP} employing MIPv6

D.3.3. MIFAv4 in Reactive Mode

The parameters and assumptions listed in table D.13 are used.

Used network topology	Mesh
BUnode	<ul style="list-style-type: none"> For performance evaluation: <ul style="list-style-type: none"> Old MA for downlink traffic New MA for uplink traffic For cost estimation: <ul style="list-style-type: none"> Old MA and HA
InNodes	No
Current MA	New MA
B	<ul style="list-style-type: none"> For performance evaluation: <ul style="list-style-type: none"> $[0 \ 0 \ 0 \ R \ 0 \ 0]$ For cost estimation: <ul style="list-style-type: none"> $[0 \ 0 \ R \ R \ 0 \ 0]$
T	$[0 \ 0 \ 0 \ T_{MA-MR} \ 0 \ 0]$, T_{MA-MR} is calculated from equations (6) and (7)
LP	$[0 \ 0 \ 0 \ LP_{MA-MR} \ 0 \ 0]$, LP_{MA-MR} is calculated from equations (12) and (14)
LUC	$[0 \ 0 \ luc_{HA} \ luc_{MA-MR} \ 0 \ 0]$, luc_{HA} and luc_{MA-MR} are calculated from equation (25)
T_{timer}	$2 * RTT$ <ul style="list-style-type: none"> RTT is the round trip time between the MN and the old MA for the downlink traffic RTT is the round trip time between the MN and the new MA for the uplink traffic
Triangular packet routing	CN→HA→MA→MN
Route optimization	No
d	<ul style="list-style-type: none"> For $pd_{CN,MN}$ <ul style="list-style-type: none"> $[0 \ 0 \ d_{HA} \ 0 \ d_{nMA} \ 0]$ For $Fc_{handoff}$ <ul style="list-style-type: none"> $[0 \ 0 \ 0 \ d_{oMA} \ d_{nMA} \ 0]$ d_{HA} is calculated from equation (34), d_{oMA} and d_{nMA} are calculated from equation (35)

η_1	1
η_2	1
λ	50 packets per second
Forwarding during the handoff	Yes
Layer 2 handoff latency	50 msec
k_9	1
ContNode	HA
RNode	Old MA
Δ	70 msec. This value is calculated as follows: the new MA sends a PFA_Not and a HA_Not simultaneously to inform the old MA and the HA, respectively. The HA needs 30 msec more than the old MA to receive the message notifying it of the new CoA. Assuming that the HA has just sent a data packet before the receipt of the HA_Not message, the data packet needs 40 msec to arrive at the old MA. Thus, the forwarding of data packets from the old MA to the new one takes duration of 70 msec.
$\bar{D}_{CurrentMA, neiMA}$	2 hops

Tab D.13: Parameters and assumptions for MIFAv4 in reactive mode

The parameters required to calculate T_{MA-MR} for downlink traffic are listed in table D.14.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	$\bar{D}_{currentMA, BUnode}$
1	2	2	1	0	1 msec	0.5 msec	0	0	4 msec	2 hops

Tab D.14: Parameters required to calculate T_{MA-MR} for downlink traffic employing MIFAv4 in reactive mode

The parameters required to calculate T_{MA-MR} for uplink traffic are listed in table D.15.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	$\bar{D}_{currentMA, BUnode}$
2	0	1	0	0	1 msec	1 msec	0	0	4 msec	0 hops

Tab D.15: Parameters required to calculate T_{MA-MR} for uplink traffic employing MIFAv4 in reactive mode

The parameters required to calculate t_{MA-MR} are listed in table D.16.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ'	$\bar{D}_{currentMA, BUnode}$
1	1	1	1	0	1 msec	0.5 msec	0	0	4 msec	2 hops

Tab D.16: Parameters required to calculate t_{MA-MR} employing MIFAv4 in reactive mode

The parameters required to calculate luc_{MA-MR} are listed in table D.17.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	$a'_{BU\text{node}}$	$a'_{In\text{Node}}$	ni	γ''	$\bar{D}_{\text{currentMA,BU\text{node}}}$
2	2	2	1	0	25	10	0	0	$2 * \rho * \delta_s$	2 hops

Tab D.17: Parameters required to calculate luc_{MA-MR} employing MIFAv4 in reactive mode

The parameters required to calculate luc_{HA} are listed in table D.18.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	$a'_{BU\text{node}}$	$a'_{In\text{Node}}$	ni	γ''	$\bar{D}_{\text{currentMA,BU\text{node}}}$
0	2	1	1	0	10	25	0	0	$a'_{MA} + N_{av} * \delta_s *$ $\bar{D}_{\text{CurrentMA,neiMA}}$	8 hops

Tab D.18: Parameters required to calculate luc_{HA} employing MIFAv4 in reactive mode

N_{av} is the average number of MAs in each L3-FHR.

D.3.4. MIFAv6 in Reactive Mode

The parameters and assumptions listed in table D.19 are used.

Used network topology	Mesh
BU\text{node}	<ul style="list-style-type: none"> For performance evaluation: <ul style="list-style-type: none"> Old MA for downlink traffic New MA for uplink traffic For cost estimation: <ul style="list-style-type: none"> Old MA, HA and CN
InNodes	No
Current MA	New MA
B	<ul style="list-style-type: none"> For performance evaluation: <ul style="list-style-type: none"> $[0 \ 0 \ 0 \ R \ 0 \ 0]$ For cost estimation: Old MA and HA <ul style="list-style-type: none"> $[0 \ 0 \ R \ R \ 0 \ R]$
T	$[0 \ 0 \ 0 \ T_{MA-MR} \ 0 \ 0]$, T_{MA-MR} is calculated from equations (6) and (7)
LP	$[0 \ 0 \ 0 \ LP_{MA-MR} \ 0 \ 0]$, LP_{MA-MR} is calculated from equations (12) and (14)
LUC	$[0 \ 0 \ luc_{HA} \ luc_{MA-MR} \ 0 \ luc_{ANP}]$, luc_{HA} , luc_{MA-MR} and luc_{ANP} are calculated from equation (25)
T_{timer}	$2 * RTT$ <ul style="list-style-type: none"> RTT is the round trip time between the MN and the old MA for the downlink traffic RTT is the round trip time between the MN and the new MA for the uplink traffic
Triangular packet routing	CN→HA→MN
Route optimization	CN→MN

d	<ul style="list-style-type: none"> For $pd_{CN, MN}$ <ul style="list-style-type: none"> $[0 \ 0 \ d_{HA} \ 0 \ 0 \ 0]$ for the triangular routing $[0 \ 0 \ 0 \ 0 \ 0 \ 0]$ for the route optimization For $F_{handoff}$ <ul style="list-style-type: none"> $[0 \ 0 \ 0 \ d_{oMA} \ d_{nMA} \ 0]$ <p>d_{HA} is calculated from equation (34), d_{oMA} and d_{nMA} are calculated from equation (35)</p>
η_1	1
η_2	1
λ	50 packets per second
Forwarding during the handoff	Yes
Layer 2 handoff latency	50 msec
k_9	1
ContNode	<ul style="list-style-type: none"> HA for the triangular routing CN for the route optimization
RNode	Old MA
Δ	<ul style="list-style-type: none"> 70 msec for the triangular routing 60 msec for the route optimization <p>This value is calculated as follows: for the triangular routing, the new MA sends a Hn_Not and a BU simultaneously to inform the old MA and the HA, respectively. The HA needs 30 msec more than the old MA to receive the message notifying it of the new CoA. Assuming that the HA has just sent a data packet before the receipt of the BU message. The data packet needs 40 msec to arrive at the old MA. Thus, the forwarding of data packets from the old MA to the new one takes duration of 70 msec. The value of Δ for the route optimization is calculated in a similar way.</p>
$\Delta_{Auto-Conf} + \Delta_{DAD}$	5 msec
$\Delta'_{Auto-Conf} + \Delta'_{DAD}$	10
$\bar{D}_{CurrentMA, neiMA}$	2 hops

Tab D.19: Parameters and assumptions for MIFAv6 in reactive mode

The parameters required to calculate T_{MA-MR} for downlink traffic are listed in table D.20.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	$\bar{D}_{currentMA, BUnode}$
1	2	2	1	0	1 msec	0.5 msec	0	0	$4 + \Delta_{Auto-Conf} + \Delta_{DAD}$	2 hops

Tab D.20: Parameters required to calculate T_{MA-MR} for downlink traffic employing MIFAv6 in reactive mode

The parameters required to calculate T_{MA-MR} for uplink traffic are listed in table D.21.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	$\bar{D}_{currentMA,BUnode}$
2	0	1	0	0	1 msec	1 msec	0	0	$4 + \Delta_{Auto-Conf} + \Delta_{DAD}$	0 hops

Tab D.21: Parameters required to calculate T_{MA-MR} for uplink traffic employing MIFAv6 in reactive mode

The parameters required to calculate t_{MA-MR} are listed in table D.22.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ'	$\bar{D}_{currentMA,BUnode}$
1	1	1	1	0	1 msec	0.5 msec	0	0	$4 + \Delta_{Auto-Conf} + \Delta_{DAD}$	2 hops

Tab D.22: Parameters required to calculate t_{MA-MR} employing MIFAv6 in reactive mode

The parameters required to calculate luc_{MA-MR} are listed in table D.23.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
2	2	2	1	0	25	10	0	0	$2 * \rho * \delta_s + \Delta'_{Auto-Conf} + \Delta'_{DAD}$	2 hops

Tab D.23: Parameters required to calculate luc_{MA-MR} employing MIFAv6 in reactive mode

The parameters required to calculate luc_{HA} are listed in table D.24.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
0	2	1	1	0	10	25	0	0	$a'_{MA} + N_{av} * \delta_s *$ $\bar{D}_{CurrentMA,neiMA}$	8 hops

Tab D.24: Parameters required to calculate luc_{HA} employing MIFAv6 in reactive mode

The parameters required to calculate luc_{ANP} are listed in table D.25.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
2	2	2	1	0	0	25	0	0	0	7 hops

Tab D.25: Parameters required to calculate luc_{ANP} employing MIFAv6 in reactive mode

D.3.5. MIPRR

The parameters and assumptions listed in table D.26 are used.

Used network topology	Hierarchical
BUnode	MRs and GW
InNodes	No
Current MA	New MA
B	$[R \ G \ 0 \ 0 \ 0 \ 0]$
T	$[T_{MR} \ T_{GW} \ 0 \ 0 \ 0 \ 0]$, T_{MR} and T_{GW} are calculated from equations (6) and (7)
LP	$[LP_{MR} \ LP_{GW} \ 0 \ 0 \ 0 \ 0]$, LP_{MR} and LP_{GW} are calculated from equations (12) and (14)
LUC	$[luc_{MR} \ luc_{GW} \ 0 \ 0 \ 0 \ 0]$, luc_{MR} and luc_{GW} are calculated from equation (25)
T_{timer}	$2 * RTT$, RTT is the round trip time between the MN and the GW
Triangular packet routing	CN→HA→GW→MR→MA→MN
Route optimization	No
d	<ul style="list-style-type: none"> For $pd_{CN, MN}$ <ul style="list-style-type: none"> $[d_{MR} \ d_{GW} \ d_{HA} \ 0 \ d_{nMA} \ 0]$ d_{HA} and d_{nMA} are calculated from equations (34) and (35), while d_{GW} and d_{MR} are calculated from equations (37) and (38)
η_1	1
η_2	1
λ	50 packets per second
Forwarding during the handoff	No
Layer 2 handoff latency	50 msec
$F_{handoff}$	0

Tab D.26: Parameters and assumptions for MIPRR

The parameters required to calculate T_{MR} and T_{GW} are listed in table D.27.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	$D_{currentMABUnode}$
2	2	2	1	0	0.1 msec	0.5 msec	0	0	4 msec	2 hops (BUnode is MR) 4 hops (BUnode is GW)

Tab D.27: Parameters required to calculate T_{MR} and T_{GW} employing MIPRR

The parameters required to calculate t_{MR} and t_{GW} are listed in table D.28.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	$a_{BU\text{node}}$	$a_{In\text{Node}}$	ni	γ'	$D_{\text{currentMABU}\text{node}}$
1	1	1	1	0	0.1 msec	0.5 msec	0	0	4 msec	2 hops (BUnode is MR) 4 hops (BUnode is GW)

Tab D.28: Parameters required to calculate t_{MR} and t_{GW} employing MIPRR

The parameters required to calculate luc_{MR} and luc_{GW} are listed in table D.29.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	$a'_{BU\text{node}}$	$a'_{In\text{Node}}$	ni	γ''	$D_{\text{currentMABU}\text{node}}$
2	2	2	1	0	10	25	0	0	$2 * \rho * \delta_s$	2 hops (BUnode is MR) 4 hops (BUnode is GW)

Tab D.29: Parameters required to calculate luc_{MR} and luc_{GW} employing MIPRR

D.3.6. HAWAII

The parameters and assumptions listed in table D.30 are used.

Used network topology	Hierarchical
BUnode	<ul style="list-style-type: none"> For calculating the handoff latency <ul style="list-style-type: none"> Old MA For calculating the expected number of the dropped packets <ul style="list-style-type: none"> For non-forwarding schemes <ul style="list-style-type: none"> Old MA for uplink traffic Crossover router for downlink For forwarding schemes <ul style="list-style-type: none"> Old MA For calculating the location update cost <ul style="list-style-type: none"> Old MA
InNodes	Yes, calculated from equation (39)
Current MA	New MA
B	$[0 \ 0 \ 0 \ R \ G \ 0]$
T	$[0 \ 0 \ 0 \ T_{MA-MR} \ T_{MA-GW} \ 0]$, T_{MA-MR} and T_{MA-GW} are calculated from equations (6) and (7)
LP	$[0 \ 0 \ 0 \ LP_{MA-MR} \ LP_{MA-GW} \ 0]$, LP_{MA-MR} and LP_{MA-GW} are calculated from equations (12) and (14)
LUC	$[0 \ 0 \ 0 \ luc_{MA-MR} \ luc_{MA-GW} \ 0]$, luc_{MA-MR} and luc_{MA-GW} are calculated from equation (25)
T_{timer}	$2 * RTT$, RTT is the round trip time between the MN and the old MA
Triangular packet routing	CN→HA→GW→MN
Route optimization	CN→GW→MN
d	<ul style="list-style-type: none"> For $pdc_{CN, MN}$ <ul style="list-style-type: none"> $[0 \ d_{GW} \ d_{HA} \ 0 \ 0 \ 0]$ for the triangular routing

	<ul style="list-style-type: none"> ▪ $[0 \ d_{GW} \ 0 \ 0 \ 0 \ 0]$ for the route optimization • For $Fc_{handoff}$ in forwarding schemes <ul style="list-style-type: none"> ▪ $[0 \ 0 \ 0 \ 0 \ 0 \ 0]$. <p>d_{HA} and d_{GW} are calculated from (34) and (37)</p>
η_1	1
λ	50 packets per second
Forwarding during the handoff	<ul style="list-style-type: none"> • No in non-forwarding schemes • Yes in forwarding schemes
Layer 2 handoff latency	50 msec
$Fc_{handoff}$	<ul style="list-style-type: none"> • 0 in non-forwarding schemes • Calculated from (32) in forwarding schemes
k_9	1
ContNode	Crossover router
RNode	Old MA
Δ	25 msec, calculated from equation (40). In the forwarding schemes, the MN informs the old MA, which starts forwarding data packets towards the new MA. As the crossover node (a MR or the GW in the worst case) is informed, it updates its routing cache and forwards data packets towards the new MA directly. Thus, the old MA will forward data packets after it gets informed and until the crossover node is informed. Thus, the average forwarding time can be calculated as $\tau_2 * (R * D_{MA-MR} + G * D_{MA-GW})$. Notice that R and G are used because the average forwarding time is affected by the applied mobility scenario, see appendix E. Assuming the crossover router has just sent a data packet before it has been informed, this data packet will need the same time calculated above to arrive at the old BS.

Tab D.30: Parameters and assumptions for HAWAII

The parameters required to calculate T_{MA-MR} and T_{MA-GW} are listed in table D.31.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	$D_{currentMABUnode}$
2	2	2	1	2	0.1 msec	0.2 msec	0.1 msec	Calculated from (39)	4 msec	4 hops (crossover router is a MR) 8 hops (crossover router is the GW)

Tab D.31: Parameters required to calculate T_{MA-MR} and T_{MA-GW} employing HAWAII

The parameters required to calculate t_{MA-MR} and t_{MA-GW} are listed in table D.32.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ'	$D_{currentMA,BUnode}$
1	1	1	1	1	0.1 msec	0.2 msec	0.1 msec	Calculated from (39)	4 msec	Non-forwarding schemes: 2 hops (crossover router is a MR). 4 hops (crossover router is the GW) Forwarding schemes: 4 hops (crossover router is a MR). 8 hops (crossover router is the GW)

Tab D.32: Parameters required to calculate t_{MA-MR} and t_{MA-GW} employing HAWAII

The parameters required to calculate luc_{MA-MR} and luc_{MA-GW} are listed in table D.33.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$D_{currentMA,BUnode}$
2	2	2	1	2	5	10	5	Calculated from (39)	$2 * \rho * \delta_s$	4 hops (BUnode is MR) 8 hops (BUnode is GW)

Tab D.33: Parameters required to calculate luc_{MA-MR} and luc_{MA-GW} employing HAWAII

D.3.7. Proxy MIPv6

The parameters and assumptions listed in table D.34 are used.

Used network topology	Mesh
BUnode	HA
InNodes	No
Current MA	New MA
B	$[0 \ 0 \ R \ 0 \ 0 \ 0]$
T	$[0 \ 0 \ T_{HA} \ 0 \ 0 \ 0]$, T_{HA} is calculated from equations (6) and (7)
LP	$[0 \ 0 \ LP_{HA} \ 0 \ 0 \ 0]$, LP_{HA} is calculated from equations (12) and (14)
LUC	$[0 \ 0 \ luc_{HA} \ 0 \ 0 \ 0]$, luc_{HA} is calculated from equation (25)
T_{timer}	$2 * RTT$, RTT is the round trip time between the current MA and the HA
Triangular packet routing	$CN \rightarrow HA \rightarrow MA \rightarrow MN$
Route optimization	No
d	<ul style="list-style-type: none"> For $pd_{CN, MN}$

	<ul style="list-style-type: none"> ▪ $[0 \ 0 \ d_{HA} \ 0 \ d_{nMA} \ 0]$ d_{HA} and d_{nMA} are calculated from equations (34) and (35)
η_1	1
η_2	1
λ	50 packets per second
Forwarding during the handoff	No
Layer 2 handoff latency	50 msec
$Fc_{handoff}$	0

Tab D.34: Parameters and assumptions for Proxy MIPv6

The parameters required to calculate T_{HA} are listed in table D.35.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	$\bar{D}_{currentMA,BUnode}$
0	2	1	1	0	0.1 msec	0.5 msec	0	0	2	8 hops

Tab D.35: Parameters required to calculate T_{HA} employing Proxy MIPv6

The parameters required to calculate t_{HA} are listed in table D.36.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ'	$\bar{D}_{currentMA,BUnode}$
0	1	0	1	0	0	0.5 msec	0	0	0	8 hops

Tab D.36: Parameters required to calculate t_{HA} employing Proxy MIPv6

The parameters required to calculate luc_{HA} are listed in table D.37.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
0	2	1	1	0	10	25	0	0	$\rho * \delta_s$	8 hops

Tab D.37: Parameters required to calculate luc_{HA} employing Proxy MIPv6

D.4. Application of the Model to Make-Before-Break Protocols

D.4.1. MIFAv4 in Predictive Mode

The parameters and assumptions listed in table D.38 are used.

Used network topology	Mesh
BNode	<ul style="list-style-type: none"> For performance evaluation: <ul style="list-style-type: none"> New MA For cost estimation: <ul style="list-style-type: none"> New MA and HA
InNodes	No
Current MA	Old MA
B	<ul style="list-style-type: none"> For performance evaluation: <ul style="list-style-type: none"> $[0 \ 0 \ 0 \ R \ 0 \ 0]$ For cost estimation: <ul style="list-style-type: none"> $[0 \ 0 \ R \ R \ 0 \ 0]$
T	$[0 \ 0 \ 0 \ T_{MA-MR} \ 0 \ 0]$ <ul style="list-style-type: none"> T_{MA-MR} is calculated from equations (6) and (7) in the first and second cases T_{MA-MR} is calculated from equation (15) in the third case
LP	$[0 \ 0 \ 0 \ LP_{MA-MR} \ 0 \ 0]$ <ul style="list-style-type: none"> LP_{MA-MR} is calculated from equations (12) and (14) in the first and second cases LP_{MA-MR} is calculated from equations (14) and (18) in the third case
LUC	$[0 \ 0 \ luc_{HA} \ luc_{MA-MR} \ 0 \ 0]$, luc_{HA} and luc_{MA-MR} are calculated from equation (25)
T_{timer}	$2 * RTT$, RTT is the round trip time between the MN and the old MA
Triangular packet routing	CN→HA→MA→MN
Route optimization	No
d	<ul style="list-style-type: none"> For $pd_{CN, MN}$ <ul style="list-style-type: none"> $[0 \ 0 \ d_{HA} \ 0 \ d_{nMA} \ 0]$ For $F_{handoff}$ <ul style="list-style-type: none"> $[0 \ 0 \ 0 \ d_{oMA} \ d_{nMA} \ 0]$ d_{HA} is calculated from equation (34). d_{oMA} and d_{nMA} are calculated from equation (35)
η_1	1
η_2	1
λ	50 packets per second
Forwarding during the handoff	Yes
Layer 2 handoff latency	50 msec
k_9	1
ContNode	HA

RNode	Old MA
Δ	91 msec. This time results as follows: assuming that a L2-LD trigger has just been raised after the old MA has received the Reg_Rqst message from the MN. The old MA sends the Reg_Rqst to the new MA, which authenticates it and informs the HA. Thus, the time required to inform the HA after the appearance of the L2-LD trigger is 51 msec. Moreover, assuming that the HA has just sent a data packet before the receipt of the HA_Not message. This packet requires 40 msec to arrive at the old MA.
$\bar{D}_{CurrentMA,neiMA}$	2 hops

Tab D.38: Parameters and assumptions for MIFAv4 in predictive mode

The parameters required to calculate T'_{MA-MR} are listed in table D.39.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	γ_{ext}	$\bar{D}_{currentMA,BUnode}$
1	2	2	1	0	1 msec	1 msec	0	0	4 msec	2 msec	2 hops

Tab D.39: Parameters required to calculate T'_{MA-MR} employing MIFAv4 in predictive mode

The parameters required to calculate t_{RNode} are listed in table D.40.

k_3	k_4	k_5	k_6	k'_4	k'_5	k'_6
1	0	0	0	1	0	0
k'_7	γ'_1	a_{MA}	a_{BUnode}	a_{InNode}	a_{RNode}	ni
0	4	1 msec	1 msec	0 msec	1 msec	0
$\bar{D}_{currentMA,BUnode}$			$\bar{D}_{RNode,BUnode}$		$\bar{D}_{currentMA,RNode}$	
2			2		0	

Tab D.40: Parameters required to calculate t_{RNode} employing MIFAv4 in predictive mode

The parameters required to calculate luc_{MA-MR} are listed in table D.41.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
3	2	2	1	0	10	25	0	0	$2*\rho*\delta_s$	2 hops

Tab D.41: Parameters required to calculate luc_{MA-MR} employing MIFAv4 in predictive mode

The parameters required to calculate luc_{HA} are listed in table D.42.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	$a'_{BU\text{node}}$	$a'_{In\text{Node}}$	ni	γ''	$\bar{D}_{\text{currentMA,BU\text{node}}}$
0	2	1	1	0	10	25	0	0	$a'_{MA} + N_{av} * \delta_S *$ $\bar{D}_{\text{CurrentMA,neiMA}}$	8 hops

Tab D.42: Parameters required to calculate luc_{HA} employing MIFAv4 in predictive mode

D.4.2. MIFAv6 in Predictive Mode

The parameters and assumptions listed in table D.43 are used.

Used network topology	Mesh
BUnode	<ul style="list-style-type: none"> For performance evaluation: <ul style="list-style-type: none"> New MA For cost estimation: <ul style="list-style-type: none"> New MA, HA and CN
InNodes	No
Current MA	Old MA
B	<ul style="list-style-type: none"> For performance evaluation: <ul style="list-style-type: none"> $[0 \ 0 \ 0 \ R \ 0 \ 0]$ For cost estimation: <ul style="list-style-type: none"> $[0 \ 0 \ R \ R \ 0 \ R]$
T	$[0 \ 0 \ 0 \ T_{MA-MR} \ 0 \ 0]$ <ul style="list-style-type: none"> T_{MA-MR} is calculated from equations (6) and (7) in the first and second cases T_{MA-MR} is calculated from equation (15) in the third case
LP	$[0 \ 0 \ 0 \ LP_{MA-MR} \ 0 \ 0]$ <ul style="list-style-type: none"> LP_{MA-MR} is calculated from equations (12) and (14) in the first and second cases LP_{MA-MR} is calculated from equations (14) and (18) in the third case
LUC	$[0 \ 0 \ luc_{HA} \ luc_{MA-MR} \ 0 \ luc_{ANP}]$, luc_{HA} , luc_{MA-MR} and luc_{ANP} are calculated from equation (25)
T_{timer}	$2 * RTT$, RTT is the round trip time between the MN and the old MA
Triangular packet routing	CN→HA→MN
Route optimization	CN→MN
d	<ul style="list-style-type: none"> For $pd_{CN, MN}$ <ul style="list-style-type: none"> $[0 \ 0 \ d_{HA} \ 0 \ 0 \ 0]$ for the triangular routing $[0 \ 0 \ 0 \ 0 \ 0 \ 0]$ for the route optimization For F_{handoff} <ul style="list-style-type: none"> $[0 \ 0 \ 0 \ d_{oMA} \ d_{nMA} \ 0]$ d_{HA} is calculated from equation (34). d_{oMA} and d_{nMA} are calculated

	from equation (35)
η_1	1
η_2	1
λ	50 packets per second
Forwarding during the handoff	Yes
Layer 2 handoff latency	50 msec
k_9	1
ContNode	<ul style="list-style-type: none"> • HA for the triangular routing • CN for the route optimization
RNode	Old MA
Δ	<ul style="list-style-type: none"> • 96 msec for the triangular routing • 122 msec for the route optimization <p>This time results for the triangular routing as follows: let us assume that a L2-LD trigger has just been raised after the old MA has received the BU message from the MN. The old MA sends the BU to the new MA, which authenticates it, executes the DAD procedure and informs the HA. Thus, the time required to inform the HA after the appearance of the L2-LD trigger is 56 msec. Moreover, assuming that the HA has just sent a data packet before the receipt of the BU message, this packet requires 40 msec to arrive at the old MA.</p> <p>For the route optimization, the principle is similar. However, the MN executes first a layer 2 handoff (50 msec). Assuming that the MN informs the CN direct after the layer 2 handoff, this takes 37 msec. Moreover, assuming that the CN has just sent a data packet before the receipt of the BU message. This packet requires 35 msec to arrive at the old MA.</p>
$\Delta_{Auto-Conf} + \Delta_{DAD}$	5 msec
$\Delta'_{Auto-Conf} + \Delta'_{DAD}$	10
$\bar{D}_{CurrentMA, neiMA}$	2 hops

Tab D.43: Parameters and assumptions for MIFAv6 in predictive mode

The parameters required to calculate T'_{MA-MR} are listed in table D.44.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	γ_{ext}	$\bar{D}_{currentMA, BUnode}$
1	2	2	1	0	1 msec	1 msec	0	0	$4 + \Delta_{Auto-Conf} + \Delta_{DAD}$	2 msec	2 hops

Tab D.44: Parameters required to calculate T'_{MA-MR} employing MIFAv6 in predictive mode

The parameters required to calculate t_{RNode} are listed in table D.45.

k_3	k_4	k_5	k_6	k'_4	k'_5	k'_6
1	0	0	0	1	0	0
k'_7	γ'_1	a_{MA}	a_{BUnode}	a_{InNode}	a_{RNode}	ni
0	4	1 msec	1 msec	0 msec	1 msec	0
$\bar{D}_{currentMA,BUnode}$			$\bar{D}_{RNode,BUnode}$		$\bar{D}_{currentMA,RNode}$	
2			2		0	

Tab D.45: Parameters required to calculate t_{RNode} employing MIFAv6 in predictive mode

The parameters required to calculate luc_{MA-MR} are listed in table D.46.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
3	2	2	1	0	10	25	0	0	$2 * \rho * \delta_S + \Delta'_{Auto-Conf} + \Delta'_{DAD}$	2 hops

Tab D.46: Parameters required to calculate luc_{MA-MR} employing MIFAv6 in predictive mode

The parameters required to calculate luc_{HA} are listed in table D.47.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
0	2	1	1	0	10	25	0	0	$a'_{MA} + N_{av} * \delta_S *$ $\bar{D}_{CurrentMA,neiMA}$	8 hops

Tab D.47: Parameters required to calculate luc_{HA} employing MIFAv6 in predictive mode

The parameters required to calculate luc_{ANP} are listed in table D.48.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
2	2	2	1	0	0	25	0	0	0	7 hops

Tab D.48: Parameters required to calculate luc_{ANP} employing MIFAv6 in predictive mode

D.4.3. Pre-Registration Method

The parameters and assumptions listed in table D.49 are used.

Used network topology	Mesh
BUnode	HA
InNodes	No
Current MA	Old MA

B	$[0 \ 0 \ R \ 0 \ 0 \ 0]$
T	$[0 \ 0 \ T_{HA} \ 0 \ 0 \ 0]$ <ul style="list-style-type: none"> ▪ T_{HA} is calculated from equations (6) and (7) in the first and second cases ▪ T_{HA} is calculated from equation (15) in the third case
LP	$[0 \ 0 \ LP_{HA} \ 0 \ 0 \ 0]$ <ul style="list-style-type: none"> ▪ LP_{HA} is calculated from equations (12) and (14) in the first and second cases ▪ LP_{HA} is calculated from equations (14) and (18) in the third case
LUC	$[0 \ 0 \ luc_{HA} \ 0 \ 0 \ 0]$, luc_{HA} is calculated from equation (25)
d	<ul style="list-style-type: none"> • For $pd_{CN, MN}$ <ul style="list-style-type: none"> ▪ $[0 \ 0 \ d_{HA} \ 0 \ d_{nMA} \ 0]$ d_{HA} and d_{nMA} are calculated from equations (34) and (35)
T_{timer}	$2 * RTT$, RTT is the round trip time between the MN and the HA (via the old and new MA).
Triangular packet routing	$CN \rightarrow HA \rightarrow MA \rightarrow MN$
Route optimization	No
η_1	1
η_2	1
λ	50 packets per second
Forwarding during the handoff	No
Layer 2 handoff latency	50 msec
$Fc_{handoff}$	0

Tab D.49: Parameters and assumptions for the pre-registration method

The parameters required to calculate T'_{HA} are listed in table D.50.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	γ_{ext}	$\bar{D}_{currentMA, BUnode}$
2	2	2	1	0	0.1 msec	0.5 msec	0	0	4.2 msec	-10.1 msec	10 hops

Tab D.50: Parameters required to calculate T'_{HA} employing the pre-registration method

The parameters required to calculate t_{RNode} are listed in table D.51.

k_3	k_4	k_5	k_6	k'_4	k'_5	k'_6
1	1	0	0	1	1	0
k'_7	γ'_1	a_{MA}	a_{BUnode}	a_{InNode}	a_{RNode}	ni
0	4.1 msec	0.1 msec	0.5 msec	0 msec	0 msec	0
$\bar{D}_{currentMA,BUnode}$			$\bar{D}_{RNode,BUnode}$		$\bar{D}_{currentMA,RNode}$	
10 hops			/		/	

Tab D.51: Parameters required to calculate t_{RNode} employing the pre-registration method

The parameters required to calculate luc_{HA} are listed in table D.52.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
2	2	2	1	0	10	25	0	0	$3 * \rho * \delta_s + a_{MA}$	10 hops

Tab D.52: Parameters required to calculate luc_{HA} employing the pre-registration method

D.4.4. FMIPv6

The parameters and assumptions listed in table D.53 are used.

Used network topology	Mesh
BUnode	<ul style="list-style-type: none"> For performance evaluation: <ul style="list-style-type: none"> Old MA for reactive mode New MA for predictive mode For cost estimation: <ul style="list-style-type: none"> Old MA for reactive mode New MA for predictive mode HA and CN for the registration using MIPv6 after completing the handoff
InNodes	No
Current MA	<ul style="list-style-type: none"> New MA for reactive mode Old MA for predictive mode
B	<ul style="list-style-type: none"> For performance evaluation: <ul style="list-style-type: none"> $[0 \ 0 \ 0 \ R \ 0 \ 0]$ For cost estimation <ul style="list-style-type: none"> $[0 \ 0 \ 0 \ R \ 0 \ 0]$ for reactive and predictive modes $[0 \ 0 \ R \ 0 \ 0 \ R]$ for the registration using MIPv6 after completing the handoff
T	$[0 \ 0 \ 0 \ T_{MA-MR} \ 0 \ 0]$ <ul style="list-style-type: none"> T_{MA-MR} is calculated from equations (6) and (7) in the first and second cases T_{MA-MR} is calculated from equation (15) in the third case
LP	$[0 \ 0 \ 0 \ LP_{MA-MR} \ 0 \ 0]$

	<ul style="list-style-type: none"> ▪ LP_{MA-MR} is calculated from equations (12) and (14) in the first and second cases ▪ LP_{MA-MR} is calculated from equations (14) and (18) in the third case
LUC	$[0 \ 0 \ 0 \ luc_{MA-MR} \ 0 \ 0]$, luc_{MA-MR} is calculated from equation (25)
T_{timer}	$2 * RTT$ <ul style="list-style-type: none"> ▪ RTT is the round trip time between the MN and old MA employing the reactive mode ▪ RTT is the round trip time between the MN and the new MA employing the predictive mode
Triangular packet routing	CN→HA→MN
Route optimization	CN→MN
d	<ul style="list-style-type: none"> • For $pd_{CN, MN}$ <ul style="list-style-type: none"> ▪ $[0 \ 0 \ d_{HA} \ 0 \ 0 \ 0]$ for the triangular routing ▪ $[0 \ 0 \ 0 \ 0 \ 0 \ 0]$ for the route optimization • For $Fc_{handoff}$ <ul style="list-style-type: none"> ▪ $[0 \ 0 \ 0 \ d_{oMA} \ d_{nMA} \ 0]$ <p>d_{HA} is calculated from equation (34). d_{oMA} and d_{nMA} are calculated from equation (35)</p>
η_1	1
η_2	1
λ	50 packets per second
Forwarding during the handoff	Yes
Layer 2 handoff latency	50 msec
k_9	1
ContNode	<ul style="list-style-type: none"> • HA for the triangular routing • CN for the route optimization
RNode	Old MA
Δ	<ul style="list-style-type: none"> • For reactive mode <ul style="list-style-type: none"> ▪ 93 msec for the triangular routing ▪ 83 msec for the route optimization <p>The value results as follows: the MN performs a layer 3 handoff to notify the old MA, which starts forwarding data packets towards the MN. The layer 3 handoff takes duration of 30 msec, while the old MA is informed after 19 msec. After the completion of the layer 3 handoff, the MN informs the HA and the CN. This takes 42 and 37 msec, respectively. To consider the packets in-flight, we assume that the HA or possibly the CN has just sent a data packet before the receipt of the BU message.</p> <ul style="list-style-type: none"> • For the predictive mode <ul style="list-style-type: none"> ▪ 132 msec for the triangular routing ▪ 122 msec for the route optimization <p>We assume here that a L2-LD trigger has just been raised at the old MA after the receipt of the F-BU message. Thus, the old MA forwards</p>

	data packets to the new location of the MN during the layer 2 handoff and during the time required to inform the HA and possibly the CN after completion of the layer 2 handoff. Moreover, the packets in-flight are forwarded as well. We assume here that the HA or possibly the CN has just sent a data packet before the receipt of the BU message.
$\Delta_{Auto-Conf} + \Delta_{DAD}$	5 msec
$\Delta'_{Auto-Conf} + \Delta'_{DAD}$	10

Tab D.53: Parameters and assumptions for FMIPv6

The parameters required to calculate T_{MA-MR} employing FMIPv6 in reactive mode are listed in table D.54.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	$\bar{D}_{currentMA,BUnode}$
1	2	2	1	0	1 msec	1 msec	0	0	Δ_{DAD}	2 hops

Tab D.54: Parameters required to calculate T_{MA-MR} employing FMIPv6 in reactive mode

The parameters required to calculate t_{MA-MR} employing FMIPv6 in reactive mode are listed in table D.55.

k_3	k_4	k'_4	k'_5	k'_6	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ'	$\bar{D}_{currentMA,BUnode}$
1	1	1	1	0	1 msec	1 msec	0	0	Δ_{DAD}	2 hops

Tab D.55: Parameters required to calculate t_{MA-MR} employing FMIPv6 in reactive mode

The parameters required to calculate T'_{MA-MR} employing FMIPv6 in predictive mode are listed in table D.56.

k_1	k_2	k'_1	k'_2	k'_3	a_{MA}	a_{BUnode}	a_{InNode}	ni	γ	γ_{ext}	$\bar{D}_{currentMA,BUnode}$
1	2	2	1	0	1 msec	1 msec	0	0	$4 + \Delta_{Auto-Conf} + \Delta_{DAD}$	3 msec	2 hops

Tab D.56: Parameters required to calculate T'_{MA-MR} employing FMIPv6 in predictive mode

The parameters required to calculate t_{RNode} employing FMIPv6 in predictive mode are listed in table D.57.

k_3	k_4	k_5	k_6	k'_4	k'_5	k'_6
1	0	0	0	1	0	0
k'_7	γ'_1	a_{MA}	a_{BUnode}	a_{InNode}	a_{RNode}	ni

0	4	1 msec	1 msec	0 msec	0 msec	0
$\bar{D}_{currentMA,BUnode}$			$\bar{D}_{RNode,BUnode}$		$\bar{D}_{currentMA,RNode}$	
2			2		0	

Tab D.57: Parameters required to calculate t_{RNode} employing FMIPv6 in predictive mode

The parameters required to calculate luc_{MA-MR} employing FMIPv6 in predictive mode are listed in table D.58.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
2	3	2	3	0	10	10	0	0	$3 * \rho * \delta_s$ $+ \Delta'_{Auto-Conf} + \Delta'_{DAD}$	2 hops

Tab D.58: Parameters required to calculate luc_{MA-MR} employing FMIPv6 in predictive mode

The parameters required to calculate luc_{MA-MR} employing FMIPv6 in reactive mode are listed in table D.59.

k_1	k_2	k'_1	k'_2	k'_3	a'_{MA}	a'_{BUnode}	a'_{InNode}	ni	γ''	$\bar{D}_{currentMA,BUnode}$
2	2	2	1	0	10	10	0	0	$2 * \rho * \delta_s$ $+ \Delta'_{Auto-Conf} + \Delta'_{DAD}$	2 hops

Tab D.59: Parameters required to calculate luc_{MA-MR} employing FMIPv6 in reactive mode

After the MN completes the handoff employing FMIPv6, it has to register again using MIPv6. The parameters required to calculate the location update cost for this registration are the same as by MIPv6 with the exception that γ'' equals 0.

D.5. Example

This section delivers an example describing how the generic mathematical model parameters can be derived. Let us analyze MIFAv6 in reactive mode. As described in chapter 5, a mesh topology is applied. The handoff is controlled either by the old or new MA. Thus, the B vector, handoff vector and packet dropped vector will be $[0 \ 0 \ 0 \ R \ 0 \ 0]$, $[0 \ 0 \ 0 \ T_{MA-MR} \ 0 \ 0]$ and $[0 \ 0 \ 0 \ LP_{MA-MR} \ 0 \ 0]$, respectively. The following describes in detail how the parameters required to calculate both the handoff latency and the expected number of dropped packets per handoff can be derived.

Basic assumptions: as assumed in the analysis provided in chapter 5, the MN always sends a solicitation after the layer 2 handoff to obtain an advertisement from the new MA. In other words, Δt is 0. Afterwards, a CoA must be auto-configured and the DAD procedure has to be executed. This consumes a delay of $\Delta_{Auto-Conf} + \Delta_{DAD}$.

Handoff latency on uplink: after completion of the DAD procedure, the MN sends a **BU** message to the new MA, which sends after processing this message a **BA** to the MN. From the generic mathematical model point of view, the new MA is the **BUnode**. There are two

messages exchanged on the wireless link between the new MA and the MN. Therefore, k_1 is 2. The **BU** message is processed once in the new MA before sending the **BA** message. This means that k_1' is 1. a_{MA} is assumed to be 1 msec. Notice that this value is protocol-specific and mainly depends on the tasks the MA has to complete. Because the new MA is the *BU*node, there are no messages sent beyond the new MA and has an impact on the handoff latency on uplink. Therefore, k_2 and k_2' are equal to 0. Mobility is processed only in MAs and the HA when employing MIPv6 in reactive mode. Thus, there are no *InNodes*, which means that k_3' , n_i and a_{InNode} are equal to 0. γ is set to $4 + \Delta_{Auto-Conf} + \Delta_{DAD}$, which is the delay resulting from exchanging solicitation and advertisement messages between the new MA and the MN, auto-configuration of the new CoA and execution of the DAD procedure. The figure below explains the deriving of the discussed parameters.

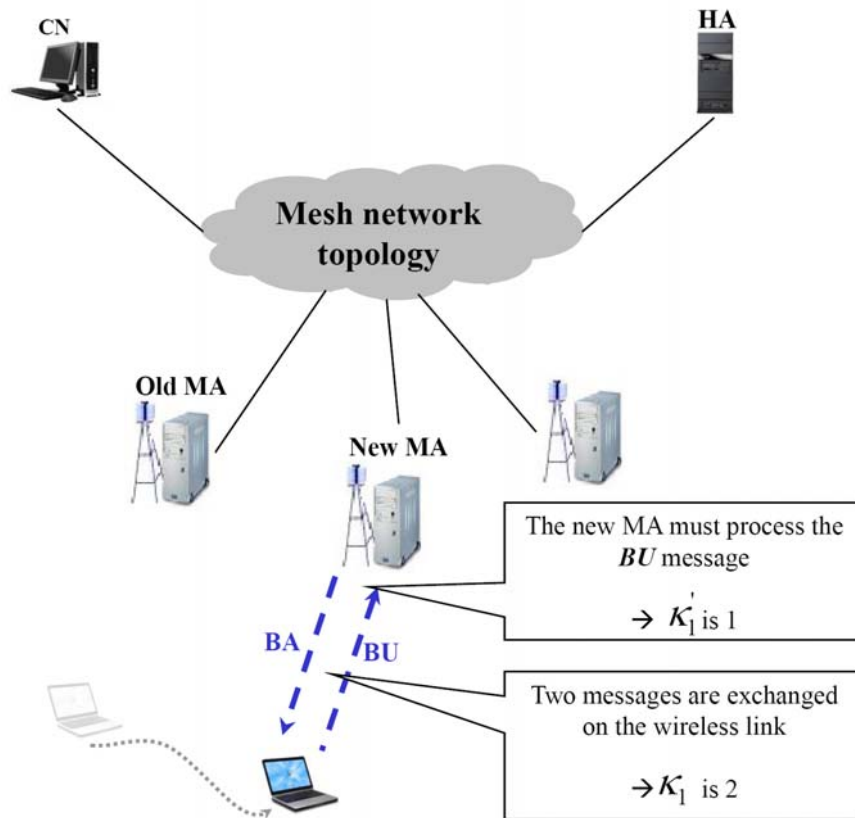


Fig D.1: Deriving the parameters required to calculate the handoff latency on uplink employing MIPv6 in reactive mode

Handoff latency on downlink: after completion of the DAD procedure, the MN sends a **BU** message to the new MA, which in turn sends after processing this message a **Hn_Not** to the old MA. After the old MA processes this message, it replies a **Hn_Ack** message and begins forwarding the MN data packets towards the new CoA. As soon as the new MA receives the **Hn_Ack** message, it declares the handoff on downlink as completed. Let us discuss this procedure from the generic mathematical model point of view. First, as the old MA is notified of the handoff, data packets will be forwarded to the new location of the MN. Thus, the old MA is the *BU*node. There is only one message exchanged on the wireless link between the new MA and the MN (the **BU** message), which means that k_1 is 1. There are two messages

exchanged on the wired link between the new and old MAs. Therefore, k_2 is equal to 2. The new MA processes the **BU** message sent from the MN before sending the **Hn_Not** message. In addition, the new MA should process the **Hn_Ack** message before declaring the handoff as completed. This means that k_1' is 2. Furthermore, upon the old MA receives the **Hn_Not** message, it processes this message and sends a **Hn_Ack** message. This means that k_2' is 1. a_{MA} is assumed to be 1 msec, while $a_{BU\text{node}}$ is supposed to be 0.5 msec. Again, these values are protocol-specific. Mobility is processed only in MAs and the HA when employing MIPv6 in reactive mode. Thus, there are no *InNodes* and k_3', ni as well as a_{InNode} are set to 0. Again, γ is set to $4 + \Delta_{Auto-Conf} + \Delta_{DAD}$. The figure below explains the deriving of the discussed parameters.

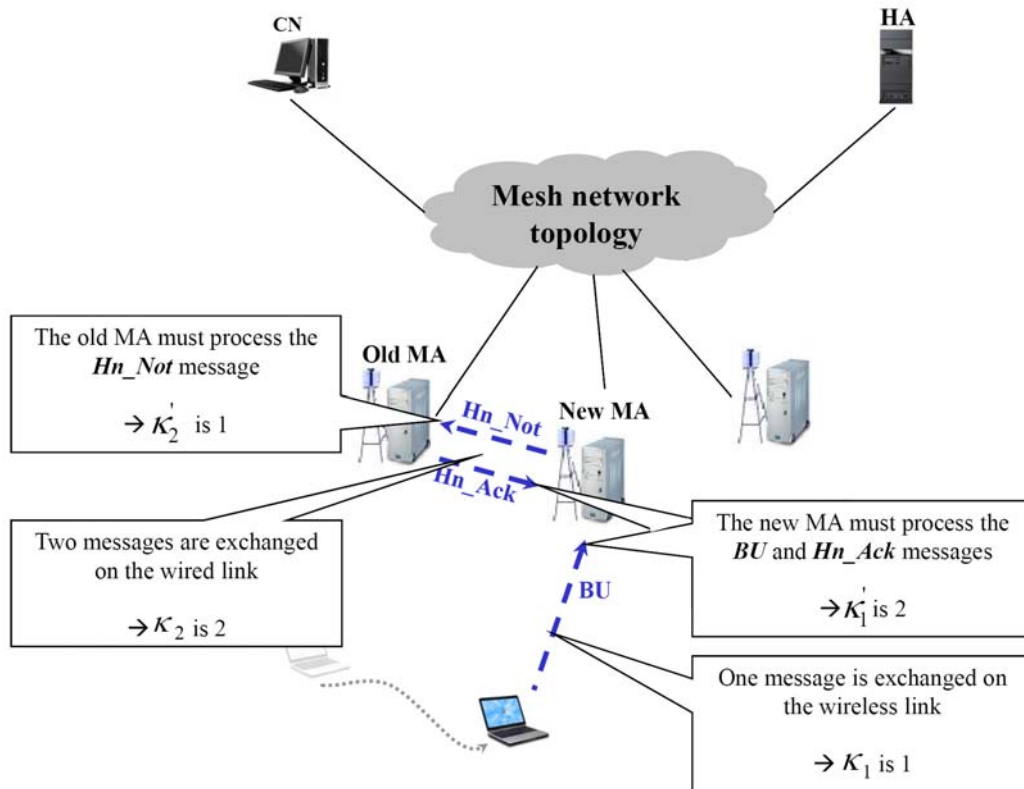


Fig D.2: Deriving the parameters required to calculate the handoff latency on downlink employing MIPv6 in reactive mode

Expected number of dropped packets per handoff on uplink: this term depends on the handoff latency on uplink discussed above. All packets the MN sends during this latency are lost.

Expected number of dropped packets per handoff on downlink: after the MN successfully completes the DAD procedure, it transmits a **BU** message to the new MA. Upon the new MA receives the **BU** message, it transmits a **Hn_Not** to the old MA, which expresses the *BUnode*. After the old MA processes the **Hn_Not** message, it is notified of the MN new CoA and no packets will be transmitted to the MN over the old wireless link. Instead, the packets will be forwarded to the new CoA. Let us discuss this procedure from the generic mathematical point of view. There is only one message exchanged on the wireless link between the new MA and

the MN (the *BU* message), which means that k_3 is 1. Notifying the old MA requires transmitting one message from the new MA (the *Hn_Ack* message). Thus, k_4 is equal to 1. Notice that the new MA processes the *BU* message sent from the MN before sending the *Hn_Not* message and the old MA processes the *Hn_Not* message once before it gets informed. Thus, k_4' and k_5' are equal to 1. For the same reasons discussed while deriving the parameters for the handoff latency on downlink, a_{MA} , $a_{BU\text{node}}$, k_3' , n_i , a_{InNode} and γ' are equal to 1 msec, 0.5 msec, 0, 0, 0 and $4 + \Delta_{Auto-Conf} + \Delta_{DAD}$, respectively. The figure below explains the deriving of the discussed parameters.

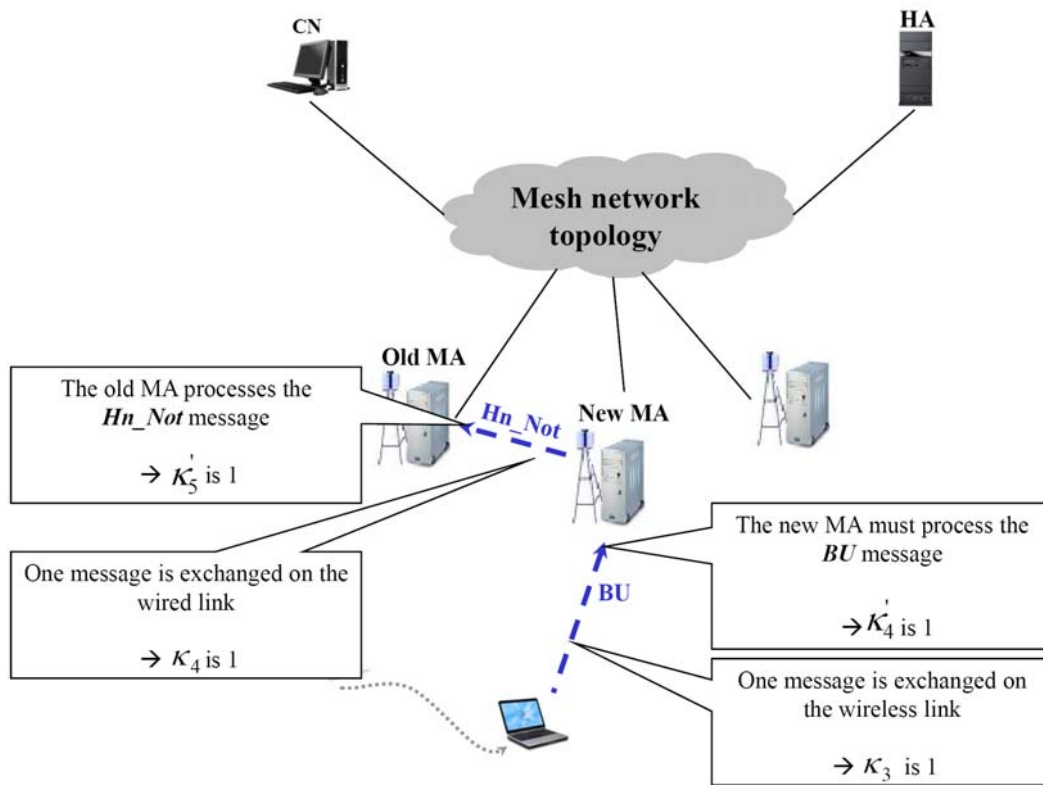


Fig D.3: Deriving the parameters required to calculate the expected number of dropped packets per handoff on downlink employing MIFAv6 in reactive mode

The parameters required to calculate the location update as well as packet delivery cost are derived in a similar way.

E. Modeling of Asymmetrical Network Topologies

This appendix discusses how asymmetrical network topologies can be considered in the generic mathematical model presented in chapter 5.

E.1. Basic Assumptions

Chapter 5 has considered symmetrical hierarchical or mesh-based network topologies. The generic model should be able, however, to consider any network topology, let us assume that there are no restrictions on the physical network topology. The access domain consists of Z MAs offering IP connectivity, \mathcal{G} MRs with optional¹ mobility support locating somewhere inside the access domain, standard IP routers and a GW interconnecting the domain with the other networks. This structure forms a logical hierarchical topology consisting of three levels of hierarchy, the GW, MRs and MAs, see figure E.1.

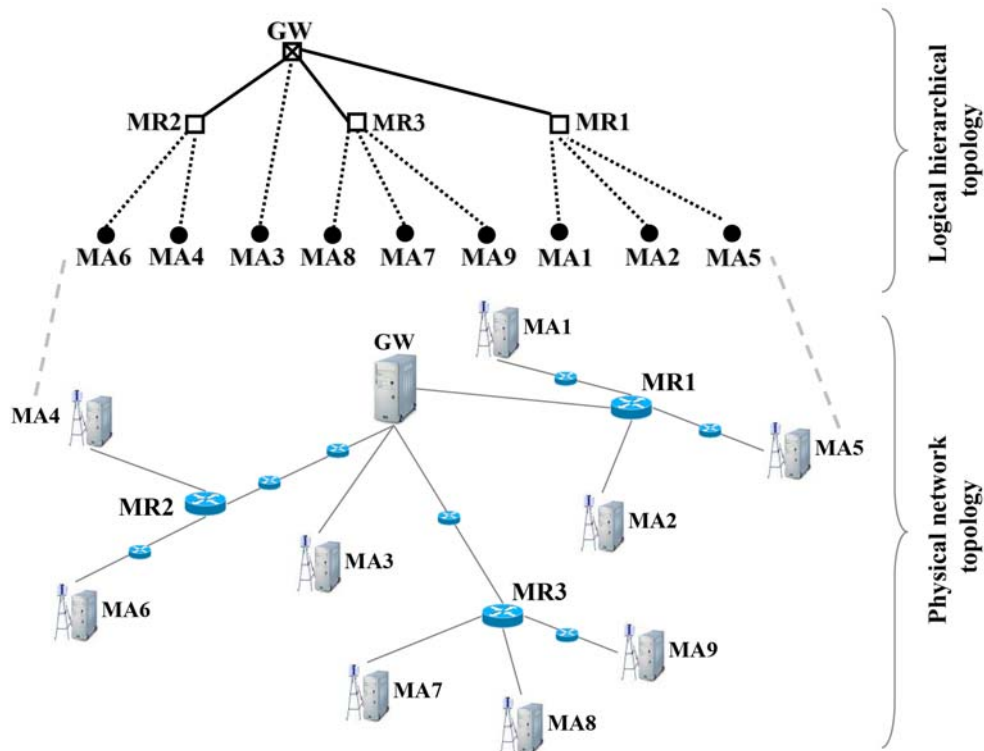


Fig E.1: An example network topology with the logical hierarchical structure for $Z = 9$ and $\mathcal{G} = 3$

E.2. Modeling of Movements Patterns

As described in section 5.3, the MN does not move from a certain MA to one of the other $(Z - 1)$ MAs with an equal probability. In reality, MNs restrict their movements to one of the MAs locating in the geographical neighborhood regardless of the used network

¹ In principle, these nodes will be used for the protocols requiring a second hierarchy level with mobility support between MAs and the GW. For others, which do not require hierarchy, MRs will be considered as standard IP-routers.

topology. Depending on the model presented in section 5.3, the probabilities that a MN is attached to each MA in the steady state of the system can be derived.

Let us assume the neighbor graph presented in figure E.2, where P_i denotes the probability that a MN is attached to MA_i and $P_{i,j}$ stands for the probability that the MN moves from MA_i to MA_j .

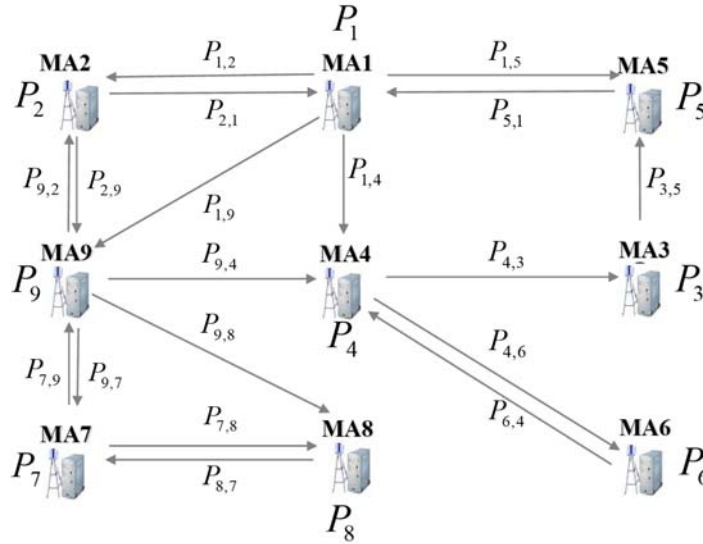


Fig E.2: An example neighbor graph with the movement probabilities for a domain containing 9 MAs

Another time, supposing $P(MR_n)$ is the probability that the MN moves between the MAs controlled by MR_n . $P(MR_n)$ can be calculated from the following equation.

$$P(MR_n) = \sum_i \sum_j P_i * P_{i,j} \text{ where } i, j \in I(MR_n) \text{ and } i \neq j \quad (46)$$

$I(MR_n)$ presents the set of MAs controlled by MR_n . By observing all movements inside the domain, R can be derived from equation (47), where R is the probability that the crossover router will be one of the MRs while moving inside the domain.

$$R = \sum_{n=1}^9 P(MR_n) \quad (47)$$

In a similar way the probability that the GW will be the crossover router while moving inside the domain can be derived.

$$G = \sum_i \sum_j P_i * P_{i,j} \text{ where } i, j \in Y(MR) \text{ and } i \neq j \quad (48)$$

where $Y(MR)$ is the set of MAs, from which the MN can move to another MA controlled by another MR inside the domain.

E.3. Modeling of Network Topologies

As mentioned in chapter 5, the distances between MAs and MRs, the GW or the old MA should be considered as average values when deploying asymmetrical topologies. However, the average values should be calculated taking the applied mobility scenario into account. Let us assume that the network topology is structured as in figure E.1 and the mobility scenario provided in section 2 is applied. The average distance between MAs and the GW ($\bar{D}_{MA,GW}$) can be calculated from the following equation, where $D_{MA_j,GW}$ is the distance between MA_j and the GW.

$$\bar{D}_{MA,GW} = \sum_i^Z \sum_j^Z P_i * P_{i,j} * D_{MA_j,GW} \quad \text{where } i \neq j \quad (49)$$

The average distance between the new MA and the old one ($\bar{D}_{oldMA,newMA}$) can be calculated from the following equation, where D_{MA_j,MA_i} is the distance between MA_j and MA_i .

$$\bar{D}_{oldMA,newMA} = \sum_i^Z \sum_j^Z P_i * P_{i,j} * D_{MA_j,MA_i} \quad \text{where } i \neq j \quad (50)$$

In order to calculate the average distance between MRs and MAs controlled by them, each MR should be considered alone. Let us consider a MR_n and the set of MAs controlled by it ($I(MR_n)$) see figure E.3.

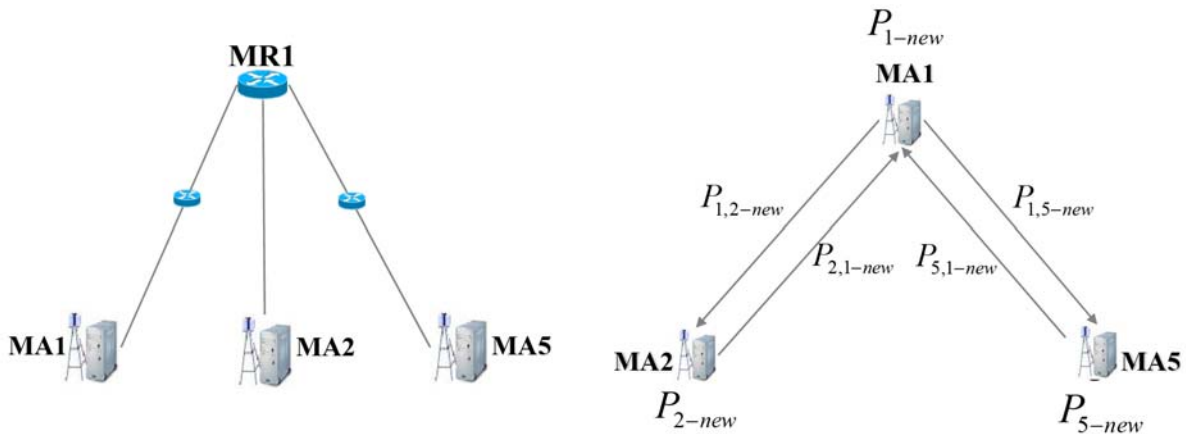


Fig E.3: An example sub-domain containing a MR and the MAs controlled by it in addition to the corresponding neighbor graph

The average distance between the MAs of the set $I(MR_n)$ and MR_n can be calculated from the equation below.

$$\bar{D}_{MA,MR_n} = \sum_i \sum_j P_{i-new} * P_{i,j-new} * D_{MA_j,MR_n} \quad \text{where } i, j \in I(MR_n) \text{ and } i \neq j \quad (51)$$

P_{i-new} and $P_{i,j-new}$ are calculated from the following equations.

$$P_{i-new} = \frac{P_i * 100}{\sum_k P_k} \text{ where } k \in I(MR_n) \quad (52)$$

$$P_{i,j-new} = \frac{P_{i,j} * 100}{\sum_k P_{i,k}} \text{ where } k \in I(MR_n) \text{ and } i \neq k \quad (53)$$

The above equations can be explained as follows. Each MR will be considered with all MAs controlled by it as a single network forming a single sub-domain. The MN is considered to move only inside the sub-domain. Therefore, the probabilities that a MN will be in the range of each MA present in this sub-domain and the probabilities of moving between these MAs should be changed accordingly.

After calculating the average distance between each MR present in the domain and the MAs controlling by it, the average distance between MAs and any MR in the domain can be calculated from the following equation.

$$\bar{D}_{MA,MR} = \sum_l^g P(MR_l) * \bar{D}_{MA,MR_l} \quad (54)$$

The average distances derived in this section should be used, after that, to calculate the average handoff latency as well as the expected average number of dropped packets.

F. Graphical Tools Supporting the Generic Mathematical Model

This appendix describes the tools developed to simplify the analysis of mobility management protocols by means of the generic mathematical model. Mainly, the tools aim to atomize the parameterization of mobility scenarios, network topologies and mobility protocols.

F.1. Mobility Scenarios Generator (MSGen)

MSGen is a powerful GUI written in Delphi and used to produce mobility scenarios that are not pure random, see figure F.1.

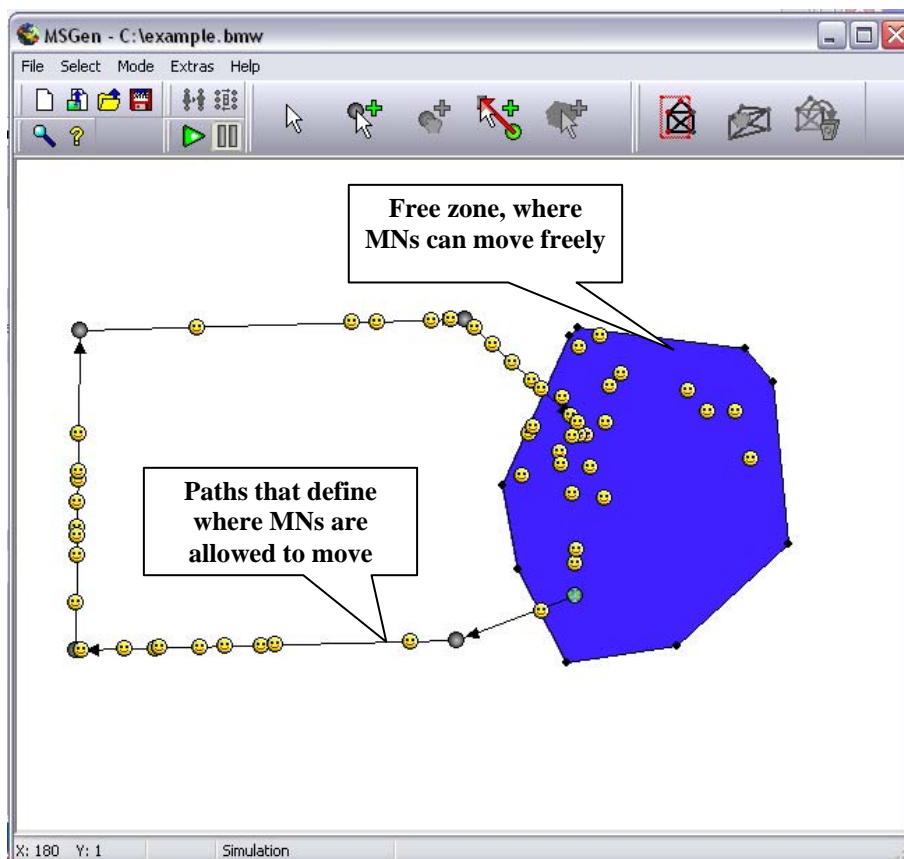


Fig F.1: GUI of MSGen containing an example mobility scenario

This tool enables uploading geographical maps to help in defining real mobility scenarios. The scenarios are constructed by means of nodes, edges and regions, see figure F.2. The nodes present routing points from the MN point of view. As the MN reaches a node, it should take a routing decision resulting in selecting either one of the possible edges to move one or one of the possible regions to move inside. The edge is defined as a path connecting two nodes. The region is a zone where MNs can move freely inside. It is used mainly to enable simulation of random walk scenarios, also ad hoc scenarios. Each region has to have at least an entrance node, from which the MN can move into the domain. The region may have an egress node enabling the MN to go outside the region.

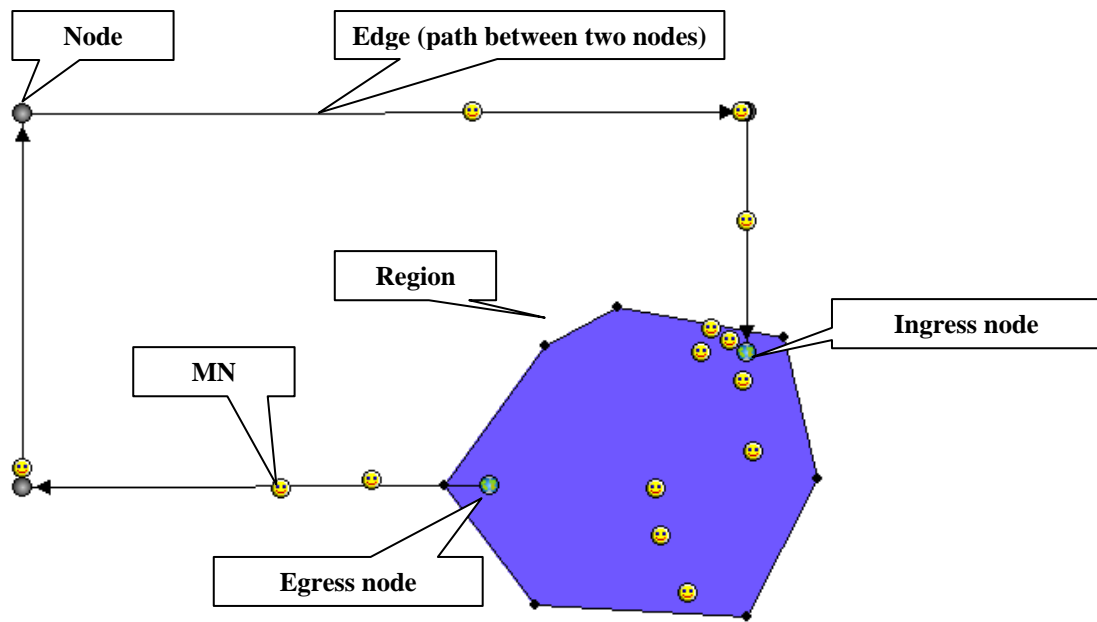


Fig F.2: Structure of mobility scenarios

MSGen enables the parameterization of mobility scenarios by defining the parameters of their components, e.g. the user can define the speed of MNs on a certain edge, probabilities of taking routing decisions from a certain node, to which many edges are connected.

F.1.1 System Requirements

MSGen requires supporting the following.

Operating system	Windows 98 or higher
CPU	Pentium I with 133 MHz or more
RAM	32 MB or more
Graphics card	256 colors or more

Tab F.1: Minimum requirements of MSGen

Operating system	Windows 2000, XP or Windows Vista – Linux 2.6 with wine >=1.0.0
CPU	Pentium II with 700 MHz or more
RAM	256 MB
Graphics card	16 Mio colors at 800x600 pixels

Tab F.2: Recommended requirements of MSGen

F.1.2. Common Interactions

Similar to most Windows applications, all important functions can be reached via a toolbar, standard and popup menus or keyboard buttons.

The menus of MSGen are operable in a classical Windows-way and contain the most common items of Windows applications, e.g. selection of the item “New” from the menu “File” results in creating an empty worksheet, etc. The menubar of MSGen is shown in figure F.3.

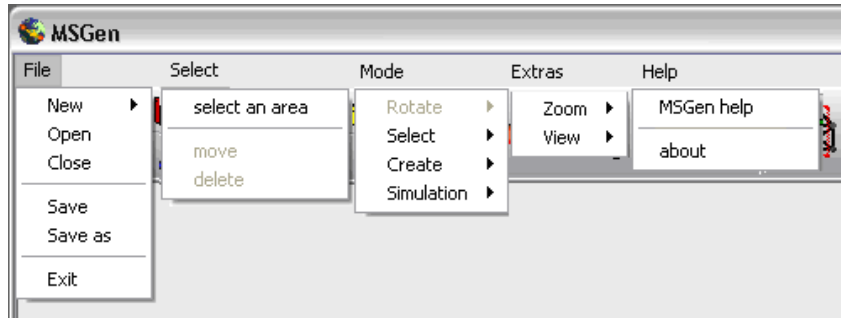









Fig F.3: MSGen menubar

The toolbar is equipped with the same functions included in the menubar, see figure F.4.



Fig F.4: MSGen toolbar

Most important buttons are presented in the table below.

 and 	These buttons create an empty worksheet, with or without a background map.
 and 	These buttons are used to load and save mobility scenarios.
	This button changes the current working mode to “select nodes”. The user should click on this button to be able to select nodes in the defined scenario.
	This button is used to connect two nodes with each other to form a path with a certain direction.
	This button is used to place, modify or erase regions.

Tab F.3: Most important buttons present in the toolbar of MSGen

F.1.3. Components of Mobility Scenarios

F.1.3.1. Nodes

Nodes are the most important components. They present routing points from the MN point of view. Upon the MN reaches a node, a routing decision should be taken and a new path is selected as a result. Each node has many properties can be accessed by means of a node’s properties dialog. This dialog can be accessed by a right mouse click on the node and selecting the item “properties” from the popup menu. This dialog contains three tabs, namely “Routing”, “Routing point” and “Region’s attributes”. The tab “Routing” enables determining the probabilities of each outgoing edge, see figure F.5.

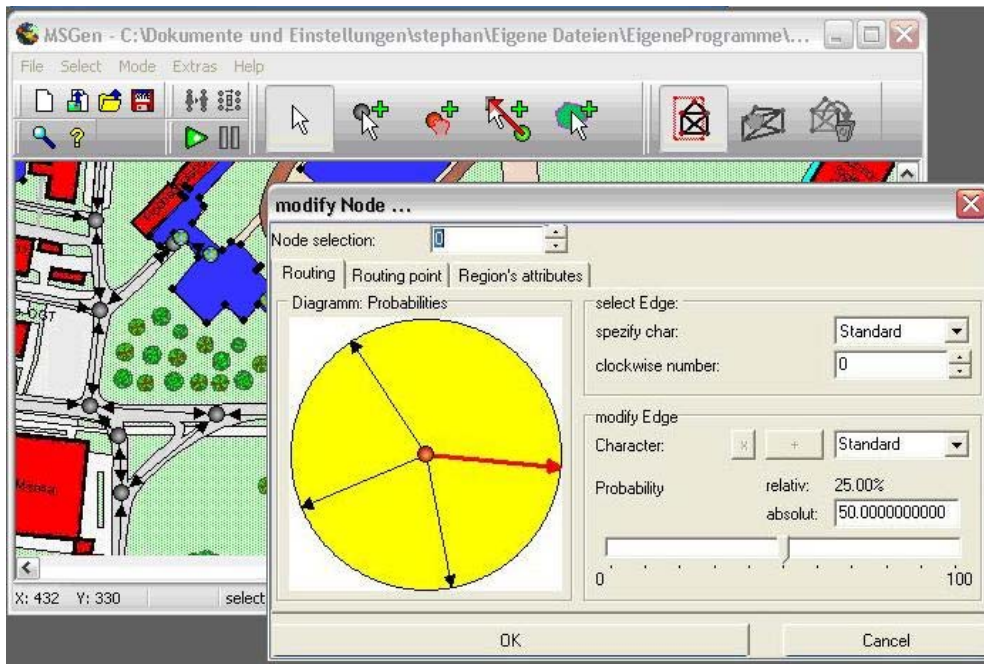


Fig F.5: Node's properties dialog - **Routing** tab

The “Routing point” tab enables adjusting the distance between the routing point (the point, at which MNs change their direction) and the node itself in pixels. In addition, it enables allocating a new common routing point for all incoming connection, see figure F.6.

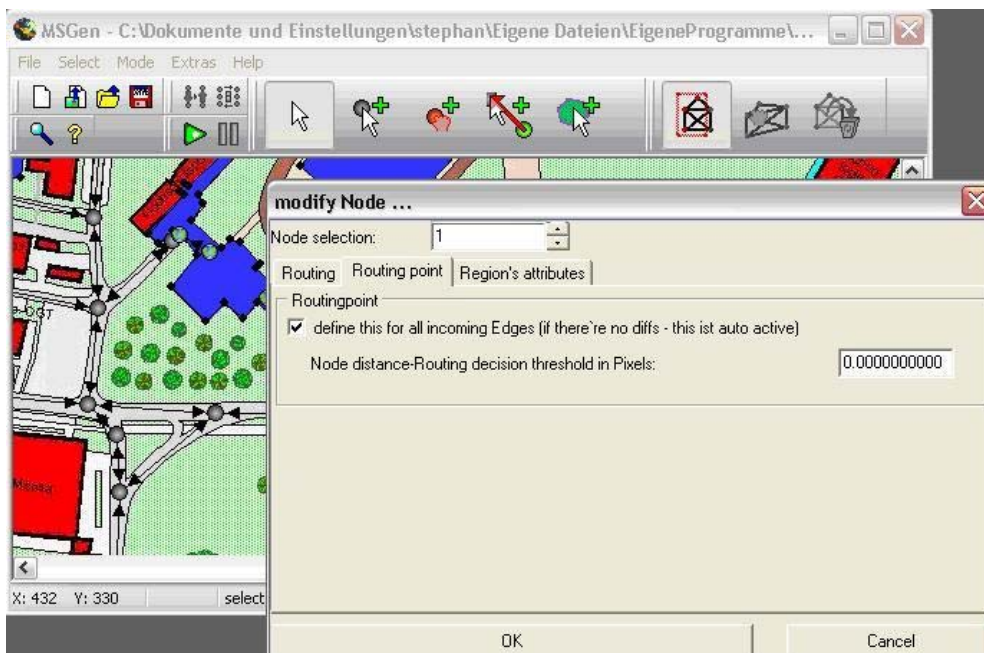


Fig F.6: Node's properties dialog - **Routing point** tab

The “Region's attributes” tab enables accessing the region the node is bound to in case the node is an ingress or egress node. In addition, the user can define the attraction radius area of the node belonging to a region. The attraction area is simply an area around the node bound to the region. This area extracts the MN coming into this area to the node to go either inside or outside the region, see figure F.7.



Fig F.7: Node's properties dialog - **Region's attributes** tab

F.1.3.2. Edges

An edge presents a path between two nodes, where MNs can move on. The properties of the edge can be accessed via an edge dialog that can be accessed by a right mouse click on the edge and selecting the item "properties" from the popup menu. This dialog contains two tabs, namely "Movement" and "Routing" tab. By means of the "Movement" tab, the user can adjust the speed of MNs moving on the edge. In addition, the user can define the variation of movements on this edge. The variation represents meanly the width of the path, see figure F.8.

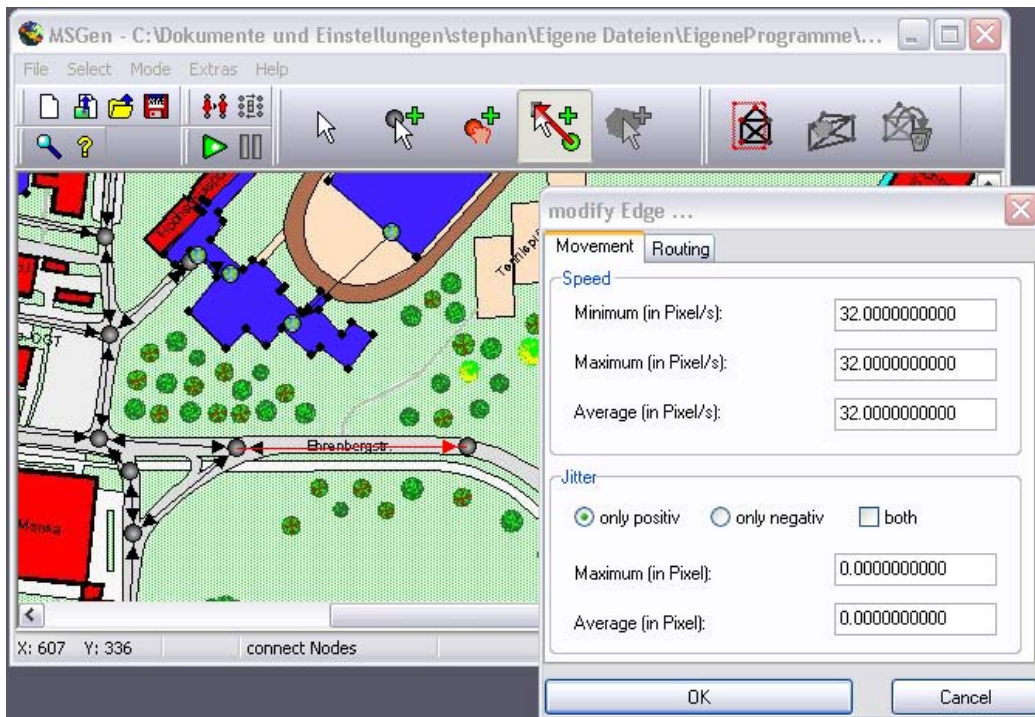


Fig F.8: Edge's properties dialog - **Movement** tab

The “Routing” tab enables setting up many properties for the connection, e.g. at which distance from the end node of the edge the MN should start taking a routing decision, the property to select the edge by the MN, etc. This tab is shown in figure F.9.

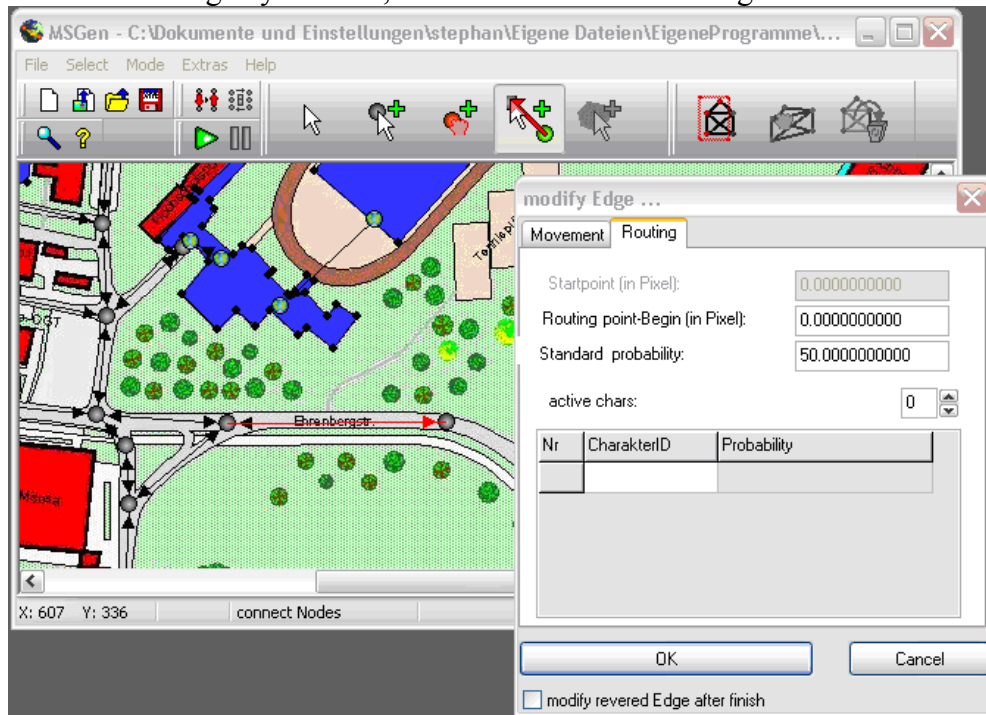


Fig F.9: Edge’s properties dialog - **Routing** tab

F.1.3.3. *Regions*

Regions are areas of the workspace where MNs move freely, i.e. in ad hoc scenarios. As mentioned previously, each region should have at least an ingress node. An egress node may be contained too. Ingress and egress nodes should be bound to the region. MSGen enables that regions overlap.

F.2. Network Generator (NetGen)

NetGen is a GUI written in Delphi and used to generate network topologies consisting of APs, MAs, MRs, GWs, routers, PCs, CNs and MNs. The user uploads the mobility scenario generated previously using MSGen. After that, a network topology is structured. The user has the ability to specify different parameters for network elements, such as link bandwidth, transmission delay, transmission cost, etc. The user can, after that, automatically generate the parameters of the mobility scenario and network topology. These parameters are used in the generic mathematical model, after that, to analyze mobility management protocols employed in the built network topology using the selected mobility scenario. figure F.10 shows the GUI of NetGen containing an example network topology.

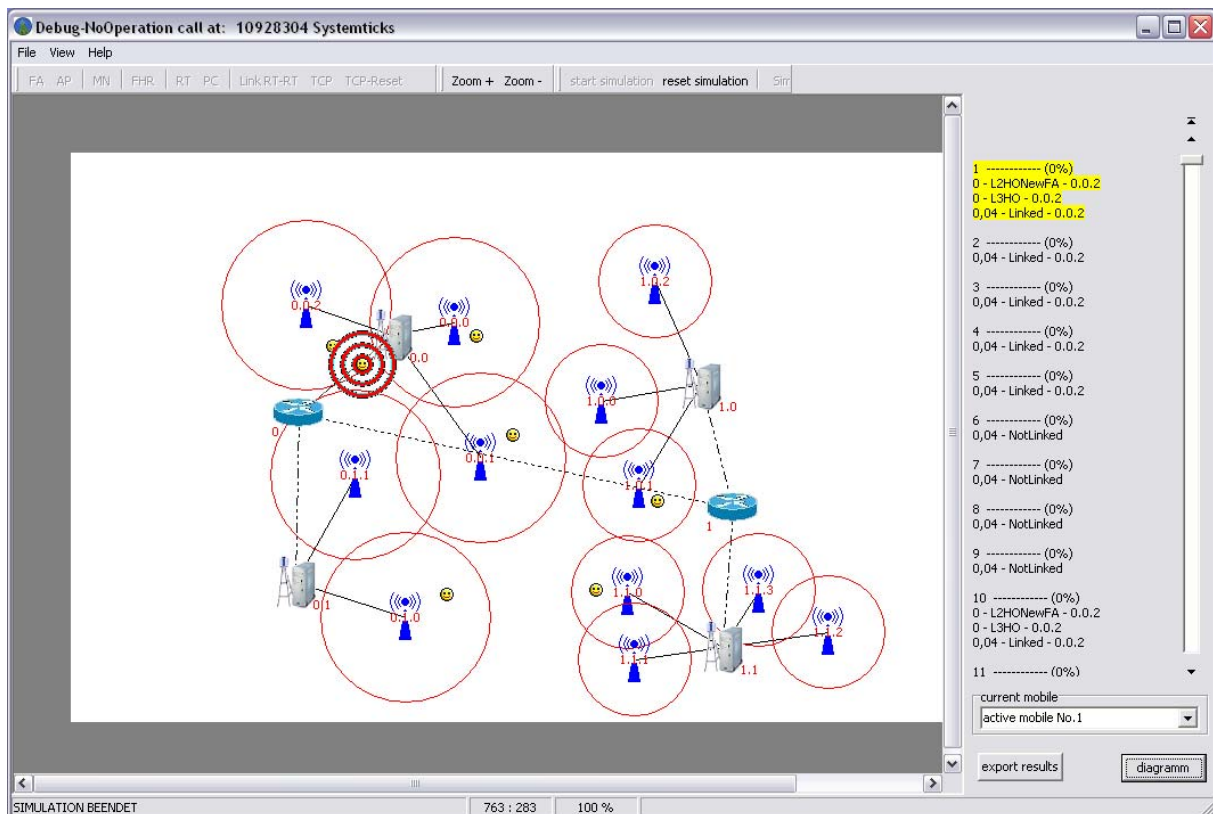


Fig F.10: GUI of NetGen containing an example network topology

F.2.1. System Requirements

NetGen requires supporting the following.

Operating system	Windows 2000 or higher
CPU	Pentium II with 700 MHz or more
RAM	256 MB or more
Graphics card	256 colors or more

Tab F.4: Minimum requirements of NetGen

Operating system	Windows 2000, XP or Windows Vista
CPU	Pentium IV with 1.5 GHz or better
RAM	1 GB (2 GB for Windows vista)
Graphics card	16 Mio colors at 800x600 Pixels

Tab F.5: Recommended requirements of NetGen

F.2.2. Common Interactions

NetGen has a classical Windows user interface. Most functions can be accessed via the menubar of NetGen and a toolbar. The menubar and the toolbar are shown in figure F.11 and figure F.12.

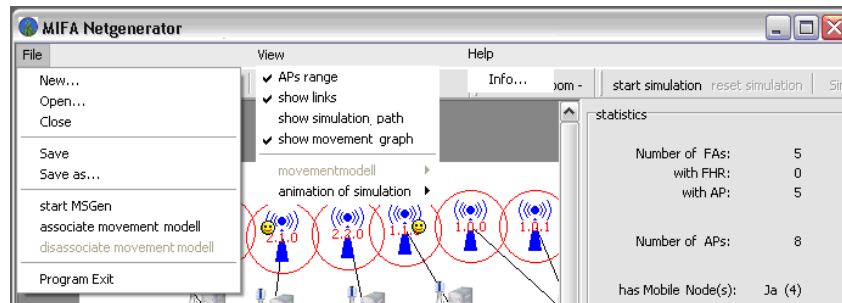


Fig F.11: NetGen menubar



Fig F.12: NetGen toolbar

The tasks of the toolbar buttons are explained below.

FA	This button adds new FAs to the network topology. Notice that FAs are bound to routers in the network. FAs correspond to MAs in the generic mathematical model.
AP	This button adds new APs to the network topology. Notice that APs are bounded to FAs in the network.
MN	This button adds new MNs.
RT	This button adds routers to the network topology. Each router has a router mode that defines the roll of the router. The roll can be a normal router, a GW, a MR or a HA.
PC	This button adds PCs to the network. PCs are connected to routers and have two rolls, a normal PC or a CN.
Link RT-RT	This button connects/disconnects two routers. The link between each two routers has a transmission delay and cost that can be determined using this button too.
start simulation	This button starts the simulation. The simulation comprises MNs moving according to a certain mobility scenario. The tool generates the parameters describing the used mobility scenario and network topology. These parameters will be used, after tat, in the generic mathematical model.

Tab F.6: Most important buttons present in the toolbar of NetGen

F.2.3. Network Elements

F.2.3.1. AP

APs are layer 2 entities from NetGen point of view. Each AP has many adjustable parameters, e.g. the range of the AP, the used access technology, etc. As mentioned previously, each AP is connected to a FA. The link to the FA has a certain transmission delay and cost, which can be set using the AP properties dialog. Figure F.13 presents the symbol used for the AP¹ and the dialog where the user can adjust AP parameters.

¹ Notice that the number written on the AP is, in principle, the name of the AP. 1.0.2 means that this AP has the number 2 and is connected to a FA numbered with 0. The FA is connected in turn to a router numbered with 1.

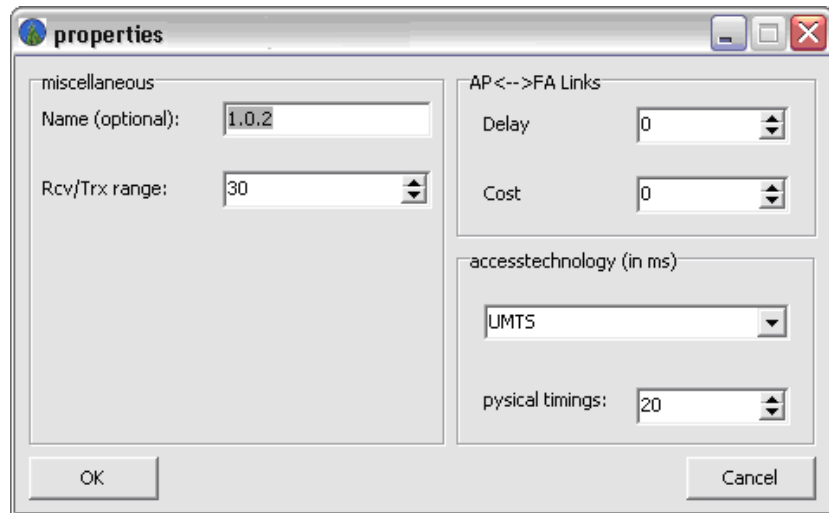


Fig F.13: AP symbol and properties dialog

F.2.3.2. FA

FAs are a layer 3 entity offering IP connectivity. Each FA controls a set of APs and is connected to a router. The user can adjust the properties of FAs by a dialog accessed by a right mouse click on the FA and selecting the “properties” item. The user can define the delay and the transmission cost on the link connecting the FA to the router. In addition, the processing delay and cost required for processing a control message in the FA can be defined too. Figure F.14 presents the symbol used for the FA¹ and the dialog where the user can adjust FA parameters. Notice that FAs correspond to MAs in the generic mathematical model.

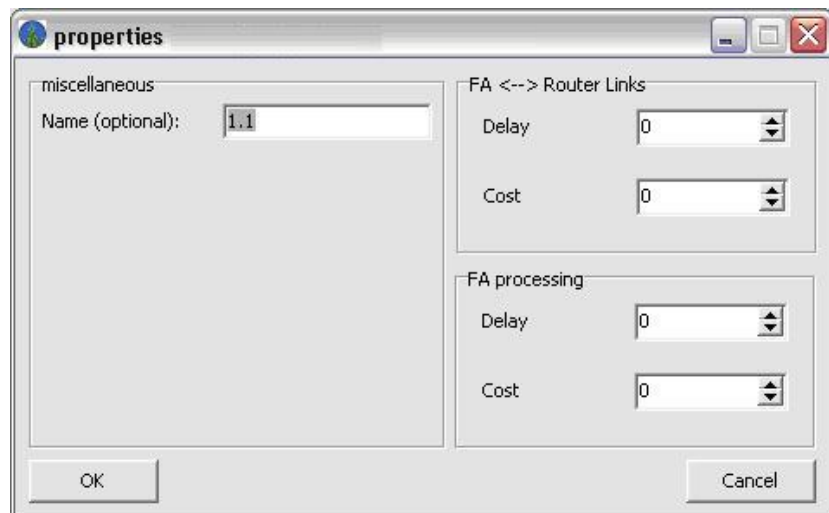


Fig F.14: FA symbol and properties dialog

F.2.3.3. Router

NetGen enables adding IP-based routers to the topology. As mentioned previously, each router is assigned a roll defining its task in the network. The roll may be a normal IP router, a GW, a MR or a HA. Notice that this roll matches the terminology of the generic mathematical

¹ Notice that the number written on the FA is, in principle, the name of the FA. 1.1 means that this FA has the number 1 and is connected to a router numbered with 1.

model. The aim is to enable generating the parameters required for the generic mathematical model. Figure F.15 presents the symbol used for each roll.



Fig F.15: Symbols used for routers

Lots of parameters can be defined for each router, e.g. IP address, number of ports, etc. The user can adjust these parameters by means of a dialog accessed by a right mouse click on the router and selecting the “properties” item. Figure F.16 presents the dialog where the user can adjust the router’s parameters.

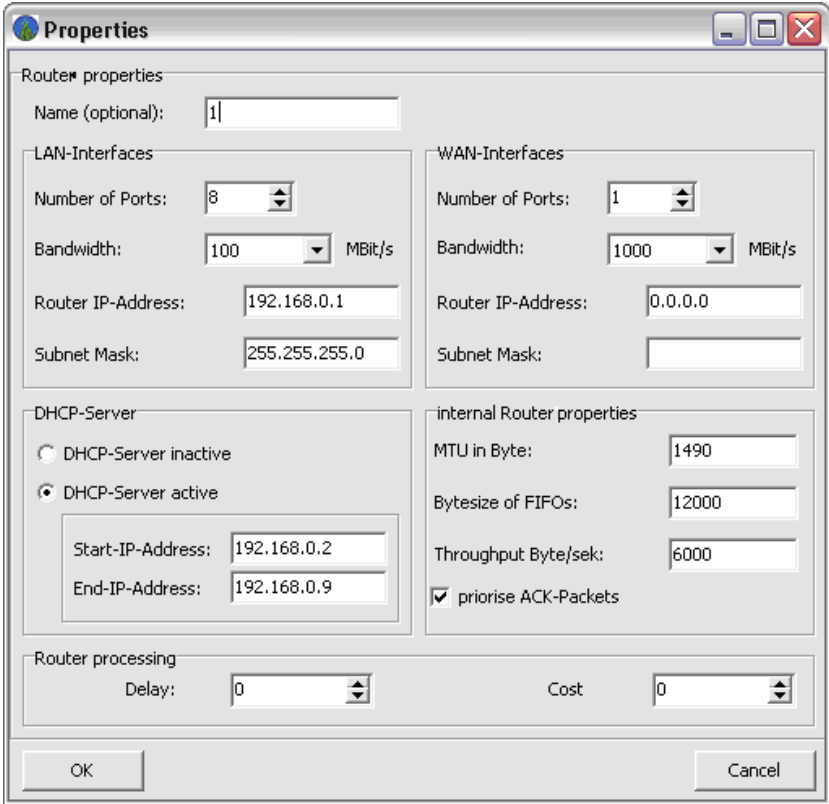


Fig F.16: Router’s properties dialog

F.2.3.4. PC

The PC is a normal computer that sends or receives data. As known, the generic mathematical model requires defining a CN in the network topology. Therefore, two rolls are defined for the PC, namely a standard PC or a CN, see figure F.17.

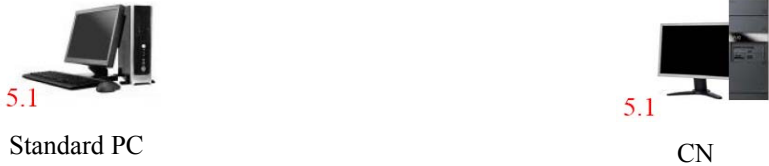


Fig F.17: Symbols used for PCs

The PC can be parameterized via a properties' dialog too, see figure F.18. Clearly, PCs are connected to routers.

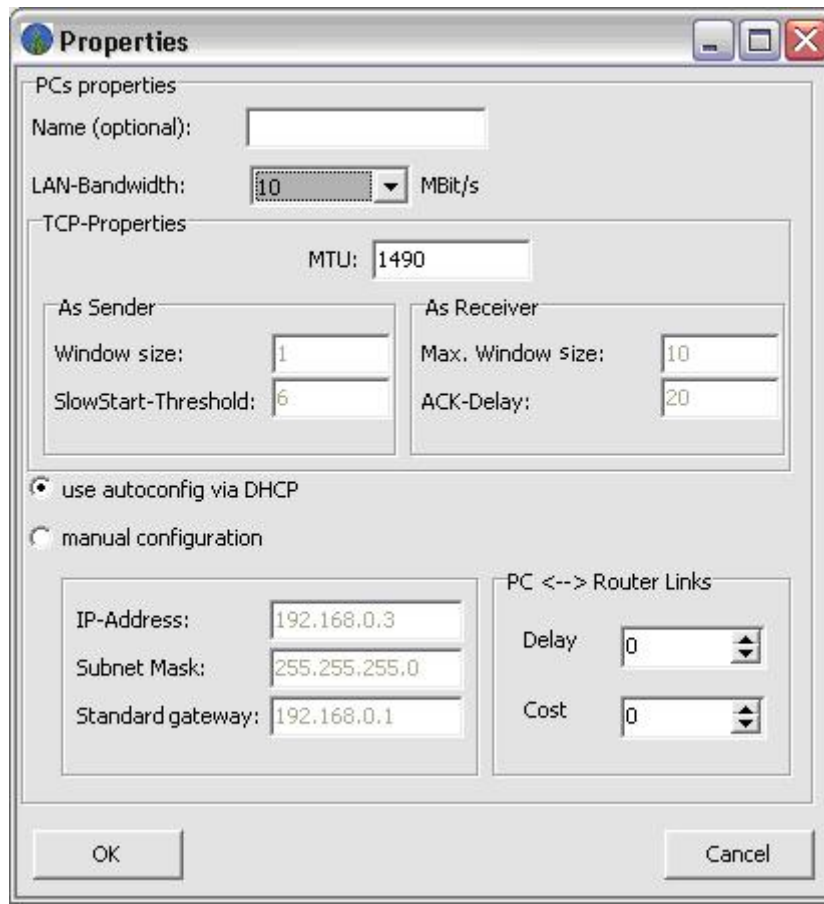


Fig F.18: PC properties dialog

F.3. Protocol Designer (ProtDes)

The main idea behind ProtDes is to develop a graphical tool that enables the parameterization of mobility management protocols by defining their messages sequence charts graphically. First, the parameters of the applied mobility scenario and network topology are imported from NetGen. After that, the user determines how control messages of the studied mobility management protocol will be exchanged between the network nodes (MAs, MR, GW, HA and anchor point). Afterwards, ProtDes generates a list of all parameters required to analyze the studied mobility management protocol and saves this list in a text file. Figure F.19 shows the GUI of ProtDes containing an example messages sequence chart for a mobility management protocol under study (MIPv4 in this example), while figure F.20 presents an example for a file for generated parameters.

Parameters files use a simple semantic. The parameter name should be preceded by “#”, where the value of the parameter should be written in a new line direct under the parameter name. Comments can be expressed in the way presented in figure F.20.

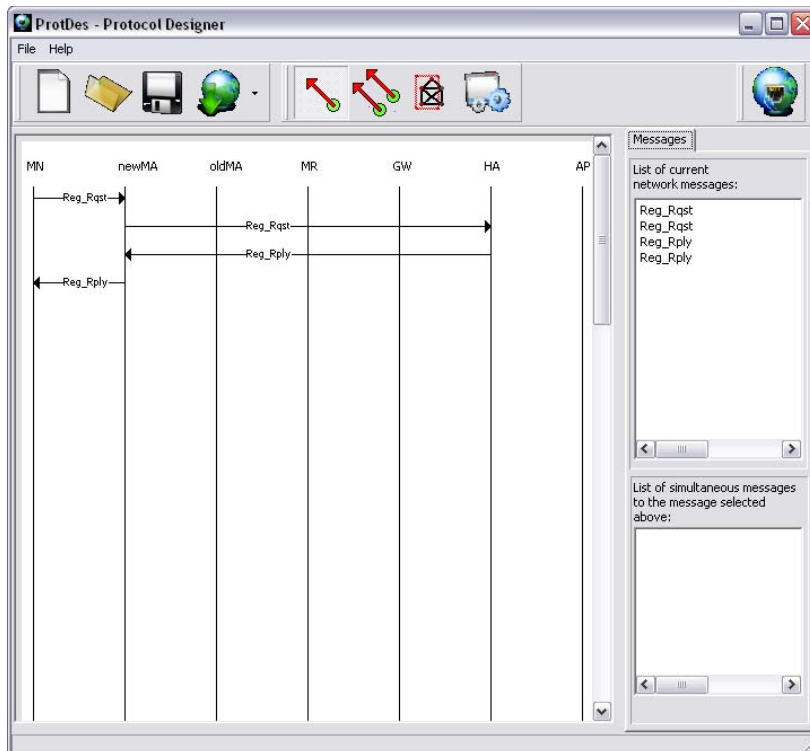


Fig F.19: GUI of ProtDes containing an example messages sequence chart for MIPv4

```

MIP Downlink.txt - Editor
Datei Bearbeiten Format Ansicht ?
Hierarchical
//*****
//*****
//Timer settings
//*****
//*****
#Timer set
Automatically
#Timer use
No
#Timer value
0
#number of retransmissions
0
//*****
//*****
//Mobility model
//*****
//*****
#R
0.75
#G
0.25
//*****
//*****
//Handoff latency
//*****
//*****
    
```

Callouts in the image point to:

- 'Comments' pointing to the //Timer settings section.
- 'Parameter name' pointing to #R.
- 'Parameter value' pointing to 0.75.

Fig F.20: Example of a parameter file

F.3.1. System Requirements

The requirements of ProtDes are the same as MSGen.

F.3.2. Common Interactions

ProtDes has a classical Windows user interface. Most functions can be accessed via a menubar and a toolbar, see figure F.21 and figure F.22.

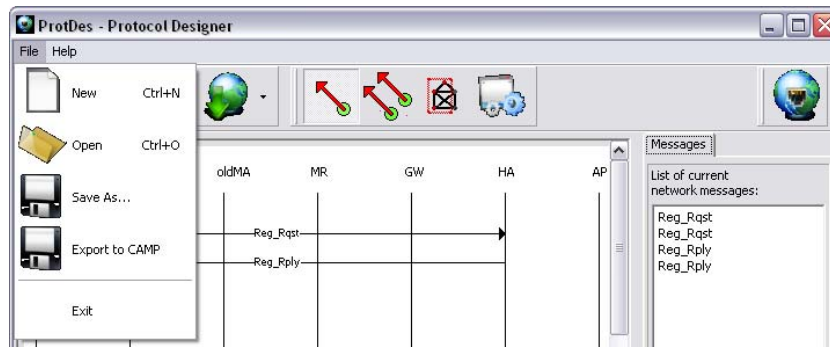











Fig F.21: ProtDes menubar



Fig F.22: ProtDes toolbar

The tasks of the toolbar's buttons are explained below.

	This button creates a new worksheet where the messages sequence chart of the studied mobility management protocol can be defined.
	This button opens an existing messages sequence chart.
	This button saves the current worksheet.
 <ul style="list-style-type: none"> Import Default Parameters Import Mobility Model Import Network Topology Import Protocol from LaTeX 	This button imports mobility, network and default parameters. In addition, a messages sequence chart predefined in Latex can be imported and further processed in ProtDes.
	This button is used to draw a control message.
	This button is used to draw a set of messages. The messages of this set are similar to the messages drawn using the previous button. However, these messages are distributed simultaneously, e.g. a certain protocol broadcasts a message to all neighbors, etc.
	This button is used to switch between the panels on the right side of the main window.
	This button is used to access the parameters used in the analysis.
	This button shows a dialog presenting some help information.

Tab F.7: Most important buttons present in the toolbar of ProtDes

The parameters of the mobility scenario as well as network topology can be accessed by the default parameters dialog. The user has the ability to change these parameters without requiring going back to MSGen or NetGen and importing the parameters again. In addition to this, some other parameters that can not be defined using the other tools, e.g. the number of

intermediate nodes, processing delays in these nodes, etc, can be defined using this dialog as well. A screenshot of this dialog is presented in figure F.23.

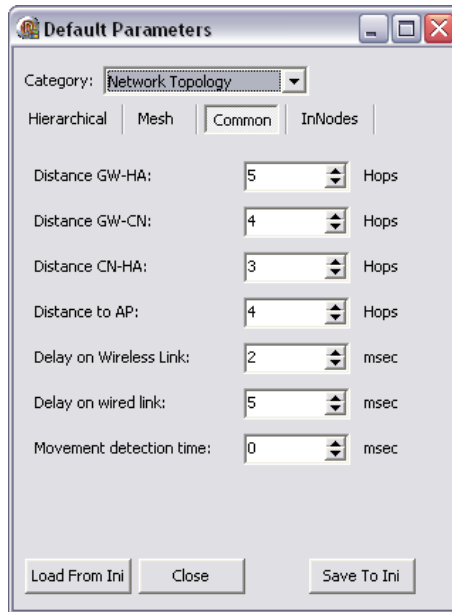


Fig F.23: Default parameters dialog

F.4. Comparative Analysis of Mobility Management Protocols (CAMP)

CAMP is a flexible tool for analysis of mobility management protocols. It has a flexible GUI, shown in figure F.24. This tool analyzes the performance and the cost of mobility management protocols. CAMP is designed in a way that enables a simple integration of new protocols and algorithms, which ensures the extensibility of this tool.

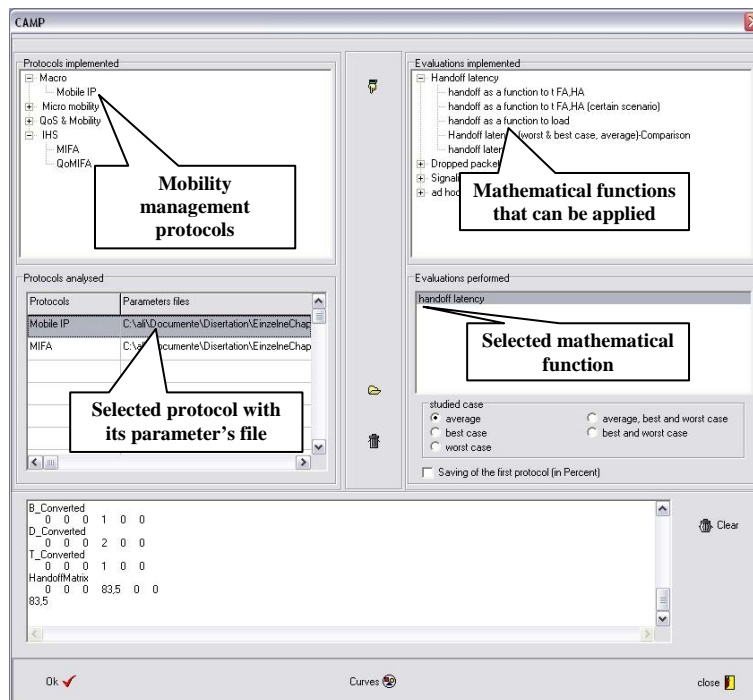


Fig F.24: GUI of CAMP

The user should first define the protocols he wants to analyze. After that, each protocol is linked to a parameters' file generated previously using ProtDes. The user has to select then the metric he wants to calculate. CAMP reads the parameters, selects the adequate libraries and performs the desired analysis. The results are saved in a stream file. These results can be shown in CAMP itself, as shown in figure F.25, or can be exported to Microsoft Excel for further analysis.

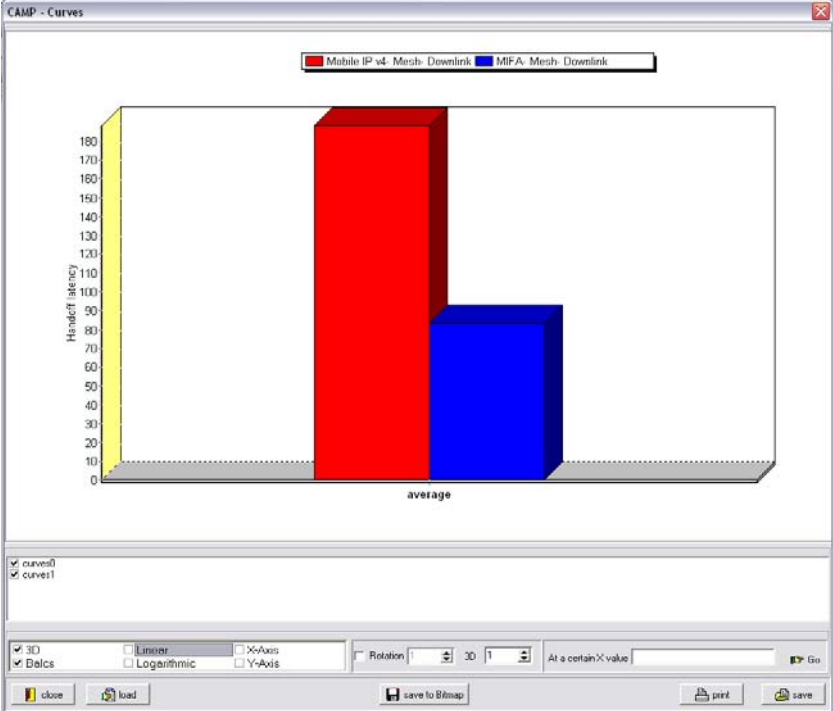


Fig F.25: Display unit of CAMP

Additional to the GUI, CAMP consists mainly of a libraries package and a configuration database, the structure of CAMP is plotted in figure F.26. All configuration parameters of CAMP are stored in the configuration database, e.g. GUI parameters, supported protocols, implemented functions, etc. The libraries' package contains the libraries required for the mathematical analysis. G-libraries include libraries implementing general functions, e.g. queuing models, etc. P-libraries are protocol-specific. Input files contain the parameters of the studied protocols. As mentioned above, these parameters are written in text files and imported from ProtDes.

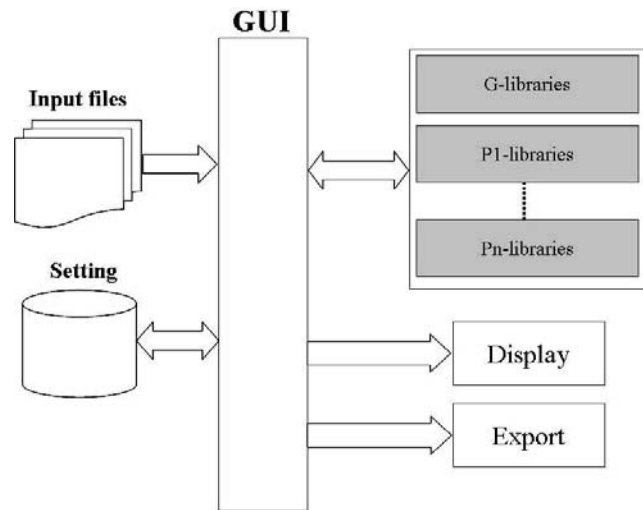


Fig F.26: Structure of CAMP

G. Impact of Network Topology

This appendix provides some extra simulation results describing the impact of network topology on the performance of HAWAII, MIP and MIFA. The scenarios used in this analysis are the same presented in section 6.4.

G.1. Handoff Latency

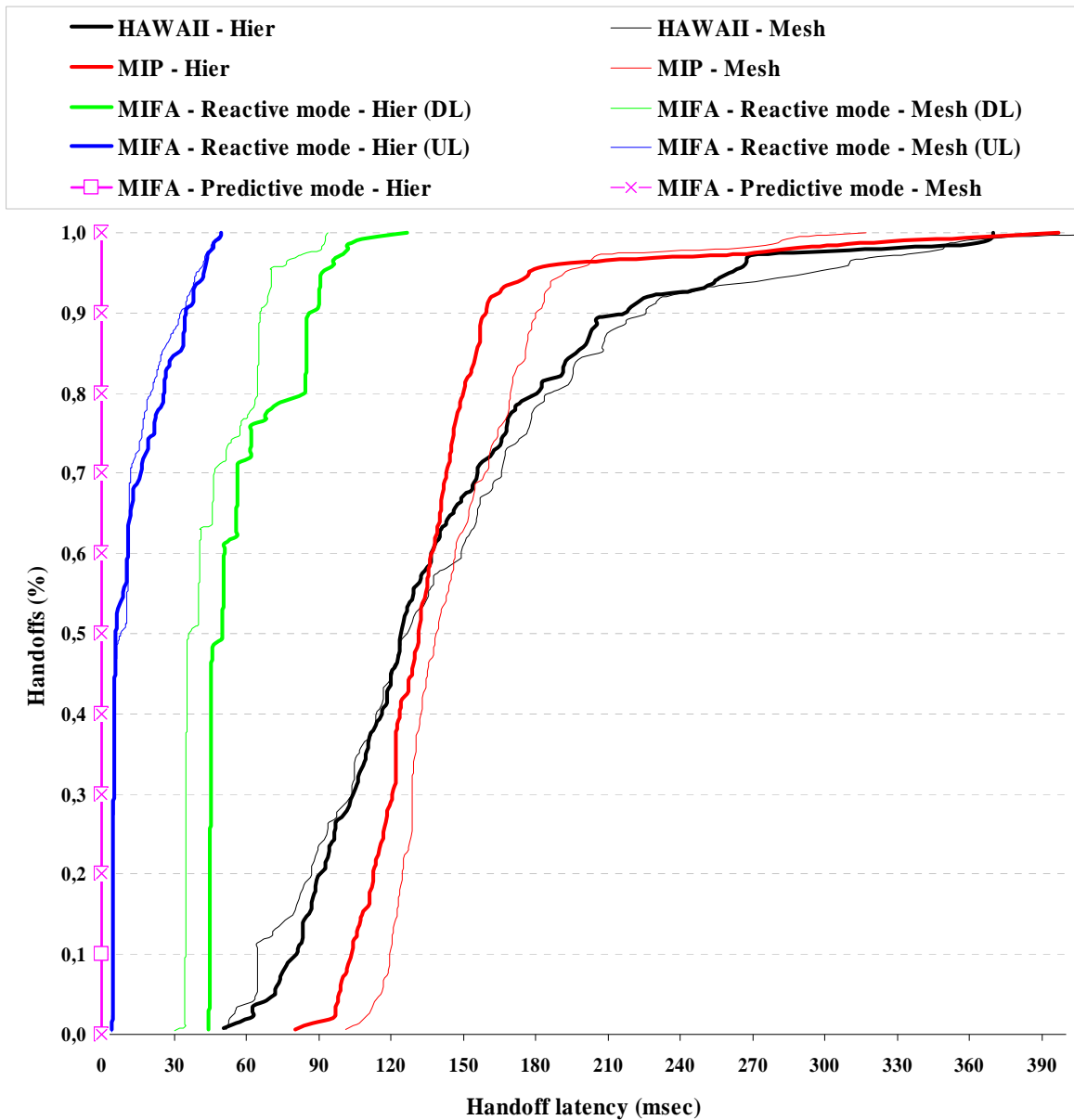


Fig G.1: Distribution function of the handoff latency experienced when employing HAWAII, MIP and MIFA in the studied mesh and hierarchical topologies under dynamically changing network conditions

G.2. Expected Number of Dropped Packets Per Handoff

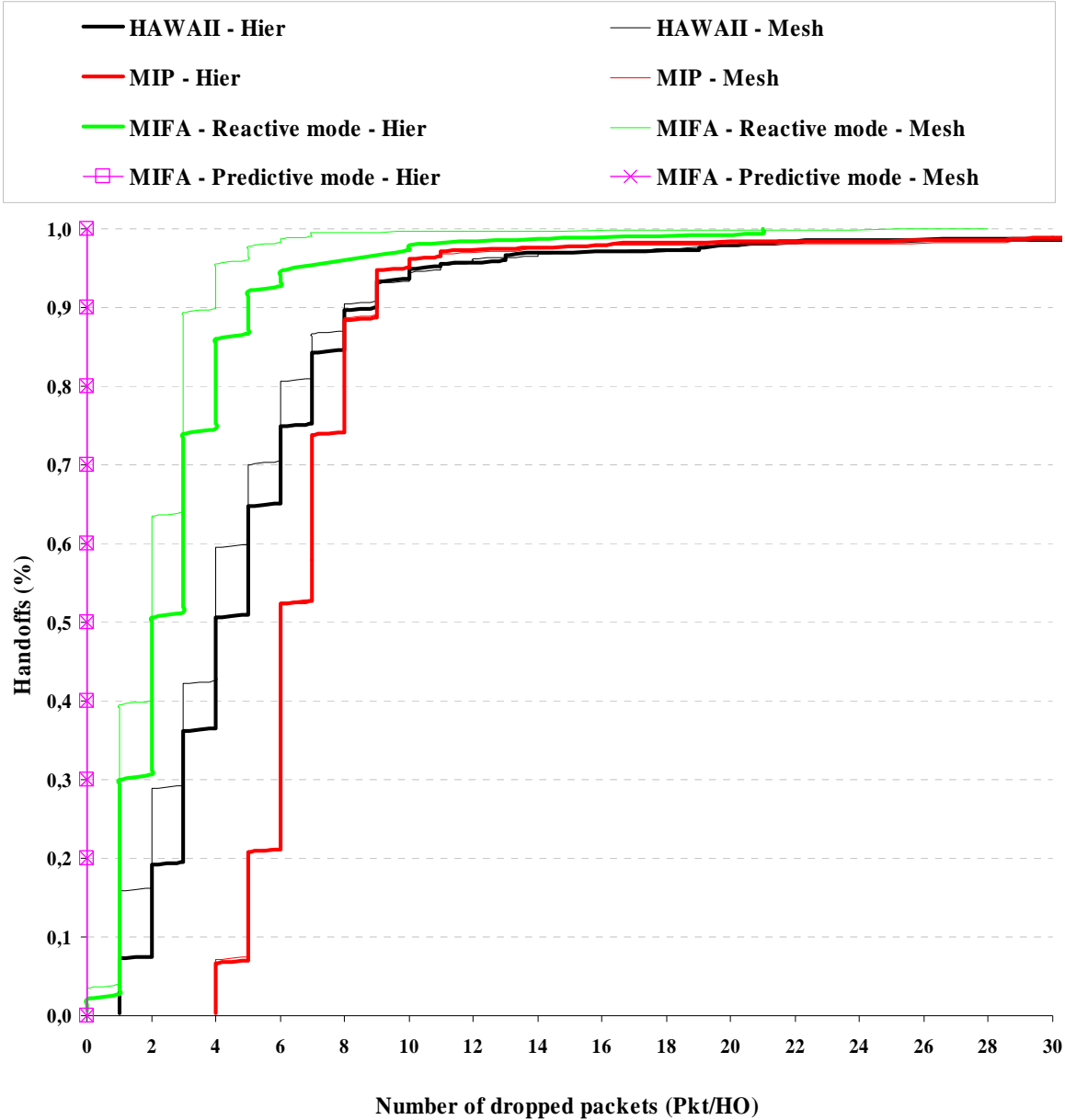


Fig G.2: Distribution function of the number of dropped packets per handoff on downlink experienced when employing HAWAII, MIP and MIFA in the studied mesh and hierarchical topologies under dynamically changing network conditions

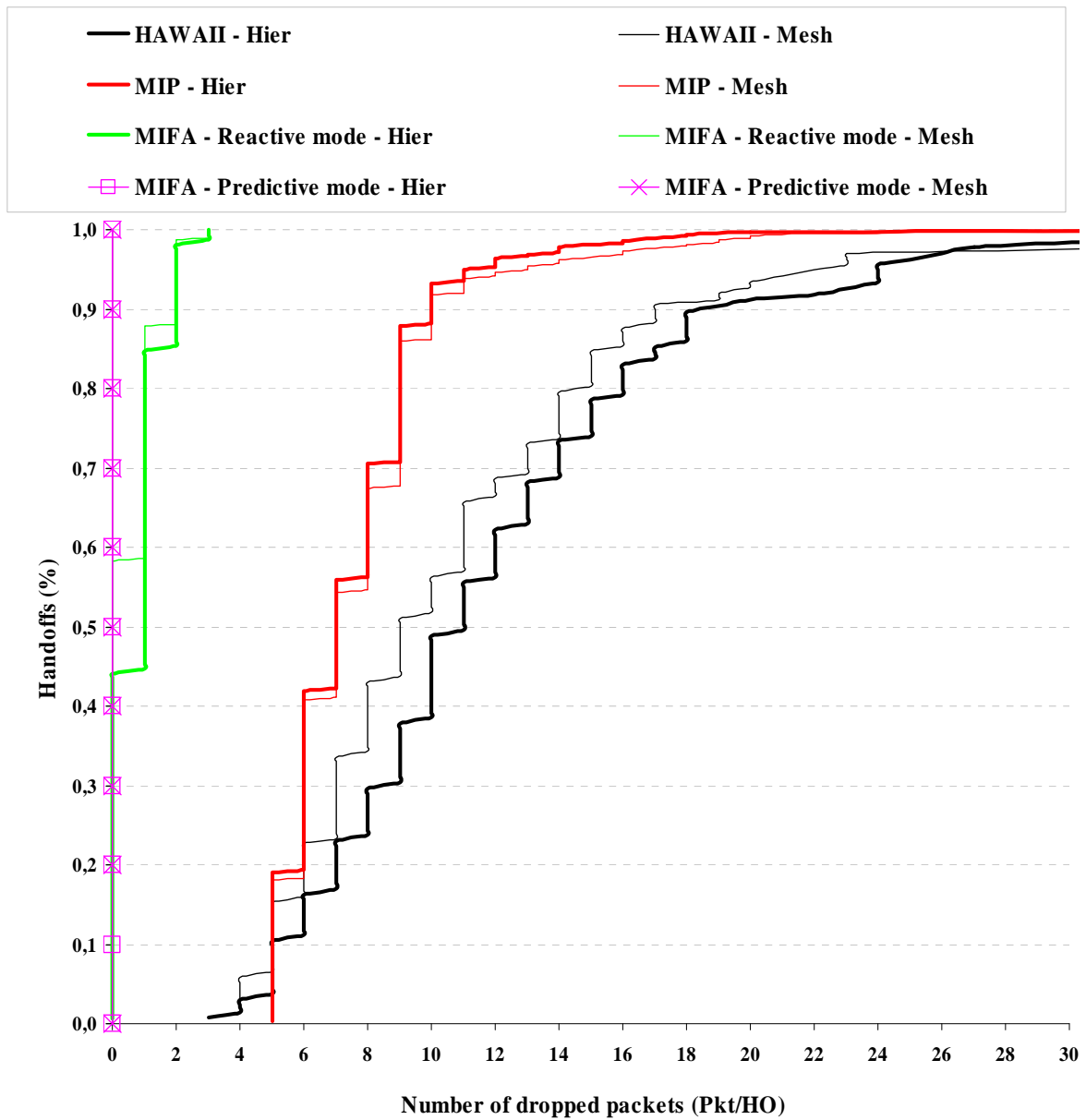


Fig G.3: Distribution function of the number of dropped packets per handoff on uplink experienced when employing HAWAII, MIP and MIFA in the studied mesh and hierarchical topologies under dynamically changing network conditions

Bibliography

- [ABG01] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.
- [Adl] Advanced Distributed Learning Initiative, official website: www.adlnet.org, accessed on 17.03.2009.
- [AFH06] W. Altmann, R. Fritz, D. Hinderink, "TYPO3: Enterprise Content Management", printed by Open Source Press, ISBN: 393751418X, 2006.
- [Aicc] Aviation Industry Computer-Based Training Committee, official website: www.aicc.org, accessed on 05.04.2009.
- [Alf06] B. A. Alfred Klampfer, "Didaktische Potentiale Wiederverwendbarer Lernobjekte", Master thesis, University of Hagen, 2006, available at: http://teaching.eduhi.at/alfredklampfer/Klal_Magisterarbeit_liz.pdf, accessed on 05.04.2009.
- [AMB06] E. Alnasouri, A. Mitschele-Thiel, R. Boeringer, A. Diab, "QoMIFA: A QoS Enabled Mobility Management Framework in ALL-IP Networks", in the proceeding of the 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06), Finland, September 2006.
- [AMD08] E. Alnasouri, A. Mitschele-Thiel, A. Diab, "MaISAM: A New Fast QoS Aware Mobility Management Protocol for ALL-IP Networks", in the proceeding of the 4th International Conference on Wireless and Mobile Communications (ICWMC'08), Greece, July 2008.
- [Apo02] M. Apolin, "Die Sprache in Physikschulbuechern unter Besonderer Beruecksichtigung von Texten zur Speziellen Relativitaetstheorie", Doctoral thesis, University of Wien, 2002, available at: <http://pluslucis.univie.ac.at/Archiv/Diplomarbeiten/Apolin/index.html>, accessed on 05.04.2009.
- [ARe05] M. Atiquzzaman, A. S. Reaz, "Survey and Classification of Transport Layer Mobility Management Schemes", in the proceeding of the 16th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'05), Germany, September 2005.
- [Ariad] ARIADNE Foundation for the European Knowledge Poll, <http://www.ariadne-eu.org/>, accessed on 05.04.2009.
- [Bal08] A. Baldranova, "Metadaten-Basierte Adaptivitaet in E-Learning", Diploma thesis, Ilmenau University of Technology, 2008.
- [Bal97] A. Ballardie, "Core Base Trees (CBT) Multicast Routing Architecture", RFC 2201, September 1997.
- [BBa95] A. Bakre, B. R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts", in the proceeding of the 15th IEEE International Conference on Distributed Computing Systems (ICDCS'95), Canada, May 1995.
- [BBM04] R. Boeringer, M. Bauer, A. Mitschele-Thiel, M. Soellner, "MPLS/RSVP-TE-Based Future UMTS Radio Access Network", in the proceeding of the 12th GI/ITG Conference on Measuring and Evaluation of Computer and Communication Systems (MMB) together with 3rd Polish-German Teletraffic Symposium (PGTS'04), Germany, September 2004.
- [BCC03] C. Blondia, O. Casals, L. Cerda, N. Van den Wijngaert, G. Willems, P. De-Cleyn, "Low Latency Handoff Mechanisms and Their Implementation in an IEEE 802.11 Network", in the proceeding of the 18th International Teletraffic Congress (ITC), Germany, 2003, available at: <http://www.pats.ua.ac.be/peter.decleyn>, accessed on 08.04.2009.

- [BCC03a] C. Blondia, O. Casals, L. Cerda, N. Van den Wijngaert, G. Willems, P. De- Cleyn. "Performance Comparison of Low Latency Mobile IP Schemes", in the proceeding of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks Workshop (WiOpt'03), France, March 2003.
- [Ber03] B. Berendt, "Vom Lehren zum Aktiven Lernen – Forschungsorientierter Beitrag zur Praxis Effektiver Hochschullehre", in: U. Welbers, T. Korytko, "Hochschuldidaktische Aus- und Weiterbildung", printed by W. Bertelsmann, ISBN: 978-3-7639-3088-3, 2003.
- [BGa99] B. Bruns, P. Gajewski, "Multimediales Lernen im Netz. Leitfaden fuer Entscheider und Planer", printed by Springer, ISBN:3-540-65428-3, 1999.
- [BGP04] V. G. Barrios, C. Guetl, A. Preis, K. Andrews, M. Pivec, F. Moedritscher, C. Trummer, "AdeLE: A Framework for Adaptive E-Learning through Eye Tracking", in the proceeding of the 4th International Conference on Knowledge Management and Knowledge Technologies (I-KNOW'04), Austria, June 2004.
- [BHH02] P. Baumgartner, K. Haefele, H. Haefele, "e-Learning. Didaktische und Technische Grundlagen", e-Learning Special Edition of bm:bwk, May 2002, available at: <http://www.qualifizierung.com/cms/materialien-mainmenu-55/downloads-mainmenu-57?func=startdown&id=10>, accessed on 06.04.2009.
- [Bla02] U. Black, "Voice Over IP. Series in Advanced Communications Technologies", printed by Prentice Hall International, 2nd Edition, ISBN: 0-13-065204-0, 2002.
- [Bru04] P. Brusilovsky, "KnowledgeTree: a Distributed Architecture for Adaptive eLearning", in the proceeding of the 13th International World Wide Web Conference (WWW'04), USA, May 2004.
- [Bru96] P. Brusilovsky, "Methods and Techniques of Adaptive Hypermedia", User Modeling and User-Adapted Interaction Journal, July 1996, available at: <http://www2.sis.pitt.edu/~peterb/papers/UMUI96.pdf>, accessed on 06.04.2009.
- [BSi96] K. Brown, S. Singh, "M-UDP: UDP for Mobile Cellular Networks", ACM SIGCOMM Computer Communication Review Magazine, October 1996.
- [BZB97] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) Version1 Functional Specification", RFC 2205, September 1997.
- [CCW03] O. Casals, L. Cerda, G. Willems, C. Blondia, N. Van den Wijngaert, "Performance Evaluation of the Post-Registration Method, a Low Latency Handoff in MIPv4", in the proceeding of the 38th IEEE International Conference on Communications (ICC'03), USA, May 2003.
- [CGK00] A. Campell, J. Gomez, S. Kim, A. Valko, C.-Y. Wan, Z. Turanyi, "Design, Implementation, and Evaluation of Cellular IP", IEEE Personal Communication Journal, August 2000.
- [CHK00] P. R. Calhoun, T. Hiller, J. Kempf, P. J. McCann, C. Pairla, A. Singh, S. Thalanany, "Foreign Agent Assisted Hand-off", Internet Draft <draft-calhoun-mobileip-proactive-fa-03.txt>, November 2000.
- [CIMS] Columbia IP Micro Mobility Suite (CIMS), available at: <http://www.comet.columbia.edu/micromobility>, accessed on 09.04.2009.
- [Cla95] H. J. Claus, "Einfuehrung in die Didaktik der Mathematik", printed by Wissenschaftliche Buchgesellschaft (WBG), ISBN: 3-534-08736-4, 1995.
- [CMD01] K. Chakrabarty, A. Misra, S. Das, A. McAuley, A. Dutta, S. K. Das, "Implementation and Performance Evaluation of TeleMIP", in the proceeding of the 35th IEEE International Conference on Communications (ICC'01), Finland, June 2001.
- [Con04] O. Conlan, "The Multi-Model, Metadata Driven Approach to Personalised eLearning Services", Doctoral thesis, University of Dublin, Trinity College, 2004.
- [DBM06] A. Diab, R. Boeringer, A. Mitschele-Thiel, "Optimized I-MPLS: A Fast and Transparent Micro-Mobility-Enabled MPLS Framework", in the proceeding of the 3rd International Symposium on Wireless Communication Systems (ISWCS'06), Spain, September 2006.

- [DBV03] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [DDr87] H. L. Dreyfus, S. E. Dreyfus, "Kuenstliche Intelligenz. Von den Grenzen der Denkmachine und dem Wert der Institution", printed by Rowohlt, ISBN: 3-499-18144-4, 1987.
- [DHi98] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [DLM07] A. Diab, F. Liers, A. Mitschele-Thiel, "Performance Analysis of Mobility Management Protocols Using a Generic Mathematical Model", in the proceeding of the 10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'07), Greece, October 2007.
- [DML08] A. Diab, A. Mitschele-Thiel, F. Liers, "Estimation of the Cost Resulting from Mobility Management Protocols Using a Generic Mathematical Model", in the proceeding of the 11th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'08), Canada, October 2008.
- [DMA00] S. Das, A. Misra, P. Agrawal, S. K. Das, "TeleMIP: Telecommunications-Enhanced Mobile IP Architecture for Fast Intradomain Mobility", IEEE personal communication Journal, August 2000.
- [DMB05] A. Diab, A. Mitschele-Thiel, R. Boeringer, "Evaluation of Mobile IP Fast Authentication Protocol Compared to Hierarchical Mobile IP", in the proceeding of the 1st Wireless and Mobile Computing, Networking and Communications (WiMob'05), Canada, August 2005.
- [DMB05a] A. Diab, A. Mitschele-Thiel, R. Boeringer, "Extension of Mobile IP for Fast Authentication and Low Latency Handoff", in the proceeding of the 11th European Wireless 2005, Cyprus, April 2005.
- [DMB06] A. Diab, A. Mitschele-Thiel, R. Boeringer, "A Framework to Support Fast Inter-Domain Mobility in ALL-IP Networks", in the proceeding of the 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06), Finland, September 2006.
- [DMD02] S. Das, A. Mcauley, A. Dutta, A. Misra, K. Chakraborty, S. K. Das, "IDMP: An Intradomain Mobility Management Protocol for Next-Generation Wireless Networks", IEEE Wireless Communications Magazine, June 2002.
- [DMG08] A. Diab, A. Mitschele-Thiel, K. Getov, O. Blume, "Analysis of Proxy MIPv6 Performance Compared to Fast MIPv6", in the proceeding of the 33rd IEEE Conference on Local Computer Networks (LCN), Canada, October 2008.
- [DMi04] A. Diab, A. Mitschele-Thiel, "Minimizing Mobile IP Handoff Latency", in the proceeding of the 2nd International Working Conference on Performance modelling and Evaluation of Heterogeneous Networks (HET-NETs'04), UK, July 2004.
- [DMi07] A. Diab, A. Mitschele-Thiel, "CAMP: A New Tool to Analyse Mobility Management Protocols", in the proceeding of the 52nd IWK-Internationales Wissenschaftliches Kolloquium, Germany, September 2007.
- [DMX04] A. Diab, A. Mitschele-Thiel, J. Xu, "Performance Analysis of the Mobile IP Fast Authentication Protocol", in the proceeding of the 7th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'04), Italy, October 2004.
- [Dotlr] LRN: Learn, Research, Network official website: <http://dotlrn.org>, accessed on 17.03.2009.
- [Dro97] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [DSM02] A. Dutta, H. Schulzrinne, S. Madhani, O. Altintas, W. Chen, "Optimized Fast-handoff Schemes for Application Layer Mobility Management", ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), 2002.

- [DSt02] G. Doerr, P. Strittmatter, "Multimedia aus Paedagogischer Sicht", in: L. J. Issing, P. Klimsa, "Information und Lernen mit Multimedia und Internet: Lehrbuch fuer Studium und Praxis", printed by Beltz PVU, 3rd Edition, ISBN:3-432-98631-9, 2002.
- [DWM08] A. Diab, H-D. Wuttke, A. Mitschele-Thiel, K. Henke, "Metadata-Based Personalization of eLearning Contents", in the proceeding of the 5th Conference of Scientific Research Outlook (SRO'08), Marocco, October 2008.
- [DYe01] G. Dommety, T. Ye, "Local and Indirect Registration for Anchoring Handoffs", Intrent Draft <draft-dommety-mobileip-anchor-handoff-02.txt>, July 2001.
- [DyMIP] Dynamics MIP, Helsinki University of Technology, available at: <http://dynamics.sourceforge.net/>, accessed on 18.06.09.
- [ECS94] D. Eastlake, S. Crocker, J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994.
- [Edd04] W. M. Eddy, "At What Layer Does Mobility Belong?", IEEE Communications Magazine, October 2004.
- [EFK06] H. Ekstroem, A. Furuskaer, J. Karlsson, M. Meyer, S. Parkvall, J. Torsner, M. Wahlqvist, "Technical Solutions for the 3G Long-Term Evolution", IEEE Communications Magazine, March 2006.
- [EGV03] P. Estrela, A. Grilo, T. Vazão, M. Nunes, "Terminal Independent Mobile IP (TIMIP)", Internet Draft <draftestrela-timip-01.txt>, January 2003.
- [EHS97] J. Ellsberger, D. Hogrefe, A. Sarma, "SDL - Formal Object-oriented Language for Communicating Systems", printed by Prentice Hall Europe, ISBN: 0-13-621384-7, 1997.
- [EMS00] P. Eardley, A. Mihailovic, T. Suihko, "A Framework for the Evaluation of IP Mobility Protocols", in the proceeding of the 11th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06), UK, September 2000.
- [ESe08] F. Evers, J. Seitz, "REACH: A Roaming-Enabled Architecture for Multi-Layer Capturing", in the proceeding of the IEEE Wireless Communications and Networking Conference (WCNC'08), USA, April 2008.
- [Est07] P. Estrela, "Transparent and Efficient IP Mobility", Ph.D. thesis, IST-Taguspark University, 2007, available at: <http://tagus.inesc-id.pt/~pestrela/>, accessed on 06.04.2009.
- [ESTI] European Telecommunications Standards Institute, official website: <http://www.etsi.org/>, accessed on 06.04.2009.
- [ESTI96] Digital cellular telecommunications system (Phase 2+); General description of a GSM Public Land Mobile Network (PLMN). GSM recommendations 01.02, March 1996.
- [EVN04] P. Estrela, T. Vazão, M. Nunes, "Micro Mobility Performance Evaluation of a Terminal Independent Mobile Architecture", in the proceeding of the 2nd International Working Conference on Performance Evaluation of Heterogeneous Networks (Het-Net'04), UK, July 2004.
- [EVN06] P. Estrela, T. M. Vazão, M. S. Nunes, "Design and Evaluation of eTIMIP – an Overlay Micro-Mobility Architecture Based on TIMIP", in the proceeding of the 2nd IEEE/IARIA International Conference on Wireless and Mobile Communications (ICWMC'06), Romania, July 2006.
- [Far01] F. Farance, "Draft Standard for Learning Technology – Public and Private Information (PAPI) for Learners (PAPI Learner)", Internet draft version 8.00, November 2001, available at: <http://www.cen-ltsa.net/Users/main.aspx?page=136>, accessed on 13.04.2009.
- [Fes03] A. Festag, "Mobility Support in IP Cellular Networks – A Multicast-Based Approach", Doctoral thesis, Technical University Berlin, Germany, 2003, available at: http://edocs.tu-berlin.de/diss/2003/festag_andreas.pdf, accessed on 06.04.2009.

- [FGo01] N. A. Fikouras, C. Goerg, "Performance Comparison of Hinted- and Advertisement-Based Movement Detection Methods for Mobile IP Hand-offs", *Computer Networks Journal*, September 2001.
- [FGo01a] N. A. Fikouras, C. Goerg, "A Complete Comparison of Algorithms for Mobile IP Hand-offs with Complex Movement Patterns and Internet Audio", in the proceeding of the 4th International Symposium on Wireless Personal Multimedia Communications (WPMC'04), Denmark, September 2001.
- [Fis06] A. Fischer, "Die Entwicklung eines Konzeptes zum Einsatz einer Webbasiereten Lehr- und Lernplattform an der Beruflichen Schule des Landkreises Mueritz", Master thesis, University of Rostock, 2006.
- [FJP07] E. Fogelstroem, A. Jonsson, C. E. Perkins, "Mobile IPv4 Regional Registration", RFC 4857, June 2007.
- [FKK02] X. Fu, H. Karl, C. Kappler, "QoS-Conditionalized Handoff for Mobile IPv6", in the proceeding of the 2nd International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols (Networking'02), Italy, May 2002, available at: <http://www.tkn.tu-berlin.de/research/SeQoMo/>, accessed on 06.04.2009.
- [FKS00] A. Festag, H. Karl, G. Schaefer, "Current Developments and Trends in Handover Design for ALL-IP Wireless Networks", Telecommunication Networks Group (TKN) Technical Report Series TKN-00-05, University of Berlin, August 2000, available at: <http://www.tkn.tu-berlin.de/research/SeQoMo/>, accessed on 06.04.2009.
- [Fle08] A. Fleischer, "Entwurf und Implementierung Innovativer Architekturen fuer Learning Content Management Systeme (LCMS) zum Einsatz im Erloesorientierten Bildungsexport", Doctoral thesis, Ilmenau University of Technology, 2008.
- [FLH00] D. Farinacci, T. Li, S. Hanks, D. Meyer, P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.
- [FMA05] S. Fu, L. Ma, M. Atiquzzaman, Y. Lee, "Architecture and Performance of SIGMA: A Seamless Handover Scheme for Data Networks", in the proceeding of the 40th IEEE International Conference on Communications (ICC'05), South Korea, May 2005.
- [FMM00] D. Forsberg, J. T. Malinen, J. K. Malinen, H. H. Kari, "Increasing Communication Availability With Signal-Based Mobile Controlled Handoffs", in the proceeding of the IP based Cellular Networks Conference (IPCN'00), Paris, May 2000.
- [FRe04] F. Feng, D. S. Reeves, "Explicit Proactive Handoff with Motion Prediction for Mobile IP", in the proceeding of the IEEE Wireless Communications and Networking Conference (WCNC'04), USA, March 2004.
- [FSe00] P. Ferguson, D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing", RFC 2827, May 2000.
- [FTo99] F. Farance, J. Tonkel, "LTSA Specification. Learning Technology System Architecture", draft 5, December 1999, available at: http://ltsc.ieee.org/wg1/files/ltsa_05.pdf, accessed on 17.03.2009.
- [FYT97] D. Funato, K. Yasuda, H. Tokuda, "TCP-R: TCP Mobility Support for Continuous Operation", in the proceeding of the 5th IEEE International Conference on Network Protocols (ICNP'97), USA, October 1997.
- [GEN01] A. Grilo, P. Estrela, M. Nunes, "Terminal Independent Mobility for IP (TIMIP)", *IEEE Communications Magazine*, December 2001.
- [Ger07] F. Gertsch, "Das Moodle 1.8 Praxisbuch : Online-Lernumgebungen einrichten, Anbieten und Verwalten", printed by Addison-Wesley, ISBN: 978-3-8273-2514-3*Gb, 2007.
- [Get08] K. Getov, "Performance Analysis of Network-Based Protocols for Localized Mobility Management", Diploma thesis, Ilmenau University of Technology, 2008.

- [GFJ03] Y. Gwon, G. Fu, R. Jain, "Fast Handoffs in Wireless LAN Networks Using Mobile Initiated Tunneling Handoff Protocol for IPv4 (MITHv4)", in the proceeding of the IEEE Wireless Communications and Networking (WCNC'03), USA, March 2003.
- [GHB08] M. C. Gonzalez, C. A. Hidalgo, A.-L. Barabasi, "Understanding individual human mobility patterns", nature journal, June 2008.
- [GLD08] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.
- [Gri00] R. Grillenbeck, "Didaktik und Methodik der Theoretischen Informatik, Motivation und Computerunterstuetztes Lernen", Doctoral thesis, University of Erlangen-Nuernberg, 2000.
- [GRu05] L. A. Galindo-Sánchez, P. M. Ruiz-Martínez, "QoS and Micromobility Coupling: Improving Performance in Integrated Scenarios", European Journal for the Informatics Professional (Upgrade), June 2005, available at: <http://www.upgrade-cepis.com/issues/2005/2/upgrade-vol-VI-2.html#summary>, accessed on 07.04.2009.
- [Gus99] E. Gustafsson, Ed., "Requirements on Mobile IP from a Cellular Perspective", Internet Draft <draft-ietf-mobileip-cellular-requirements-02>, June 1999.
- [GWo08] GSM World, official website: <http://www.gsmworld.com/roaming/gsminfo/index.shtml>, accessed on 08.04.2009.
- [Haa02] J. Haack, "Interaktivitaet als Kennzeichen von Multimedia und Hypermedia", in: L. J. Issing, P. Klimsa, "Information und Lernen mit Multimedia und Internet", printed by Beltz, 3rd Edition, ISBN: 3-621-27449-9, 2002.
- [HAg97] Z. J. Haas, P. Agrawal, "Mobile-TCP: An Asymmetric Transport Protocol Design for Mobile Systems", in the proceeding of the 32th IEEE International Conference on Communications (ICC'97), Canada, June 1997.
- [Hal96] F. Halsall, "Data Communications, Computer Networks and Open Systems", printed by Addison-Wesley, 4th Edition, ISBN: 0-201-42293-X, 1996.
- [HCa98] D. Harkins, D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.
- [HCJ03] Y. Han, J. Choi, H. Jang, S. Park, "Advance Duplicate Address Detection", Internet Draft <draft-han-mobileip-adad-01.txt>, July 2003.
- [Hei08] P. Heilwagen, "Performance Analysis of Mobility Management Protocols Using a Generic Mathematical Mode", Diploma thesis, Ilmenau University of Technology, 2008.
- [Hen03] T. R. Henderson, "Host Mobility for IP Networks: A Comparison", IEEE Network Magazine, November 2003.
- [HGP00] H. Hersent, D. Gurle, J. -P. Petit, "IP Telephony Packet-Based Multimedia Communications Systems", printed by Pearson Education Limited, ISBN: 0-201-61910-0, 2000.
- [HJM06] K. Hametner, T. Jarz, W. Moriz, J. Pauschenwein, H. Sandtner, I. Schinnerl, A. Sfiri, M. Teufel, "Qualitaetskriterien fuer E-Learning - Ein Leitpfaden fuer Lehrer/innen, Lehrende und Content-Ersteller/innen", 2006, available at: http://www.bildung.at/ext/bmbwk/news/index.php?article_id=198&itk_sid=a538f3b462aa6fbc3b236b0ca830c02d, accessed on 07.04.2009.
- [HKZ03] H. Hsieh, K. Kim, Y. Zhu, R. Sivakumar, "A Receiver-Centric Transport Protocol for Mobile Hosts with Heterogeneous Wireless Interfaces", in the proceeding of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom'03), USA, September 2003.
- [Hor02] W. Horton, "Don't Bother Me With Objects! I've Got a Course to Teach!", August 2002, available at: <http://teacode.com/biblio/lo/objectkeynote.pdf>, accessed on 07.04.2009.
- [HSS99] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, March 1999.

- [HZS03] R. Hsieh, Z. G. Zhou, A. Seneviratne, "S-MIP: A Seamless Handoff Architecture for Mobile IP", in the proceeding of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03), USA, March 2003.
- [Ibr02] J. Ibrahim, "4G Features", Bechtel Telecommunications Technical Journal, December 2002, available at: <http://www.comsec.uwaterloo.ca/~flchiu/CDMA/4g%20standard.pdf>, accessed on 07.04.2009.
- [IEEEin] Institute of Electrical and Electronic Engineers, official website: <http://grouper.ieee.org/groups/802/11/>, accessed on 07.04.2009.
- [IEStd] IEEE Computer Society, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", Standard IEEE 802.11, ISBN: 0-7381-5656-6 SS95708, June 2007.
- [Ilias] ILIAS, official website: <http://www.ilias.de>, accessed on 17.03.2009.
- [ImS] Instructional Management System, official website: www.imsproject.org, accessed on 17.03.2009.
- [IPSec] IP Security Protocol, official website: <http://www.ietf.org/html.charters/OLD/ipsec-charter.html>, accessed on 07.04.2009.
- [IPv6WG] IPv6 Working Group, official website: <http://www.ietf.org/html.charters/OLD/ipv6-charter.html>, accessed on 7.04.2009.
- [JAK02] X. Jiang, I. F. Akyildiz, "A Novel Distributed Dynamic Location Management Scheme for Minimizing Signaling Costs in Mobile IP", IEEE Transactions on Mobile Computing Journal, July 2002.
- [JPA04] D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.
- [KAt98] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [KAt98a] S. Kent, R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.
- [KBA96] R. H. Katz, E. A. Brewer, E. Amir, H. Balakrishnan, A. Fox, S. Gribble, T. Hodes, D. Jiang, G. T. Nguyen, V. Padmanabhan, M. Stemm, "The Bay Area Research Wireless Access Network (BARWAN)", in the proceeding of the 41st IEEE Computer Society International Conference (COMPCON'96), USA, February 1996.
- [KBC97] H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.
- [KCL04] S. J. Koh, M. J. Chang, M. Lee, "mSCTP for Soft Handover in Transport Layer", IEEE Communications Letters, March 2004.
- [Ken05] S. Kent, "IP Encapsulating Security Protocol (ESP)", RFC4303, December 2005.
- [Ker01] M. Kerres, "Multimediale und Telemediale Lernumgebungen: Konzeption und Entwicklung", printed by Oldenbourg, 2nd Edition, ISBN: 3-486-25055-8, 2001.
- [KGT01] C. Keszei, N. Georganopoulos, Z. Turanyi, A. Valko, "Evaluation of the BRAIN Candidate Mobility Management Protocol", in the proceeding of the 10th IST Mobile Communications Summit, Spain, September 2001.
- [KMA03] K. Kaneko, H. Morikawa, T. Aoyama, "Session Layer Mobility Support for 3C Everywhere Environments", in the proceeding of the 6th International Symposium on Wireless Personal Multimedia Communications (WPMC '03), Japan, October 2003, available at: <http://www.mlab.t.u-tokyo.ac.jp/publications/2003/en-US>, accessed on 09.04.2009.
- [KNo00] R. Keil-Slawik, O. Nowaczyk, "Von der Geschlossenen Multimediaproduktion zur Offenen Lernumgebung" in: F. Scheuermann, "Campus 2000: Lernen in neuen Organisationsformen", printed by Waxmann, ISBN:3-89325-925-2, 2000, available at: <http://iug.upb.de/rks//Publikationen/.2000/2000-ksn-campus2000.pdf>, accessed on 07.04.2009.
- [Koo05] R. Koodli, Ed., "Fast Handovers for Mobile IPv6", RFC 4068, July 2005.

- [KR01] J. Kurose, K. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", printed by Addison Wesley Longman, ISBN: 0-201-47711-4, 2001.
- [Lei01] T. Leidig, "L3 - Towards an Open Learning Environment", ACM Journal of Educational Resources in Computing (JERIC), March 2001.
- [LGG00] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence, "Generic AAA Architecture", RFC 2903, August 2000.
- [LLI99] B. Landfeldt, T. Larsson, Y. Ismailov, A. Seneviratne, "SLM: A Framework for Session Layer Mobility Management", in the proceeding of the 8th IEEE International Conference Computer Communications and Networks (ICCCN'99), USA, November 1999.
- [LSV99] B. Lampson, V. Srinivasan, G. Varghese, "IP Lookups Using Multiway and Multicolumn Search", IEEE/ACM Transactions on Networking Journal, June 1999.
- [Lts] Institute for Electrical and Electronic Engineers Learning Technology Standards Committee, official website: <http://ieeeltsc.org/>, accessed on 17.03.2009.
- [Lts-a] IEEE LTSC group, "Draft Standard for Learning Object Metadata", available at: http://lts.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, accessed on 17.03.2009.
- [Lud05] N. Ludwig, "Multimediale Lernumgebung fuer Remote Labs-Konzept und Implementierung", Diploma thesis, Ilmenau University of Technology, 2005.
- [Mal07] K. El Malki, "Low Latency Handoffs in Mobile IPv4", RFC 4881, June 2007.
- [MBh97] J. Mysore, V. Bharghavan, "A New Multicasting Based Architecture for Internet Host Mobility", in the proceeding of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'97), Hungary, September 1997.
- [MBh98] D. Maltz, P. Bhagwat, "TCP Splicing for Application Layer Proxy Performance", IBM Research Report RC 21139, IBM T.J. Watson Research Center, March 1998.
- [MBh98a] D. A. Maltz, P. Bhagwat, "MSOCKS: An Architecture for Transport Layer Mobility", in the proceeding of the 17th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'98), USA, March 1998.
- [MCN02] Y. Ma, D. Chae, W. Noh, J. Lee, Y. Kim, S. An, "A Multi-Path Support for Mobile IP with the Hierarchical Architecture", in the proceeding of the 17th International Conference on Information Networking(ICOIN'02), Korea, January 2002.
- [MDa03] N. Moore, G. Daley, "Fast Address Configuration Strategies for the Next-Generation Internet", in the proceeding of Australian Telecommunications, Networks and Applications Conference (ATNAC'03), Australia, December 2003.
- [Metac] Metacoön, official website: <http://www.metacoön.net/>, accessed on 17.03.2009.
- [mextWG] Mobility EXTensions for IPv6 (mext), IETF working group, official website: <http://www.ietf.org/html.charters/mext-charter.html>, accessed on 07.04.2009.
- [MFa04] W. Ma, Y. Fang, "Dynamic Hierarchical Mobility Management Strategy for Mobile IP Networks", IEEE Journal on Selected Areas in Communications, 2004.
- [MGr05] R. Messerschmidt, R. Grebe, "Zwischen Visionaerer Euphorie und Praktischer Erneuechterung: Informations- und Bildungstechnologien der Vergangenen Fuenfzig Jahre", QUEM-report, Berlin 2005.
- [MHa02] K. Maier-Haeefe, H. Haeefe, "Learning-, Content- und Learning-Content-Management-Systeme: Gemeinsamkeiten und Unterschiede", IDEE-Haeefe KEG, 2002, available at: <http://rk-web.de/data/pdf/LCMS.pdf>, accessed on 17.03.2009.
- [Mil92] D. L. Mills, "Network Time Protocol (Version 3) Specification, Implementation and Analysis", RFC 1305, March 1992.
- [mip4WG] Mobility for IPv4 (mip4), IETF working group, official website: <http://www.ietf.org/html.charters/mip4-charter.html>, accessed on 18.04.2009.

- [Mit01] A. Mitschele-Thiel, "Systems Engineering with SDL: Developing Performance-Critical Communication", printed by John Wiley & Sons, 2001.
- [MKM08] J. Manner, G. Karagiannis, A. McDonald, "NSLP for Quality-of-Service Signaling", Internet Draft <draft-ietf-nsis-qos-nslp-16.txt>, February 2008.
- [MMN05] J. Montavont, N. Montavont, T. Noel, "Enhanced Schemes for L2 Handover in IEEE 802.11 Networks and their Evaluations", in the proceeding of the 16th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'05), Germany, September 2005, available at: <http://lsiit.u-strasbg.fr/Publications/2005/MMN05/>, accessed on 07.04.2009.
- [Mon01] G. Montenegro, "Reverse Tunneling for Mobile IP, revised", RFC 3024, January 2001.
- [Moo06] N. Moore, "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, April 2006.
- [Mood1] Moodle, official website: <http://www.moodle.org>, accessed on 17.03.2009.
- [Moy98] J. Moy, "OSPF Version 2", RFC 2328, April 1998.
- [MSA00] A. Mihailovic, M. Shabeer, A. H. Aghvami, "Multicast for Mobility Protocol (MMP) for Emerging Internet Networks", in the proceeding of the 11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'00), UK, September 2000.
- [MSA02] A. Mishra, M. Shin, W. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process", ACM SIGCOMM Computer Communication Review Magazine, April 2003.
- [MSA04] A. Mishra, M. Shin, W.A. Arbaugh, "Context Caching Using Neighbor Graphs for Fast Handoffs in a Wireless Network", in the proceeding of the 23th Conference of the IEEE Communications Society (INFOCOM'04), China, March 2004.
- [MSA04a] A. Mishra, M. Shin, N. L. Pertroni, T. C. Clancy, W. Arbaugh, "Pro-Active Key Distribution Using Neighbor Graphs", IEEE Wireless Communications Magazine, February 2004.
- [MYO03] H. Matsuoka, T. Yoshimura, T. Ohya, "End-to-End Robust IP Soft Handover", in the proceeding of the 38th IEEE International Conference on Communications (ICC'03), Alaska, May 2003.
- [Nam] Network Animator, official website: <http://www.isi.edu/nsnam/nam/>, accessed on 18.04.2009.
- [NETWG] NETLMM, IETF work group, official website: <http://www.ietf.org/html.charters/netlmm-charter.html>, accessed on 03.03.2009.
- [NHH04] H. M. Niegermann, S. Hessel, D. Hochscheid-Mauel, K. Aslanski, M. Deimann, G. Kreuzberger, "Kompendium E-Learning", printed by Springer, ISBN: 3-540-43816-5, 2004.
- [NIS95] National Institute of Standards and Technology, "Secure Hash Standard", Federal Information Processing Standards Publication (FIPS PUB) 180-1, April 1995, available at: <http://www.itl.nist.gov/fipspubs/fip180-1.htm>, accessed on 09.04.2009.
- [NNS98] T. Narten, E. Nordmark, W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.
- [NS2] Network Simulator 2 (NS2), official website: <http://www.isi.edu/nsnam/ns/>, accessed on 07.04.2009.
- [Pae93] G. Paetzold, "Lehrmethoden in der Beruflichen Bildung", printed by Sauer, ISBN:3-7938-7086-3, 1993.
- [Paw04] J. M. Pawlowski, "Lerntechnologiestandards: Gegenwart und Zukunft", in: S. O. Tergan, P. Schenkel, "Was Macht E-Learning Erfolgreich? Grundlagen und Instrumente der Qualitätsbeurteilung", printed by Springer, ISBN: 3-540-20676-0, 2004.
- [PCa01] C. E. Perkins, P. R. Calhoun, "Generalized Key Distribution Extensions for Mobile IP", Internet Draft < draft-ietf-mobileip-gen-key-01.txt >, August 2001.
- [PCA04] C. Politis, K. A. Chew, N. Akhtar, M. Gerogiades, R. Tafazolli, T. Dagiuklas, "Hybrid Multilayer Mobility Management with AAA Context Transfer Capabilities for All-IP Networks", IEEE Wireless Communications Magazine, August 2004.

- [PCh02] S. Pack, Y. Choi, "Fast Inter-AP Handoff Using Predictive-Authentication Scheme in a Public Wireless LAN", in the proceeding of IEEE Networks Conference 2002, USA, August 2002.
- [PCh02a] S. Pack, Y. Choi, "Pre-Authenticated Fast Handoff in a Public Wireless LAN Based on IEEE 802.1x Model", in the proceeding of the 7th IFIP International Conference on Personal Wireless Communications (PWC'02), Singapore, October 2002.
- [PCT03] C. Politis, K. Chew, R. Tafazolli, "Multilayer Mobility Management for All-IP Networks: Pure SIP vs. Hybrid SIP/Mobile IP", in the proceeding of the 57th IEEE Semiannual Vehicular Technology Conference (VTC'03), Korea, April 2003.
- [Per02] C. Perkins, "IP Mobility Support for IPv4", RFC 3344, August 2002.
- [Per96] C. Perkins, "IP Encapsulation within IP", RFC 2003, October 1996.
- [Per96a] C. Perkins, "Minimal Encapsulation within IP", RFC 2004, October 1996.
- [Per96b] C. Perkins, "IP Mobility Support", RFC 2002, October 1996.
- [Per98] C. Perkins, "MOBILE IP - Design Principles and Practices", printed by Addison-Wesley, ISBN: 0-201-63469-4, 1998.
- [PJA00] C. Perkins, D. B. Johnson, N. Asokan, "Registration Keys for Route Optimization", Internet Draft <draft-ietf-mobileip-regkey-03.txt>, July 2000.
- [PJa02] S. R. Pandey, S. Jamadagni, "Improved Low Latency Handoff in Mobile IPv4", Internet Draft <draft-shiva-improved-lowlatency-handoff-v4-01.txt>, February 2002.
- [PJo01] C. Perkins, D. B. Johnson, "Route Optimization in Mobile IP", Internet Draft <draft-ietf-mobileip-optim-11.txt>, September 2001.
- [PKi01] T. Pagtzis, P. Kirstein, "A Framework for Proactive Mobility in Mobile IPv6", Internet Draft <draft-pagtzis-mobileiproactivev6-00.txt>, July 2001.
- [PMD04] L. Peters, I. Moerman, B. Dhoedt, P. Demeester, "MEHRM: Micromobility Support with Efficient Handoff and Route Optimization Mechanism", in the proceeding of the 16th ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems, Belgium, August, 2004.
- [Pos80] J. Postel, "User Datagram Protocol", RFC 768, August 1980.
- [Pos81] J. Postel, "Internet Protocol", RFC 791, September 1981.
- [Pos81a] J. Postel, "Transmission Control Protocol", RFC 793, September 1981.
- [PWa99] C. Perkins, K. -Y. Wang, "Optimized Smooth Handoffs in Mobile IP", in the proceeding of the 4th IEEE Symposium on Computers and Communications, Egypt, July 1999.
- [Raa07] P. Raatikainen, "Design Issues in All-IP Mobile Networks", Final Report of DAIMON project, printed by Julkaisija utgivare, ISBN: 978-951-38-6929-8, 2007, available at: <http://www.vtt.fi/inf/pdf/tiedotteet/2007/T2376.pdf>, accessed on 09.04.2009.
- [RBa98] J. Redi, P. Bahl, "Mobile IP: A Solution for Transparent Seamless Mobile Computer Communications", Fuji-Keizai's Report on Upcoming Trends in Mobile Computing and Communications, July 1998.
- [RFD06] M. Rey-López, A. Fernández-Vilas, R. Díaz-Redondo, J. Pazos-Arias, "Providing SCORM with Adaptivity", in the proceeding of the 15th international Conference on World Wide Web (WWW'06), Scotland, 2006.
- [Riv92] R. Rivest, "The MD5 Message Digest Algorithm", RFC1321, April 1992.
- [RKT94] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks", in the proceeding of the 13th Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM'94), Canada, June 1994.
- [RLi95] Y. Rekhter, T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

- [RLL00] R. Ramjee, T. La Porta, L. Li, "Paging Support for IP Mobility", Internet Draft <draft-ietf-mobileip-paging-hawaii-01.txt>, July 2000.
- [RLT00] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, "IP Micro-Mobility Support Using HAWAII", Internet Draft <draft-ietf-mobileip-hawaii-01>, July 2000.
- [Ruh07] M. Ruhwedel, "Konzeption und Prototypische Realisierung einer Adaptiven e-Learning-Umgebung", Diploma thesis, Ilmenau University of Technology, 2007.
- [RVC01] E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RVS02] 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, June 2002.
- [SBa00] A. C. Snoeren, H. Balakrishnan, "An End-to-End Approach to Host Mobility", in the proceeding of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom'00), USA, August 2000.
- [SBH01] S. Seufert, A. Back, M. Haeusler, "E-Learning - Weiterbildung im Internet: Das "Plato-Cookbook" fuer Internetbasiertes Lernen", printed by SmartBooks, 1st Edition, ISBN: 3-908490-53-7, 2001.
- [SBK97] S. Seshan, H. Balakrishnan, R. H. Katz, "Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience", Kluwer Journal on Wireless Personal Communications, January 1997.
- [SCF96] H. Schulzrinne, S. Casner, R. Fredrick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.
- [Sch03] J. Schiller, "Mobile Communication", printed by Addison-Wesley, 2nd Edition, ISBN: 0-321-12381-6, 2003.
- [Sch04] R. Schulmeister, "Didaktisches Design aus Hochschuldidaktischer Sicht", in: U. Rinn, M. D. Meister, "Didaktik und Neue Medien. Konzepte und Anwendungen in der Hochschule", printed by Waxmann, ISBN: 3-8309-1216-1, 2004.
- [Sch06] R. Schulmeister, "eLearning: Einsichten und Aussichten", printed by Oldenbourg, ISBN: 3-486-58003-5, 2006.
- [SCM05] H. Soliman, C. Castelluccia, K. El-Malki, L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)", RFC 4140, August 2005.
- [Scorm] Sharable Content Object Reference Model (SCORM), 3rd Edition Documentation Suite, <http://www.scormsoft.com/scorm>, accessed on 08.04.2009.
- [Ses95] S. Seshan, "Low-Latency Handoff for Cellular Data Network", Ph.D. thesis, Graduate Division of the University of California at Berkeley, 1995.
- [SFR04] S. Shin, A. G. Forte, A. S. Rawat, H. Schulzrinne, "Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs", in the proceeding of the 2nd ACM International Workshop on Mobility management & Wireless Access Protocols (MobiWac'04), USA, October 2004.
- [SMM04] D. Saha, A. Mukherjee, I. S. Misra, M. Chakraborty, N. Subhash, "Mobility Support in IP: A Survey of Related Protocols", IEEE Network Magazine, December 2004.
- [SRo99] H. Schulzrinne, J. Rosenberg, "Internet Telephony: Architecture and Protocols – an IETF perspective", Computer Networks and ISDN Systems Journal, February 1999.
- [SWC02] E. Shim, H. Wie, Y. Chang, R. D. Gitlin, "Low Latency Handoff for Wireless IP QoS with Neighborcasting", in the proceeding of the 37th IEEE International Conference on Communications (ICC'02), USA, April 2002.
- [SWe00] H. Schulzrinne, E. Wedlund, "Application-Layer Mobility Using SIP", ACM SIGMOBILE Mobile Computing and Communications Review, July 2000.

- [Tec06] Technical Paper, “UTRA-UTRAN Long Term Evolution (LTE) and 3GPP System Architecture Evolution (SAE)”, last update October 2006, available at: ftp://ftp.3gpp.org/Inbox/2008_web_files/LTA_Paper.pdf, accessed on 08.04.2009.
- [The04] G. Tulodziecki, B. Herzig, “Allgemeine Didaktik und Computerbasierte Medien”, in: U. Rinn, D. M. Meister, “Didaktik und Neue Medien. Konzepte und Anwendungen in der Hochschule”, printed by Waxmann, ISBN:3-8309-1216-1, 2004.
- [TNa98] S. Thomson, T. Narten, “IPv6 Stateless Address Auto-Configuration”, RFC 2462, December 1998.
- [TPL99] C. Tan, S. Pink, K. Lye, “A Fast Handoff Scheme for Wireless Networks”, in the proceeding of the 2nd ACM International Workshop on Wireless Mobile Multimedia, USA, August 1999.
- [TPR99] H.-Y. Tzeng, T. Przygienda, “On Fast Address-Lookup Algorithms”, IEEE Journal on Selected Areas in Communications, June 1999.
- [TRM03] J. Trnkova, G. Roeßling, M. Muehlhaeuser, “L4 - A Sophisticated Adaptive E-Learning Laboratory”, in the proceeding of Workshop Ontologie-basiertes Wissensmanagement (WOW'03), Switzerland, April 2003.
- [TWY06] C. Tseng, Y. Wong, L. Yen, K. Hsu, “Proactive DAD: A Fast Address-Acquisition Strategy for Mobile IPv6 Networks”, IEEE Internet Computing Journal, November 2006.
- [Typo3] TYPO3 official website: <http://typo3.com>, accessed on 17.03.2009.
- [Val99] A. G. Valko, “Cellular IP - A New Approach to Internet Host Mobility”, ACM Sigcomm Computer Communication Review Magazine, January 1999.
- [VKa04] H. Velayos, G. Karlsson, “Techniques to Reduce the IEEE 802.11b Handoff Time”, in the proceeding of the 39th IEEE International Conference on Communications (ICC'04), June 2004, available at: <http://winternet.sics.se/workshops/sncnw2003/proceedings/30T-Techniques%20to%20reduce%20IEEE%2080211b%20handoff%20time.pdf>, accessed on 07.04.2009.
- [VPK03] D. Vali, S. Paskalis, A. Kaloxylas, L. Merakos, “An Efficient Micro-Mobility Solution for SIP Networks”, in the proceeding of the 46th IEEE Global Telecommunications Conference (GLOBECOM '03), USA, December 2003.
- [VPK03a] D. Vali, S. Paskalis, A. Kaloxylas, L. Merakos, “A SIP-Based Method for Intra-Domain Handoffs”, in the proceeding of the 58th IEEE Vehicular Technology Conference (VTC'03), USA, October 2003.
- [WAb03] Q. Wang, M. A. Abu-Rgheff, “Integrated Mobile IP and SIP Approach for Advanced Location Management”, in the proceeding of the 4th International Conference on 3G Mobile Communication Technologies (3G'03), UK, June 2003.
- [WAG03] D. Wisely, H. Aghvami, S. L. Gwyn, T. Zahariadis, J. Manner, V. Gazis, N. Houssos, N. Alonistioti, “Transparent IP Radio Access for Next-Generation Mobile Networks”, IEEE Wireless Communications Magazine, August 2003.
- [WDM02] W. Wu, S. K. Das, A. Misra, S. Das, “Performance Evaluation of IDMP's QoS Framework”, in the proceeding of 50th IEEE Global Telecommunications Conference (GLOBECOM'02), Taiwan, November 2002.
- [WGu09] R. Wakikawa, S. Gundavelli, “IPv4 Support for Proxy Mobile IPv6”, Internet Draft <draft-ietf-netlmm-pmip6-ipv4-support-10.txt>, March 2009.
- [Wol00] A. Wolisz, “Wireless Internet Architecture: Selected Issues”, in the proceeding of the 8th IFIP International Conference on Personal Wireless Communications (PWC'00), Poland, September 2000.
- [WOL05] E. Weiss, A. Otyakmaz, E. López, B. Xu, “Design and Evaluation of a New Handoff Protocol in IEEE 802.11 Networks”, in the proceeding of the 11th European Wireless Conference 2005, Cyprus 2005.

- [XGra] Trace Graph - network simulator ns trace files analyser, official website: <http://www.geocities.com/tracegraph/>, accessed on 09.04.2009.
- [ZCB01] X. Zhao, C. Castelluccia, M. Baker, "Flexible Network Support for Mobile Hosts", ACM Mobile Networks and Applications Journal, March 2001.
- [ZCC00] X. Zhang, J. Castellanos, A. Campbell, K. Sawada, M. Barry, "P-MIP: Minimal Paging Extensions for Mobile IP", Internet Draft <draft-zhang-pmip-00.txt>, July 2000.
- [ZCC02] X. Zhang, J. G. Castellanos, A. T. Campbell, "P-MIP: Paging Extension for Mobile IP", ACM Mobile Networks and Applications Journal, April 2002.
- [ZVT02] B. T. Zahariadis, G. K. Vaxevanakis, P.C. Tsantilas, A. N. Zervos, A. N. Nikolaou, "Global Roaming in Next-Generation Networks", IEEE Communications Magazine, February 2002.
- [3GPP] 3rd Generation Partnership Project, official website: <http://www.3gpp.org/>, accessed on 05.04.2009.
- [3GPP-L] 3rd Generation Partnership Project Long Term Evolution (LTE), official website: <http://www.3gpp.org/Highlights/LTE/LTE.htm>, accessed on 05.04.2009.

Theses

- Ubiquitous access to information anywhere, anytime and anyhow, support large data volumes, minimal delay, low costs, etc. are the key features of future All-IP networks.
- One of the main challenges that should be overcome to reach the ambitious goals of future All-IP networks is the support of efficient and smooth mobility management.
- The mobility problem in IP-based mobile communication networks lies in the dichotomy of IP addresses since an IP address represents the point of attachment. Changing the IP address during an ongoing session typically results in a disruption of the application.
- Mobility management can be implemented in different layers of the TCP/IP reference model. Link layer mobility is responsible for the establishment of radio links after movements. If the new point of attachment belongs to a new subnet, additional mobility procedures are required. These procedures can be supported by the network layer, transport layer or application layer or by a hybrid approach involving several layers.
- Network layer mobility management solutions seem to be most suited to satisfy the requirements of future All-IP networks.
- The classical solution used to support mobility in IP-based networks is Mobile IP. This solution suffers, however, from many drawbacks, which have triggered the development of new mobility management solutions. These solutions attempt to improve the performance either by making constraints on the network or on the MN itself.
- The aim to satisfy the requirements of future All-IP networks has led to the development of a new layer 3 mobility management solution named Mobile IP Fast Authentication protocol (MIFA).
- MIFA advances the state of the art and supports a continuous communication between the mobile and its communication partner while the registration is in progress. It is a very fast mobility management scheme and capable of achieving smooth handoffs even while moving at high speeds.
- Analyses of mobility management protocols can be done using mathematical models, simulation studies or real implementations. Implementation and simulation are accurate and time consuming, while mathematical models can be developed more quickly and result in a good estimation of the performance.
- The development of a generic mathematical model that allows for the analysis of a wide range of mobility management solutions will be a major contribution. This dissertation has proposed such a model. The parameters of the generic model should be selected based on the characteristics of the studied protocols, mobility scenarios and network topologies.
- It is a great contribution to provide an adaptive eLearning environment capable of personalizing courses on mobility management issues, so that researchers quickly become involved in current research trends, while other learners are provided with courses containing the topics required to eliminate gaps in their knowledge. Such an environment has been developed in the scope of this dissertation.

