

Comparative Evaluation of the Mobile Internet Protocol 6 Suite

by

Johan Pieterse

*Thesis presented in partial fulfilment of the requirements for
the degree Master of Science in Engineering at Stellenbosch
University.*



Department of Electrical and Electronic Engineering,
University of Stellenbosch

Supervisor: Dr. R. Wolhuter

March 2015

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2015

Copyright © 2015 Stellenbosch University
All rights reserved.

Abstract

Comparative Evaluation of the Mobile Internet Protocol 6 Suite

J. Pieterse

*Department of Electrical and Electronic Engineering,
University of Stellenbosch*

Thesis: MSc(Eng)

March 2015

Mobile IPv6 is a proposed mobility standard for Next Generation Wireless Access Networks that allows mobile nodes, such as laptops, tablets, smart phones to stay reachable while moving around in an IPv6 Internet network. The need for MIPv6 exists because a mobile device cannot maintain the previously connected link when changing location and IP address. The initial IP Mobility protocol was first presented in 1993 for IPv4 and in 2004 for IPv6. The Mobile IP protocol solves the TCP/IP Stack Layer 3 mobility issue, by assigning a permanent IP Home Agent address to the mobile node. IPv4 has some drawbacks, the main one being IP address exhaustion, making MIPv6 the future option for mobility protocol in IP Networks. The main goal of the mobility protocol is to enable network applications to operate continuously at the required quality of service for both wired and wireless networks while the mobile node moves around in the network.

MIPv6 on its own needs optimization techniques to improve the handover latency of the protocol and to minimize the latency. This thesis investigates MIPv6 simulated using OMNeT++ Network Simulator Framework and the implementation thereof on a Linux IPv6 test bed. The test bed was used to test handover latency, overhead added by the MIPv6 extensions and packet loss. The developed test set up can also be used to evaluate different handover schemes that might enhance the MIPv6 protocol, decreasing handover latency and enabling real-time IPv6 applications such as Voice over IP. FMIPv6 and PMIPv6 are extensions to MIPv6 to enhance its functionality. These protocols are investigated and evaluated against MIPv6 in order to make recommendations on possible improvements of these mobility protocols.

Uittreksel

Comparative Evaluation of the Mobile Internet Protocol 6 Suite

J. Pieterse

*Departement Elektries en Elektroniese Ingenieurswese,
Universiteit van Stellenbosch*

Tesis: MSc(Ing)

Maart 2015

Mobiele IPv6 is 'n voorgestelde standaard vir mobiele netwerke of die sogenaamde Volgende Generasie Netwerke wat mobiele nodes sal toelaat om bereikbaar te bly wanneer die nodes rondbeweeg in 'n IPv6 Internet omgewing. Die behoefte aan 'n kontinue netwerksessie is baie groot en dit kan toegeskryf word aan die vinnige toename in mobiele nodes, soos skootrekenaars, tablette en slimfone. Die oorspronklike IP Mobiele protokol was voorgestel in 1993 vir IPv4 en in 2004 vir IPv6. Mobiele IP dien as 'n oplossing vir netwerk mobiliteit deur te fokus op Laag 3 van die TCP/IP Stapel. Kontinue sessies word bereik deur 'n permanente Basis Adres aan die mobiele node te bind. IPv4 het heelwat nadele, waarvan die grootste een verseker IP adres uitputting is, ander nadele sluit in driehoekige roetering en invloed filtering wanneer die mobiele node rondbeweeg. Weens die genoemde nadele van MIPv4 en die stelselmatige oorgang van IPv4 na IPv6 word die fokus gerig op MIPv6 vir toekomstige verbetering en implementering. Die hoofdoel van MIPv6 is om te sorg dat mobiele nodes kan rondbeweeg in 'n netwerk sonder om netwerk konneksie te verloor en ook om die gehalte van diens te handhaaf.

MIPv6 benodig addisionele optimalisering tegnieke om die oorhandigings latensie van die protokol te verbeter en dus die gehalte van diens ook te verbeter. In die tesis ondersoek ons die elemente wat oorhandigingstydperk beïnvloed en verhoog, deur MIPv6 in 'n OMNeT++ Simuleerder te evalueer. Nadat die nodige simulasiere resultate verkry is, word MIPv6 geïmplementeer op 'n toets netwerk om die oorhandigingstydperk te toets wanneer die node rondbeweeg, oorhoofse inligting wat bygevoeg word deur MIPv6 en die aantal netwerk pakkies wat verlore gaan tydens die oorhandigingsproses. Hierna word

die optimalisering tegnieke genaamd PMIPv6 en FMIPv6 ook geïmplementeer op die toets netwerk om die effektiwiteit en optimaliserings voordele teenoor die toets resultate van MIPv6 te vergelyk. Die resultate kan gebruik word om verbeterings en voorstelle te maak rakende die mobiele protokols.

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations...

Pioneers in Network Mobility:

Prof. Thomas Noël
Dr. Emil Ivov

University of Stellenbosch:

Dr. Riaan Wolhuter
Jaco du Toit
Joseph Wamicha
Anita van der Spuy

University of the North West (Potchefstroom):

Albert Sorgdrager
Hester Oelofse

University of Strasbourg:

Prof. Thomas Noël
Prof. Antoine Gallais
Dr. Emil Ivov
Dr. Damien Roth

INRIA:

Dr. Nathalie Mitton
Priyanka Rawat

Conferences:

IEEE-APS: Topical Conference on Antennas and Propagation in Wireless Communications, Cape Town, South Africa - 2012

4th International Conference on Ad Hoc Networks, Paris, France - 2012

Industry related:

Altech Academy

Altech ISIS

INRIA (National Institute for Research in Computer Science and Control)

DSP lab (University of Stellenbosch)

Niel Bester

Contents

Declaration	i
Abstract	ii
Uittreksel	iii
Acknowledgements	v
Contents	vii
List of Figures	ix
List of Tables	xi
Nomenclature	xii
1 Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Problem Description	4
1.4 Research Objectives	4
1.5 Methodology	5
1.6 Summary of Results and Complementary Work	6
1.7 Outline of Thesis	7
2 Literature Review	9
2.1 Introduction	9
2.2 Internet Protocol Version 6	10
2.3 Wireless Sensor Networks	23
2.4 OSI Layer Mobility	28
2.5 Mobile IPv6	31
2.6 Network Mobility	44
2.7 Fast Handover for Mobile IPv6	46
2.8 Proxy Mobile IPv6	55
2.9 Hierarchical Mobile IPv6	60

2.10	Discrete Event Simulator	61
2.11	Conclusion	68
3	Design and Implementation	70
3.1	Introduction	70
3.2	Simulation Model	71
3.3	MIPv6 Network	77
3.4	FMIPv6 Network	83
3.5	PMIPv6 Network	86
3.6	Simulation Model MIPv6 for 6LoWPAN	89
3.7	Conclusion	90
4	Results	92
4.1	Introduction	92
4.2	Evaluation Methodology	92
4.3	Simulation Results	93
4.4	MIPv6 Protocol Results	95
4.5	FMIPv6 Protocol Results	98
4.6	PMIPv6 Protocol Results	101
4.7	Comparing Protocol Results	105
4.8	MIPv6 for 6LoWPAN	110
4.9	Conclusion	111
5	Conclusion, Outcomes and Recommendations	112
5.1	Conclusion	112
5.2	Outcomes of Completed Work	114
5.3	Recommendations for Future Work	116
	Appendices	117
A		118
A.1	xMIPv6 Runtime Parameters	118
B		123
B.1	Router Advertisement Test Bed Configuration	123
B.2	MIPv6 Test Bed Configuration	123
B.3	FMIPv6 Test Bed Configuration	126
B.4	PMIPv6 Test Bed Configuration	128
C		132
C.1	MIPv6 Contiki Runtime Parameters	132
	List of References	133

List of Figures

2.1	IPv6 Packet Structure	15
2.2	IPv4 and IPv6 Header Differences [1]	16
2.3	Chaining Extension Headers in IPv6 Packets [1]	20
2.4	ICMPv6 Router Advertisement Message	23
2.5	Wireless Sensor Network (6LoWPAN Architecture)	24
2.6	IEEE 802.15.4 Uncompressed Frame	25
2.7	IEEE 802.15.4 HC1 Encoded Frame	26
2.8	IEEE 802.15.4 LOWPAN_IPHC Encoded Frame	27
2.9	MIPv6 Network	32
2.10	MIPv6 Message Flow	37
2.11	MIPv6 Bidirectional Tunneling Mode	38
2.12	MIPv6 Route Optimization Mode	39
2.13	Mobility Header for IPv6 Protocol	40
2.14	Message Data field in Mobility Header for Binding Update	42
2.15	Message Data in Mobility Header for BU Acknowledgement	43
2.16	MIPv6 Message Flow in 6LoWPAN Network	44
2.17	NEMO Network [2]	45
2.18	FMIPv6 Protocol	47
2.19	FMIPv6 Predictive Handover [3]	49
2.20	FMIPv6 Reactive Handover [3]	50
2.21	PMIPv6 Network	55
2.22	PMIPv6 Message Flow	58
2.23	PMIPv6 Handover Signaling Flow	59
2.24	HMIPv6 Network	61
2.25	OMNeT++ Simulation Environment	63
2.26	OMNeT++ IPv6 Compound Module	64
2.27	Analysis Editor	65
2.28	Analysis Editor Charts	65
2.29	ns-2 Network Animator	66
3.1	OMNeT++ MIPv6 Simulation Network	71
3.2	OMNeT++ Mobile Node Modules	72
3.3	MN Movement in Simulation	74
3.4	Components of Handover Latency - OMNeT++ Implementation	77

3.5	MIPv6 Testbed Overview	78
3.6	MIPv6 Testbed with IP Assignment	79
3.7	FMIPv6 Testbed with IP Assignment	84
3.8	PMIPv6 Testbed with IP Assignment	87
3.9	PMIPv6 Software Architecture [4]	88
3.10	MIPv6 6LoWPAN Testbed [5]	90
4.1	Handover Latency in MIPv6 OMNeT++ Simulation	93
4.2	RR and CN Handover Delay MIPv6 Simulation	94
4.3	UDP Throughput in MIPv6 OMNeT++ Simulation	94
4.4	TCP Throughput in MIPv6 OMNeT++ Simulation	95
4.5	UDP Throughput in MIPv6	96
4.6	UDP Throughput in MIPv6 During Handover	96
4.7	UDP Jitter in MIPv6	97
4.8	TCP Throughput in MIPv6	98
4.9	UDP Throughput in FMIPv6	99
4.10	UDP Throughput in FMIPv6 During Handover	99
4.11	UDP Jitter in FMIPv6	100
4.12	TCP Throughput in FMIPv6	101
4.13	UDP Throughput in PMIPv6	102
4.14	UDP Throughput in PMIPv6 During Handover	102
4.15	UDP Jitter in PMIPv6	103
4.16	TCP Throughput in PMIPv6	104
4.17	IP Camera TCP Throughput in PMIPv6	104
4.18	IP Camera TCP Packet Throughput in PMIPv6	105
4.19	MIPv6 UDP Throughput - OMNeT++ Simulation vs Testbed . . .	106
4.20	MIPv6 TCP Throughput - OMNeT++ Simulation vs Testbed . . .	106
4.21	UDP Throughput in MIPv6, FMIPv6 and PMIPv6 during Handover	107
4.22	UDP Throughput MIPv6, FMIPv6 and PMIPv6	108
4.23	TCP Throughput MIPv6, FMIPv6 and PMIPv6	108
4.24	Handover Latency of MIPv6, FMIPv6 and PMIPv6	109
4.25	UDP Packet Loss MIPv6, FMIPv6 and PMIPv6	109
4.26	UDP Jitter Comparison	110
4.27	MIPv6 WSN Handover Latency	111

List of Tables

2.1	Differences between IPv4 and IPv6 [6]	12
2.2	Special Syntax for IPv6 Addressing	14
2.4	IPv6 Address Type Identification [1]	15
2.5	IP protocol numbers for Next Header field [1]	18
2.6	Extension Headers for Next Header field	19
2.7	Differences in IPv4 and IPv6 Header Fields [7]	21
2.8	Types of ICMPv6 messages [7]	22
2.9	Mobility Header Types [8]	41
2.10	Comparison of OMNeT++ and ns-2 [9]	66
3.1	MIPv6 Parameters for Test Scenarios	75
3.2	IP Assignment for MIPv6 Network Interfaces	82
3.3	MIPv6 Test Bed Setup	83
3.4	FMIPv6 Test Bed Setup	85
3.5	IP Assignment for FMIPv6 Network Interfaces	86
3.6	IP Assignment for PMIPv6 Network Interfaces	89
3.7	PMIPv6 Test Bed Setup	89
4.1	Protocol Results	110
A.1	MIPv6 OMNeT++ Simulation Parameters	118
B.1	MIP6D MN Parameters	124
B.2	MIP6D HA Parameters	125
B.3	MIP6D CN Parameters	125
B.4	FMIPv6 MN Parameters	126
B.5	FMIPv6 PAR Parameters	127
B.6	FMIPv6 NAR Parameters	127
B.7	PMIPv6 LMA Parameters	128
B.8	PMIPv6 MAG1 Parameters	129
B.9	PMIPv6 MAG2 Parameters	130
C.1	WSN MIPv6 Contiki Simulation Parameters	132

Nomenclature

Acronyms: General Network Terms

3GPP	3rd Generation Partnership Project
API	Application programming interface
ARP	Address Resolution Protocol
BDP	Bandwidth Delay Product
DAD	Duplicate Address Detection
DES	Discrete Event Simulator
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol Version 6
DNS	Domain Name System
DOS	Denial Of Service
FTP	File Transfer Protocol
GNED	Graphical Network Editor
HMIPv6	Hierarchical Mobile Internet Protocol 6
ICMPv6	Internet Control Message Protocol for IPv6
ICMPv4	Internet Control Message Protocol for IPv4
IEFT	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IoF	Internet of Things
IPsec	Internet Protocol security
IPv6	Internet Protocol 6
ITS	Intelligent Transportation Systems
LAN	Local Area Network
LTE	Long Term Evolution
MLD	Multicast Listener Discovery
NED	Network Description
NDP	Neighbor Discovery Protocol
NIC	Network Interface Card
NUD	Neighbor Unreachability Detection
OSI	Open Systems Interconnection
OTcl	Object oriented Tool Command Language
PDU	Protocol Data Unit

QoS	Quality of Service
TCP	Transmission Control Protocol
TOS	Type of Service Field
UDP	User Datagram Protocol
WSN	Wireless Sensor Network

Acronyms: Mobile Internet Protocol 6

AP	Access Point
BU	Binding Update
CN	Correspondent Node
CoA	Care-of Address
CoT	Care-of Test
FA	Foreign Agent
FN	Foreign Network
HA	Home Agent
HN	Home Network
HoT	Home Test
Kbm	Binding Management Key
MIPv6	Mobile Internet Protocol 6
MH	Mobility Header
MN	Mobile Node
RA	Router Advertisement
RR	Return Routability

Acronyms: Fast Handover for Mobile Internet Protocol 6

AR	Access Router
FBack	Fast Binding Acknowledgment
FBU	Fast Binding Update
FMIPv6	Fast Handover for Mobile Internet Protocol 6
HAck	Handover Acknowledge
HI	Handover Initiate
NAR	New Access Router
NCoA	New Care-of Address
PAR	Previous Access Router
PCoA	Previous Care-of Address
PrRtAdv	Proxy Router Advertisement
RtSolPr	Router Solicitation for Proxy Advertisement
UNA	Unsolicited Neighbor Advertisement

Acronyms: Proxy Mobile Internet Protocol 6

AAA	Authentication, Authorization, and Accounting
LMA	Local Mobility Anchor
LMAA	LMA Address
MAG	Mobile Access Gateway
MN-Identifier	Mobile Node Identifier
MN-HNP	Mobile Node's Home Network Prefix
MN-HoA	Mobile Node's Home Address
MN-LL-Identifier	Mobile Node Link-layer Identifier
NETLMM	Network-based Localized Mobility Management
PBA	Proxy Binding Acknowledgement
PBU	Proxy Binding Update
PMIPv6	Proxy Mobile Internet Protocol 6
Proxy-CoA	Proxy Care-of Address
RADIUS	Remote Access Dial In User Service

Acronyms: Hierarchical Mobile Internet Protocol 6

LCoA	On-Link Care-of Address
MAP	Mobility Anchor Point
RCoA	Regional Care-of Address

Acronyms: IPv6 over Low power WPAN

6LoWPAN	IPv6 over Low Power Wireless Personal Network
BR	Border Router
IPHC	IP Header Compression
NHC	Next Header Compression
WSN	Wireless Sensor Networks

Acronyms: Network Mobility Basic Support Protocol

NEMO	Network Mobility Basic Support Protocol
------	---

Chapter 1

Introduction

1.1 Background

The initial IP Mobility protocol was first presented in 1993 for IPv4 [10]. Mobile IP protocol solves TCP/IP Layer 3 mobility, because the mobile node has a permanent IP address assigned to it even when moving around in networks. Mobile IP includes both MIPv4 and MIPv6 but IPv4 has some drawbacks for mobility, the main drawback being IP address exhaustion, which makes MIPv6 the future option for mobility protocols in IP Networks [10]. The main goal of the mobility protocol is to enable network applications to operate continuously at the required quality of service for both wired and wireless networks [11]. MIPv6 uses the existing IPv6 protocol to enable seamless roaming between different access points [12]. MIPv6 on its own needs optimization techniques to improve the handover latency of the protocol and to minimize latency caused by control message exchange. To date, there are various technologies under development for enhancing and optimizing the existing MIPv6 protocol. The Mobile IPv6 Suite comprises various variations of the standard MIPv6. The main variations being:

- Proxy Mobile IPv6 (PMIPv6)
- Fast Handovers for Mobile IPv6 (FMIPv6)
- Hierarchical Mobile IPv6 (HMIPv6)

FMIPv6 proposes some enhancements to minimize the handover latency of a MIPv6 network by reducing latency caused by control message exchange and Duplicate Address Detection between two nodes. PMIPv6 focuses on the mobility of the network as a whole and requires no adaptation by the mobile node to accommodate mobility. HMIPv6 focuses to reduce the signaling between the mobile node, the correspondent node and the home agent in the specified network. HMIPv6 reduces binding update message overhead in large scale mobility networks [13].

The following list indicates the main requirements for MIPv6 protocols and mobile networks [14]:

- **Migration Transparency:** Permanent connectivity to the Internet has to be provided to all times, since continuous sessions are expected by the mobility protocol.
- **Performance Transparency and Seamless Mobility:** Limited network overhead should be added by mobility messages needed for MIPv6. The mobility protocol needs to decrease handover latency and packet loss during handover.
- **Operational Transparency:** MIPv6 is to be implemented on the IP layer and is expected to be transparent to upper layers. This means that any upper-layer protocol can run unchanged on top of the extended mobile IP layer.
- **Arbitrary Configurations:** This should allow the mobile node to change their point of attachment within the mobile network. Should allow connection of foreign mobile nodes in the network.
- **Scalability:** A mobile network is expected to scale to a large number of mobile nodes, correspondent nodes and HA.
- **Backward Compatibility:** A mobile network should be able to function with established IPv6 standards. Address allocation, configuration mechanisms, security protocols and access control protocols should function normally.
- **Secure Signaling:** A mobile network should comply with usual IETF security policies and recommendations.

MIPv6 is also being adapted to be used in Sensor Networks. Due to the evolution of sensor networks to accommodate non-static sensor nodes, the need for a mobility mechanism is ever increasing. Sensor nodes are very low powered nodes and also not very powerful when it comes to computational power. Sensor nodes thus use scaled down versions of the IPv6, namely 6LoWPAN. 6LoWPAN makes use of header compression techniques allowing the sending and receiving of packets over IEEE 802.15.4 based networks [15]. MIPv6 can be used as a mobility protocol for these networks but control signaling and neighborhood discovery needs to be adapted to accommodate low power devices [5].

1.2 Motivation

The main problem with IPv4 is the relatively small address space that's available, this is caused by the decision to only use 32 bits to compile the IP ad-

dress [16]. Under the original classful addressing allocation scheme for IPv4, the number of available addresses decreased rapidly due to the vast expanding number of internet devices. Moving to classless addressing has helped postpone the exhaustion of addresses, as has technology like IP Network Address Translation (NAT), that allowed privately-addressed hosts Internet access.

The current 32-bit address space is too small for the present and future size of the Internet. The only solution to this problem is to enlarge the address space to 64-bit addressing. This led to the next version of the Internet Protocol, IPv6 [17]. Implementation of IPv6 was officially launched on 6 June 2012, and since then the global use of IPv6 has more than doubled [18]. It is clear that IPv6 will be the future network layer protocol used by standard networks, as well as by mobile networks.

As the number of mobile devices is increasing very rapidly, the need for a network mobility protocol becomes ever more demanding. The increasing number of mobile users leads to a problem in that when the mobile user moves from one location to another, the messages between the user and the network must be able to follow him. IPv6 mobility support is particularly important, as laptops, smart phones, tablets etc. are likely to account for the biggest or at least a substantial number of the Internet population. MIPv6 [8] was introduced as a proposed layer 3 mobility solution. During the handover period, there is a delay caused by IP protocol operations and link switching, where the mobile node is unable to send or receive packets. The handover latency resulting from Mobile IPv6 procedures needs to be decreased to allow satisfactory Quality of Service (QoS) for users. The primary challenge of MIPv6 is to eliminate the handover latency caused by ISO layer 3 control messages, this has led to various extensions of MIPv6.

FMIPv6 is proposed in [19] which aims at reducing the handover latency and related packet losses by performing operations such as movement detection and new CoA configuration (including Duplicate Address Detection (DAD) for the newly formed Address) in advance. In order to determine whether FMIPv6 achieves this, the testing and evaluation of the protocol on test platforms are of utmost importance. PMIPv6 introduces a network based mobility management approach, with minimal client participation. As a result of this, the Proxy Mobile IPv6 [20], a network-based mobility management protocol was standardized in 2008. The protocol was designed with the goal that the network would perform the mobility management on behalf of the client, PMIPv6 would keep the client involvement to minimal proportions, such as allowing the client to perform inter-technology handoffs or allowing the client to express handoff or flow preferences [21]. Various link layer technologies exist today, but this thesis will focus on the use of WLAN, 802.11n. Wireless LAN (Wi-Fi) is arguably becoming one of the most popular and easiest ways to connect to the Internet, because of its easy setup, relatively low deployment cost and fast network speed.

Wireless sensor networks (WSN) have recently been proposed for a large

range of applications in home automation, industrial or environmental monitoring and health care [5]. Recently, the research community has tried to enable IPv6 connectivity in WSN. Such support would provide numerous advantages to WSN. Enabling MIPv6 on low power devices need a great need of investigation into adapting MIPv6 for better support.

The subject of this work is the investigation, analysis, implementation and testing of the above mentioned mobility protocols, which allow for network mobility in an all mobile Internet world.

1.3 Problem Description

The standard IP network design introduces one main issue that can be addressed by Mobile IP. The issue being that a mobile node will always have a different IP address when moving between networks. This introduces a problem in the transport layer session, used by TCP connections and UDP based transactions. Data connections are lost when a node moves between networks because of the change in IP address. For example when a mobile node has an TCP video stream connection with a correspondent node and the mobile moves to a different network the data connection will be lost. The Mobile IP design aims to solve these issue by allowing the mobile node to always be reachable with a static IP, which is called the Home of Address (HoA). By allowing the mobile node to be reachable at the HoA the transport session layer stays connected even when the mobile node moves to a visited network.

MIPv6 do not improve the Layer 2 handover between a mobile node and an access point. Thus FMIPv6 is introduced to decrease handover latency caused by MIPv6 and standard IPv6 messaging. FMIPv6 proposes a solution to remove the latency caused by Duplicate Address Detection and to decrease the Layer 2 handover latency between the access point and the mobile node, but still keeping the mobile connected via the same IP address.

For a network to be MIPv6 compatible, the mobile node, correspondent node and the home agent needs to have the MIPv6 IP stack. This introduces the need for modification of every mobile node's IP layer, which is not ideal. PMIPv6 is proposed to move the mobility from the mobile node to the network. Thus only the network needs to be MIPv6 compatible and the mobility is transparent for the mobile node. The objectives for this thesis are set out in the next section.

1.4 Research Objectives

No real comparative information could be found in literature on the relative performance of the mobility protocols and in view of the increasing importance of these protocols in current and new envisaged applications in mobile data

communications, the main objectives for the work done under this project were defined as follows:

- To investigate and evaluate mobility protocols, as well as the associated and required handover strategies, in a Mobile IPv6 network.
- To evaluate the capability of the IPv6 network protocol in a mobile environment was also evaluated.
- To evaluate the efficiency and reliability of the different handover strategies for MIPv6, as this is the main cause of packet loss and thus a reduction in service quality in mobile networks. A particular effort in this regard was made to implement FMIPv6 on the network to test reduction in handover latency and evaluate possible improvements over the handover latency of MIPv6.
- To establish a simulation environment suitable for present and future trial purposes for MIPv6.
- To create a physical and realistic testbed for the different protocols as a complementary evaluation platform to the simulation environment.
- The implementation of MIPv6 in Wireless Sensor Networks, was seen as an outside mainstream application, but nevertheless important, in view of the increasing deployment of WSN's. This was investigated by implementation and simulation of MIPv6 in the Contiki OS wireless environment. It was shown that MIPv6 is effective as a WSN mobility protocol.
- The dual simulation/hardware based approach enabled the comparison of different strategy performance and enhanced the credibility of the results.

The establishment of a dual evaluation vehicle was seen as important in terms of more detailed application driven future work.

1.5 Methodology

The following methodologies were applied:

- Provide the relevant theory and working of the following Protocols:
 - Internet Protocol version 6 (RFC 2460)
 - IPv6 over Low Power WPAN (RFC 6282)
 - Mobile IPv6 (RFC 6275)

- Fast Handover for Mobile IPv6 (RFC 5568)
 - Proxy Mobile IPv6 (RFC 5949)
 - Network Mobility (RFC 3963)
 - The Internet of Things
- Provide the relevant theory and working of the simulation engine to be used as simulation platform for MIPv6.
 - Design and develop a IPv6 testbed, in order to evaluate different proposed mobility protocols.
 - Implement MIPv6 on the IPv6 testbed to evaluate protocol performance and handover latency.
 - Implement FMIPv6 on the network to evaluate the improvements on the handover latency of MIPv6.
 - Implement PMIPv6 on the network to compare the protocol with MIPv6.
 - Investigate MIPv6 for low power devices such as sensor networks with the use of the Cooja simulation program.
 - Evaluate and present possible improvements regarding MIPv6.
 - Evaluate the protocols to conclude the best mobility approach for mobile IPv6 networks and propose enhancement for mobility in IPv6 networks.

The various protocols will be tested with real world data, like VoIP or video streaming over the test network.

1.6 Summary of Results and Complementary Work

The main outcomes of the thesis are outlined as follows:

- A detailed investigation into IPv6 and the enhancement of IPv6 for mobility of networks.
- Explanations of why mobility in networks is necessary and problems existing today in non-mobile networks.
- A thorough investigation and explanation of proposed network layer and non-network layer mobility protocols.
- Implementation and performance evaluation of MIPv6 in OMNeT++ network simulator.

- Comparison of simulation results and physical results of testbed to improve the performance of the MIPv6 simulation module.
- Design and building of an IPv6 test platform to evaluate different network protocols in the IPv6 environment.
- Implementation, configuration and testing of MIPv6 in a physical testbed environment.
- Implementation, configuration and testing of Proxy Mobile IPv6 on top of MIPv6 in a physical environment.
- Implementation, configuration and testing of FMIPv6 to decrease handover latency in MIPv6.
- Investigation of handover strategies to reduce handover latency in 802.11 networks.
- Identification of the most suitable mobility protocol for further enhancements and development.
- Complete evaluation of all the mobility protocols and proposed problems with large scale implementation of protocols.
- Investigation of a compression mechanism to enable IPv6 on low power devices.
- Investigation of MIPv6 on a 6LoWPAN architecture.
- It was also found interesting enough by the general research community, to accept two research papers for an European conference and an IEEE conference:
 - IEEE-APS: Topical Conference on Antennas and Propagation in Wireless Communications, Cape Town, South Africa - 2012 [22].
 - 4th International Conference on Ad Hoc Networks, Paris, France - 2012 [23].

1.7 Outline of Thesis

In this section, an overview of the topics is presented and they are explained. An overview is given in Chapter 2 of all the relevant network layer mobility protocols, these protocols include Mobile IPv6, Proxy Mobile IPv6 and Fast Handover for Mobile IPv6. An introduction is also given to the changes and main advantages of Internet Protocol Version 6 compared to Internet Protocol Version 4. IPv6 is also explained in more detail and mobility support in IPv6 is explained in detail. Furthermore Chapter 2 looks at Simulation Modules

and Engines. The main focus of this Chapter is to explain mobility protocols and IPv6.

The simulation module, MIPv6, FMIPv6 and PMIPv6 implementation are presented in Chapter 3. Detailed implementation of the OMNeT++ MIPv6 simulation module and functioning thereof is described. Chapter 3 includes detailed network designs for all the mobility protocols and how these protocols are implemented in the physical test environment. The design includes hardware setups, software configuration, network configuration, network performance tools and handover scenarios for testing. In Chapter 4 all results obtained from the simulation and testbed are presented and evaluated. Comparisons are made between the MIPv6 simulation and the MIPv6 Testbed. The different mobility protocols are compared as regards to TCP, UDP, Jitter and Handover Latency results. The results also show that FMIPv6 reduces the handover latency tremendously compared to MIPv6 and PMIPv6. Lastly, Chapter 4 shows the advantages of PMIPv6 due to the low effect and no changes on the Mobile Node, performance is also relatively good compared to that of MIPv6.

Finally, in Chapter 5, the research findings and results are discussed in detail. A conclusion is arrived at using the related results to identify a mobility protocol for future research and development. A conclusion is also given on the performance of these protocols with live data streaming, such as VoIP and Video.

Chapter 2

Literature Review

2.1 Introduction

As the number of mobile devices is increasing very rapidly, the need for a network mobility protocol becomes ever more urgent. The current exhaustion of IPv4 addresses paves the way for a worldwide implementation of IPv6. The increasing number of mobile users presents the problem that when the mobile user moves from one location to another, the messages between the user and the network must be able to follow them. Mobility was not kept in mind during the design and implementation of IPv4, but later mobility extensions were added to allow for more mobility in IPv4. IPv6 has a better designed solution for network mobility. It is clear that IPv6 will be the future network layer protocol used by standard networks, as well as by mobile networks. Various mobility protocols are introduced to accommodate the movement of nodes and to allow continuous throughput when moving around. The main mobility protocol is Mobile IPv6.

Recently many alternative additions have been introduced to the MIPv6 framework to allow various enhancements. Some of the proposed enhancements to MIPv6 can be seen as Proxy Mobile IPv6, Fast Handovers for MIPv6 and Hierarchical MIPv6. Hierarchical MIPv6 focusses on reducing network mobility messages in large scale networks and will thus not be evaluated in this thesis. Mobility features are also being considered for low power devices in order to solve issues introduced by the Internet of Things [24]. To accommodate IPv6 in low power devices, 6LoWPAN [15] was introduced to solve IPv6 connectivity in a simple low cost communication network. The main reason low power nodes are not able to use standard IPv6 is because of their limited processing power, the energy efficiency of nodes and the huge decrease in header space. Various IPv6 header encoding mechanisms, such as HC1 [15] and LOWPAN_IPHC [25] were introduced to decrease the header space and enable IPv6 for low power devices. MIPv6 is also an option as mobility protocol for 6LoWPAN networks, due to availability of IPv6 connectivity in

6LoWPAN networks.

This chapter presents a literature review looking at various differences between and enhancements of IPv4 and IPv6, such as auto-configuration, expanded IP addresses, enhanced mobility and better Quality of Service(QoS). An in-depth explanation of the mobility protocols, namely MIPv6, FMIPv6 and PMIPv6 is included in this chapter. IPv6 encoding mechanisms for 6LoWPAN networks is discussed and explained. Lastly the chapter focuses on various network simulation engines that are available.

2.2 Internet Protocol Version 6

With the rapid expansion of the Internet the current IPv4 is becoming exhausted and will make way for the next generation IP version 6 that is designed to enable ongoing expansion of the Internet. The current IP architecture requires an evolution to handle the constant growth of the Internet, the increasing number of users and applications, as well as the accommodating new technologies [6, 26].

2.2.1 IPv6 New Features

IPv6 offers integrated IP services such as auto-configuration, expanded IP addresses, end-to-end security, enhanced mobility and QoS. IPv6 was intentionally design to introduce minimal impact on lower and upper layer protocols. This is accomplished by building the functionality into to the architecture and avoiding the random addition of new features as in IPv4 [6].

- Large address space: IP address size is increase from 32 bits to 128 bits in IPv6, as seen in Figure 2.2. This increase in addressing space allows for a greater number of addressable nodes, more levels of addressing hierarchy and easier auto-configuration of addresses. A scope field was added to the multicast addresses to improve the scalability of multicast routing. A new type of address called the anycast address was also defined in IPv6 and is used for sending a packet to a group of nodes.
- Extensibility: IPv6 has the ability to easily add new features, this is done by adding new extension headers after the IPv6 header. The number of extension headers that can be added to IPv6 is only constrained to the packet size of IPv6, but in IPv4 only 40 bytes of options is supported.
- Improved support for prioritized delivery of packets: Packets belonging to specific traffic flow will be labeled if the sender requested special handling of the packets. The special handling request can be presented by non-default quality of service or real-time service. IPsec, mobility and support for prioritized delivery was built into IPv6.

- New header format: IPv6 offers a new header format that provides more efficient processing at intermediate routers and also reduces header overhead.
- Efficient routing infrastructure: A hierarchical, efficient and summarizable routing infrastructure is created by the use of Global Addresses on the IPv6 portion of the Internet.
- New neighbouring node interaction protocol: The Neighbour Discovery protocol for IPv6 is a series of Internet Control Message Protocol for IPv6 (ICMPv6) messages. These messages are responsible for interaction between neighbouring nodes. In IPv4 the broadcast-based Address Resolution Protocol (ARP), ICMPv4 Redirect and ICMPv4 Router Discovery messages are replaced in IPv6 with efficient multicast and unicast Neighbour Discovery messages.
- Security: IPv6 uses Built-in security provided with Internet Protocol security (IPsec) which is not optional, as in IPv4
- Address configuration: IPv6 host makes use of stateless and stateful address configuration.
- Duplicate Address Detection (DAD): DAD is defined to avoid address duplication on links where stateless address autoconfiguration is used [19].

Table 2.1 lists the differences between IPv4 and IPv6. There are more differences than mentioned in Table 2.1, but all are not necessary for the scope of this work.

Table 2.1: Differences between IPv4 and IPv6 [6]

IPv4	IPv6
IPsec is optional security in IPv4.	IPsec support is required.
IP Address is 32-bits long.	IP address is 128-bits long.
IPv4 header doesn't support packet flow identification by routers for QoS	Flow Label Field is used for packet flow identification for QoS handling by routers and is included in the IPv6 Header Field.
Routers and sending host do fragmentation.	Fragmentation is only done by the sending host and not by routers.
Checksum is included in the header.	Checksum is not included in the header.
Header include Options field.	No Options field in IPv6, but Extension Headers are used for optional data.
ARP uses broadcast request frames to resolve IPv4 addresses.	Multicast Neighbour Solicitation messages replace ARP request frames.
Internet Group Management Protocol (IGMP) is used to maintain subnet groups.	IGMP is replaced by Multicast Listener Discovery messages.
To determine best default gateway IPv4 address, ICMP Router Discovery is used, but is optional.	ICMP Router Discovery is replaced by ICMPv4 Router Solicitation and Router Advertisement messages, which are not optional.
Broadcast messages are used to send messages to all nodes on subnet.	No IPv6 broadcast messages. Link-Local scope all-nodes multicast address is used.
Configuration only manually or with DHCP.	Address autoconfiguration, does not require manual or DHCP.
DNS - Uses host address (A) resource records to determine host name IPv4 address	DNS - Uses host address (AAAA) resource records to map host name IPv6 address.
Packet size of 576-bytes must be supported and can be fragmented.	Packet size of 1280-bytes must be supported without fragmentation.

2.2.2 IPv6 Addressing

In this section the IPv6 addressing model, addresses, definition of IPv6 unicast addresses, anycast addresses and multicast addresses is looked at. IPv6

addresses are 128-bit identifiers and consist of three types of addresses:

- **Unicast:** Unicast addresses are used for a individual interface. If a packet is sent with an unicast address, it can be delivered only to the interface identified by that unicast address.
- **Anycast:** Anycast addresses is an identifier for multiple interfaces (typically belonging to different nodes). A packet with an anycast address will be delivered to anyone of the interfaces, usually the closest node calculated by the routing protocol.
- **Multicast:** Multicast addresses is an identifier for multiple interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces containing the multicast address.

IPv6 does not have Broadcast addresses, as the function of broadcast messages is replaced by multicast messages. An interface can be assigned with all the IPv6 addressing types. Each interface belongs to an individual node, and this means that any of that node's interface unicast addresses can be used to identify the node. Each interface must have at least one Link-Local unicast address, but can also have multiple addresses which can be of any type or scope. In IPv4 one link is associated with a subnet prefix and IPv6 currently continues with this model. IPv6 has the functionality to assign multiple prefixes to the same link [1].

IPv6 addresses can written in three conventional forms of text string:

- The preferred IPv6 addressing form is $x:x:x:x:x:x:x$, where the x 's represent one to four hexadecimal digits of the eight 16-bit pieces that form the address. Leading zeros can be left out, but there must be at least one numeral in every field. Examples:

ABCD:EF02:1234:5678:ABCD:EF02:1234:5678

2001:DB6:0:0:9:800:200C:417A

- The methods of allocating the IPv6 addresses can cause an address to contain a long string of zero bits. To make the reading and writing of the address easier, a special syntax can be used for this. The use of '::' indicates single or multiple groups of 16 bits of zeros and can only appear once in an address. The previous mentioned syntax can also be used to compress leading or trailing zeros in an address. Table 2.2 indicates how the special syntax can be applied to IPv6 addressing.

Table 2.2: Special Syntax for IPv6 Addressing

Original Address	Special Syntax Address	Address Type
2001:0DB8:0:0:8:800:100C:500A	2001:DB8::8:800:100C:500A	Unicast address
FF01:0:0:0:0:0:0:1	FF01::1	Multicast address
0:0:0:0:0:0:0:1	::1	Loopback address
0:0:0:0:0:0:0:0	::	Unspecified address

- An alternative form can be used when the environment consist of both IPv4 and IPv6, x:x:x:x:x:x:d.d.d.d. The x's represent the six high-order 16-bit pieces in hexadecimal value and the d's represent the lower-order 8-bit pieces in decimal value. Examples:

- 0:0:0:0:0:0:12.2.70.4
- 0:0:0:0:0:FFFF:128.133.42.28

or in special syntax form:

- ::12.2.70.4
- ::FFFF:128.133.42.28

IPv6 prefix addresses can be written as follows: IPv6-address/prefix-length. Were IPv6-address represent the actual IPv6 address of the interface and the decimal value prefix-length specifies how many of the address bits are used for the prefix. For example, the following are legal representations of the 60-bit prefix 20010DB90000CE4 (hexadecimal):

- 2001:0DB9:0000:CE40:0000:0000:0000:0000/60
- 2001:0DB9::CE40:0:0:0:0/60
- 2001:0DB9:0:CE40::/60

If you want to combine the node address:

- 2001:0DB9:0:CE40:123:4567:88AB:CDEG

with the subnet number:

- 2001:0DB9:0:CE40::/60

the address result in:

- 2001:0DB8:0:CD30:123:4567:88AB:CDEG/60

Table 2.4 shows how the different types of IPv6 address are identified. The high-order bits are used to identify the type of the address. No distinguish between anycast and unicast addresses can be made, because anycast addresses are derived from unicast messages. Unicast addresses can be of several types, in particular global, link-local and site-local unicast. The latter type has been deprecated. A link-local address is intended only for communications within the segment of a local IPv6 network or a point-to-point connection that a host is connected to. Link-local addresses will not be forwarded by routers [27].

Table 2.4: IPv6 Address Type Identification [1]

Address type	Binary prefix	IPv6 format
Unspecified	00...0 (128 bits)	::/128
Loopback	00...0 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Site-Local unicast	1111111011	FEC0::/10

2.2.3 IPv6 Header

Figure 2.2 shows the differences between the IPv4 and IPv6 header. The advantage of the simpler and more efficient IPv6 header is that it greatly reduces processing costs. The IPv6 header is stripped of unnecessary or rarely used fields. The header also provides better support for real-time applications by the addition of new fields to the header. This section looks into the IPv6 header and shows the differences between the IPv4 and IPv6 header.

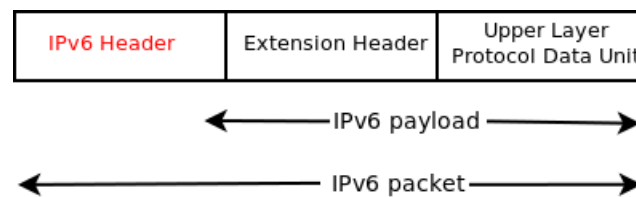


Figure 2.1: IPv6 Packet Structure

Figure 2.1 illustrates the structure of an IPv6 packet. The most important part of the packet is the IPv6 header which is described in more detail. The structure of the packet consists of the following components, as seen in Figure 2.1:

- IPv6 Header: The IPv6 header is a fixed size of 40 bytes.

- Extension Headers: The extension header format is very flexible. This flexibility ads support for future needs and capabilities for IPv6. The extension headers replace the Option field in the IPv4 header, as seen in Figure 2.2.
- Upper Layer Protocol Data Unit (PDU): The Upper Layer PDU is constructed of a PDU header and a payload, the payload can consist of an UDP message, TCP segment or ICMPv6 message.

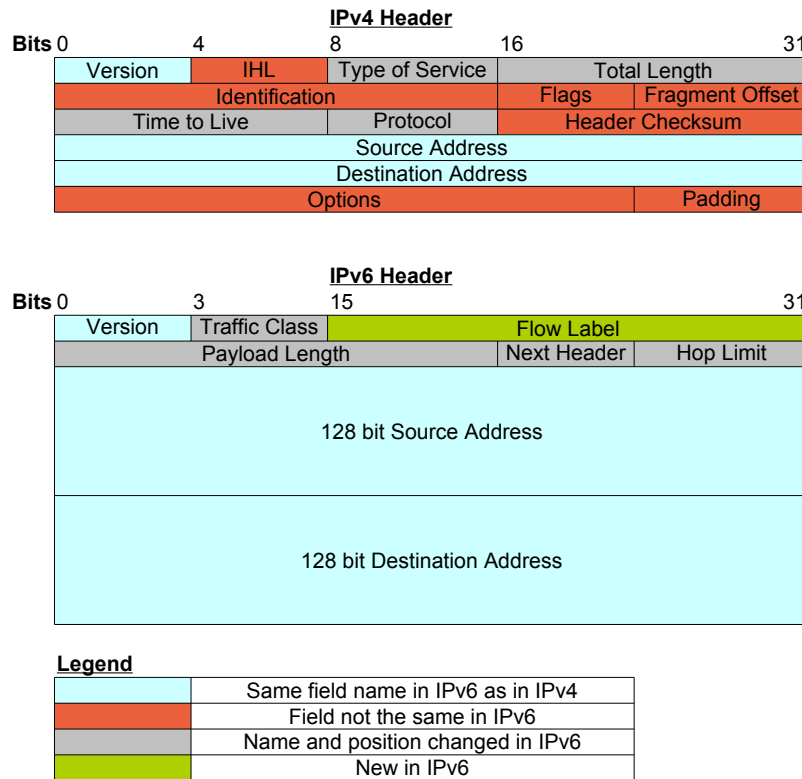


Figure 2.2: IPv4 and IPv6 Header Differences [1]

The IPv6 header is made up of eight fields, as shown in Figure 2.2. IPv6 header fields are:

- Version: This field contains the version number of the Internet Protocol version used, in our case it is version 6. This is a 4-bit field.
- Traffic Class: The Type of Service Field (TOS) in IPv4 is replaced by this 8-bit field. The traffic class field is used by forwarding routers and/or

originating nodes to identify and distinguish IPv6 packets that have different priorities or classes. The Traffic Class field is important to facilitate in the handling of data that requires special handling, such as real-time data.

- Flow Label: Flow Label is a 20-bit field. This field is used to indicate that a packet belongs to a certain sequence of packets between a source and destination address. Flow Label is a newly added field.
- Payload Length: The size of the payload field is 16 bits. The length consist of the length of the extension headers, plus the upper layer PDU's.
- Next Header: The field was renamed in IPv6 to reflect the new organization of IP packets. This 8-bit field specifies the type of the next header, if it was UDP or TCP the value will be 17 or 6. This field will contain the various types of Extension headers when used. Table 2.5 lists some of the Next Header value options available.

Table 2.5: IP protocol numbers for Next Header field [1]

Value	Description
0	IPv6 Hop-by-Hop Option
1	Internet Control Message Protocol v4 - IPv4 Support
2	Internet Control Message Protocol v4
4	IPv4 Encapsulation
5	Internet Stream Protocol
6	Transmission Control Protocol
8	Exterior Gateway Protocol
17	User Datagram Protocol
41	IPv6 (encapsulation)
43	Routing Header for IPv6
44	Fragment Header for IPv6
45	Inter-Domain Routing Protocol
46	Resource Reservation Protocol
50	Encapsulating Security Payload (ESP)
51	Authentication Header (AH)
55	IP Mobility (Min Encap)
58	ICMP for IPv6
59	No Next Header for IPv6
60	Destination Options for IPv6
88	Enhanced Interior Gateway Routing Protocol
89	Open Shortest Path First
108	IP Payload Compression Protocol
115	Layer Two Tunneling Protocol Version 3
132	Stream Control Transmission Protocol
135	Mobility Header
140	Site Multihoming by IPv6 Intermediation
141-252	Unassigned
255	Reserved

- Hop Limit: Replaces the Time to Live field in IPv4. The value of this field is decremented by one at each intermediate node. As soon as the hop limit reaches zero, the packet will be discarded.
- Source Address: The 128-bit IPv6 address of the sending node and not a 32-bit address as in IPv4.
- Destination Address: The 128-bit IPv6 address of the destination node(s) and not a 32-bit address as in IPv4.

Optional internet-layer information in IPv6 is encoded in separate headers. These headers can be placed between the IPv6 header and the upper-layer header. The current IPv6 specification (RFC 2460) defines a small number of such extension headers, each identified by a unique Next Header value as listed in Table 2.6. If there are no more extension headers, the next header field will be a upper-layer header, like UDP, TCP, ICMPv6 header or other next header values[17].

Table 2.6: Extension Headers for Next Header field

Value	Name	Description
0	IPv6 Hop-by-Hop Option	This option must be examined by all devices on the path.
43	Routing Header for IPv6	Methods to specify the route for a datagram (used with Mobile IPv6).
44	Fragment Header for IPv6	Contains parameters for fragmentation of datagrams.
50	Encapsulating Security Payload (ESP)	Carries encrypted data for secure communication.
51	Authentication Header (AH)	Contains information used to verify the authenticity of most parts of the packet.
60	Destination Options for IPv6	Only examined by the destination of the packet.
135	Mobility Header	Parameters used with Mobile IPv6.

Figure 2.3 shows how the Extension Headers of 8 octets are used. The number of Extension Headers used can be zero, one or multiple. Extension headers are only processed by the node that is identified as the Destination Address field of the IPv6 header. Extension headers of a multicast destination address are processed by all the nodes in the multicast group. The order in which the Extension Headers appear is also the order in which they must be executed. If a node cannot process the Next Header field, the packet must be discarded and a ICMPv6 Parameter Problem message is sent back to the IPv6 Source Address [7].

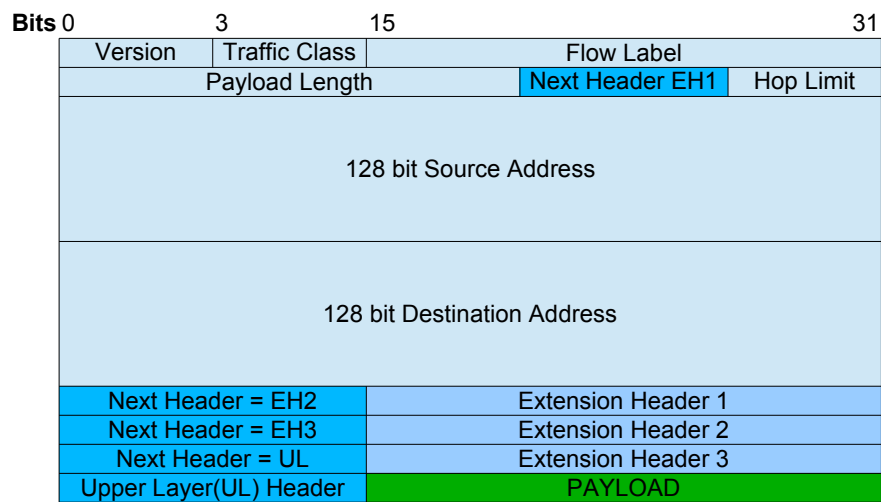


Figure 2.3: Chaining Extension Headers in IPv6 Packets [1]

Table 2.7 provides the differences between the structure of an IPv4 header and IPv6 header. Some of the important changes are the addition of Extension Headers and the removal of the header checksum.

Table 2.7: Differences in IPv4 and IPv6 Header Fields [7]

IPv4 Header	IPv6 Header
Version	Change from 4 to 6.
Internet Header Length	Removed, Header Length is a fixed size of 40 bytes.
Type of Service	Replaced by Traffic Class field.
Total Length	Payload Length replaces this field in IPv6.
Identification Fragmentation Flags Fragment Offset	Removed in IPv6, fragmentation data is included in the Fragment Extension Header.
Time to Live	Hop Limit field in IPv6 serves as replacement.
Protocol	Next Header field in IPv6 serves as replacement.
Header Checksum	Do not exists in IPv6.
Source Address	Same as in IPv4 but changes to a 128-bit address.
Destination Address	Field is the same as in IPv4 but changes to a 128-bit address.
Option	Removed, Extension Headers are used for this.

2.2.4 IPv6 Autoconfiguration

In current IPv4, addresses can only be assigned by ways of manual configuration by a administrator or by the use of DHCP. IPv6 autoconfiguration is a new feature that provides the facility to assign IPv6 addresses to network interfaces without the use of DHCP or manual intervention. Autoconfiguration comprises of two types, stateless and stateful autoconfiguration. The computer builds a link local address as soon as the network interface comes up. This address is concatenated with the link local prefix 'fe80::' and another 64-bit number which is derived from the 48-bit MAC address of the network interface. Packets with a link local address will never be relayed by a router and can only be used in the same local network. The link local address is assumed to be unique because the MAC addresses are supposedly unique worldwide.

Table 2.8 shows the different types of ICMPv6 messages.

Table 2.8: Types of ICMPv6 messages [7]

Value	Description
1	Destination Unreachable
4	Parameter Problem
128	Echo Request
129	Echo Reply
130	Multicast Listener Query
131	Multicast Listener Report
132	Multicast Listener Done
133	Router Solicitation (NDP)
134	Router Advertisement (NDP)
135	Neighbour Solicitation (NDP)
136	Neighbour Advertisement (NDP)
137	Redirect Message (NDP)
138	Router Renumbering
139	ICMP Node Information Query
144	Home Agent Address Discovery Request Message
145	Home Agent Address Discovery Reply Message
146	Mobile Prefix Solicitation
147	Mobile Prefix Advertisement
151	Multicast Router Advertisement
152	Multicast Router Solicitation
140	Site Multihoming by IPv6 Intermediation

- **Stateless Autoconfiguration:** Nodes can connect to a network and automatically generate global IPv6 addresses without the need for manual configuration or the help of a server, such as a Dynamic Host Configuration Protocol (DHCP) server. With IPv6, a device on the link advertises any global prefixes in Router Advertisement (RA) messages, as well as its willingness to function as a default device for the link. RA messages are sent periodically and in response to device solicitation messages, which are sent by hosts at system startup. A node on the link can automatically configure global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the RA messages. The resulting 128-bit IPv6 addresses configured by the node are then subjected to duplicate address detection to ensure their uniqueness on the link. If the prefixes advertised in the RA messages are globally unique, then the IPv6 addresses configured by the node are also guaranteed to be globally unique. Device solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, as seen in Table 2.8, are sent by

hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message [28]. When the M-flag (managed address configuration) bit in Figure 2.4 is set to 0, the stateless autoconfiguration protocol will be used to assign an address to the node. If the M-flag is set to 1, the node must not use stateless autoconfiguration to build the address, but should use the configuration server to assign the IPv6 address to the node. If the node uses stateless autoconfiguration but needs to receive additional information from the configuration server, the O-flag bit will be set to 1. Figure 2.4 shows the ICMPv6 Router Advertisement Message construction.

Type (134)	Code (0)			Checksum
Cur Hop Limit	M	O	Reserved	Route Lifetime
Reachable Time				
Retransmit Timer				
Options				

Figure 2.4: ICMPv6 Router Advertisement Message

- **Stateful Autoconfiguration:** In this autoconfiguration process DHCPv6 is used to supply an address and other configuration information to the host. This mechanism is called stateful because there is bidirectional and reliable communication between the server and client. The server that hosts DHCPv6 keeps track of all the addresses being assigned and to which nodes they are assigned. DHCPv6 is capable of being used under two sets of circumstances, with or without stateless autoconfiguration. If stateless autoconfiguration is already used on the link, a DHCPv6 client can be used to obtain DHCPv6 options only, such as DNS and NTP servers. If DHCPv6 is used to obtain options, addresses, and routes, the process becomes slightly more complex, and uses link-local and well-known multicast addresses to communicate with the DHCPv6 servers. It is not the preferred method of autoconfiguration, because it requires additional resources to assign an IPv6 address[29].

2.3 Wireless Sensor Networks

This section describes the functioning of wireless sensor networks and the 6LoWPAN IPv6 stack used in these devices. A wireless sensor network (WSN) is constructed of specifically placed nodes and is used to measure physical or environmental conditions, such as sound, temperature, air pressure, and humidity. The network will cooperatively pass the measured data back to a main element where the data will be stored. Recently, the research community has

been trying to enable IPv6 connectivity in WSN. Such support would provide numerous advantages to WSN. Compression techniques used to minimize packet length and header compression mechanism is also be described. These techniques allow low-power devices to send IPv6 packets over low-rate wireless personal area networks (LR-WPAN) based networks and to participate in the Internet of Things[30]. The Internet of Things is further describe in Section 2.3.1.2. Figure 2.5 shows an example of a basic WSN that can be used to monitor various conditions or to capture information.

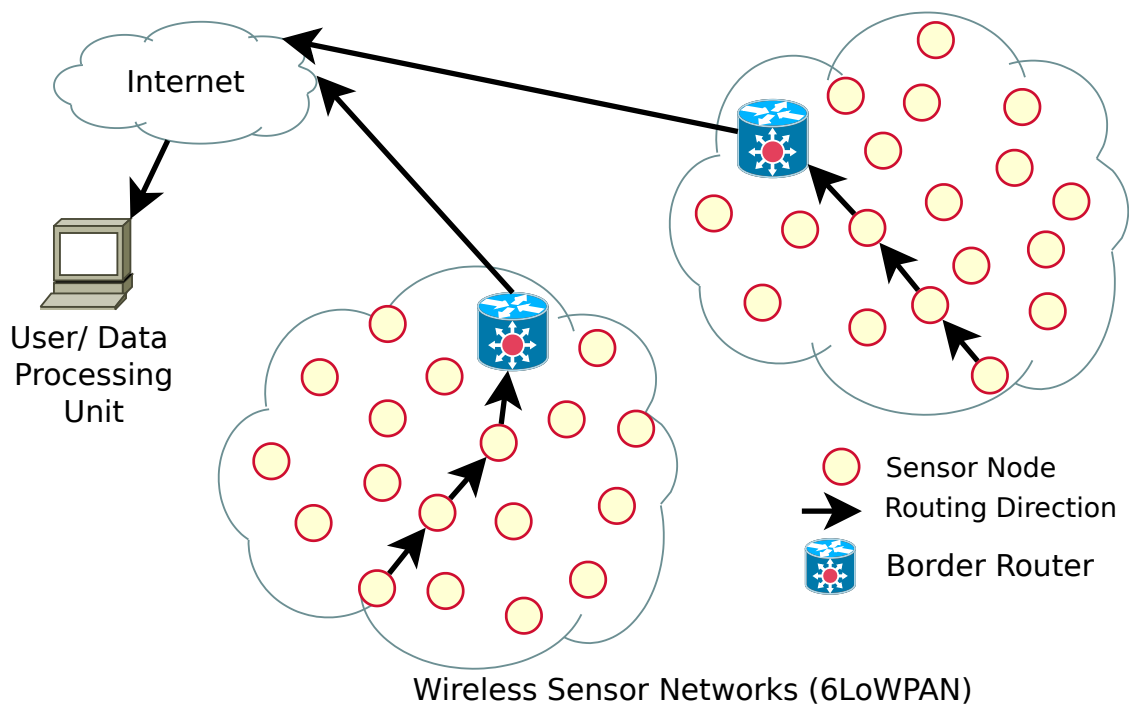


Figure 2.5: Wireless Sensor Network (6LoWPAN Architecture)

2.3.1 IPv6 over Low Power Wireless Personal Networks

A 6LoWPAN network [15] as seen in Figure 2.5 is a simple low cost communication network that allows IPv6 connectivity. It is formed by devices that are compatible with the IEEE 802.15.4 standard [31] and characterized by short range, low bit rate, low memory usage and low cost. Unlike standard IP access networks, a 6LoWPAN network has a tree structure[5]. WSN cannot fully support IPv6, one of the main problems being the relatively limited frame size available on WSN (127 bytes) as shown by Figure 2.6. IPv6 has a fixed header size of 40 bytes and this means that the overhead imposed by multiple headers would drastically reduce the number of bytes left for data transmission.

6LoWPAN consists of a sensor node, router and border router. The border router is the root of the 6LoWPAN network and connects the network to the rest of the IPv6 Internet. The border router is also responsible for assignment of IPv6 prefixes inside the 6LoWPAN network.

There are two routing mechanisms for 6LoWPAN networks, namely route-over and mesh-under. The route-over mechanism uses the network layer for routing, thus each hop inside the 6LoWPAN network is seen as one IP hop. By using this routing mechanism sensor nodes can communicate only with routers. Mesh-under router mechanism enables layer 2 routing. As a result, each sensor node is one IP hop away from the border router, regardless of the number of layer 2 transmissions needed to reach the BR[32].

In a standard IPv6 packet the maximum transmission size for a packet is 1024 bytes; however, in 6LoWPAN this size is reduced to 127 bytes. This shows clearly that the IPv6 packet does not fit into an IEEE 802.15.4 frame. Figure 2.6 shows that 25 bytes are used for the MAC header and another 40 bytes are used for the IPv6 header. This leaves only 62 bytes for the payload. Another 8 bytes can be used for the UDP headers and this leaves a maximum payload size of only 54 bytes. Looking at the small frame size available for a payload, the IETF has defined an adaptation layer in order to reduce the overhead of the IPv6 header [15]. This adaptation layer is placed between the MAC layer and the network layer[5].

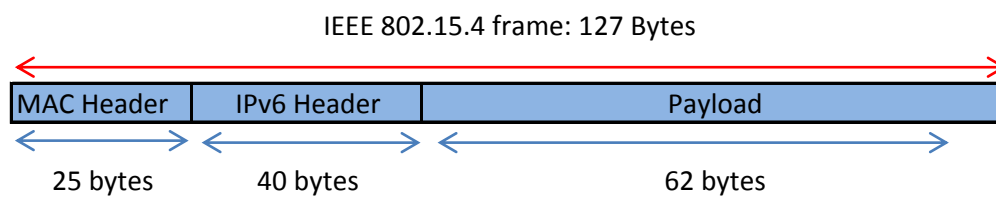


Figure 2.6: IEEE 802.15.4 Uncompressed Frame

2.3.1.1 HC1 Encoding

The first header compression mechanism proposed, was referred to as HC1 [25]. Figure 2.7 shows how the resulting 6LoWPAN packet is constructed when using this compression mechanism. The HC1 mechanism introduces the option to omit some of the IPv6 header fields. This is possible because certain fields are always the same and can either be retrieved implicitly, for example version field is always 6, or from other layers, e.g. the header length can be computed from the MAC header. The only field that will always stay uncompressed is the hop limit field [5]. HC1 encoding makes use of 2 bytes to compress the header. The first byte is used to differentiate the next content (IPv6 header, fragmentation

header, etc.) and the second byte is used to describe the fields in the IPv6 header that are compressed. The third byte is the hop limit field and is never compressed. By using this compression the IPv6 header can be reduced from 40 bytes as shown in Figure 2.7 to only 3 bytes. An IPv6 address has a size of 128 bits and is composed of an 64 bit IPv6 prefix and the 64-bit Extended Unique Identifier of the network interface. The extended unique identifier can easily be retrieved from the MAC header. By using HC1 encoding the extended unique identifier part of the IPv6 address can systematically be omitted for on-link source or destination. The IPv6 prefix can be either the link-local prefix or a global prefix. The HC1 encoding can omit only the local prefix and thus global communication will systematically include a global IPv6 prefix of 64 bits in the source and destination address fields. This constraint seriously limits the utility of HC1, because the interest into bringing IPv6 support into WSN is to enable global communication. This constraint introduced a new compression mechanism referred to as LOWPAN_IPHC, defined by the IETF [15].

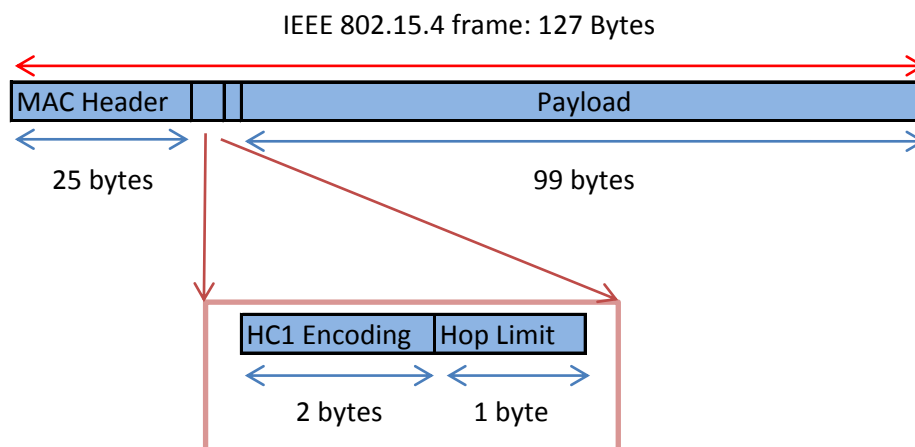


Figure 2.7: IEEE 802.15.4 HC1 Encoded Frame

2.3.1.2 LOWPAN_IPHC Encoding

The LOWPAN_IPHC encoding mechanism uses two bytes to define each field of the IPv6 Header, whether the fields are omitted or included as an uncompressed field. Two modes exist that can be used to compress the IPv6 addresses, namely stateless compression and a context-based compression [5]. Stateless compression can be seen as a compression mechanism similar to the one used in HC1 encoding. Context-based compression is defined as a mechanism used to compress global IPv6 addresses. For this mechanism a context

ID which consists of a set of 4 bits, is defined for each global prefix used in the 6LoWPAN network. For off-link communication, the context ID of 4 bits is used to replace the IPv6 prefix of 64 bits. When using this configuration, an additional byte is added after the LOWPAN_IPHC encoding field, which contains the prefixes of the source and the destination [15]. For the remainder of the IPv6 address the same techniques are used as in stateless compression mode. Figure 2.8 shows LOWPAN_IPHC encoding message format as well as the additional byte added during off-link communication.

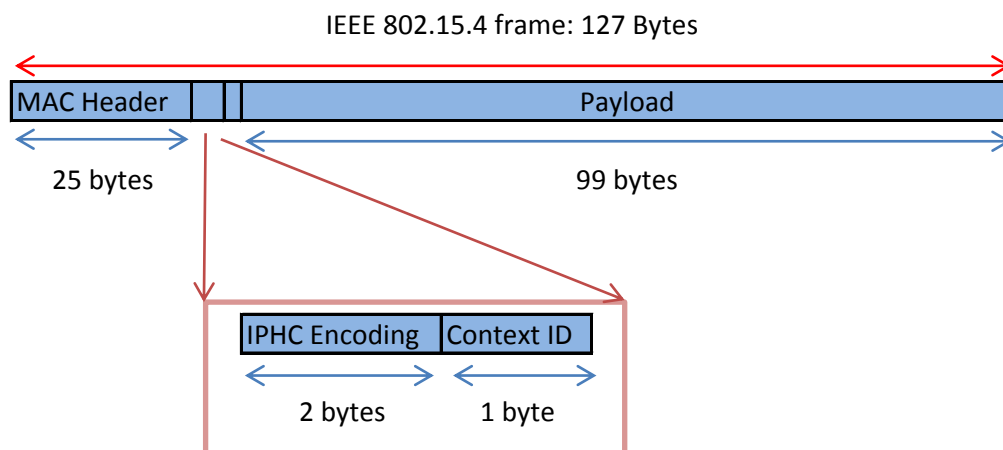


Figure 2.8: IEEE 802.15.4 LOWPAN_IPHC Encoded Frame

2.3.2 Internet of Things

"If we had computers that knew everything there was to know about things-using data they gathered without any help from us we would be able to track and count everything, and greatly reduce waste, loss, and cost. We would know when things needed replacing, repairing or recalling, and whether they were fresh or past their best. The Internet of Things has the potential to change the world, just as the Internet did. Maybe even more so." -Kevin Ashton, originator of the term, Internet of Things - 1999 [30].

IPv6 is emerging as the actual solution for scaling up the Internet to host an almost unlimited number of devices with globally unique reachable addresses [24]. Development of RFID tags, sensors, actuators and smart phones makes it possible to materialise an Internet of Things. The Internet of Things and devices like these mentioned can improve service and can be accessible anytime, from anywhere by interacting and co-operating with each other [33].

2.4 OSI Layer Mobility

This section looks at mobility options across the different layers of the OSI Stack. When we look at the classical TCP/IP stack, mobility has no well-defined place. From the application layer to the link layer a lot of different types of support for mobility have been proposed. The physical layer is not involved in mobility. The reason being that the wireless access technology is implicitly responsible for handling mobility at the physical layer, e.g. the movement of mobile nodes in the area surrounding a wireless access point [34].

2.4.1 Physical Layer Mobility

During physical transmission the message is actually sent out and transported across the network. The physical layer is mainly used for encoding, sending of data(transmission), signaling and the receiving(decoding) of data. Encoding and signaling have to do with transforming the data from bytes, to a signal that can be transmitted over a physical medium. After this has been done, data is also received on this layer and decoded back to bytes that can be read by the computer. It is obvious that mobility is not applicable, because the data still needs to be send over the medium while the mobile node is moving. Mobility challenges are not associated with the physical layer [34].

2.4.2 Link Layer Mobility

Link layer mobility is where the actual handoff occurs between different nodes and is also known as layer handoffs. Two types of handoff exist, namely horizontal handoffs and vertical handoffs [34]. Horizontal handoffs are invisible to the higher layers, whereas vertical handoffs are the handoffs between different access technologies and are only visible to the network layers. If one looks at mobility on the link layer, the link layer is responsible for scanning, detecting of potential access technologies, monitors channel conditions, authentication and re-association. The link layer is also useful in detecting different link layer technologies, supplying the signal to noise ratio from different access points available, the channel of operation and information on overlaying networks. Information on channel conditions is very useful in making decisions about routing, queuing, QoS and packet drop [35]. Information about the link and link properties is useful in the case of overlaying networks. The responsibility for communication between different link layer devices, lies with the different link layer technologies. This enables heterogeneity and proactive context caching for fast handoffs [34]. In order to detect link layer handoffs, the handoff scheme makes use of the signal strength level.

2.4.3 Network Layer Mobility

The network layer is used to connect different and remote networks and to allow communication between them. The two main services of the network layer that could affect mobility are location management (e.g. interface change) and IP-address assigning (e.g. inter-network and subnet mobility). QoS is provided by the network layer and this could be impacted by the service of mobility. As soon as a mobile node enters a new network, the mobile node undergoes an IP change and this can lead to huge challenges in location management which is also the responsibility of the network layer. When a mobile node changes its network, the change requires the mobile node to be configured with new network settings. The mobile node must be in a position to receive a new IP address, this allows the node to be reachable by corresponding nodes. The mechanisms for location management are Domain Name System (DNS), Dynamic DNS and home agent binding in Mobile IP. Another problem that is posed by mobility is dynamic routing of packets to reach their destinations. For dynamic routing the approach to overcome this is mentioned in [34, 8].

2.4.4 Transport layer Mobility

The network layer mobility affects the transport layer mobility directly due to link capacity, packet loss and change of IP address. Packet loss affects the congestion control algorithms. The connection's window size is determined by the Bandwidth Delay Product (BDP). The BDP is used for the rest of the connection, but the link capacity might change when the device moves to a new network. This change will affect the BDP directly and causes a new window size negotiation between the nodes. The mobility aware transport layer protocol are used to deal with the issues of BDP. The most affected conventional layer when mobility management is used, will be the transport layer. The reason being the tight binding of the the transport layer with the layer above and below it. The transport layer mobility is affected by the following processes [34]:

- Inter-subnet mobility
- Inter-network mobility
- Horizontal handoff
- Vertical handoff
- Address change
- Session mobility and
- Application/flow mobility

It is the responsibility of the transport layer and higher layer mobility management mechanism's to handle all the issue produced by the above mentioned processes [36].

2.4.5 Session Layer Mobility

The maintaining of parameters related to communication and session state is handled by the session layer and this serves as the main purpose of the session layer [34]. Unexpected termination of the transport layer can be caused by mobility and this can lead to the corruption or loss of the state of the session. Session layer mobility is affected only when application mobility is present. When using the session layer, the following is stored as session state information:

- The byte size already transferred and written to disk during a file transfer.
- Information related to a secure login session. This information can consist of encryption keys and security associations set up during the secure login.
- Maintaining synchronization data used for combining incoming network streams.

Session layer mobility ensures that the above mentioned information is not lost when the transport layer has an unexpected connection termination. The relevant session layer mobility handling mechanism is required to pause and restart the current session. The mobility handling mechanism must be granted access by the application to all the required information. This information can then be reused when a new connection socket has been set up [34].

2.4.6 Presentation layer Mobility

The Presentation layer is mainly responsible for data formatting, as required by the Application layer. In the case of mobility, the Presentation layer will have to deal with the following issues like screen resolution, codec and application versions. The mobility management of the application should be capable of detecting the capabilities of the new device and modify and adapt the content accordingly for display on the new device [34].

2.4.7 Application layer Mobility

Application layer mobility solutions, or also known as application specific mobility solutions have no set requirements for mobility management. The application layer is dependent on the underlying layers for network access and

socket establishment, which makes the application layer highly flexible. If the presentation layer is not responsible for the session mobility control, the drawback is that the application needs to be redesigned so that the services provided by the session layer can be used [34].

2.5 Mobile IPv6

Mobile IPv6 is a proposed mobility standard for IPv6 networks [8, 37]. MIPv6 enables a mobile node to stay connected when moving around in different networks while maintaining one IP address. MIPv6 is suitable for mobility in heterogeneous networks, for example it allows a mobile node to transfer between different Ethernet segments, as well as from an Ethernet segment to a Wifi cell. The mobile node is able to change between different 802.11 Access Points while maintaining its home network IP address, even if the new point of attachment is in a new network. The home address is used to identify the mobile node and the node will always be reachable through this address. The network consists of a home network and a foreign network also called a visiting network. The home network assigns an IPv6 home address to the mobile node. This address is then used to form a binding between the mobile node and the home network. This enables the home network to always know the location of the mobile node when it is in a foreign network. Further in this section the functioning of the protocol is described in more detail. The basic setup of a MIPv6 network can be seen in Figure 2.9 with route optimization and triangular routing visible.

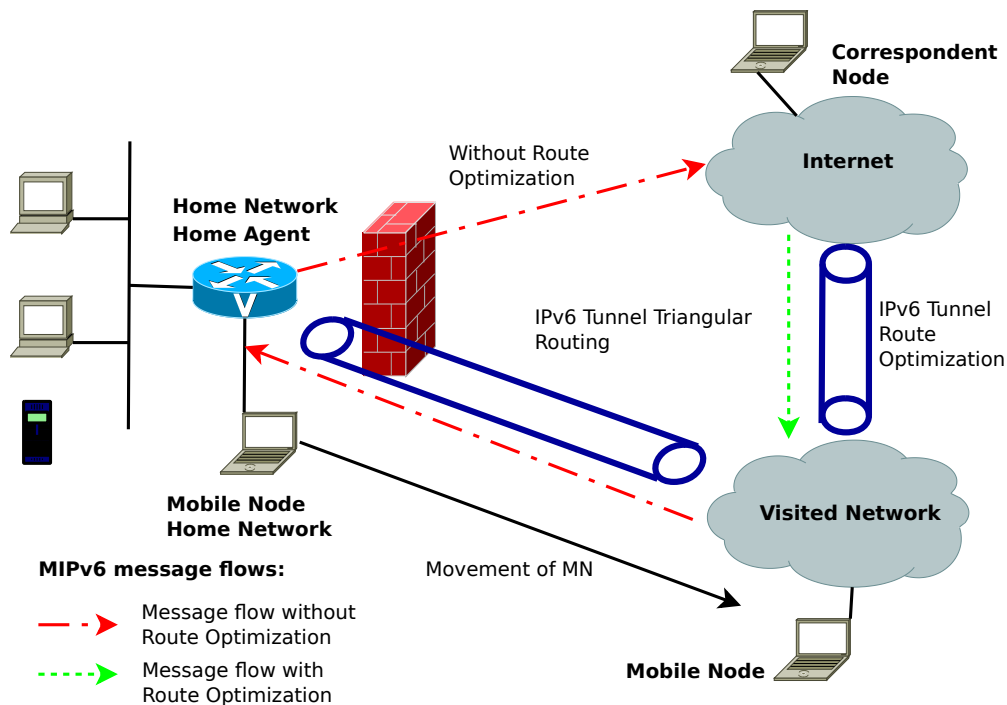


Figure 2.9: MIPv6 Network

2.5.1 MIPv6 Network Elements

- **Mobile Node (MN):** The MN is an IPv6 enabled node that passes around in an IPv6 network and thus stays connected through its home address. The MN creates a binding with the HA when moving to a visiting network where it receives a Care-of address.
- **Home Agent (HA):** The HA is a router located in the Home Network and stores information about MNs whose permanent home address is in the HA's network. The HA stores the binding messages, the Care-of Addresses (CoA) of the MN and performs forwarding of data destined for the MN in triangular routing. Both the correspondent node and the HA keep binding information in their binding caches.
- **Home Network (HN):** The HA resides in the HN. The HN uses the home subnet prefix and assign the mobile node's home address from this prefix.
- **Correspondent Node (CN):** The CN is any node that wants to establish a connection with the mobile node. The correspondent node needs only IPv6 capabilities and not MIPv6 capabilities to communicate with the MN. If the CN is MIPv6 capable, route optimization can be used to reduce overhead and direct communication with the mobile node.

- Visited Network (VN): This is any network that the MN visits outside its home network. The mobile node receives a CoA from this network which has a different network prefix to that of the home network.
- Home Address (HoA): The home address is the IPv6 address the mobile node receives when connected to the home network. The mobile node will always be reachable through this address, even if the mobile node has moved to a visiting network. The mobile node is always logically connected to the home network because the mobile node is always assigned with the home address. If the mobile node moves to a visiting network, the home agent will intercept packets destined for the mobile node and forward the packets to the CoA.
- Care-of Address (CoA): The CoA is the IPv6 address the mobile node receives when in a visiting network. A mobile node can have multiple CoAs but only one can be linked as the primary CoA at the home agent. The binding that is made between the mobile node and the home agent is an association of these two addresses. Both the correspondent node and the home agent keep binding information in their binding caches.
- Route Optimization (RO): Route Optimization is a process that removes much of the overhead caused by the MIPv6 protocol, as seen in Figure 2.11, by the use of Bidirectional Tunneling. Route Optimization can be seen in Figure 2.9 and Figure 2.12. Communication is directly between the mobile node and the correspondent node, without the home agent having to intercept the packets destined for the mobile node. An IPv6 tunnel is established between the mobile node and correspondent node that makes communication possible.

2.5.2 Comparison between MIPv6 and MIPv4

Mobile IPv4 is the first implementation of mobility in IP networks and was implemented on the Internet Protocol version 4. The design of MIPv6 benefits a lot from the experience gained from the implementation of the MIPv4 version. IPv6 also offers a lot of advantages for mobility and the following section describes the differences between MIPv4 and MIPv6 implementations [38].

- In MIPv4 the network consists of a foreign agent, this is a router in the FN that acts as the agent when the MN roams to a new network. In MIPv6 there is no need for a foreign agent because of the Neighbour Discovery and Address Auto-configuration features that enable an MN to function in any location without the services of any special router (FA) in that location.

- Route optimization is a very important part of mobile IP and for MIPv4 it is a non-standard extension of the protocol. This way route optimization was better built into the IPv6/MIPv6 protocols.
- Route optimization is also implemented in such a way that it coexists efficiently with routers that implement ingress filtering. In MIPv4 this will happen because the CN sends a packet that contains a source address that is set to the home address and is discarded by ingress filtering. In MIPv6 the CN puts the CoA as the source address and uses the Home Address Destination option in the IP packet header for the home address and this enables CoA to be transparent over the IP layer.
- Symmetric reachability is ensured between the MN and the default router in the current location by IPv6 Neighbour Unreachability Detection.
- The overhead of packets is reduced by avoiding IP encapsulation and instead using the IPv6 routing header options.
- Mobile IPv6 is not linked to any particular link layer, as it does not use Address Resolution Protocol (ARP) but instead IPv6 Neighbour Discovery. This ability improves the robustness of the protocol [39, 40].

2.5.3 MIPv6 Drawbacks and Known Issues

Mobile IPv6 has some known limitations and this leads to the argument that Mobile IP could be unsuccessful and impractical. The following section explains the challenges faced by Mobile IP and possible solutions to these challenges [12].

2.5.3.1 Security Attacks

- Denial Of Service Attacks: These attacks occur when a large number of packets are sent from an attacker to a certain host. This tremendous number of packets being sent to a host may cause the host CPU to go down. During this period of processing the bogus packets, no useful information can be transmitted on the network. DOS attacks can also happen when the attacker interferes with packets flowing between two nodes on a network. The attacker can generate bogus Registration Request packets specifying his own IP address as the CoA of the node. This will cause all the packets destined for the CoA to be routed by the Home Agent to the attacker's IP. The possible prevention method for this is to require cryptographically strong authentication in all registration messages exchanged by a mobile node and its home agent [12]. IPsec is also a mandatory requirement for MIPv6 to enable secure communication [41].

- **Theft of Information Passive Eavesdropping:** This will allow an attacker to gain wired or wireless access to the network infrastructure. Passive Eavesdropping can be prevented by using End-to-End encryption technologies like IPSec [41].
- **Replay Attack:** An attacker host machine can obtain a copy of a Registration Request message and by using this at a later stage be able to register a bogus CoA. Prevention of this is easy by making use of the Identification field. The identification field is generated in such a way that when the packet is received by the Home Agent it will be recognized as being out of date [12].

2.5.3.2 Triangulation Problem

The triangular routing problem is responsible for delays in the delivery of data packets to mobile nodes. This delay of data packets causes a lot of overhead in the network and thus lowering the performance of the network and the relative routers in the network. Route optimization can be used to improve the overhead by avoiding a packet route through the home network, but instead delivering the packets directly to the CoA of the mobile node. The triangular routing problem can be seen in Figure 2.9 as can the solution. In [42] an analytical evaluation of these the methods can be seen. From this research it is clearly visible that route optimization has a great importance especially for real-time application. The functioning of triangular routing and route optimization in MIPv6 will be explained in more detail in the following section.

2.5.4 Basic Functioning of Protocol

A Mobile node must also be addressable at its home address, even when moving around in different subnets. The home address is the IPv6 address that the mobile node received in the home network and it is within the home network's subnet prefix. If the MN is in the home network, the packets are addressed to the MNs home address and use conventional routing mechanism to deliver the packets. When the mobile node moves to a different network, it receives an IPv6 address that is in the same range as the foreign network's prefix. This new IPv6 address is called the CoA of the MN. The CoA is assigned to the MN using conventional mechanisms like stateless or state-full auto-configuration. For the period that the MN is in the foreign network packets destined for the MN is addressed to the MN's CoA [8, 37].

The association between the MN's home address and Care-of address is called a binding. The MN and the home agent will make a binding every time the MN moves to a new foreign network. The MN forms this binding with the home agent by sending a binding update (BU) to the home agent. This

process registers the MN CoA with the MN and the home agent knows that the MN is reachable at the CoA. The home agent sends a binding update acknowledgement to the MN after receiving the BU and processing of the BU is successful. A correspondent node is any node in the network that wants to communicate with the MN, and can be a stationary or mobile node. The MN can send information of its location to the CN by a process called correspondent registration. This way the CN will know where the MN is located and return routability is used to authorize the binding between the MN and the CN [8, 43].

The message flow of MIPv6 with Route Optimization is shown in Figure 2.10. The mobile node travels to a foreign network, using IPv6 Router Advertisement and Neighbour Discovery to detect the movement to a new network. The mobile node can ask the foreign node for router advertisement by sending a router solicitation message to the default router in the foreign network. A router on the foreign network will then reply with a Router Advertisement message to the mobile node that includes the network prefix of the FN. The mobile node uses this message to detect movement to a new subnet, and is assigned a Care-of address by the router. After the mobile node receives a CoA, the node sends a BU message to the Home Agent. The Bu message is encrypted and authenticated using IPSec [44].

If the BU message is correct and verified by the Home Agent, the Home Agent sends a Binding Acknowledgement (BAck) message to the mobile node if the A-bit has been set in the BU as seen in Figure 2.14. Now that the binding is formed between the mobile node and home agent, data packets can be sent between the mobile node and correspondent node via bidirectional tunneling as seen in Figure 2.11. To enable Route optimization between the mobile node and correspondent node, the mobile node sends a Home Test Init (HoTI) message to the correspondent node via the HA. The MN also sends a Care-of Test Init (CoTI) message directly to the correspondent node. By exchanging these messages between the HA, MN and the CN, the CN can obtain some reasonable assurance that MN are reachable at its HoA and CoA. The correspondent node returns a Home Test and Care-of Test message to the MN. After these message have been accepted, the mobile node sends a new BU message to the CN to enable Route Optimization. The data packets can now be sent directly from the CN to the MN and from the MN to CN [8, 43].

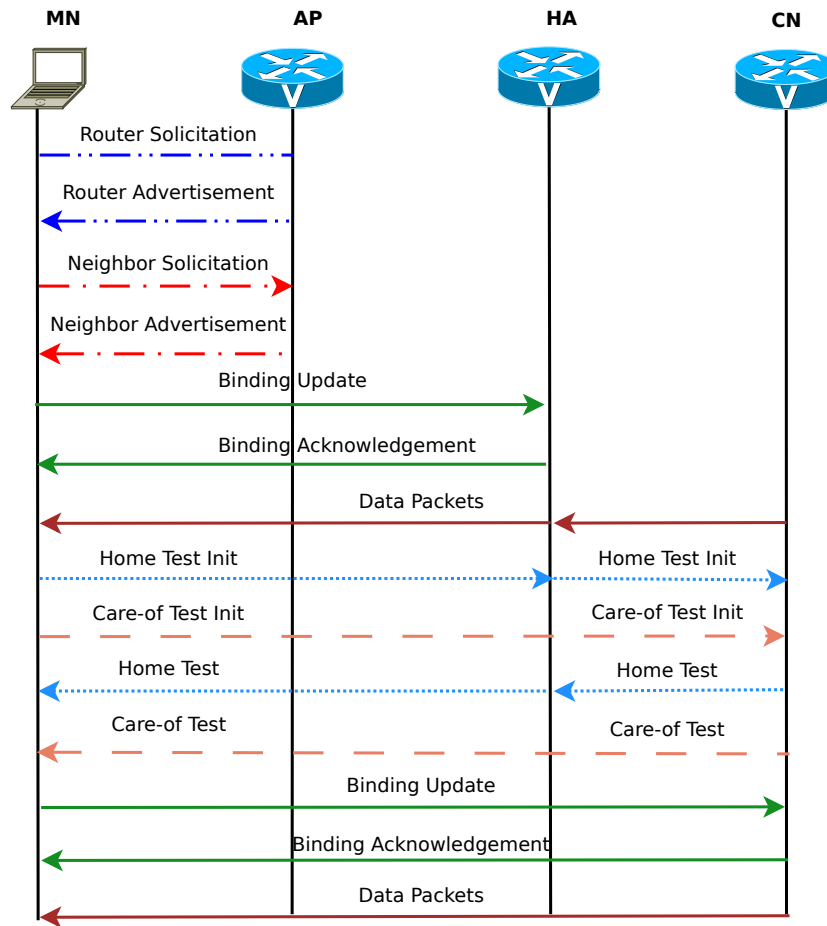


Figure 2.10: MIPv6 Message Flow

The MN and the CN can communicate via two options known as bidirectional tunneling mode or route optimization mode. Figure 2.11 shows the Bidirectional Tunneling Mode in MIPv6.

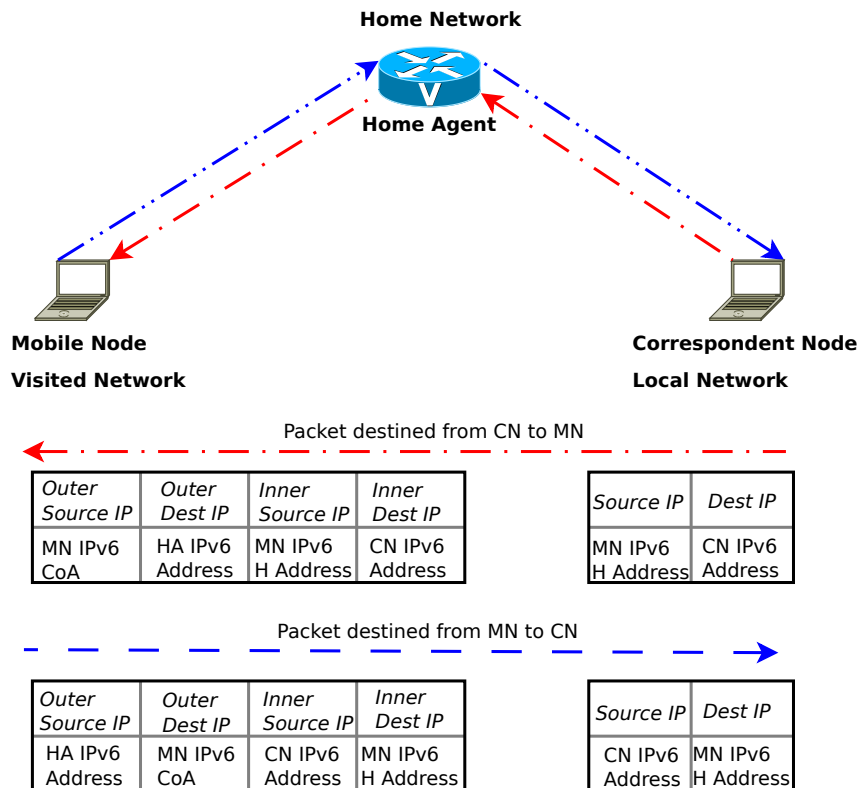


Figure 2.11: MIPv6 Bidirectional Tunneling Mode

In bidirectional tunneling the CN needs only to be IPv6 enabled to communicate with the MN and there is also no binding between the MN and the CN. The CN routes packets to the HA and from here packets are tunneled to the MN. The red line in Figure 2.11 indicates the path of the packets. The MN tunnels packets to the HA and from here it is routed to the CN using normal routing protocols. The blue line in Figure 2.11 indicates the path of the packets destined for the CN. This process is also called reverse tunneling mode. In order for the HA to intercept packets destined for the mobile node, it uses proxy Neighbour Discovery and intercepts all the packets addressed to the MN's home address. Intercepted packets are then tunneled to the MN current CoA by the use of IPv6 encapsulation [6, 8].

The second mode is called route optimization and for this the CN also needs to support MIPv6 [43]. This mode can be seen in Figure 2.12.

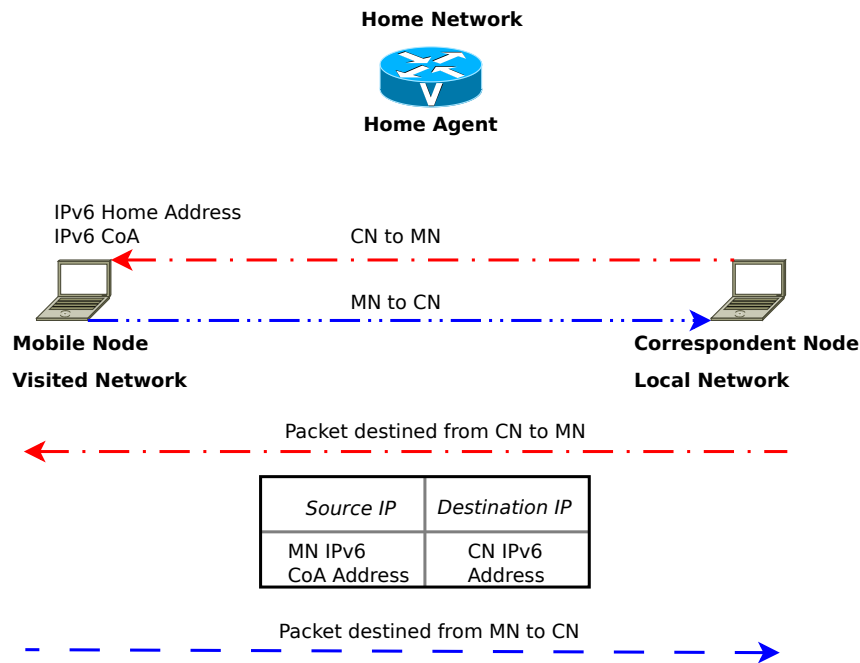


Figure 2.12: MIPv6 Route Optimization Mode

In this mode the MN and the CN configure an BU between them. When the CN wants to send any IPv6 packet it first checks the cache bindings to see whether any binding entry exists for the IPv6 address. If the CN finds an BU for that specific address, a new routing header is used by the CN to route the packet directly to the MN's information captured in the binding. The advantages of this mode are that the overhead in the network is hugely reduced and congestion at the home agent is also minimized. This mode also ensures that the packets reach the targeted node via the shortest possible route and this eliminates the home agent as a Single Point of Failure in the network [10].

If packets are routed directly to the MN, the CN will set the Destination Address of the IPv6 header to the CoA of the MN as seen in Figure 2.12. For packets that are sent from the MN to CN the MN uses the CoA as the source address of the packet and adds a new IPv6 home address destination option to carry the MN's home address. By adding the home address destination option to the packet the use of the CoA is transparent above the network layer.

MIPv6 also supports multiple home agents and has limited support for home network reconfiguration. If the mobile node is not sure who the home agent is in the network, the MN can make use of dynamic home agent address discovery to identify the home agent in the network [6, 8, 43]

2.5.5 Mobility Header for IPv6 Protocol

Mobile IPv6 defines a new Mobility Header (MH) to carry various MIPv6 messages. The Mobility Header is used by MN, CN, and HA during binding creation and management of bindings. The MH uses a value of 135 for the Next Header field to be identified in the immediately preceding header. This can be seen in Table 2.5. The MH has the same alignment requirements as any IPv6 protocol header, an 8-octet boundary [8]. Figure 2.13 illustrates the format of a MH in IPv6.

Payload Proto	Header Length	MH Type	Reserved
Checksum			
Message Data			

Figure 2.13: Mobility Header for IPv6 Protocol

- Payload Proto: 8-bit Selector field. This field uses the same values as in Table 2.5 and identifies the header type following the MH.
- Header Length: This 8-bit unsigned integer field is used to represent the length of the MH and must always be in a multiples of 8 octets.
- MH Type: 8-bit Selector. This field is used to identify the type of mobility message. If this indicator contains an invalid MH type, an error indication will be sent. The various mobility header types can be seen in Table 2.9.
- Reserved: 8-bit field reserved for future use.
- Checksum: This field contains a 16-bit unsigned integer checksum of the Mobility Header and is composed of the 16-bit ones complement of the ones complement sum of this string. The home address of the mobile node will be used as the value of the IPv6 source address field if available [8].
- Message Data: This variable length field contains the relevant data for the indicated MH type.

Table 2.9: Mobility Header Types [8]

Value	Description
0	Binding Refresh Request
1	Home Test Init
2	Care-of Test Init
3	Home Test
4	Care-of Test
5	Binding Update
6	Binding Acknowledgement
7	Binding Error
8	Fast Binding Update
9	Fast Binding Acknowledgment
10	Fast Neighbour Advertisement (Deprecated)
11	Experimental Mobility Header
12	Home Agent Switch Message
13	Heartbeat Message
14	Handover Initiate Message
15	Handover Acknowledge Message
16	Binding Revocation Message
17	Localized Routing Initiation
18	Localized Routing Acknowledgment

The following list of messages is carried by the new Mobility Header and Table 2.9 indicates the MH Type values for all the messages.

- **Binding Refresh Request:** A binding update is valid only for a certain period of time. If the binding time is reached, the CN or HA will send a Binding Refresh Request to re-establish the binding with the MN. This message is typically used when the cache binding is active and the CN sees that the binding's lifetime is reaching its expiry date. An indication by which the CN checks to see whether the connections are in active use, is to check for recent traffic and open transport layer connections on the network.
- **Home Test Init:** This message is used to trigger the RR procedure by the MN. During this procedure the MN will request a home keygen token from the CN. An MH Type of 1 is used by this message.
- **Care-of Test Init:** The care-of Test Init (CoTI) message is used to trigger the RR procedure by the MN. During this procedure the MN will request a care-of keygen token from the CN.

- Home Test: The Home Test (HoT) message is sent from the CN to the MN and is a response to the Home Test Init message. It uses the MH Header type 3 and it contains a 64-bit keygen token and home init cookie that is used in the RR procedure.
- Care-of Test: The Care-of Test (CoT) message is sent from the CN to the MN and is a response to the CoTI message. Uses MH Header type 4 and it contains a 64-bit keygen token and care-of init cookie used in the RR procedure.
- Binding Update: A Binding Update (BU) is an authorization message between the MN and the CN, or the MN and the HA to notify them of its current binding and to notify other nodes of a new CoA. A BU uses the value 5 as MH Header type and Figure 2.14 indicates the format of the Message Data field in the MH. The Acknowledge (A) bit is set by the MN to request a Binding Acknowledgement message upon receipt of the Binding Update by the CN or HA. This option is mandatory when a BU is sent to a CN. The H-bit, or Home Registration bit is used to request the node to act as a HA. The node must be a router sharing the same network prefix as the home network of the MN. The Link-Local Address Compatibility bit (L) is set if the MN home address has the same interface identifier as the mobile node's link-local address. If the Key Management Mobility Capability (K) is cleared, the protocol used to establish IPsec security associations between the MN and the HA does not survive movement. The bit must be cleared when manual IPsec configuration is used. The 16-bit Sequence integer is used to sequence BU so that the received Binding Update Acknowledgement message can be matched to a certain U. The 16-bit Lifetime field is used to indicate the lifetime of a Binding before it will expire.

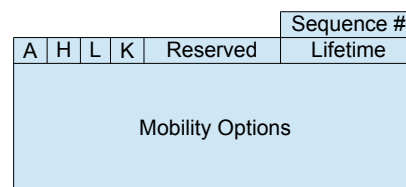


Figure 2.14: Message Data field in Mobility Header for Binding Update

- Binding Acknowledgement Message: A Binding Acknowledgement (BAck) message acknowledges the receipt of a BU message. An BU message will trigger a BAck message if requested in the BU message. The BAck has the MH Type value 6 and the Message field is shown in Figure 2.15. The

Status is an 8-bit integer indicating the nature of the BU. The Status field can have different values, but the most used value is 0, which means the BU has been accepted. A value of 1 also means the BU was accepted, but prefix discovery is necessary. Values of 174 and 131 mean that the BU was not successful because the MN use an invalid CoA or home registration is not supported. The Key Management Mobility Capability (K) bit is cleared, thus the IPsec security associations between the MN and the HA do not support movement. The Sequence field is used to match a BU Acknowledgment to a BU. The Lifetime field, in time units of 4 seconds, is used to indicate the time to retain MN entry in its Binding Cache.

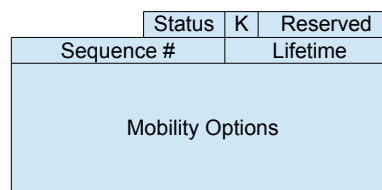


Figure 2.15: Message Data in Mobility Header for BU Acknowledgement

- Binding Error Message: The CN and HA uses binding error messages to indicate errors relating to binding associations. The HA use this message to indicate to the MN that it has received an unrecognized Mobility Header Message Type. The CN node will use the binding error message if an attempt is made to use a home address destination option without the existence of a valid binding.

2.5.6 MIPv6 for WSN

MIPv6 easily fits into the 6LoWPAN network structure due to the availability of IPv6. For IPv6-in-IPv6 tunneling in MIPv6 the standard IPv6 features are used, which are available in the 6LoWPAN adaptation layer. As described in the previous sections, MIPv6 introduces some new control messages called BU and BAck. These messages, however, use the IPv6 extension headers and this increases the overhead and reduces frame space available for payload. BU and BAck messages have a minimum size of 12 bytes. Furthermore a BU also includes a destination option with a home address size of 20 additional bytes. An BAck message includes a type 2 routing header with an additional size of 12 bytes [5]. By using the HC1 compression mechanism, an encoding mechanism was proposed to reduce the overhead of the Mobile IPv6 control messages [45]. For compression of the mobility headers the adaptation layer is the same as the 6LoWPAN adaption layer. The gain of this compression

mechanism depends on the type and size of header to compress. An uncompressed binding update uses 32 bytes:

- 12 bytes for the mobility header.
- 20 byte for options.

A compressed binding update uses 23 bytes:

- 3 bytes for the mobility header.
- 20 byte for options.

Figure 2.16 illustrates the message flows in MIPv6 for 6LoWPAN networks.

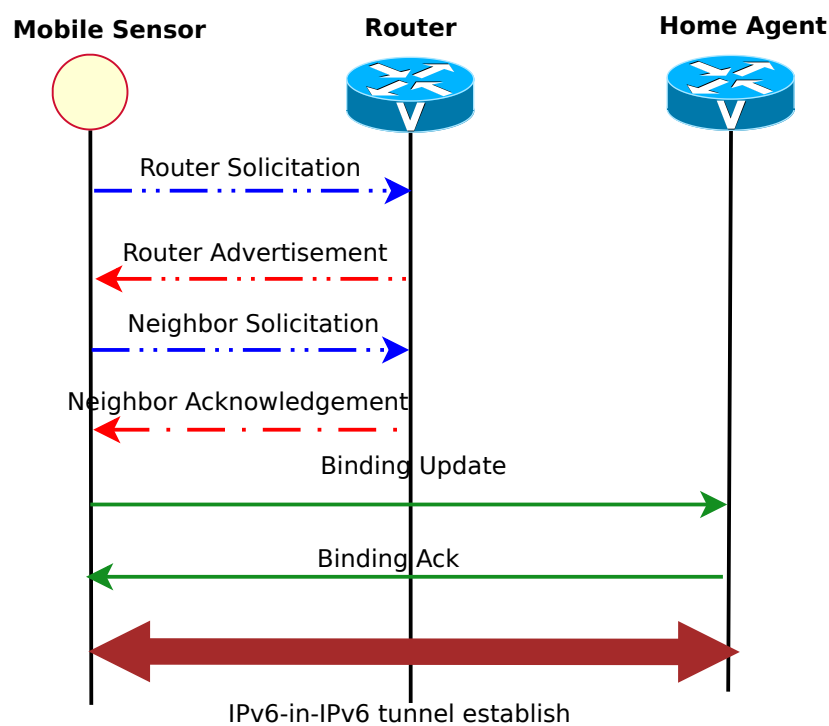


Figure 2.16: MIPv6 Message Flow in 6LoWPAN Network

2.6 Network Mobility

In the previous section MIPv6 was described in detail and diagrams was supplied to illustrate the functioning of the protocols. In this section NEMO Basic Support Protocol is described. NEMO [46] is a protocol extension to MIPv6

to enable persistent connection of moving networks (e.g., trains with wireless hosts of passengers inside the carriages) to the Internet [2]. The main goal of this scheme is to preserve ongoing internal and external communication sessions of nodes attached to a moving network during the network's movement. The NEMO Basic Support ensures session continuity for all the nodes in the Mobile Network, even as the Mobile Router changes its point of attachment to the Internet. As in MIPv6 all the traffic between the Mobile Node and the Correspondent Node are routed through the Home Agent[46].

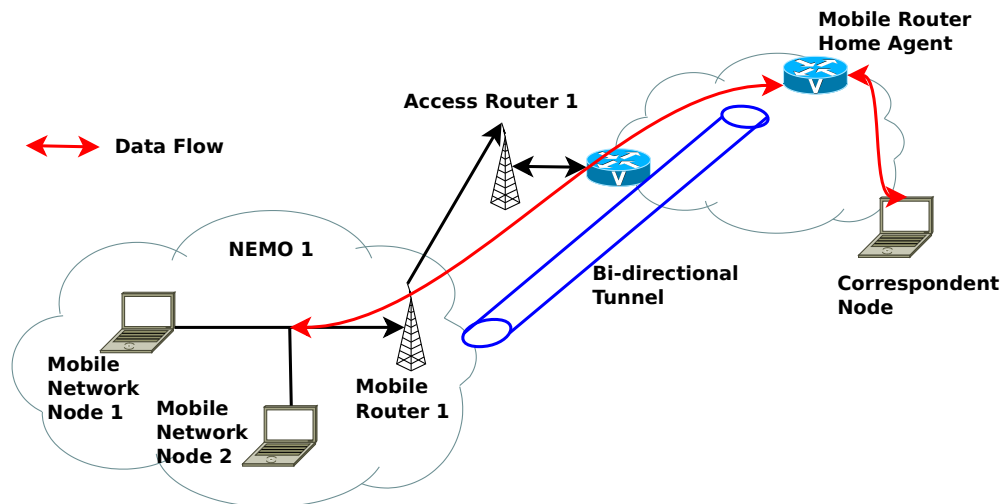


Figure 2.17: NEMO Network [2]

A Mobile Network can be described as a network segment or subnet that has the ability to move and attach to arbitrary points in the routing infrastructure. The architecture of NEMO can be seen in Figure 2.17. NEMO makes it possible that only the MR must be involved in the handover operations on behalf of the whole moving structure. To access the Mobile Network there are specific Gateways called Mobile Routers that is responsible for the managing of mobile network movements. The Mobile Router also forms the default gateway of a Mobile Network. The Home Agent advertises an aggregation of Mobile Networks to the rest of the network infrastructure and this enables connectivity to the network (Visited Network). A Mobile Network can also comprise of multiple and nested subnets. A router without mobility support may be permanently attached to a Mobile Network for local distribution. Mobile Routers are also allowed to connect to Mobile Networks owned by different Mobile Routers. NEMO Basic Mobility Protocol is defined as a mobility protocol network solution where the Mobile Router can at any time act either as a Mobile Host or as a Mobile Router.

NEMO routing optimization techniques are used to further improve the solution and enabling the roaming of whole networks and provides transparent provision of Internet access in public transportation systems for passengers. The widest scale of Intelligent Transportation Systems (ITS) scenarios (e.g., road safety on the move entertainment) or even in personal area networks (PAN) where various electronic devices (like tablets,digital cameras, e-health sensors, etc.) would connect to the Internet through a smartphone simulating the role of the mobile router [46].

In the following section PMIPv6 is described, this forms part of the NEMO network mobility extension. PMIPv6 defines a Mobile Network and not just mobile nodes as with MIPv6.

2.7 Fast Handover for Mobile IPv6

Mobile IPv6, as described in RFC6275 and in Section 2.5, is a protocol used to enable an MN to stay connected in a IPv6 network while changing its point of attachment. But this protocol does not perform well when tested against real-time application, due to a long handover latency. The operations involved in MIPv6 that introduce the longest handover latency are movement detection of the node, link-layer procedures, IP address configuration and location management. For MIPv6 to handle real-time traffic and to enable a seamless handover, Mobile IPv6 Fast Handover(FMIPv6) was introduced in RFC5568 [19]. FMIPv6 is used to minimize the operations mentioned earlier, thus reducing the handover latency. FMIPv6 is used to address two common problems that exist in current IP operations. The first one is to enable the MN to send data as soon as the MN connects to a new subnet and the second one is to deliver packets to an MN as soon as the MN attaches to the new access router [3]. FMIPv6 is not dependent on link layer technologies and does not address improving the link-layer switching latency [37, 47]. Figure 2.18 shows the basic elements of an FMIPv6 network and the various elements are described in Subsection 2.7.1.

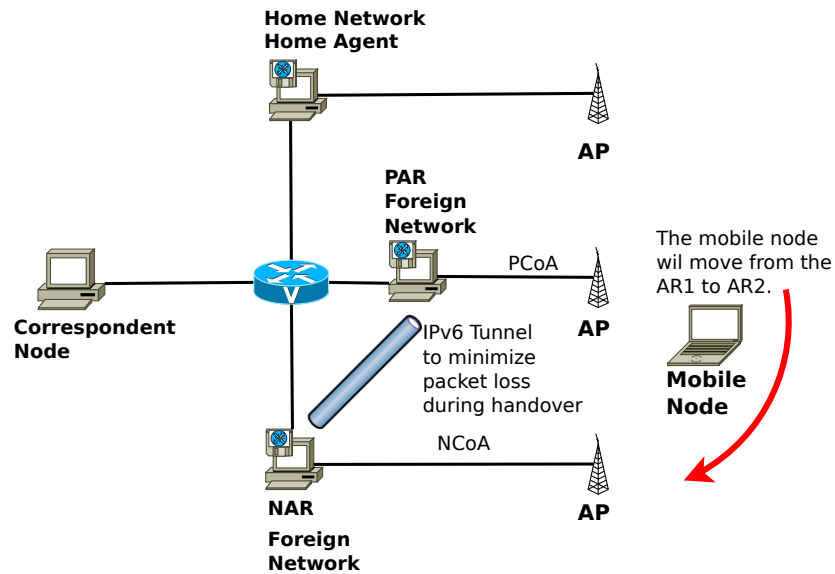


Figure 2.18: FMIPv6 Protocol

2.7.1 FMIPv6 Terminology

Some terminology to help in the understanding of the messaging of FMIPv6 [19].

- Previous Access Router(PAR): The PAR is the router the MN is connected to before a handover is triggered.
- New Access Router (NAR): The NAR is the router the MN is connected to after a handover is triggered.
- Previous Care-of Address (PCoA): The MN's CoA that is only valid on the subnet of the PAR.
- New Care-of Address (NCoA): The MN's CoA that is only valid on the subnet of the NAR.
- Router Solicitation for Proxy Advertisement (RtSolPr): An RtSolPr message is used to retrieve information of a potential new access router and is sent from the MN to the PAR.
- Proxy Router Advertisement (PrRtAdv): An PrRtAdv message is used to gather information about the neighbouring links and is sent from the PAR to the MN.
- Neighbourhood Discovery: This process is used to resolve neighbourhood AP-IDs to AR-Info.

- Fast Binding Update (FBU): The MN sends an FBU to the PAR, this message is used to trigger a redirection of the MN's traffic to the NAR.
- Fast Binding Acknowledgment (FBAck): When a PAR receives an FBU, it will respond with an FBAck message.
- Predictive Fast Handover: When the MN is able to send an FBU when its still connected to the PAR is called Predictive Fast Handover. In this case the MN can predict the NAR before the handover process starts.
- Reactive Fast Handover: When the MN is only able to send an FBU when its connected to the NAR is called Reactive Fast Handover. In this case the MN can not predict the NAR before the handover process starts.
- Unsolicited Neighbour Advertisement (UNA): An UNA will be sent to the NAR by the MN as soon as the MN establishes connectivity with the NAR. The UNA will trigger any buffered or arriving packets to be forwarded to the MN.
- Handover Initiate (HI): A message stating that the MN's handover process that is about to start and is send from the PAR to the NAR.
- Handover Acknowledge (HACK): A response message on an HI message and is send from NAR to PAR.
- AP-ID, AR-Info tuple: The tuple consist of the AR's IP addresses, L2 information and valid prefixes on the AR's interface.

2.7.2 Protocol Overview

The FMIPv6 protocol starts when the MN sends a RtSolPr message to the PAR to resolve one or more AP identifiers to subnet-specific information. In response to a RtSolPr message, the AR, in this case the PAR, will send a PrRtAdv message to the MN. The PrRtAdv consists of one or more [AP-ID, AR-Info] tuples. The rate of RtSolPr messages is determined by the MN and can be sent at any convenient time, like when performing router discovery or as a response to some link-specific triggers. Before the MN can send a RtSolPr message, the MN must first discover the available APs by link-specific methods.

The MN uses information from the PrRtAdv message to formulate a NCoA. After the NCoA has been formulated the MN sends a FBU update to the PAR. FBU's are used to authorize the PAR to make a binding between PCoA and the NCoA to start tunneling packets to the new location of the MN. FMIPv6 has two modes of operation, predictive and reactive handover mode. If possible, the FBU should be sent from the PAR's link to trigger predictive handover

mode, else the FBU can be sent from the NAR during the reactive handover mode.

2.7.2.1 Predictive Mode

Figure 2.19 shows how messages are exchanged in the Predictive mode of FMIPv6. During predictive handover mode the FBU and FBack was received on a link from the PAR and will trigger packet tunneling between the PAR and the NAR. As soon as the MN is connected to the NAR, the MN must send a UNA message to start the forwarding of new and buffered packets to the MN immediately. The PAR can first test if the NCoA is valid and available on the NAR by exchanging HI and HAck messages, once this validation process is complete, the PAR will send back the FBack. The HI message contains the new proposed NCoA, if the HAck contains the proposed NCoA the PAR can start using this address to forward packets to the NAR. The PAR then sends a FBack message to the MN to confirm the use of the NCoA, and the MN must use the assigned IPv6 address and not the one that was used in the FBU message. After the MN is connected to the NAR, a UNS message is sent to start the forwarding of buffered and arriving packets [3, 19, 37].

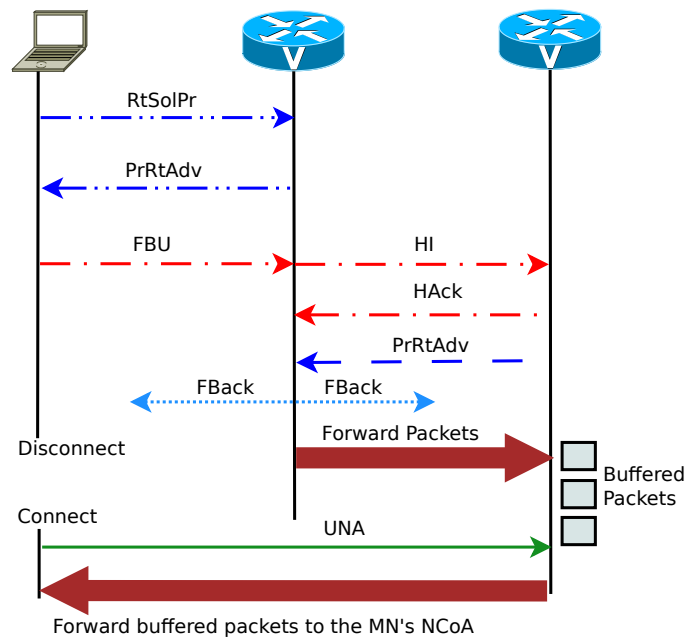


Figure 2.19: FMIPv6 Predictive Handover [3]

2.7.2.2 Reactive Mode

In Figure 2.20 an FBack is not send back to the MN. The reason for not receiving an FBack on the previous link can be due to the fact that the MN did not sent an FBU or the MN disconnected from the link before receiving the FBack. The MN is not sure if the FBU has been received and processed successfully due to the fact that no FBack was received from the PAR. In the case where the FBack has not been received, a new FBU message will be send immediately after the UNA message has been sent [3, 19, 37].

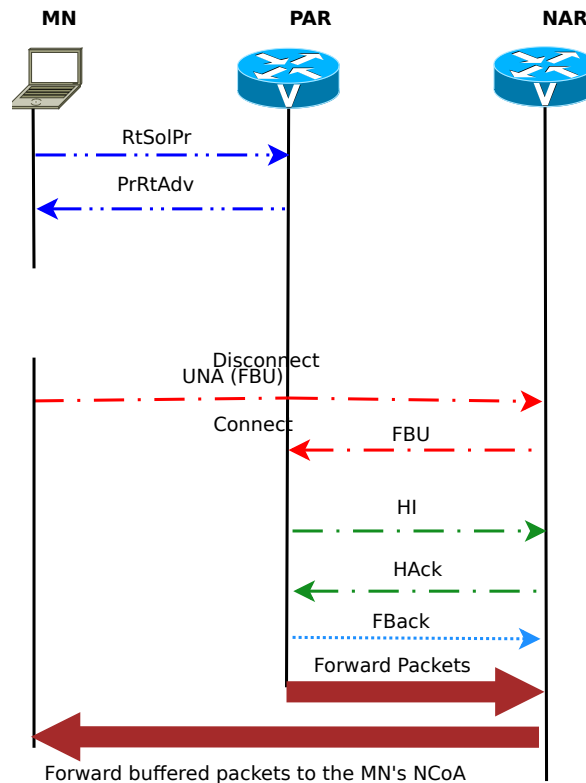


Figure 2.20: FMIPv6 Reactive Handover [3]

2.7.3 Detail Protocol Functioning

For the detailed functioning of the protocol, we refer to Figure 2.18. After the MN has finished discovering nearby APs, **RtSolPr** messages are send by the MN to the PAR, to determine AP identifiers and subnet router information. **RtSolPr** messages are send by the MN at any time one or more APs have been discovered, but the most convenient way is right after router discovery has completed. The trigger for sending **RtSolPr** messages can be link-specific data, like a stronger signal strength at a nearby AP and fading signal strength

at its current AP. When the PAR receives a RtSolPr message, it will reply with a PrRtAdv message. This message will indicate the following conditions:

- If the PAR cannot find an entry for a certain AR contained in the RtSolPr message, the PAR must reply that the AR is unknown. FMIPv6 will then stop on the MN because the surrounding ARs do not support FMIPv6.
- When the new AR is connected on the same link as the PAR, the PAR will indicate to the MN that the AR is on the same link and will not include any prefix information in the message. This scenario may occur when multiple APs are bridged into a wired network.
- If the new AR is identified by the PAR, the PAR will respond to the MN indicating that the AP is recognized and also supplies the [AP-ID, AR-Info] tuple.
- If the AP is not recognized the PAR should supply neighbourhood [AP-ID, AR-Info] tuples that are subject to path maximum transmission unit (MTU) restrictions.

If the router supports a fast handover mechanism, PrRtAdv and RtSolPr messages must be implemented by the MN and the the access routers. However when the MN attaches to the NAR and all the information required to send packets immediately is supplied by the link layer handover technology, the above messages are optional. After the MN has received a PrRtAdv message, the MN sends an FBU to the PAR router. The timing of this message depends on link-specific events like signal strength. The FBU includes a proposed NCoA for the MN ones connected to the NAR. The MN must, whenever possible transmit the FBU from the link associated with the PAR. When the MN cannot anticipate when the handover will occur, the FBU can still be sent from the NAR link. If the MN does not receive a FBack from the previous link, the MN must send a FBU immediately after attaching to the NAR. After the PAR receives the FBU from the MN, an HI message must be transmitted from the PAR to the NAR. In reply to the HI message the NAR will send a HAck message to the PAR. In order for the PAR to determine the address of the NAR, the PAR can make use of longest prefix match on the NCoA prefix list of the neighbouring access routers. The HI message is composed of the PCoA, link-layer address and the proposed NCoA of the MN. The HI message sent from the PAR link will contain a Code 0, and if the HI message was sent from the NAR link it will contain the Code 0.

An HI message with Code 0 will be processed as follows: The NAR determines whether the proposed NCoA is valid on the link and unique. DAD is used to determine the uniqueness of the address. DAD can also be avoided in the protocol if the probability of the address not being unique on the network

is low. Once the address has been identified as unique, the NAR can start proxying for the lifetime that the MN is expected to be connect to the NAR. The NAR will then create a proxy neighbour cache entry for the NCoA and will also begin to defend the NCoA. If the NCoA cannot be assigned or accepted a host route must be set up between the PCoA. In this case a reverse tunnel must be set up between the PAR and the NAR to intercept packets destined for the PCoA, after which the NAR sends a HAcK message to the PAR. If the Code of the HAcK message is 1, all of the above operations must be skipped.

If an HI message with code 1 is used, the NAR will validate the created neighbour cache entry for the MN.

This means that the NAR can use the knowledge that its trusted peer, PAR, has a trust relationship with the MN. When the NAR assigns a different NCoA as proposed by the MN in the FBU message, the NAR will include this NCoA in the HAcK message to the PAR. The PAR must then include this address in the FBacK message to the MN and the MN must use this NCoA. The result of a FBU and FBacK is that the PAR starts to tunnel packets to the NCoA of the MN. When the MN establishes link connectivity at the NAR the following protocol operations start:

- The MN sends a UNA message to the NAR. If no FBacK has been received by the MN, it must send a FBU message following the UNA message.
- MN joins the solicited-node and all-nodes multicast groups corresponding to the NCoA.
- At this point, DAD probing can start depending on the configuration of the protocol.
- As soon as the NAR receives the UNA message it will delete the neighbour cache entry and the NAR starts forwarding the new and buffered packets to the MN.

For data forwarding, the PAR tunnels packets using its global IP address valid on the interface to which the MN was attached. The MN reverse tunnels all of its packets to the same global IP address of the PAR. When the PAR receives a reverse-tunneled packet, it must verify whether a secure binding exists for the MN identified by the PCoA in the tunneled packet, before forwarding the packet.

2.7.4 Other Considerations of FMIPv6

- Handover Capability Exchange: In scenarios where the PAR does not support FMIPv6 capabilities, the MN will send RtSolPr until the maximum number of RtSolPr retries are reached. If the MN did not receive

any PrRtAdv messages in response to the RtSolPr messages, the MN will assume that the PAR does not support FMIPv6. FMIPv6 will then be terminated and stop sending RtSolPr messages.

- **DAD Handling:** The biggest part of handover latency is caused by Duplicate Address Detection to verify the uniqueness of a IPv6 address on a link. The probability of an address not being unique is very low, but cannot be ignored, and thus the need for DAD does exist. In FMIPv6 the NAR sends a DAD probe before the NAR starts to defend the NCoA, and thus minimizing DAD delay. The NAR will perform DAD before the MN is connected and the success of this process is translated to the MN with the use of a HAck message.
- **Assignment of NCoA:** The NCoA is formulated by the MN with the information received in the PrRtAdv message. However, if the MN is not sure whether this address is usable at the NAR, the MN will send a FBU message to the PAR. This indicates to the PAR that the MN wants to do a handover. The reason for handover is not fixed and depends on different factors. The PAR will then send a HI message to the NAR and it must use the NCoA configured by the MN in the HI message. The NAR will then use DAD to verify the correctness of the NCoA address and whether the address can be used on the link. The NAR will then reply to the HI message with a HAck message, which contains either the original IP address formulated by the MN or the new address formulated by the NAR due to DAD failure or wrong formulation. The PAR will receive the HAck message and will forward these data by using an FBack message. The MN is then required to use the NCoA in the FBack message. If the MN does the handover after sending the FBU, the address can be verified after the connection to the NAR. However handover latency will increase due to extra time spend on DAD. When DHCP is used the MN must perform DHCP operations once it attaches to the NAR even though it formulates an NCoA for transmitting the FBU.
- **Prefix Management:** The prefix that is valid on the link to which the MN connects, is contained in the prefix part of the AR-Info message. FMIPv6 works regardless of the length, the assignment policies or the management of the prefix on the network. When per-mobile prefix assignment is used, the AR-Info advertised in PrRtAdv still includes the (aggregate) prefix valid on the interface to which the target AP is attached, unless the access routers communicate with each other (using HI and HAck messages) to manage the per-mobile prefix [19]. The NCoA will still be formulated by the MN using the aggregate prefix, but the return prefix in the HI and HAck messages will use the per-mobile prefix.

- Packet Loss: Handover involves link switching, which may not be exactly coordinated with fast handover signaling. The arrival of a pattern of packets is dependent on many factors, including application characteristics, network queuing behaviours, etc. This means that packets is lost if the packets arrive at the NAR before the MN has completed the handover process and the NAR is not able to buffer these packets. Similarly, if the MN attaches to the NAR and then sends an FBU message, packets arriving at the PAR, until the FBU is processed, will be lost unless they are buffered. FMIPv6 supports the option to request buffering at the NAR in the HI message. If the NAR supports buffering this protocol can support a smooth handover, but the buffer size and the rate at which buffered packets are eventually forwarded are important considerations when providing buffering support.
 - Some applications will transmit fewer packets over a certain period and this will continuously change the buffering requirements. Voice over IP will need less buffer space than live video streaming because of the difference in packet size and packet rate.
 - When the MN has completed handover to a new link, the buffering router may bottle neck the network routers or the MN with buffered packets. In particular, transmitting a large number of buffered packets in succession can congest the path between the buffering router and the mobile node. Furthermore the base station between the MN and NAR, may drop some of the packets or even buffer the packets itself. This will also cause an increase in the jitter of the packets arriving after the buffered packets.
 - Routers are not involved in end-to-end communication, so this means that they will not have any knowledge of the transport conditions of the network.
 - The wireless connectivity of the mobile node can also change and can cause a decrease in bandwidth. This may cause congestion when the buffered packets are being forwarded.

The above points just show why it is very difficult to formulate an algorithm to handle buffering under all these scenarios. The purpose of FMIPv6 is to reduce packet loss. Routers that are implementing buffering of packets, must at least use the default algorithm, which is based on the original arrival rates of the buffered packets. A maximum of 5 packets may be sent one after another, but all subsequent packets should use a sending rate that is determined by metering the rate at which packets have entered the buffer, potentially using smoothing techniques such as recent activity over a sliding time window and weighted averages [19, 47, 48].

- Fast or Erroneous Movement: IP layer mobility introduces its own limit in the speed of movement by the MN. The handovers should occur at such a rate that the binding message can be made to the Home Agent and the Correspondent Node in the network. Packet loss may occur if the MN moves faster than this speed.

2.8 Proxy Mobile IPv6

Proxy Mobile IPv6 [20] is the standardized mobility protocol by IETF for network-based localized mobility management in IPv6 networks. PMIPv6 is a solution especially for inter-access system handover between 3GPP and non-3GPP networks. The advantage of PMIPv6 over the previously mentioned Mobility Protocols is that PMIPv6 requires only changes to the edge routers and no changes to the IPv6 stack of the MN. This makes PMIPv6 a very promising network mobility option and can support mobility service to a wide range of network users. All the network mobility intelligence resides in the network thus PMIPv6 becomes questionable and requires a solution for deployment in large scale networks [49]. A basic setup of PMIPv6 can be seen in Figure 2.21 and the new terminology is described in Subsection 2.8.1.

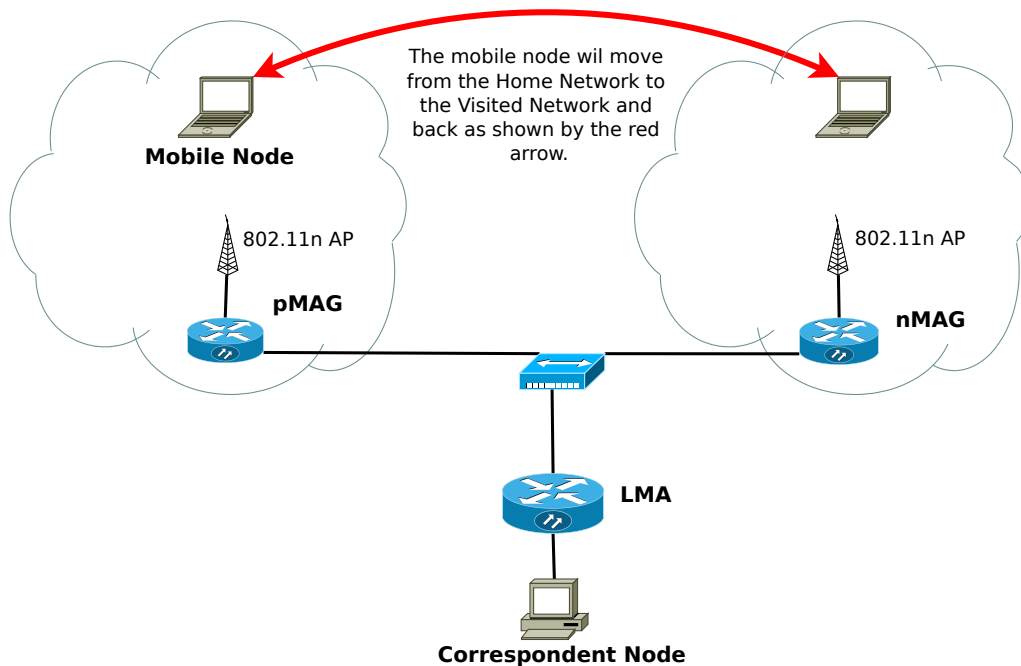


Figure 2.21: PMIPv6 Network

2.8.1 PMIPv6 Terminology

The following describes the new terminology introduced by PMIPv6.

- **Mobile Access Gateway (MAG):** The MAG is used to detect the movement of the MN to and from the access link and to initiate mobility messages with the LMA on behalf of the MN.
- **Local Mobility Anchor (LMA):** The LMA is like the Home Agent in MIPv6. The LMA is the topological anchor point for the MN's home network prefixes and manages the MN's binding state. The LMA has the same functionality as that of the HA but can also support PMIPv6.
- **Proxy Binding Update (PBU):** The MAG sends a PBU message to the LMA to initiate and setup a binding between MAG and LMA.
- **Proxy Binding Acknowledgement (PBA):** The MAG waits for the PBA message from the LMA as conformation of binding status.
- **Proxy Mobile IPv6 Domain (PMIPv6-Domain):** This domain refers to the network where the mobility management of the MN is handled by PMIPv6.
- **LMA Address (LMAA):** This is the global address that is configured on the LMA. This address forms the endpoint of the bidirectional tunnel between the LMA and the MAG. The MAG send PBU to this address.
- **Proxy Care-of Address (Proxy-CoA):** The Proxy-CoA is the global address configured on the MAG that forms the transport endpoint of the bidirectional tunnel between MAG and LAM. This address is viewed by the LMA as the Care-of Address of the mobile node [49].
- **Mobile Node's Home Network Prefix (MN-HNP):** The MN-HNP is the prefix assigned to the link between the mobile node and the MAG. Multiple prefixes can be assigned between the link and all of the assigned prefixes are managed as a set associated with a mobility session. Multiple prefixes assigned to one interface will be managed as one mobility session.
- **Mobile Node's Home Address (MN-HoA):** This is the address the mobile node received on the home network's prefix. The MN is able to use this address as long as the MN is attached to the PMIPv6 network domain.
- **Mobile Node Identifier (MN-Identifier):** The identity of the MN in the PMIPv6 network domain. This is usually a Media Access Control (MAC) address or a Network Access Identifier (NAI) and is used to predictably identify a MN.

- Mobile Node Link-layer Identifier (MN-LL-Identifier): An identifier that identifies the attached interface of a mobile node. Typically the link-layer identifier will be used as the MN-LL-Identifier.
- Mobile Node's Home Link: A layer 3 address is obtained from this link with address configuration once the MN has moved into the PMIPv6 network domain [49].

2.8.2 Protocol Overview

This description refers to Figure 2.21 and Figure 2.22. PMIPv6 is designed to implement network mobility without requiring any changes to the IPv6 stack of the MN. The MN is also not required to participate in any of the mobility signaling. The mobility entities in the network like the LMA and MAG, tracks the movement of the mobile node in the PMIPv6 domain and initiates the mobility messaging and sets up the required routing. The main functional entities in the NETLMM network are the Local Mobility Anchor and the Mobility Access Gateway. The LAM is mostly responsible for maintaining the MN's connection and reachability on the PMIPv6 domain. LMA forms the topological anchor point for the MN's home network prefixes.

The MAG is used to detect the MN's movements and to initiate mobility messages to the LMA on behalf of the MN. The MAG is used to track the movement of the MN's link to and from the access link. The MAG also performs binding registrations to the LMA on behalf of the MN. The PMIPv6 architecture can be seen in Figure 2.21.

When the MN enters the PMIPv6 domain and connection to the access link is complete, the MAG will acquire the identity of the MN. The MAG will then determine if the MN is authorized to use the network-based mobility management service. After the authorization of the MN has been completed, the MN will obtain address configuration by using any of the address configuration mechanisms available on the network. The address configuration consists of obtaining IPv6 address/es from its home network prefixes/es, the default router address on the link and various other configuration parameters which is discussed later in the section. The entire PMIPv6 domain appears as a single link for the MN. This ensures that the MN does not detect any changes with respect to its layer-3 attachment even if the MN moves around in the network.

In the scenario where the MN connects to the PMIPv6 domain using multiple interfaces, the network will allocate a unique set of home prefixes for each of the interfaces of the MN. The home addresses that is configured on the interfaces will make use of the different home prefixes. A handover can be performed by either changing from one interface to another on the same MAG, or by changing from one MAG to a different MAG in the PMIPv6 domain.

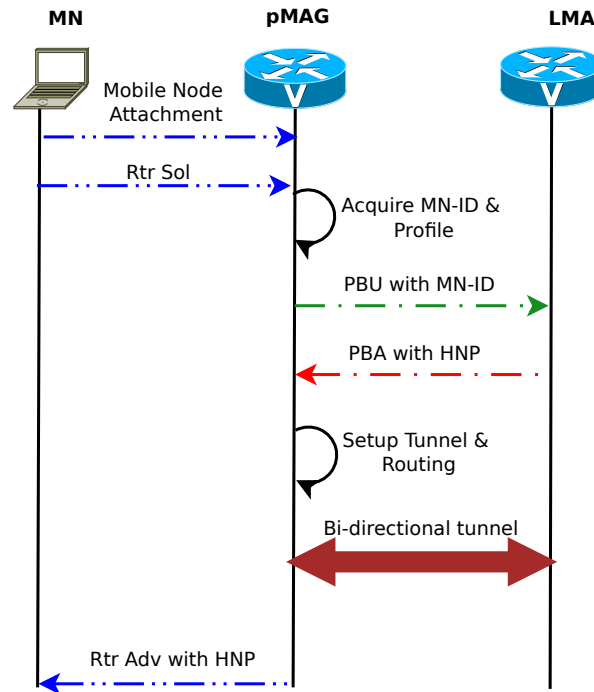


Figure 2.22: PMIPv6 Message Flow

The message flow is shown in Figure 2.22. The Router Solicitation message sent from the MN may arrive at any time after the connection has been made to the MAG and has no ordering restriction towards the message call flow. The local mobility anchor needs to be updated about the MN movement. The MAG sends a Proxy Binding Update message to the LMA and if the LMA accepts the Proxy Binding Update it will reply with a Proxy Binding Acknowledgement. The Proxy Binding Acknowledgement includes the home network prefixes on the interface/s. After completion of the Proxy Binding Acknowledgement the LMA also sets up the bidirectional tunnel with the tunnel endpoint as the MAG address. After the MAG has received the Proxy Binding Acknowledgement, it sets up its bidirectional tunnel endpoint as the LMA. Forwarding will also be set up for the MN's traffic. The MAG will start to send Router Advertisement to the MN, advertising the home network prefixes as the hosted on-link prefixes.

When the MN receives the Router Advertisement it will attempt to configure its interface using either stateful or stateless address configuration. This decision is based on the configuration modes that are permitted on the access link as indicated by the Router Advertisement. If the address configuration has been completed successfully, the MN will have one or more addresses from its home network. The MAG serving the MN and the LMA also set up proper routing states for handling traffic to and from the MN. The LMA is the topo-

logical anchor point of the PMIPv6 domain and thus intercepts all the packets destined for the MN from a source in or outside of the PMIPv6 domain. These packets are then forwarded to the MAG serving the MN with the use of the bidirectional tunnel that was setup. The MAG receives the packets from the LMA destined for the MN, removes the outer header from the packets and forwards the packets to the MN. If data is sent from a correspondent node connected locally to the MAG, it is forwarded to the MN and not to the LMA.

If the MN sends packets to a correspondent node, the MAG forwards packets to the LMA with the use of the bidirectional tunnel. When the packets are received at the LMA, the outer header is removed and the packets are forwarded to the destination address. The reason for removing the outer addresses is to ensure IP transparency and tunneling.

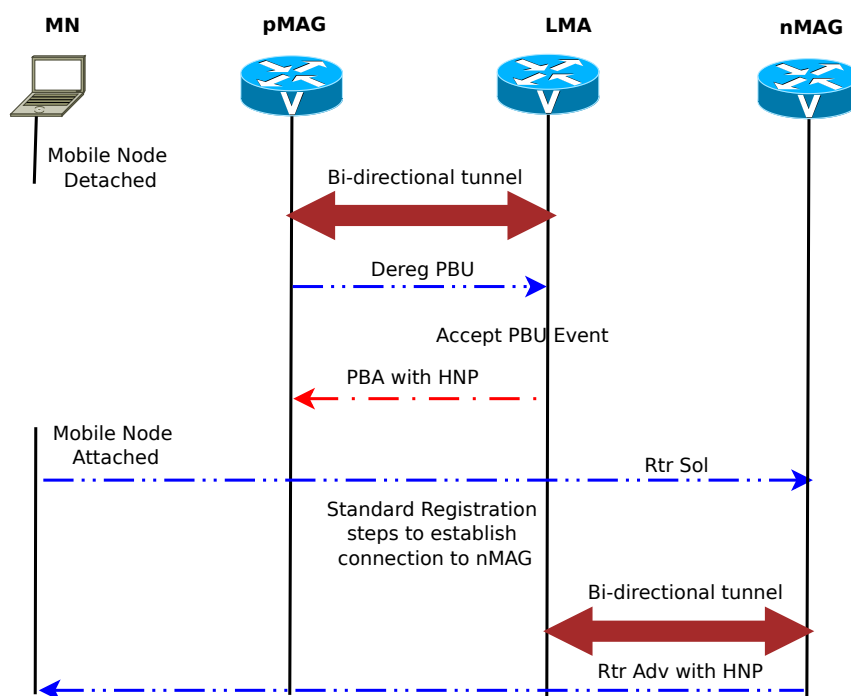


Figure 2.23: PMIPv6 Handover Signaling Flow

Figure 2.23 shows the signaling flow during handover from a previous MAG to a new MAG. When the MN moves from a MAG to a new MAG, the MN has already configured its IPv6 address for the PMIPv6 domain. The previous MAG will register the movement of the MN and signal to the LMA to remove the binding and routing state for the MN. The LMA will identify the mobility session for the specific MN, after this the LMA will wait for certain amount of time to allow the MAG on the new access link to update the binding of the MN. If the LMA doesn't receive a PBU from the new MAG, the LMA

will delete the binding cache for that MN. The message signaling will be the same from here as illustrated in Figure 2.22. The new MAG will set up a bidirectional tunnel with the LMA as the endpoint and the handover will be transparent to the application layer on the MN and CN. All of the signaling messages must be protected by a end-to-end security association and data origin authentication. Proxy Binding Update, Proxy Binding Acknowledgement and other signaling messages make use of IPSec for protection and is the mandatory security mechanism.

2.9 Hierarchical Mobile IPv6

Internet engineering task force(IETF) has proposed hierarchical mobile IPv6 (HMIPv6) in order to reduce a frequent location registration of a mobile node in MIPv6 [50]. The main focus point of HMIPv6 is to reduce the number of signaling required when changing access points and thus improving handover speed and network overhead caused by MIPv6. HMIPv6 also separate local mobility from global mobility, this is achieved by using MIPv6 mechanism for global mobility, while local handoffs are managed locally. HMIPv6 is necessary to improve the scalability of MIPv6, a large number of overhead is added in large scale MIPv6 networks due to the exchange of binding updates messages. HMIPv6 is not included and evaluated in the thesis because the thesis do not focus on the scalability of these protocols. One disadvantage of HMIPv6 is the MAP, the MAP forms a single point of failure in the network. A basic setup of HMIPv6 can be seen in Figure 2.24 and the new terminology is described in Subsection 2.9.1.

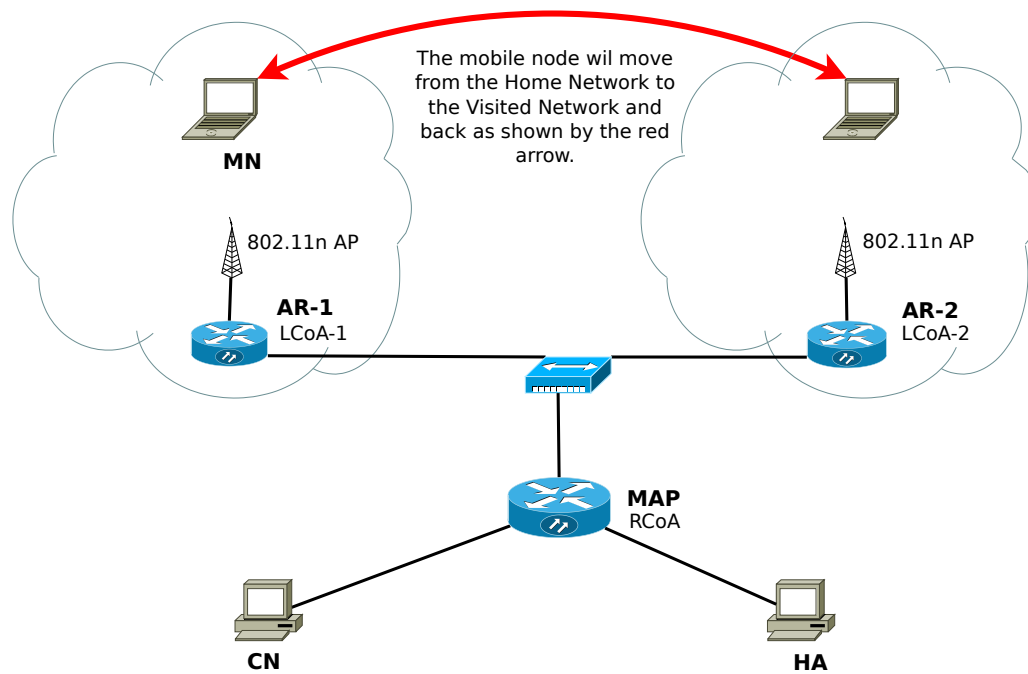


Figure 2.24: HMIPv6 Network

2.9.1 HMIPv6 Terminology

The following describes the new terminology introduced by HMIPv6 [13].

- Access Router (AR): The AR is the access point the MN connects to and is used to aggregate the outbound traffic of mobile nodes.
- Mobility Anchor Point (MAP): An MAP is a router located in a network visited by the MN. The MAP acts as local home agent for the MN and there can be more than one located in a visited network.
- Regional Care-of Address (RCoA): An RCoA is an address allocated by the MAP to the MN.
- On-Link Care-of Address: The LCoA is the on-link CoA configured on a MN's interface based on the prefix advertised by its default router. The LCoA is basically the same as the CoA in MIPv6.
- Local Binding Update (LBU): The LBU is used to establish a binding between the RCoa and the LCoA.

2.10 Discrete Event Simulator

In this section different Discrete Event Simulator (DES) software was examined for use to simulate MIPv6 networks to determine the most appropriate

simulator. In section 2.10.3 a comparison is made between OMNeT ++ and ns-2. The comparison focuses on the framework structure, model management, programming model and tracing ability of each simulation framework. In classical thinking there are three types of simulation: discrete event, continuous, and MonteCarlo. A DES simulation is described by Mike Albrecht the following way:

Discrete event simulation utilizes a mathematical/logical model of a physical system that portrays state changes at precise points in simulated time. Both the nature of the state change and the time at which the change occurs mandate precise description [51].

A DES simulator is composed of the following components [52]:

- **Clock:** A clock is used so that the simulation can keep track of the current simulation time. In discrete-event simulations, as opposed to real time simulations, time 'hops' because events are instantaneous - the clock skips to the next event start time as the simulation proceeds.
- **Events List:** The simulation will at least maintain one list of all the simulation events. This is sometimes called the pending event set because it lists events that are pending as a result of previously simulated events but has yet to be simulated themselves. Events are described by the time at which it occurs and its type.
- **Random-Number Generators (RNG):** Random numbers needs to be generated for the simulation and is dependent on the system model. For the simulation to be rerun with the same values, pseudorandom number generators are used. This allows the simulation to be rerun with the exact same behavior over and over. To overcome one problem of random number generators in DES, bootstrapping needs to be performed on the simulation model. The problem that exists with the use of random number generators in DES is that the steady-state distributions of event times may not be known in advance. As a result, the initial set of events placed into the pending event set will not have arrival times representative of the steady-state distribution. These initial events, however, schedule additional events, and with time, the distribution of event times approaches its steady state and this is called bootstrapping [53].
- **Statistics:** The simulation model will also keep track of certain statistic information during simulation.
- **Ending Condition:** A DES could theoretically run for ever because of bootstrapping. Ending conditions must be used in the simulation to stop the simulation when certain conditions are reached.

2.10.1 OMNeT++

OMNeT++ is a modular, component based and open-architecture DES and is used for the simulation of communication networks, complex IT systems and queuing networks. The OMNeT++ 4.x Integrated Development Environment is based on the Eclipse platform and basic layout of the environment can be seen in Figure 2.25. As mentioned the component structure of OMNeT++, models can be assembled from reusable components and can be combined in various ways like Lego blocks to build the desired model. OMNeT++ have a wide range of simulation packages that can be used to simulate different types of simulation. For network simulations one of the most popular packages is the INET framework [54]. The INET framework contains several models for both wireless and wired communications.

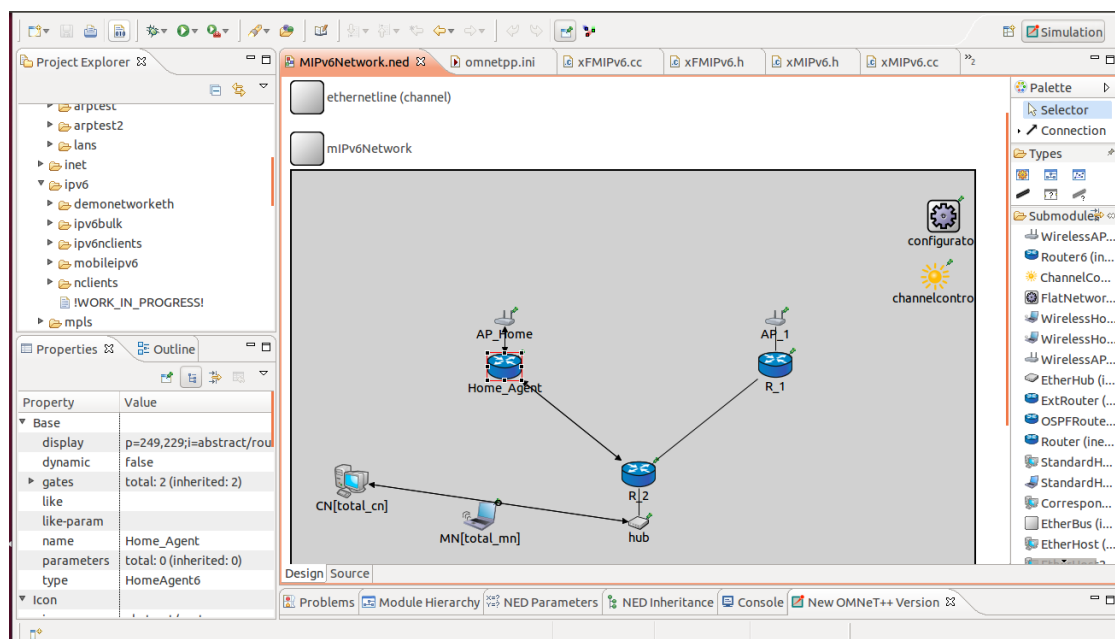


Figure 2.25: OMNeT++ Simulation Environment

Modules can be connected with each other via gates and combined to form compound modules. An example of a compound module, in this case the IPv6 compound module can be seen in Figure 2.26. Connections are created within a single level of module hierarchy: a submodule can be connected with another, or with the containing compound module. Every simulation model is an instance of a compound module type. In Figure 2.26 illustrates how the modules are connected together to form a working IPv6 compound module.

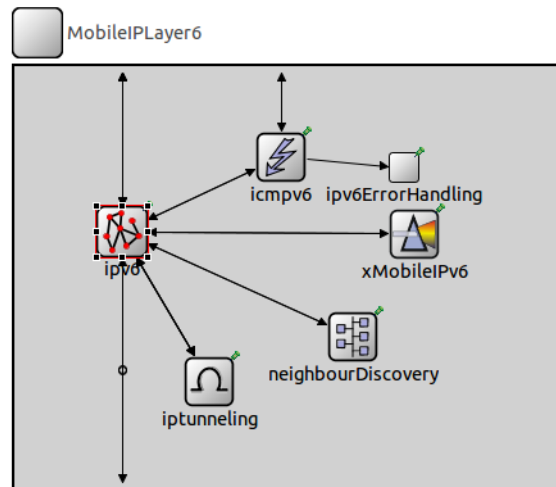


Figure 2.26: OMNeT++ IPv6 Compound Module

As mentioned in Section 2.10, a DES makes use of random number generators. OMNeT++ default random number generator is Mersenne Twister. The other RNG is a legacy LCG generator with a 231-1 long sequence generator. The Mersenne Twister RNG provides fast generation of very high-quality pseudorandom numbers and is based on a matrix linear recurrence over a finite binary field [55]. The statistics that are recorded in OMNeT++ are in the form of output vectors and output scalars. These statistics can either be visualized in OMNeT++ with the use of the Analysis Editor in the IDE environment, or with your own personal choice of analysis software. The Analysis Editor for OMNeT++ can be seen in Figure 2.27 and an example of a line chart in the analysis editor can be seen in Figure 2.28.

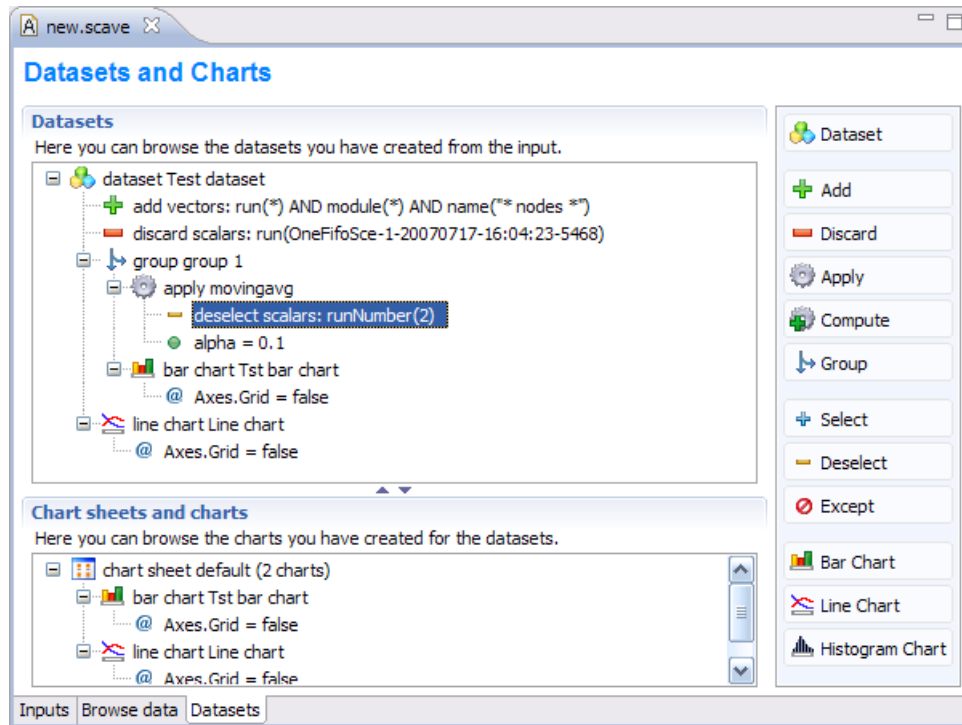


Figure 2.27: Analysis Editor

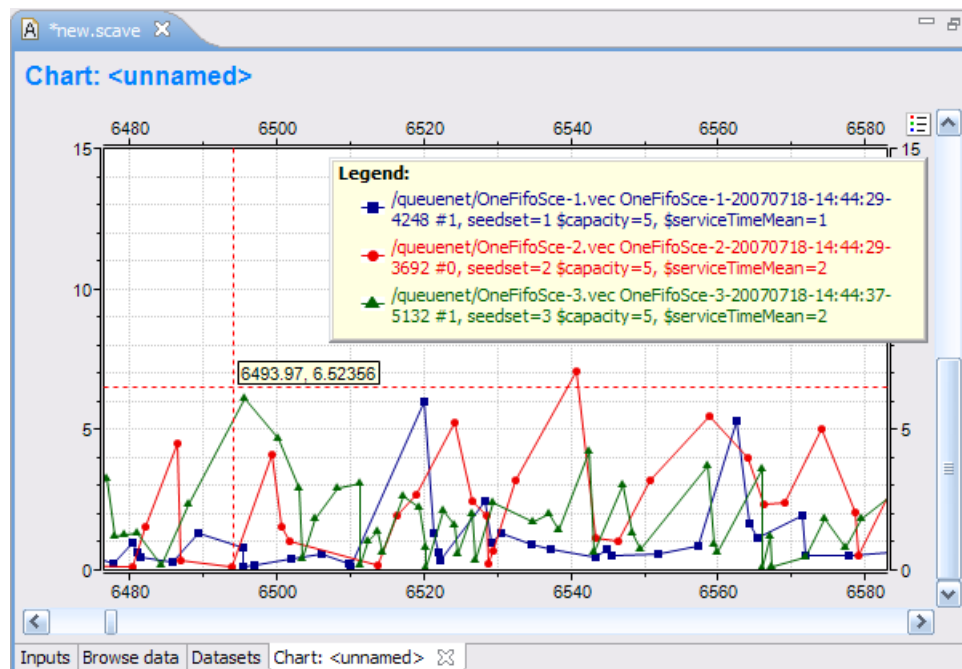


Figure 2.28: Analysis Editor Charts

2.10.2 ns-2

Network Simulator 2 (ns-2) [56] is used for simulation of routing, TCP, ad hoc, propagation and multicast protocols in wireless and wired networks. ns-2 works together with Network Animator (NAM) and is used to visualize the simulation as animations. NAM is a Tcl/TK based animation tool viewing ns-2 simulation traces and real world packets [57]. NAM supports topology layout, packet level animation and data inspection tools. Figure 2.29 shows an example of the Network Animator for ns-2.

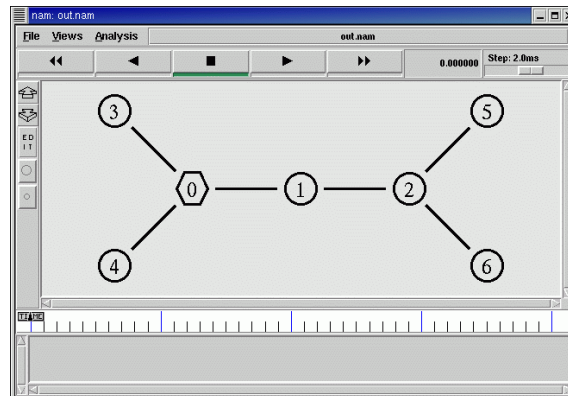


Figure 2.29: ns-2 Network Animator

2.10.3 Comparison of OMNeT++ and ns-2

Table 2.10: Comparison of OMNeT++ and ns-2 [9]

Value	OMNeT++	ns-2
Flexibility	OMNeT++ was build as flexible and generic simulation network. Flexibility of of OMNeT++ is made easy by using components, this allows OMNeT++ to simulate basically everything that can be mapped to components. OMNeT++ consist of several model frameworks that can be used for different simulation domains, like INET, MACSimulator, MIXIM etc.	ns-2 was designed and build as a (TCP/IP) simulator and restricts ns-2 from simulating any other models except packet-switching networks and protocols.

Continued on next page

Table 2.10 – continued from previous page

Value	OMNeT++	ns-2
Debugging and Tracing Support	Packet transmission in OMNeT++ can be seen while the simulation is running. An interactive execution environment is provided by OMNeT++'s Tkenv, this allows live examining of the simulation and changing of parameters.	Any standard C++ debugger will work and ns-2 also supports TCL level debugging with Don Libs' Tcl debugger. ns-2 do not have an interactive simulation environment. [57].
Variety of Models Available	OMNeT++ consists of a number of models for simulating queuing systems, network systems etc.	ns-2 however have a rich set of communication protocols models.
Experiment Design	In OMNeT++ the model and experiment is separated. This behavior is enforced by adding parameters to .ini files.	In ns-2 the separation of model and experiment is somewhat more difficult to achieve. Topology, parameters, models and results are all based in the same Tcl script.
Support for Hierarchical Models	Hierarchical models can be achieved by making use of simple and compound models. Compound models is defined by combining several simple models together and can be reused in the simulation.	Creating composition units in ns-2 and reusing these units is impossible. ns-2 have a flat model structure and thus impossible to create composition models.
Model Management	The simulation kernel is a class library, which means that the models are independent of the simulation kernel. Models are written against the simulation API and simple models can be combined like building blocks to create simulations.	No boundary is defined in ns-2 between the models and the simulation core. Combining models in ns-2 is not possible.
Programming Model	OMNeT++ is an object-oriented, event-driven simulator and is written in C++. Topology can either be created as text files (NED language) or dynamically during runtime. Network topologies can be created with the use of GNED, which is a graphical editor for OMNeT++.	ns-2 consists of OTcl with underlying C++ classes. The creation and configuration of network is done in OTcl.

2.10.4 Cooja

Cooja [58] is a sensor network simulator designed and used in Contiki. Cooja allows large and small networks of Contiki motes¹, or more widely known as sensor nodes to be simulated. Motes can be emulated at the hardware level, which is slower but allows precise inspection of the system behavior, or at a less detailed level, which is faster and allows simulation of larger networks. Cooja controls and analyzes a Contiki system via a few functions and this allows Cooja to simulate motes at hardware level. The main reason for choosing Cooja is because it can simulate the hardware (actual mote) on the hardware level. These simulations can then be easily ported over to the actual mote to be used in further experiments. The Cooja simulator was used to investigate the implementation of 6LoWPAN for low power devices and the possible implementation of MIPv6 in the 6LoWPAN network stack.

2.11 Conclusion

IPv4 addresses are being depleted rapidly because of the fast growing numbers of mobile devices worldwide such as smart phones, tablets, notebooks and sensor nodes. IPv6 [17] is introduced as a replacement for IPv4 due to its more extensive range of IP addressing. Particular attention was given to the differences between IPv4 and IPv6 as well as the improvements presented by IPv6. IPv6 introduces enhancements like auto-configuration, expanded IP addresses, enhanced mobility capabilities and better QoS. This chapter also introduced the need for mobility protocols due to the fast increasing amount of mobile devices worldwide. Network mobility was explained for all the layers of the ISO stack and how mobility can be accommodated on each layer, except for the physical layer where mobility is handled implicitly by the wireless access technology.

Network layer mobility protocols were introduced and explained in detail. These protocols consist of MIPv6 [8], PMIPv6 [20] and FMIPv6 [19]. Mobile IPv6 makes it possible for a mobile node to move around seamlessly in different networks. MIPv6 does not solve the handover latency introduced by the physical layer. MIPv6 introduces several new network elements like the Home Agent, correspondent node, access routers, mobile nodes and handshake messages namely bindings. FMIPv6 focuses to minimize the handover latency by removing duplicate address detection during address assignment and to formulate care of addresses for mobile nodes before the actual handover to the new network will occur. PMIPv6's attention is more focused on network mobility

¹A mote, also known as a sensor node, is a node in a wireless sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. A mote is a node but a node is not always a mote.

than of the mobility of the node itself. In PMIPv6 the network is responsible for mobility handling, the main advantage of this being that mobility is transparent to the mobile nodes.

Furthermore the chapter looked at IPv6 adoption for low power devices. Due to the limited processing power and limited frame size of only 127 bytes, which is a lot less than the 1024 bytes of IPv6, 6LoWPAN [15] is introduced to solve this issue. 6LoWPAN introduces several encoding and compression mechanisms. These mechanisms are used to compress the IPv6 headers and to allow IPv6 communication over low power devices. Due to the availability of IPv6, MIPv6 can also be used on low power devices for seamless movement across networks. This mobility of mobile nodes contributes hugely to the Internet of Things [30]. The Internet of Things refers to uniquely identifiable objects/nodes and their virtual representations in an Internet-like structure.

Lastly, attention was given to various simulation frameworks that can be used for the above mentioned protocols. Cooja was selected for the 6LoWPAN simulations and OMNeT++ for IEEE 802.11n networks. A comparison of the ns-2 and OMNeT++ can be seen in Section 2.10.3. The main advantage of OMNeT++ is the re-usability of models and the combining of models to form compound models. OMNeT++ consists of various simulation packages and as mentioned in Section 2.10.1 the INET package will be used as the base network framework. OMNeT++ was also chosen above ns-2 because of previous experience with ns-2 where results were invalidated because of undocumented behavior by ns-2. The simulation and testbed designs for the various networks as well as the implementation of the IPv6 testbed will be presented in the following chapter.

Chapter 3

Design and Implementation

3.1 Introduction

In the preceding chapter an overview of Mobility Protocols in an IPv6 network was given. The basic and fundamental protocol is MIPv6, this protocol also has a few variations to improve handover latency and overall network performance. This chapter focuses on the design and implementation of the following network setups, simulations and implementation for a test environment:

- Simulation Model: MIPv6 Network
- IPv6 Network
- MIPv6 Network
- FMIPv6 Network
- PMIPv6 Network

The simulation model will make use of the INET Framework [54], this is an open-source communication networks simulation package for OMNeT++. The INET Framework contains several models for wire and wireless connectivity. The simulation model will consist of the basic MIPv6 setup. The network is composed of two foreign networks, a home network and visiting network. The results comprises of handover latency, UDP and TCP throughput.

The Chapter also describes the methodology for the comparison of MIPv6, FMIPv6 and PMIPv6 by using IPv6 Software tools to measure the protocol performances with different traffic source categories such as video, FTP, VoIP and TCP/UDP transfers. The various protocol implementations used for this work were also described and explained. Testbed architectures are presented in this chapter, together with the implementation setup and hardware used for the testbed.

3.2 Simulation Model

The simulation for MIPv6 in OMNeT++ consists mainly of the Extensible Mobile IPv6 (xMIPv6) Simulation Model for OMNeT++ [59]. This model implemented the MIPv6 protocol in OMNeT++ 3.2, but is also available in OMNeT++ Version 4. The network setup for the MIPv6 network can be seen in Figure 3.1. The network consists of one HA, one MN, and multiple CNs.

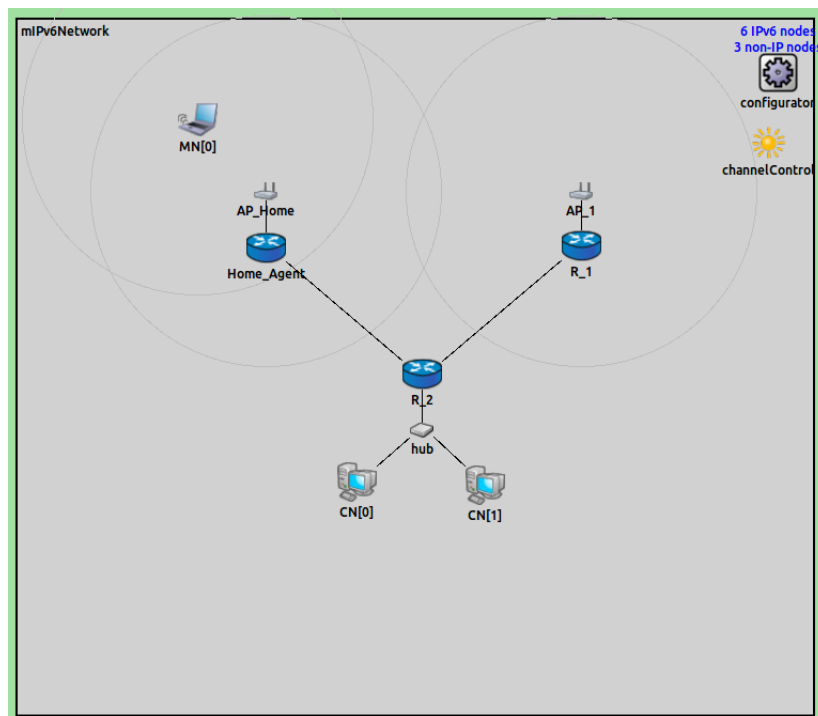


Figure 3.1: OMNeT++ MIPv6 Simulation Network

3.2.1 Home Agent

The HA is a standard IPv6 router, which was developed from the standard Router6 model in INET. Parameters for the HA include `isMobileNode` and `isHomeAgent`. If the first parameter is set to true, the router acts as an MN and if the `isHomeAgent` parameter is set, the router acts as the HA for that network. Both parameters cannot be set to true. The HA will carry a single instance of the `BindingCache` module and this is used to hold the binding data between the HA and the MNs.

3.2.2 Mobile Node

The MN is derived from the INET `StandardHost6` compound node and is an IPv6 enabled node. The MN has a WLAN interface (`Ieee80211NicSTA`) and a

mobility module to handle the mobility of the MN between different subnets. For the MN the `isMobileNode` must be set to true, while the `isHomeAgent` and `isRouter` must be set to false. MN consists of a `BindingUpdateList` to hold the information of bindings made between the MN, CN and HA. In Figure 3.2 the MN IPv6 and BuList module can be seen. The `xMobileIPv6` module is connected to a modified IPv6 INET module. The IPv6 INET module is modified to receive, recognize and forward mobility packets to and from `xMobileIPv6` module.

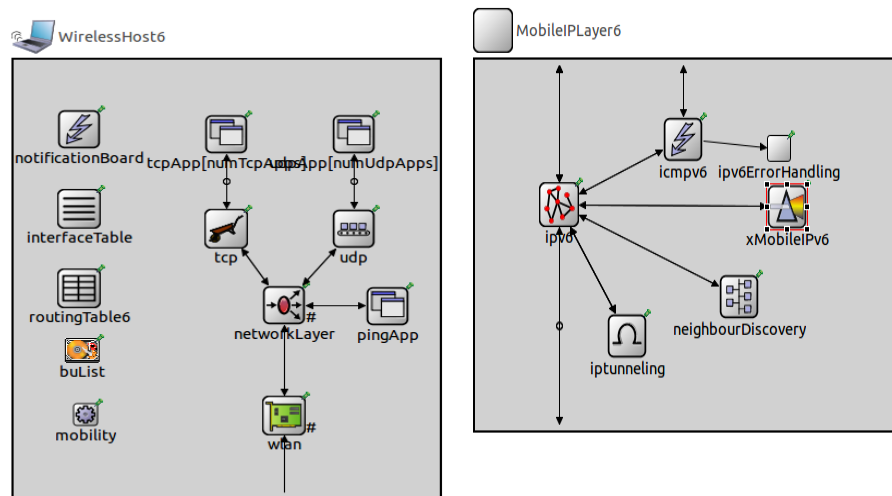


Figure 3.2: OMNeT++ Mobile Node Modules

3.2.3 Correspondent Node

The CN is a generic INET `StandardHost6` node and all of the mobility parameters mentioned are set to false. The CN also includes a `BindingCache` to capture all the binding information for Route Optimisation between the MN and the CN.

3.2.4 Runtime Parameters

The runtime parameters for the simulation can be seen in Appendix A.1. These include all the parameters for the simulation and the parameters that are directly linked to MIPv6 modules are described briefly. A number of different scenarios and parameter values were used to see the effect of the parameters on the handover latency of MIPv6.

- `neighbourDiscovery.minIntervalBetweenRAs`: This parameter is described in RFC 6275 and this specifies the minimum interval between Router Advertisements in the network. The parameter value was varied in the

simulation to see the effect on the handover latency. If the minimum time of the RA is shorter, the MN will receive router information more quickly after handover, and this may cause a decrease in handover latency. However, if the interval becomes too low, it can cause a flooding of the network and a decrease in the throughput of the network. The value for this parameter can be determined by looking at the network configuration and the throughput the network requires to operate, if the throughput is not the main attribute required of the network, the RA interval can be increased.

- `neighbourDiscovery.maxIntervalBetweenRAs`: This parameter is described in RFC 6275 and it specifies the maximum interval between Router Advertisements in the network. See further explanation as described in the paragraph above at `neighbourDiscovery.minIntervalBetweenRAs`.
- **Traffic Generators**: Traffic on the network will be generated using TCP, UDP and PING connections between the MN and the CN. The MN will act as a TCP host and the CN will be the TCP server. This was used to measure throughput and handover latency during full loaded network.
- **AP Wireless Parameters**: The APs in the simulation use the parameters for the wireless environment as described in Appendix A.1. The parameters preceded by `**radio.` relates to the wireless parameters.
- **Mobility**: The mobility used for the MN is rectangular mobility. The MN will move from a preset starting position in a rectangular form. The movement of the MN can be seen in Figure 3.3. The MN starts at position (180,100) and stops at (530,110). From here the MN moves down (530,285) and returns to the home network (180,285). This is done to simulate the MN returning home and to break the bindings between the CN and HA nodes. The speed of the MN was also varied to see the impact of an MN traveling at different speeds on the handover latency. The speed of the MN is also aligned to the update speed of the mobility in the simulation. The update interval will be set very small to overcome any impact on the simulation results. An update interval that is too big will not allow a smooth movement of the MN and will cause a 'jumping' style of movement. The update interval was set to 0.1 s for all the simulations after experimenting with the mobility simulation.

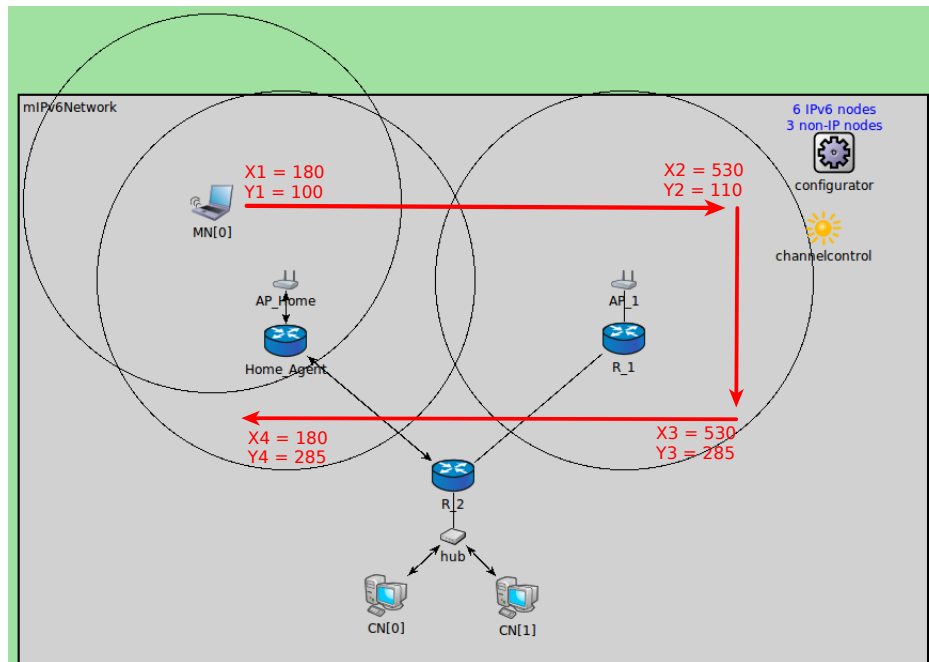


Figure 3.3: MN Movement in Simulation

Table 3.1 shows the different tests that were done on the simulation. The simulation comprised of 8 tests with different parameter values. The first four tests were set up to determine the influence of the RA interval time on the handover latency. The first three tests had random values for the RA time, test four used the RA interval time as described by RFC 6275 [8]. In tests 1-4 no data was generated on the network, because the handover time with no data was the variable tested. During tests 5 and 6 TCP and UDP performance was determined using the prescribed RA interval. Throughput, jitter and packet loss was measured. In tests 7 and 8, the moving speed of the MN was changed from 1m/s to 8m/s and lastly to 15m/s. This showed the effect that the moving speed of the MN had on the protocol.

Table 3.1: MIPv6 Parameters for Test Scenarios

Test Runs	Min RA Int(s)	Min RA Int(s)	TCP	UDP	PING	Speed(ms)
Test 1:	5	10	No	No	Yes	1
Test 2:	1	4	No	No	Yes	1
Test 3:	0.1	1.5	No	No	Yes	1
Test 4:	0.03	0.07	No	No	Yes	1
Test 5:	0.03	0.07	Yes	No	No	1
Test 6:	0.03	0.07	No	Yes	No	1
Test 7:	0.03	0.07	No	No	Yes	8
Test 8:	0.03	0.07	No	No	Yes	15

3.2.5 Statistic Capturing

OMNeT++ offers basic classes which compute basic statistics such as mean and standard deviation; some classes deal with density estimation, and other classes support automatic detection of the end of a transient, and automatic detection of accuracy of collected statistics. For the simulation the following classes in OMNeT++ will be used:

- `cOutVector` is used to record vector simulation results (an output vector, containing (time, value) pairs) to file.
- `cStdDev` keeps mean, standard deviation, minimum and maximum value etc.

$$T_{THO} = T_{HRD} + T_{CRD} + T_{L2D} + T_{RDD} + T_{RRD} + T_{DAD} \quad (3.2.1)$$

The components of handover latency for MIPv6 can be seen in Equation 3.2.1. T_{THO} comprises the total handover latency and this is the sum of the Layer 2 (T_{L2D}) delay and the Layer 3 (T_{L3D}) delay. Figure 3.4 shows the message exchange of MIPv6 and the delays related to the messaging. Figure 3.4 also shows the statistic functions implemented in the simulation to capture the required delay times. The components are:

- T_{THO} = Total Handover Delay: Comprises the complete Layer 2 and Layer 3 handover latency.
- T_{L2D} = Layer 2 Handover Delay: The handover latency incurred by L2 handoff process. The delay was measured as soon as the MN started a scan request and dissociation had been completed with the old AP. The delay was completed when the MN had successfully associated with the new AP. The Layer 2 delay stayed almost the same in all the simulations

because this is not part of the protocol testing. MIPv6 does not propose a way to improve the Layer 2 delay and the parameters for propagation loss will stay the same.

- T_{RDD} = Router Discovery Delay: The time it takes the MN to detect its movement and discover a new access router, which is dependent on the RA interval. The router discovery delay starts when the MN has finished the Layer 2 handover and is completed when the MN receives an RA and neighbour advertisement message. The values of the RA intervals will be varied as seen in Table 3.1.
- T_{DADD} = Duplicate Address Detection Delay: Delay caused by the verification of the uniqueness of IPv6 address(CoA). This delay is combined with the Router discovery delay but was measured independently. DAD forms the biggest part of the total handover delay for MIPv6.
- T_{HRD} = Home Registration Delay: This delay is the time it takes the MN to make a binding association with the HA and receive a BAcK back.
- T_{RRD} = Return Routability Delay: The latency caused by the return routability procedure. CoTI message and HoTI message exchange between the MN, HA and CN. The delay finishes when the MN receives CoT and HoT messages from the CN and HA nodes.
- T_{CRD} = Correspondent Registration Delay: The delay which occurs when the MN makes a binding association with the CN until a BAcK message has been received. This delay takes place only when return routability is used. If Return Routability is not used, there will not be any delay here.

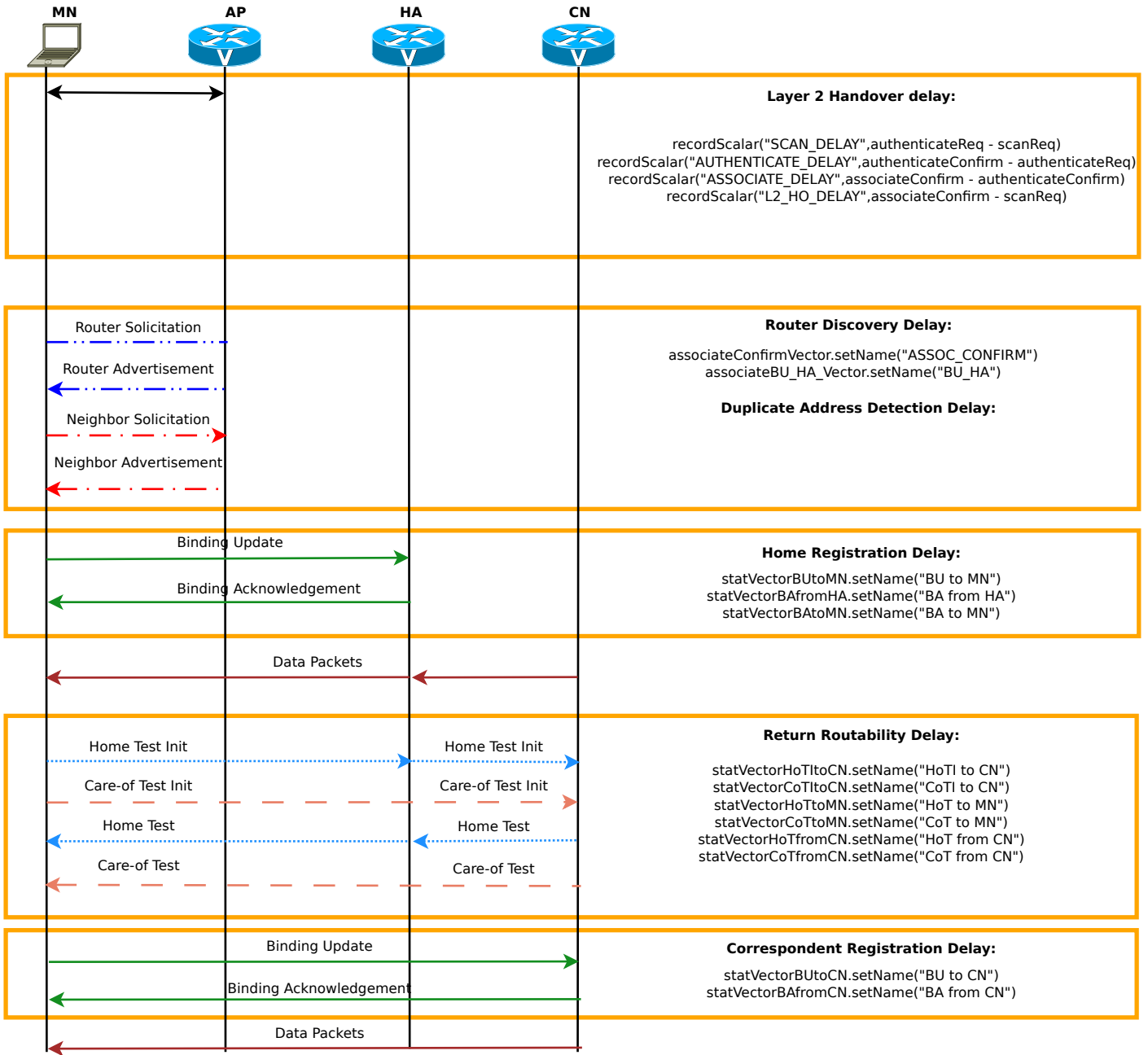


Figure 3.4: Components of Handover Latency - OMNeT++ Implementation

3.3 MIPv6 Network

This section describes the methodology and hardware setup for the MIPv6 implementation. The MIPv6 source code used for the testbed is UMIP 0.4 from UMIP.org [60]. The main components of the network can be seen in Figure 3.5 and are the Correspondent Node, Home Agent and Mobile Node. Performances

of IP mobility protocols may vary widely depending on the mobility of the MNs and traffic related characteristics. It is thus necessary to analyse and evaluate the IP mobility protocols under various conditions and to do a in-depth study on these protocols.

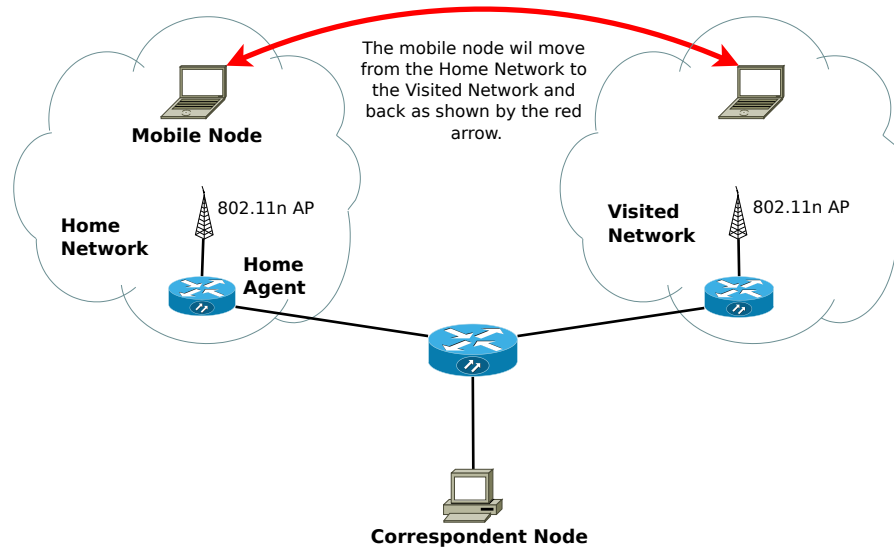


Figure 3.5: MIPv6 Testbed Overview

3.3.1 Methodology: Comparing Protocols

This section is used to describe the evaluation methods for comparing the performances between the various protocols. A single mobile node will be used and the MN moves between different subnets. The MIPv6 network will consist of two APs, the FMIPv6 network of two ARs and the PMIPv6 network of two MAGs. The Operating System of the routers and MN will be different versions of Ubuntu Linux, each running with its own recompiled kernel to accommodate hardware requirements and mobility modules. The movement of the MN will be repeated for each of the networks described in the following sections. The measurements of the testbed will be constructed from the following elements:

- Using real-time applications such as video streaming and File Transfer Protocol (FTP) to generate network traffic in order to determine the TCP/UDP throughput and packet loss of the testbed.
- Using tools such as Wireshark to quantitatively measure the performance of MIPv6, FMIPv6 and PMIPv6.
- Using Ping6 command to determine the latencies between MN and the hosts during the handover process.

- Using Jperf [61] to determine the network performance through tests of both TCP and UDP for comparison between the MN and the CN.

3.3.2 IPv6 Network

The IPv6 Network setup can be seen in Figure 3.6. The IP assignment for each of the network interfaces was configured with the use of RADVD, the router advertisement daemon in Linux. The configurations of the various RADVD daemons for different interfaces can be seen in Appendix B.1.

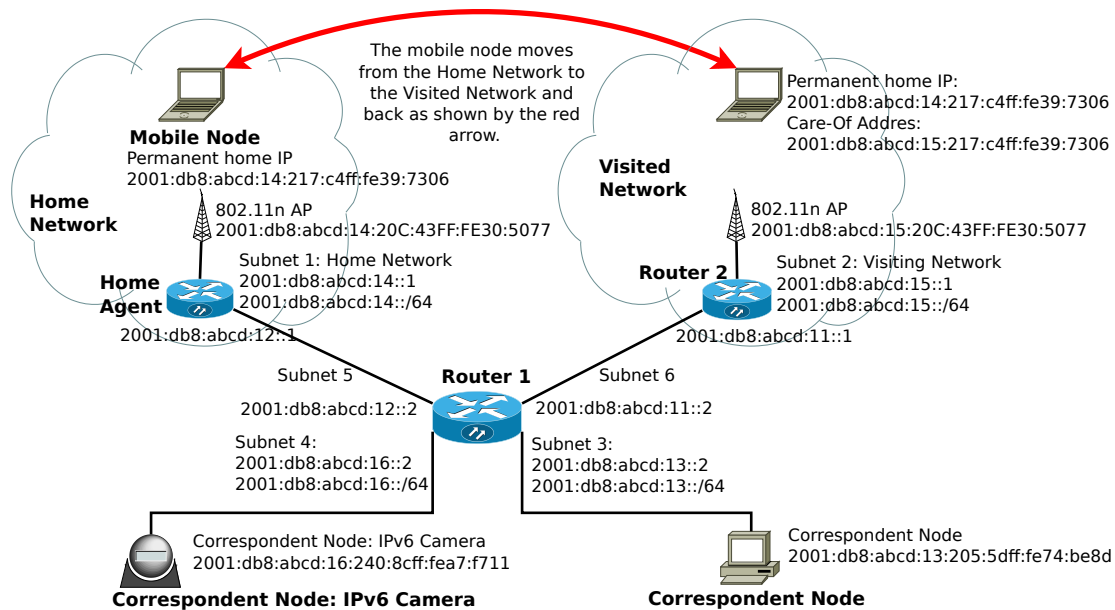


Figure 3.6: MIPv6 Testbed with IP Assignment

The following paragraph gives a detailed explanation of the network setup and it refers to Figure 3.6. The network is made up of six different subnet allocations. The subnet allocations for the network are shown in Table 3.2. Router 1 was used to connect all the network elements. The Router was a standard Linux PC with multiple network cards which had been set up as a router. Router 1 made use of Quagga routing daemon to dynamically assign routes for the network. Router 1 used RADVD to assign IPv6 addresses to Subnets 3 and 4. All the subnets used a prefix assignment of /64, which means the first 64 bits are used for subnet prefix assignment and the last 64 bits of the

address are derived from the MAC address of the hardware. Subnets 3 and 4 were used by the two Correspondent Nodes that were communicating with the Mobile Node. Router 1 did not need to run the MIPv6 software because the router is not involved with MIPv6 mobility messages. The Linux kernels for the following components were recompiled to include some mobility features and important kernel modules that are used by the MIPv6 software.

The Home Agent was a standard Linux OS PC with multiple network cards. The HA was used for mobility management in the home network and ran the MIPv6 software to capture Binding update messages related to the movement of the Mobile Node. The HA consists of two subnets, 1 and 5. Subnet 1(2001:db8:abcd:14::/64) was used as the Home Network subnet and allowed connection to the Mobile Node and to Access Points devices. Subnet 5(2001:db8:abcd:12::/64) was used as a manual setup between the Home Agent router and Router 1, this formed the connection to the Correspondent Node and the Visiting Network. The router advertisement configuration for the HA is shown in Appendix B.1, the Home Agent flag for the router advertisement was turned on because this router acted as the Home Agent for the Mobile nodes. The Home Agent Lifetime for the HA is set to 1800, this value is used to specify the time the HA will be offering MIPv6 Home Agent services. In this case the value was large to avoid interruption during operation. The configuration for the MIPv6 software can be seen in Appendix B.2.2. The configuration file also explains all the values and uses of the configuration parameters for the Home Agent setup. Some of the important parameters are listed and described below:

- **Interface:** Configures which network device to use on the MN
- **DoRouteOptimizationCN/MN:** Enables route optimization between the other MN's or between the MN and the CN. In all the tests route optimization will be enabled.
- **UseMnHaIPsec:** Enables IPsec for communication between the MN and the HA.
- **OptimisticHandoff:** Used by MN to indicate whether route optimization should start before receiving the Back.
- **MnMaxHa/CNBindingLife:** Specifies the maximum time of the binding lifetime.
- **MnHomeLink:** Used to set up the Home addresses of the MN. Multiple home addresses can be used.
- **IPsecPolicySet:** Configuration of the IPsec setup.

The mobile node consisted of only one wireless network card that was used to establish connection to the access points. The mobile node used the recompiled kernel and also the MIPv6 software. No routing daemon and RADVD was used by the mobile node, the configuration file for the MIPv6 software can be seen in Appendix B.2.1. The mobile node moved between two access points, as seen in Figure 3.5. During the movement of the mobile node the correspondent node had an active TCP or UDP connection with the mobile node. The connection was established and measurements were made with the use of Iperf. Iperf is a network protocol analyzer/measuring tool and was used to measure the handover latency, UDP/TCP throughput and UDP jitter. During the movement process the mobile node would hand over to different access points. When the mobile node was in the home network, it received an HoA from the Home Agent using RADVD. The mobile node must be reachable at all times at the HoA address. When the mobile node moved into the range of the visiting network, the mobile node dissociated from the home network's AP and established a connection with the new access point. The mobile node kept the HoA and received a CoA from Router 2 via RADVD. Before Return Routability is set up, all packets destined for the mobile and back are routed via triangular routing, which caused some overhead in the network. The return routability procedure would then use the Care-of-Init and Home-of-Init messages to determine whether the mobile node is the owner of the HoA and the CoA. When the return routability was successfully established, route optimization would then allow direct packet exchange between the mobile node and the correspondent node. A secondary correspondent node consisted of a IPv6 camera to stream live video over the network to the mobile node. This was used to investigate the performance of MIPv6 with real-time data and to see how the handover latency would affect the QoS. The camera acting as the second correspondent node is not able to implement MIPv6, so triangular routing was used to route packets in the network.

Table 3.2: IP Assignment for MIPv6 Network Interfaces

Component	Interfaces	IP Assignment	RA Config	Subnet
HA	Eth0	2001:db8:abcd:12::1	None	1
	Eth1	2001:db8:abcd:14::1	2001:db8:abcd:14::/64	5
AP1	Eth0	2001:db8:abcd:14:20C:43ff:fe30:5077	None	1
	Wlan0	2001:db8:abcd:14:20C:43ff:fe30:5077		
AP2	Eth0	2001:db8:abcd:15:20C:43ff:fe30:5077	None	2
	Wlan0	2001:db8:abcd:15:20C:43ff:fe30:5077		
MN	Wlan0	HoA: 2001:db8:abcd:14:217:c4ff:fe39:7306	None	1
		CoA: 2001:db8:abcd:15:217:c4ff:fe39:7306		2
CN	Eth0	2001:db8:abcd:13:205:5dff:fe74:be8d	None	3
CN: Camera	Eth0	2001:db8:abcd:16:240:8cff:fea7:f711	None	4
Router 1	Eth0	2001:db8:abcd:12::2	None	5
	Eth1	2001:db8:abcd:11::2	None	6
	Eth2	2001:db8:abcd:13::2	2001:db8:abcd:13::/64	3
	Eth4	2001:db8:abcd:16::2	2001:db8:abcd:16::/64	4
Router 2	Eth0	2001:db8:abcd:11::1	None	6
	Eth1	2001:db8:abcd:15::1	2001:db8:abcd:15::/64	2

Router 2, which forms the visiting network, consists of two subnets. Subnet 2 (2001:db8:abcd:15::/64) was used as the visiting network. RA was used to assign IPv6 addresses to Access Points and the CoA of the Mobile Node when moving from the Home Network. This router was also not involved in MIPv6 messaging and thus did not require the MIPv6 software and updated Linux kernel. Router 2 also had a second Subnet 6 (2001:db8:abcd:11::/64), a manual setup between Router 2 and Router 1. Router 2 also used Quagga to assign dynamic routes on the router.

The Mobile Node was moved around to switch between Subnet 1 and 2. The Mobile Node was running the MIPv6 software and updated kernel ready for IPv6 mobility. Software configuration can be seen in Appendix B.2.1. The mobile node receives an HoA when in the home network and a CoA when in the visiting network. During handover the connectivity of the mobile node was tested to make sure the mobile node stayed reachable through the HoA. The mobile node received live video stream data from the correspondent IPv6 camera.

Table 3.3: MIPv6 Test Bed Setup

Component	Network Configuration	Software setup
Home Agent	Consists of multiple NICs and IPv6 forwarding is enabled to act as router.	Linux MIPv6 2.6.38 compiled kernel, Radvd 1.7, Quagga routing daemon and MIPv6 UMIP 0.4 home agent setup.
Access Point	IPv6 802.11n access point	OpenWrt Firmware and Linux 3.2.5 compiled kernel with IPv6 enabled.
Router 2	Consists of multiple NICs and IPv6 forwarding is enabled to act as router.	Linux 2.6.38-generic kernel, Radvd 1.7 and Quagga routing daemon.
Mobile Node	IPv6 802.11n access point	Linux MIPv6 2.6.38 compiled kernel and MIPv6 UMIP 0.4 mobile node setup.
Correspondent Node	Standard PC with ethernet connection	Linux MIPv6 2.6.38 compiled kernel and MIPv6 UMIP 0.4 CN setup.
CN IPv6 Camera	IPv6 ready IP Camera	Axis IPv6 ready camera.
Router 1	Consists of multiple NICs and IPv6 forwarding is enabled to act as router.	Linux 2.6.38-generic kernel, Radvd 1.7 and Quagga routing daemon.

Iperf was used to generate traffic on the network between the mobile node and the correspondent node. The Iperf application was set up to make use of the new IPv6 ready features. The correspondent node was set up as the host server and the mobile node as the client. The size of the data packages has significant impact on the required results. To determine the UDP jitter caused by the various protocols, the build feature of Iperf was used to measure the jitter. To measure the performance of the network with live video stream feed and to determine the impact of the handover latency on video streaming or VoIP, the mobile node connected to a IPv6 ready streaming camera.

3.4 FMIPv6 Network

In this section the Fast Mobile IPv6 network setup and protocol implementation is discussed. The FMIPv6 network setup is similar to the MIPv6 described in Section 3.3. For the network to be FMIPv6 capable, the following nodes needs to be FMIPv6 ready:

- Mobile Node
- Previous Access Router (PAR)
- New Access Router (NAR)

The results and measurement of the network were similar to those of the MIPv6 network in order to compare the protocols and to measure the improvement introduced by FMIPv6. The full network setup can be seen in Figure 3.7. The network consisted of the same hardware as used by MIPv6, but some extra nodes were introduced into the network. The measurements of the network included UDP and TCP throughput, UDP jitter, and most important: The handover latency caused by FMIPv6. The protocol implementation was done by the FMIPv6.org [62]. The implementation was built on top of the MIPv6 implementation and also required the MIPv6 software to handle the basic mobility messages. The FMIPv6 was responsible for handover between the Access Routers, assignment of PCoA, NCoA and to set up a tunnel between the PAR and the NAR to minimize packet loss during the handover process.

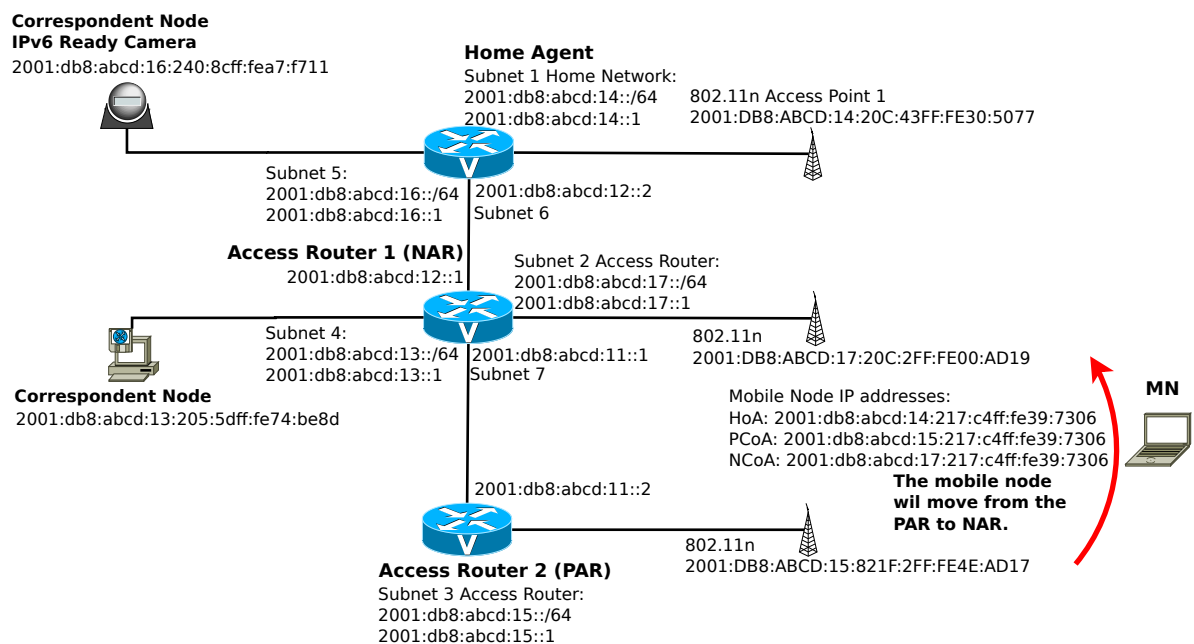


Figure 3.7: FMIPv6 Testbed with IP Assignment

The network consisted of various nodes with different software requirements for each of the nodes. Table 3.4 shows all the various nodes and the software setup for each node. The Home Agent for FMIPv6 was set up similarly to the Home Agent in Section 3.3. This is possible because the Home Agent was not involved in FMIPv6 protocol operation. The Home Agent implemented MIPv6 to allow standard mobility in the network.

Router 1 and Router 2 acted as the Previous Access Router and the New Access Router when looking at FMIPv6 node definitions. The PAR/NAR implemented FMIPv6 and the FMIPv6 runtime configuration for PAR can be

seen in Appendix B.3.2 and that for the NAR in Appendix B.3.3. The mobile node changed its point of attachment from the PAR to the NAR. During this handover process the PAR and the NAR constructed an IPv6 tunnel to forward packets destined for the mobile node to the NAR. The data was buffered at the NAR, as soon as the NAR received the UNA message from the mobile node. This process was used to minimize packet loss in FMIPv6 and improved the QoS of the network. The PAR was also used to assign a New Care-of Address to the mobile node. The NCoA needs to be verified by the NAR. This process of verification makes use of the HI, HAcK and FBack messages to confirm the use of the NCoA by the mobile node. This process improves the handover latency, because DAD and IP assignment latency are avoided.

Table 3.4: FMIPv6 Test Bed Setup

Component	Network Configuration	Software setup
Home Agent	Consists of multiple NICs and IPv6 forwarding is enabled to act as router.	Linux MIPv6 2.6.38 compiled kernel, Radvd 1.7, Quagga routing daemon and MIPv6 UMIP home agent setup.
Router 1	Consists of multiple NICs and IPv6 forwarding is enabled to act as router.	Linux 2.6.38-generic kernel, Radvd 1.7 and Quagga routing daemon. FMIPv6 NAR node setup.
Router 2	Consists of multiple NICs and IPv6 forwarding is enabled to act as router.	Linux 2.6.38-generic kernel, Radvd 1.7 and Quagga routing daemon. FMIPv6 PAR node setup.
Correspondent Node	Standard PC with ethernet connection	Linux MIPv6 2.6.38 compiled kernel and MIPv6 UMIP 0.4 CN setup.
CN IPv6 Camera	IPv6 ready IP Camera	Axis IPv6 ready camera.
Access Point 1	IPv6 802.11n access point	OpenWrt Firmware and Linux 3.2.5 compiled kernel with IPv6 enabled.
Mobile Node	2 x IPv6 802.11n network adapter	Linux MIPv6 2.6.38 compiled kernel and MIPv6 UMIP 0.4 mobile node setup. FMIPv6 software setup as mobile node.

The MN was configured with FMIPv6. The MN consists of two wireless network adapters (wlan0 and wlan1). The FMIPv6 configuration parameters can be seen in Appendix B.3.1. The MN used the second wireless card to scan for available networks in the surrounding area. The MN can be configured to base its handover decision on the signal strength level of the current connection or the handover can be initiated by a hard handover process. The MN was configured for only hard handover, this means that the handover was initiated from the MN operator only and not from the signal level. The secondary network scans the area before the previous connection is dropped, this is used

to obtain information from the available access routers. The MN recorded the information on the NAR during scan and also knew the subnet prefix of the NAR network subnet. This information was used to suggest an NCoA during the handover process. The home agent is set up the same way as described in the previous section for MIPv6. The home agent does not need any additional configuration other than standard MIPv6 to accommodate FMIPv6.

Table 3.5: IP Assignment for FMIPv6 Network Interfaces

Component	Interfaces	IP Assignment	RA Config	Subnet
HA	Eth0	2001:db8:abcd:12::2	None	6
	Eth1	2001:db8:abcd:14::1	2001:db8:abcd:14::/64	1
	eth2	2001:db8:abcd:16::1	2001:db8:abcd:16::/64	5
AP1	Eth0	2001:db8:abcd:14:20C:43ff:fe30:5077	None	1
	Wlan0	2001:db8:abcd:14:20C:43ff:fe30:5077		
AP2 PAR	Eth0	2001:db8:abcd:15:20C:43ff:fe30:5077	None	3
	Wlan0	2001:db8:abcd:15:20C:43ff:fe30:5077		
AP3 NAR	Eth0	2001:db8:abcd:17:20c:2ff:fe00:ad19	None	2
	Wlan0	2001:db8:abcd:17:20c:2ff:fe00:ad19		
MN	Wlan0	HoA: 2001:db8:abcd:14:217:c4ff:fe39:7306	None	1
		PCoA: 2001:db8:abcd:15:217:c4ff:fe39:7306		3
		NCoA: 2001:db8:abcd:17:217:c4ff:fe39:7306		2
CN	Eth0	2001:db8:abcd:13:205:5dff:fe74:be8d	None	4
CN: Camera	Eth0	2001:db8:abcd:16:240:8cff:fea7:f711	None	5
Router 1	Eth0	2001:db8:abcd:12::1	None	6
	Eth1	2001:db8:abcd:11::1	None	7
	Eth2	2001:db8:abcd:13::1	2001:db8:abcd:13::/64	4
	Eth4	2001:db8:abcd:17::1	2001:db8:abcd:17::/64	2
Router 2	Eth0	2001:db8:abcd:11::2	None	7
	Eth1	2001:db8:abcd:15::1	2001:db8:abcd:15::/64	3

3.5 PMIPv6 Network

The following section describes the implementation of Proxy Mobile IPv6, as described in Section 2.8.1. In PMIPv6 the mobile node requires no mobility implementation, which is the main advantage of PMIPv6. The protocol is implemented on the same testbed as MIPv6 and FMIPv6. The implementation runs on top of the standard MIPv6 protocol, with minor patches to the UMIP 0.4 MIPv6 implementation. The network design architecture for the PMIPv6 testbed is shown in Figure 3.8

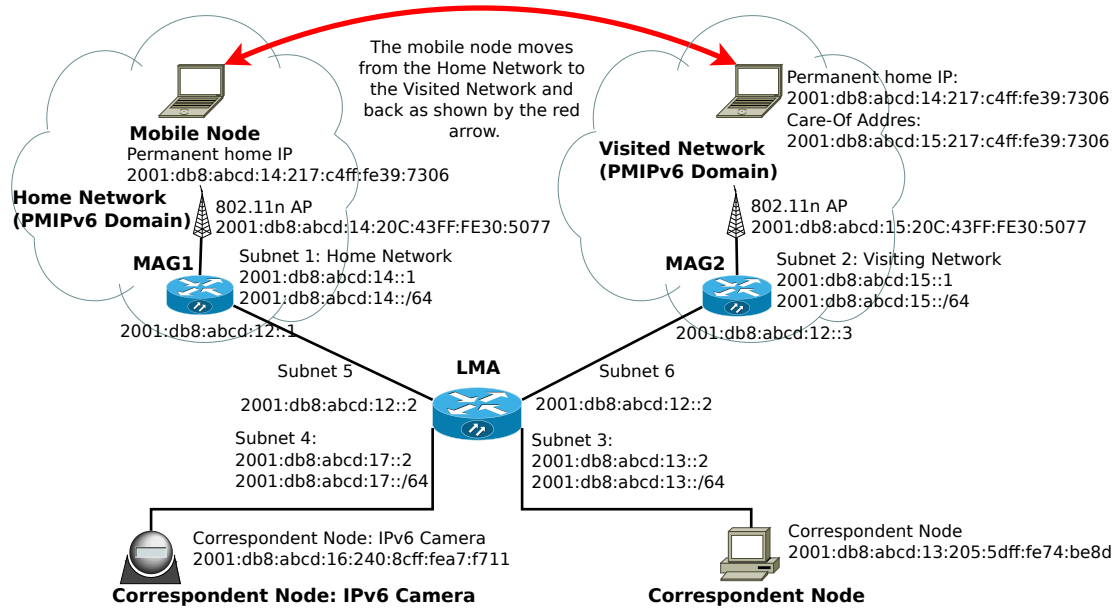


Figure 3.8: PMIPv6 Testbed with IP Assignment

Figure 3.9 illustrates the software architecture of the Open Air Interface Proxy Mobile IPv6 (OAI PMIPv6) [4]. OAI PMIPv6 is built on top of the UMIP modules for Linux. The UMIP implementation makes use of different threads for the handling of the various mobility messages, one thread is used for ICMPv6 messages, one thread is used for the Mobility header messages and the last thread is used for handling tasks and timing events. A handler is used to handle all the mobility messages and these are then passed to the finite state machine. The state machine makes appropriate decisions and controls all other elements to provide a correct predefined protocol behaviour depending on the input received from the handler. The PMIPv6 binding cache is used to store all the information about an MN and is kept up-to-date as the MN moves around in the network.

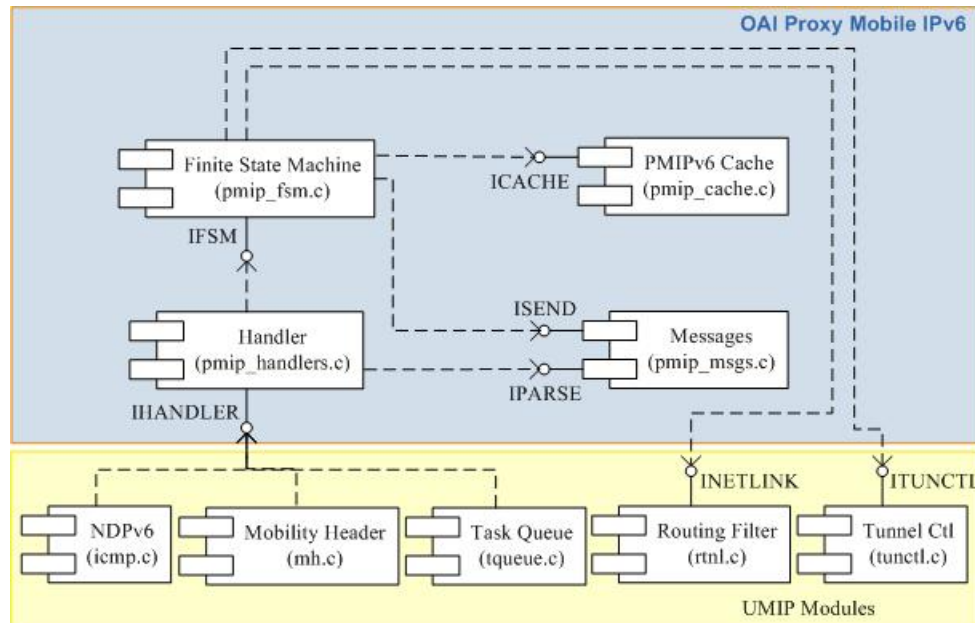


Figure 3.9: PMIPv6 Software Architecture [4]

The network components and IP assignments for the network are listed in Table 3.6. The network consisted of two correspondent nodes, the one being a standard Linux PC and the other being an IPv6 ready camera that was used to stream live video feed over the network. The two CNs are located in subnets 3 and 4 respectively. The CNs both connected to the LMA, as seen in Figure 3.8. The various router advertisement configurations for each subnet can also be seen in Table 3.6, RADVD (The Router Advertisement Daemon) was used. An example of a configuration file can be seen in Appendix B.1. For PMIPv6 the Home Agent configuration items are not used.

The LMA, which is like the Home Agent in MIPv6, is mainly responsible for the MN's home network prefixes and manages the MN's binding state. The LMA comprises of four subnets, each on its own network card. Subnets 3 and 4 were used for the two CNs. A Freeradius server was also installed on the LMA; this was used to provide AAA management for MNs. Subnet 5 was used to connect the LMA with MAG1, and subnet 6 was used for the connection to MAG2. For the MAGs, Freeradius clients were installed to provide AAA management for the MAGs. The Freeradius server configuration can be found in Appendix B.4.4 and client configuration in Appendix B.4.5. During protocol operation the MN connected to two APs, one located in the home network in subnet 1 and the other AP in the visited network in subnet 2. The MN continuously moved between the two APs. The various configuration parameters for PMIPv6 can be seen in Appendix B.4.

Table 3.6: IP Assignment for PMIPv6 Network Interfaces

Component	Interfaces	IP Assignment	RA Config	Subnet
CN	Eth0	2001:db8:abcd:13:205:5dff:fe74:be8d	None	3
CN: Camera	Eth0	2001:db8:abcd:16:240:8cff:fea7:f711	None	4
LMA	Eth0	2001:db8:abcd:13::2	2001:db8:abcd:13::/64	3
	Eth1	2001:db8:abcd:16::2	2001:db8:abcd:16::/64	4
	Eth2	2001:db8:abcd:11::2	none	6
	Eth3	2001:db8:abcd:12::2	none	5
MAG1	Eth0	2001:db8:abcd:12::1	None	5
	Eth1	2001:db8:abcd:14::1	2001:db8:abcd:14::/64	1
MAG2	Eth0	2001:db8:abcd:11::1	None	6
	Eth1	2001:db8:abcd:15::1	2001:db8:abcd:15::/64	2
AP1	Eth0	2001:db8:abcd:14:20C:43ff:fe30:5077	None	1
	Wlan0	2001:db8:abcd:14:20C:43ff:fe30:5077		
AP2	Eth0	2001:db8:abcd:15:20C:43ff:fe30:5077	None	2
	Wlan0	2001:db8:abcd:15:20C:43ff:fe30:5077		
MN	Wlan0	HoA: 2001:db8:abcd:14:217:c4ff:fe39:7306	None	1
		CoA: 2001:db8:abcd:15:217:c4ff:fe39:7306		2

Table 3.7: PMIPv6 Test Bed Setup

Component	Network Configuration	Software setup
Local Mobility Anchor	Consists of multiple NICs and IPv6 forwarding is enabled to act as router.	Linux 11 MIPv6 2.6.38 compiled kernel, Radvd 1.7, Quagga routing daemon and OAI PMIPv6 0.4.1 LMA setup.
Access Point	IPv6 802.11n access point	OpenWrt Firmware and Linux 3.2.5 compiled kernel with IPv6 enabled.
Mobile Access Gateways	Consists of multiple NICs and IPv6 forwarding is enabled to act as router.	Linux 11 MIPv6 2.6.38 compiled kernel, Radvd 1.7, Quagga routing daemon and OAI PMIPv6 0.4.1 MAG setup.
Mobile Node	IPv6 802.11n access point	Standard Linux 12.10
Correspondent Node	Standard PC with Ethernet connection	Standard Linux 12.10.
CN IPv6 Camera	IPv6 ready IP Camera	Axis IPv6 ready camera.

3.6 Simulation Model MIPv6 for 6LoWPAN

In [5] MIPv6 was implemented in Contiki. Contiki is an open source OS and is designed to be used in low power sensor networks and embedded systems. Contiki uses the uIPv6 network stack and is certified for IPv6 Ready Phase 1 [63]. In Section 2.3.1, two compression mechanisms were described, of which the latter (LOWPAN_IPHC) is used in the uIPv6 stack for IPv6 header compression. The simulation model 6LoWPAN architecture is shown in Figure

3.10. The network consisted of two visiting networks (each containing 1 border router), a home agent, 1 mobile sensor (moving between the two visiting networks) and 1 correspondent node. The nodes used for the mobile sensor and the border router were TelosB [64] developed by Crossbow. For movement detection, MIPv6 relies on router advertisement and from [8] it is suggested that the time between two consecutive router advertisements should be varied between 30ms to 70ms. For WSN, this rate will increase the power use of a node, as well as the contention on the wireless medium. WSN should thus rely on other mechanisms to detect node movement. By using the movement detection adaptation proposed in [5], MIPv6 was simulated in Contiki. The various runtime parameters used for the simulation can be seen in Appendix C.1.

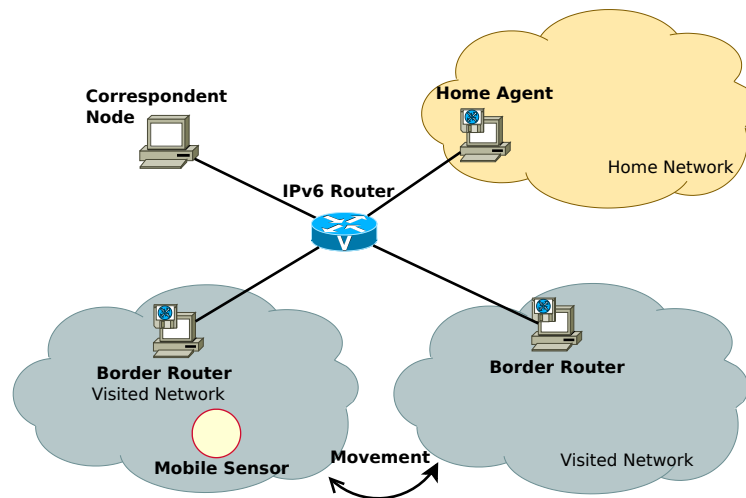


Figure 3.10: MIPv6 6LoWPAN Testbed [5]

3.7 Conclusion

In this chapter the testbed setup and network designs for the various protocols have been illustrated and explained in detail. The three protocols consisted of basically the same network design with some small changes regarding network layout and configuration. All the configurations for the various test setups can be seen in Appendix B. Network design for MIPv6 can be seen in Figure 3.6 and the UMIP0.4 MIPv6[60] protocol was implemented on top of the IPv6 network setup beforehand. The functioning of this protocol relies heavily on the hardware/software support of the network.

The network design for FMIPv6 can be seen in Figure 3.7 and was implemented on top of the MIPv6 network setup. Once again the implementation of this protocol relies heavily on the hardware/software support of the network and is functional only in certain versions of Linux OS. To simulate a reactive

handover in FMIPv6 the MN was manually switched to the new access point. For predictive handovers, the signal strength of the previous access point was lowered.

The PMIPv6 network design can be seen in Figure 3.8. The network design for this implementation differs widely from the previous two implementations. In PMIPv6, the mobility resides in the network and not on the mobile nodes.

Lastly, the simulation for the wireless sensor MIPv6 network can be seen in Figure 3.10. The simulation consisted of two border routers that acted as the outward point for the two 6LoWPAN networks. The results of the tests done using the simulation module and the testbed with the various test network setups, will be discussed in the next chapter.

Chapter 4

Results

4.1 Introduction

In this chapter various results of the network protocol tests are discussed. The results provide handover latency, as determined by the simulation module. The testbed implementation results consists of UDP/TCP throughput, UDP jitter, and handover latency for MIPv6, FMIPv6 and PMIPv6. Iperf [61] was used to generate traffic between the mobile node and the correspondent node. The measurements were done for all the protocols under the same conditions and network load. From these results a very accurate conclusion can be reached in order to evaluate the performance of each protocol and to identify a future protocol for mobility.

4.2 Evaluation Methodology

The evaluation of the testbed for MIPv6, FMIPv6 and PMIPv6 consisted of the following performance metrics:

- Handover Latency (Using Iperf)
 - Measure the handover latency produced by MIPv6 when moving from home network to visiting network.
 - Measure the handover latency produced by FMIPv6 for the predictive case when moving from home network to visiting network.
 - Measure the handover latency produced by PMIPv6 when moving from home network to visiting network.
 - Compare the average handover latency produced by MIPv6, FMIPv6 and PMIPv6 during handover.
- TCP Throughput (Using Iperf)
 - Comparison of TCP throughput for the various protocols.

- UDP Throughput, UDP Jitter and Packet loss (Using Iperf)
 - Measure UDP throughput while running IPv6 application
 - Measure Packet loss during handover process.
 - Comparison of UDP throughput for the various protocols.
 - Comparison of UDP jitter for the various protocols.
- Live video streaming evaluation
 - Measure throughput while streaming live video feed.
 - Compare protocol performance under live streaming.

4.3 Simulation Results

This section shows the results of the OMNet++ simulation module for MIPv6. The UDP and TCP throughput results obtained from the simulation model for MIPv6 are compared to the MIPv6 testbed in Section 4.7.

4.3.1 Handover Latency

Figure 4.1 shows total handover latency caused by MIPv6 messaging during 5 simulation runs. The layer 2 handover delay is around 1.5s for each of the simulation runs. The delay caused by binding messages is around 2s and the other main contribution to handover delay is caused by duplicate address detection. The delay caused by DAD is between 1.5 - 2s.

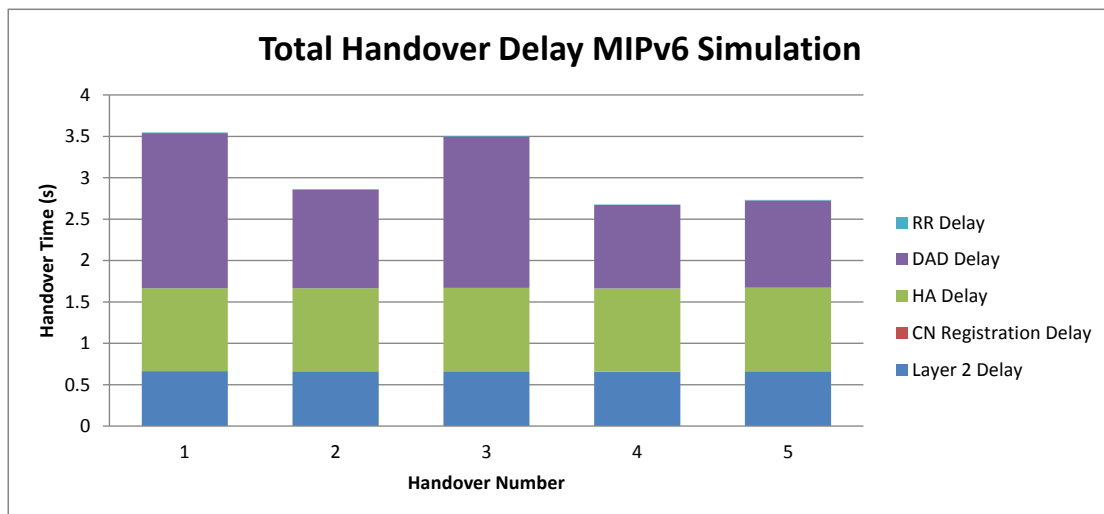


Figure 4.1: Handover Latency in MIPv6 OMNeT++ Simulation

Figure 4.2 shows total handover latency caused by MIPv6 return routability. The latency introduced by this messaging are very small and do not have any real impact on the overall latency.

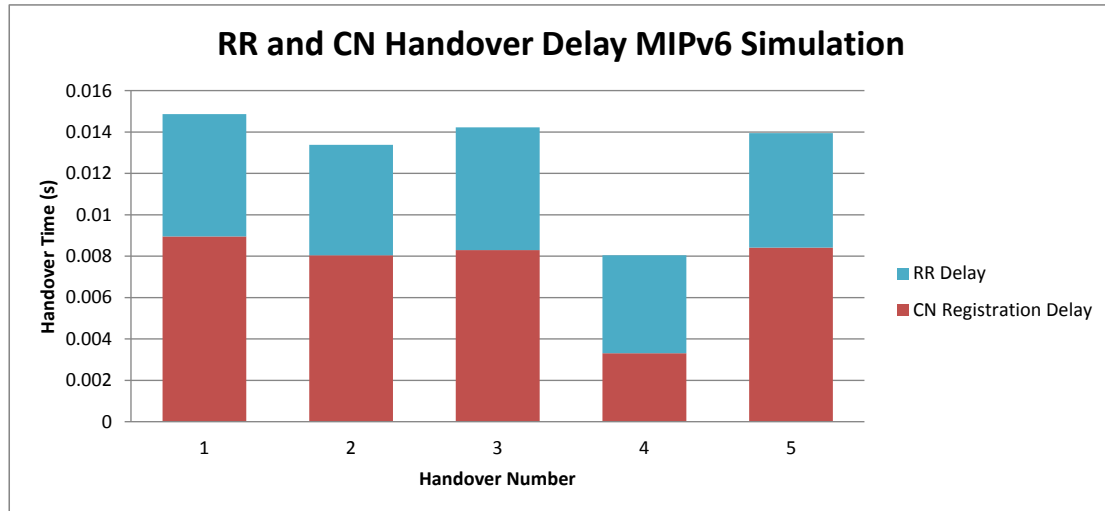


Figure 4.2: RR and CN Handover Delay MIPv6 Simulation

4.3.2 UDP Throughput

Figure 4.3 shows the UDP throughput for MIPv6 when the MN performs a handover to a new access router in the visited network.

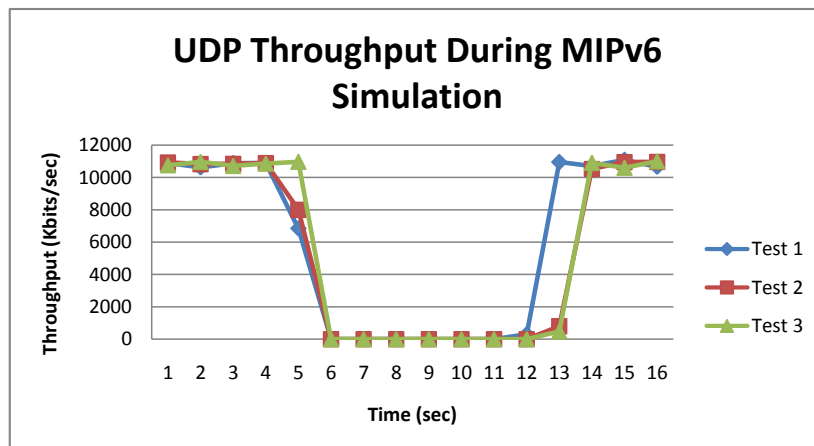


Figure 4.3: UDP Throughput in MIPv6 OMNeT++ Simulation

4.3.3 TCP Throughput

Figure 4.4 shows the TCP throughput for MIPv6 when the MN performs a handover to a new access router in the visited network.

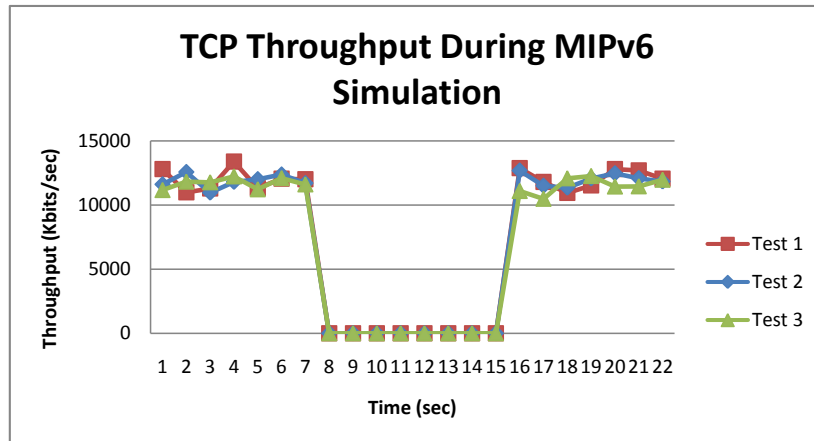


Figure 4.4: TCP Throughput in MIPv6 OMNeT++ Simulation

4.4 MIPv6 Protocol Results

The following section provides the various testbed results for MIPv6. The results obtained from the testbed are used to evaluate the performance of MIPv6 by comparing the UDP and TCP measurements to the results of the other protocols presented later in this chapter. The evaluation methodology for testing can be seen in Section 4.2.

4.4.1 UDP Throughput

In the first experiment the UDP throughput, UDP jitter, TCP throughput and packet loss were measured for MIPv6. With the use of JPerf, the MN was set up as the client and the CN as the server. The JPerf setup for UDP was kept at the default values, with the UDP bandwidth set to 10 Mbps, buffer size set to 41 Kbytes and a packet size of 1500 bytes. Figure 4.5 shows the average throughput on the MIPv6 if no handover is performed. The average throughput of the network is 9960 Kbits/sec with a transmission rate of 10 Mbps.

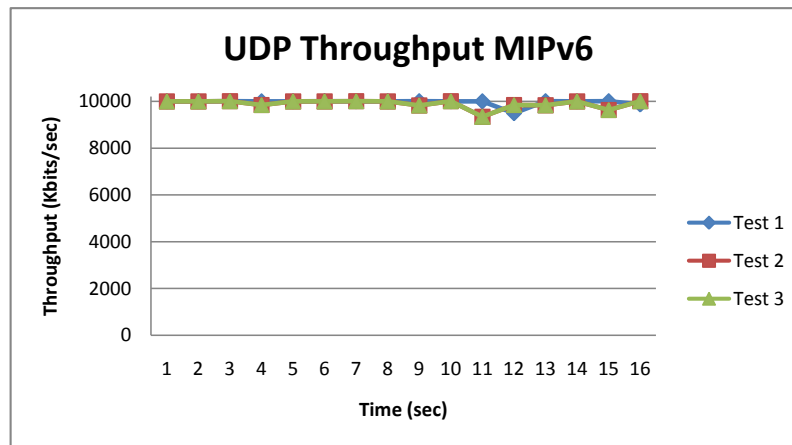


Figure 4.5: UDP Throughput in MIPv6

In Figure 4.6 the UDP throughput of MIPv6 can be seen. The MN moved from one access point to a new access point at 4 seconds and established connection to the new network at around 11 seconds. The test results show the average values of three different tests of the network. The average handover time for the UDP transmission was 7 seconds. The average throughput of the network was 5850 Kbits/sec with a transmission rate of 10 Mbps.

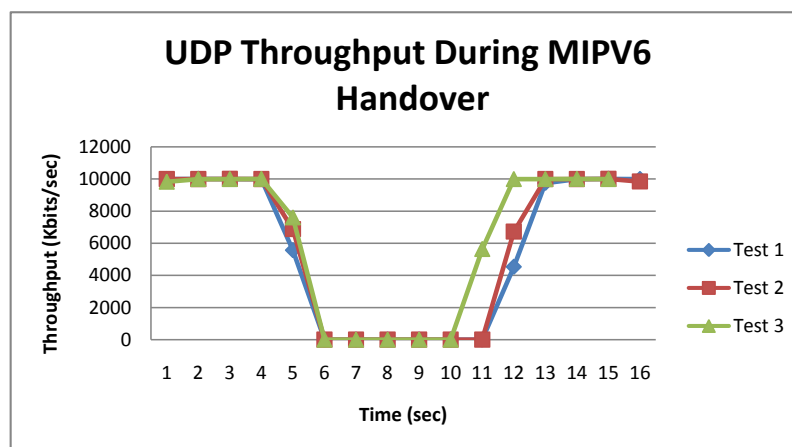


Figure 4.6: UDP Throughput in MIPv6 During Handover

4.4.2 UDP Jitter

"Jitter(UDP) of a packet stream is defined as the mean deviation of the difference in packet spacing at the receiver compared to the sender, for a pair of packets[65]."

Figure 4.7 shows the comparison of MIPv6 UDP jitter when no handover is performed and when a handover is performed by the mobile node. It is clearly visible how the jitter increases when the mobile node goes into the handover process. The average jitter when a handover is performed is in the order of 1.1666 ms, compared with 0.654 ms when no handover is performed on the network. The reason for the increase in jitter after the handover process has completed is that Route Optimization has not yet completed, which causes more overhead on the network. This increases the jitter, as packets have to be routed via the Home Agent, because the bidirectional tunnel setup has not yet been completed. The average jitter also increases when Route Optimization is used, because additional routing headers need to be added to the packet for encapsulation.

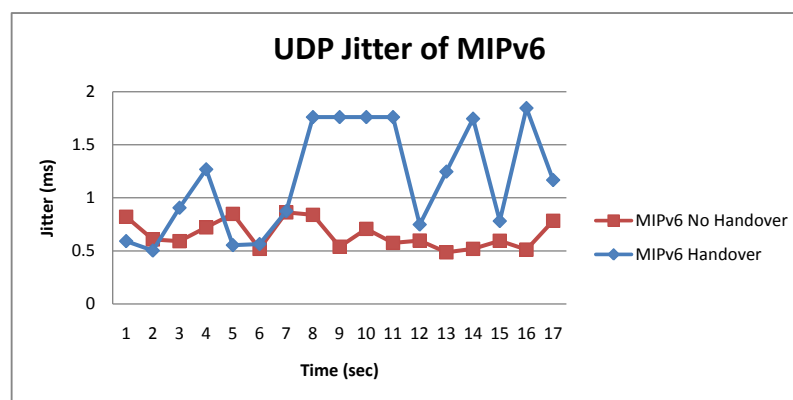


Figure 4.7: UDP Jitter in MIPv6

4.4.3 TCP Throughput

TCP performance of the network was also tested with the use of JPerf. All Jperfs default values for the TCP buffer length (2), TCP Windows Size (56) and TCP Max Segment Size (1) were used in the setup. Figure 4.8 shows the throughput of the network when MIPv6 is used. The handover period is clearly visible from the figure and total handover latency is around 12 s. After handover completion the throughput is lower than before the handover occurred. This is for a number of reasons, one being the new access point and access router. Other reasons include the overhead added by MIPv6.

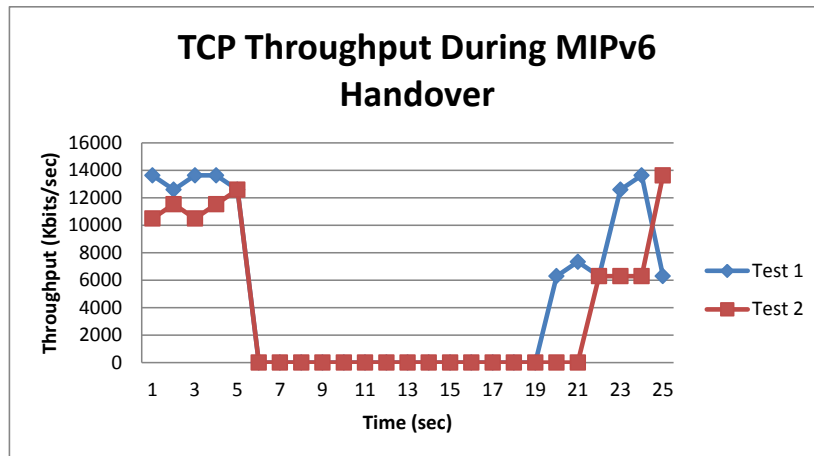


Figure 4.8: TCP Throughput in MIPv6

4.5 FMIPv6 Protocol Results

This section presents the results of the testbed after FMIPv6 had been implemented on the testbed. The results presented are similar to those measured in Section 4.4 and an in depth comparison is made in Section 4.7.

4.5.1 UDP Throughput

The second experiment consisted of FMIPv6 UDP throughput, UDP jitter, TCP throughput and packet loss measurements. The same experiment and measurement setup was used as in Section 4.4.1. The JPerf setup for UDP was kept at the default values, with the UDP bandwidth set to 10 Mbps, buffer size set to 41 Kbytes and a packet size of 1500 bytes. Figure 4.9 shows the average throughput on the FMIPv6 if no handover is performed. The average throughput of the network is 9914 Kbits/sec with a transmission rate of 10 Mbps.

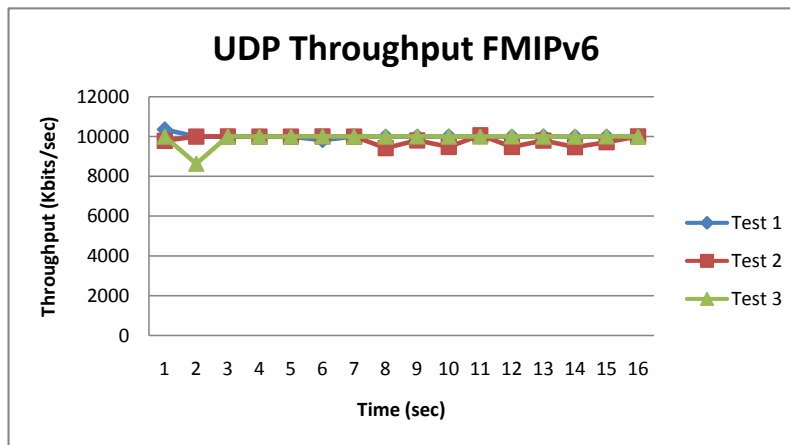


Figure 4.9: UDP Throughput in FMIPv6

In Figure 4.10 the UDP throughput of FMIPv6 can be seen. The MN moved from the Previous Access Router to the New Access Router at 5.5 seconds and established connection to the new network at around 7.5 seconds. The test results show the average values of three different tests of the network. The average handover time for the UDP transmission was 2 seconds. The average throughput of the network was 7436 Kbits/sec with a transmission rate of 10 Mbps.

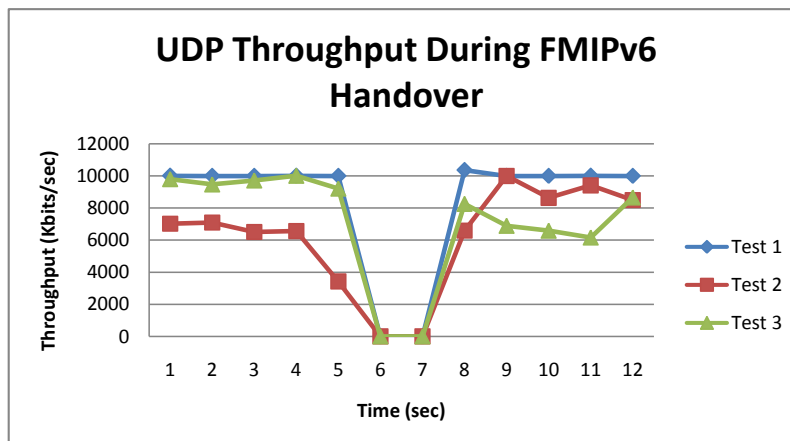


Figure 4.10: UDP Throughput in FMIPv6 During Handover

4.5.2 UDP Jitter

Figure 4.11 shows the comparison of FMIPv6 UDP Jitter when no handover is performed and when a handover is performed by the mobile node. It is clearly

visible how the jitter increases when the mobile node goes into the handover process. The average jitter when a handover is performed is the order of 1.006 ms compared to 0.671 ms when no handover is performed on the network. Handover increases the jitter as packets have to be routed via the Home Agent because the bidirectional tunnel setup has not yet been completed.

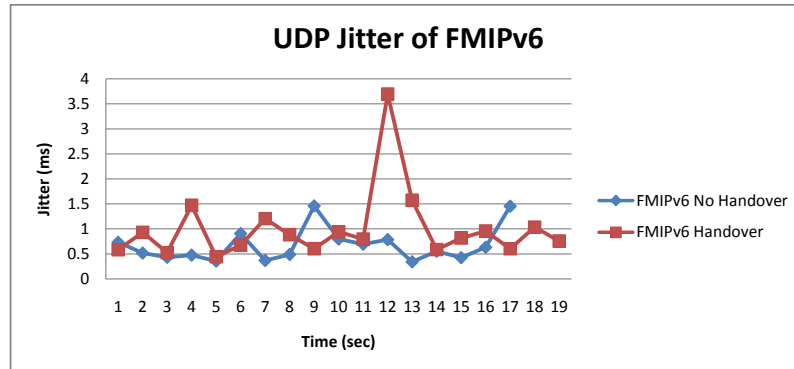


Figure 4.11: UDP Jitter in FMIPv6

4.5.3 TCP Throughput

TCP performance of the network was also tested with the use of JPerf. All JPerf's default values for the TCP buffer length (2), TCP Windows Size (56) and TCP Max Segment Size (1) were used in the setup. Figure 4.12 shows the throughput of the network when FMIPv6 was used. The handover period is clearly visible from the figure and total handover latency is around 8 s. After handover completion the throughput is lower than before the handover occurred. This could be caused by a number of reasons, one being the location of the new access point which could contribute to a weaker signal strength, another that the NAR uses the same channel as the surrounding AP. These overlapping channels can lead to more congestion which can cause lower throughput. Another reason could be the overhead added by FMIPv6. Figure 4.8 shows the throughput of the MIPv6 network, it can be seen that after handover the throughput is also lower. This adds support to the belief that the main cause is the secondary AP used and not MIPv6 or FMIPv6 overhead.

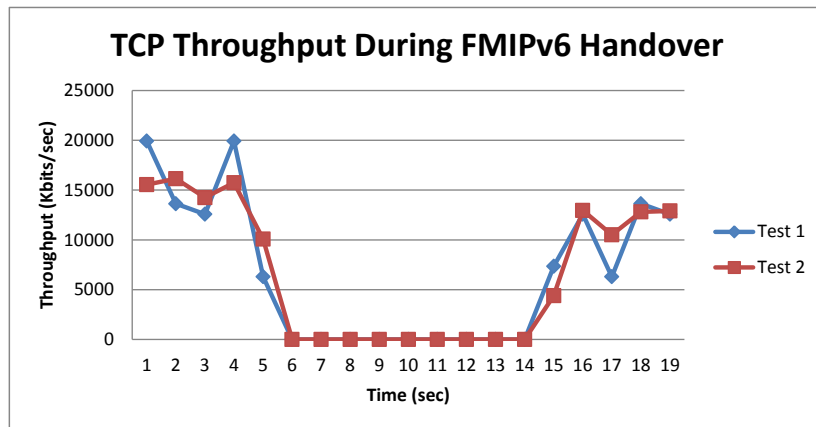


Figure 4.12: TCP Throughput in FMIPv6

4.6 PMIPv6 Protocol Results

The following section provides the various testbed results for PMIPv6. The results obtained from the testbed were used to evaluate the performance of PMIPv6 by comparing the UDP and TCP measurements with those of MIPv6 and FMIPv6 presented earlier in this chapter.

4.6.1 UDP Throughput

The third experiment comprised of PMIPv6 UDP throughput, UDP jitter, TCP throughput and packet loss measurements. The same experiment and measurement setup were used as in Section 4.4.1. The JPerf setup for UDP was kept at the default values, with the UDP bandwidth set to 10 Mbps, buffer size set to 41 Kbytes and a packet size of 1500 bytes. Figure 4.13 shows the average throughput on the PMIPv6 if no handover is performed. The average throughput of the network was 9971.8 Kbits/sec with a transmission rate of 10 Mbps.

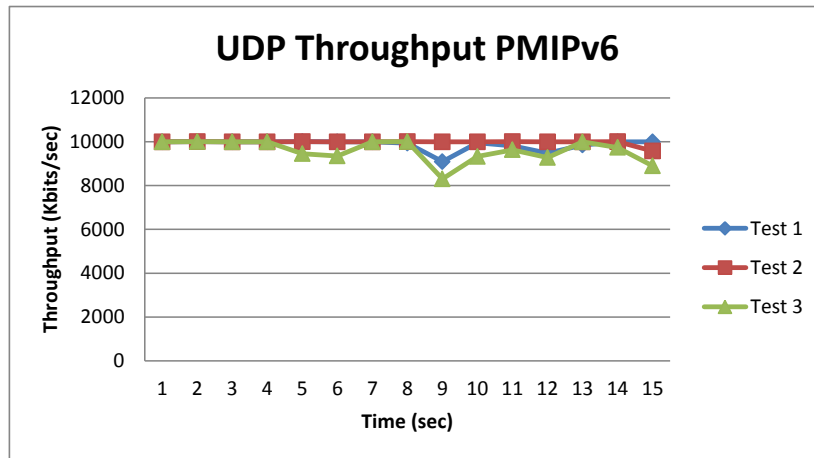


Figure 4.13: UDP Throughput in PMIPv6

In Figure 4.14 the UDP throughput of PMIPv6 when a handover is performed can be seen. The MN moved from LMA1 to LMA2 at 4.5 seconds and established connection to the new network at around 11.5 seconds. The test results showed the average values of three different tests of the network. The average handover time for the UDP transmission was 6.5 seconds. The average throughput of the network was 4983 Kbits/sec with a transmission rate of 10 Mbps.

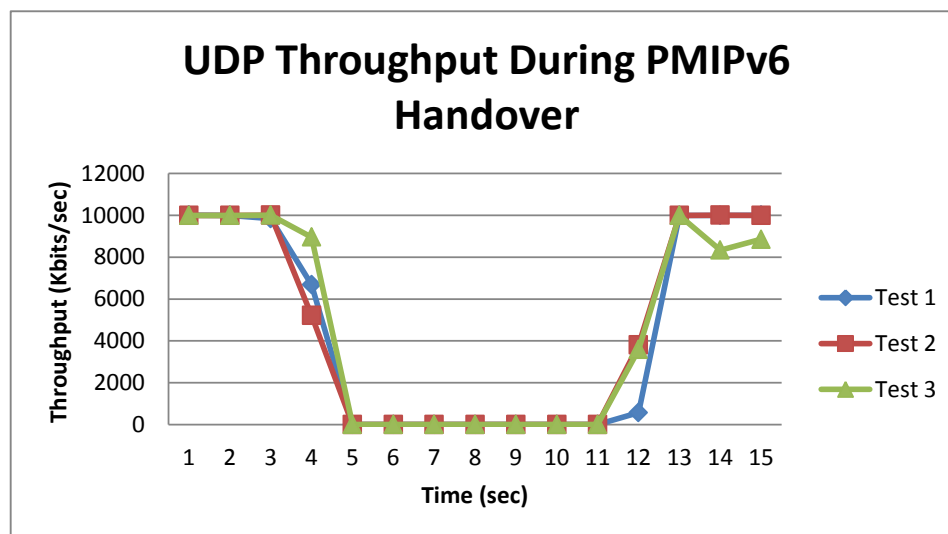


Figure 4.14: UDP Throughput in PMIPv6 During Handover

4.6.2 UDP Jitter

Figure 4.15 shows the comparison of PMIPv6 UDP jitter when no handover is performed and when a handover is performed by the mobile node. Increase in jitter can be seen when the mobile node enters the handover process. The average jitter when a handover is performed equals 0.758 ms compared with 0.413 ms when no handover is performed on the network.

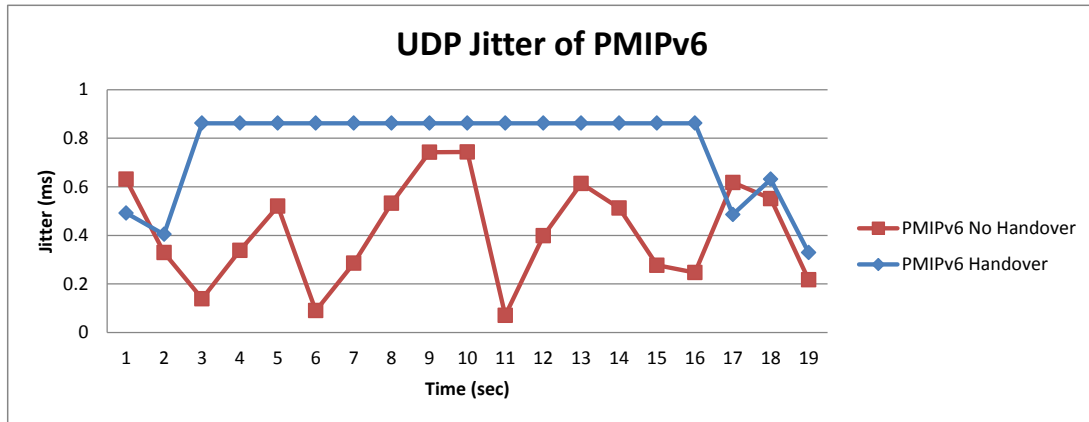


Figure 4.15: UDP Jitter in PMIPv6

4.6.3 TCP Throughput

The setup for TCP measurement in PMIPv6 is similar to that described for FMIPv6. Figure 4.16 shows the throughput of the network when PMIPv6 is used. The handover period is clearly visible from the figure and total handover latency is around 13 s. The higher throughput after handover is quite possibly caused by a better wireless link quality on the new access point and small discrepancies in hardware performance by the access routers.

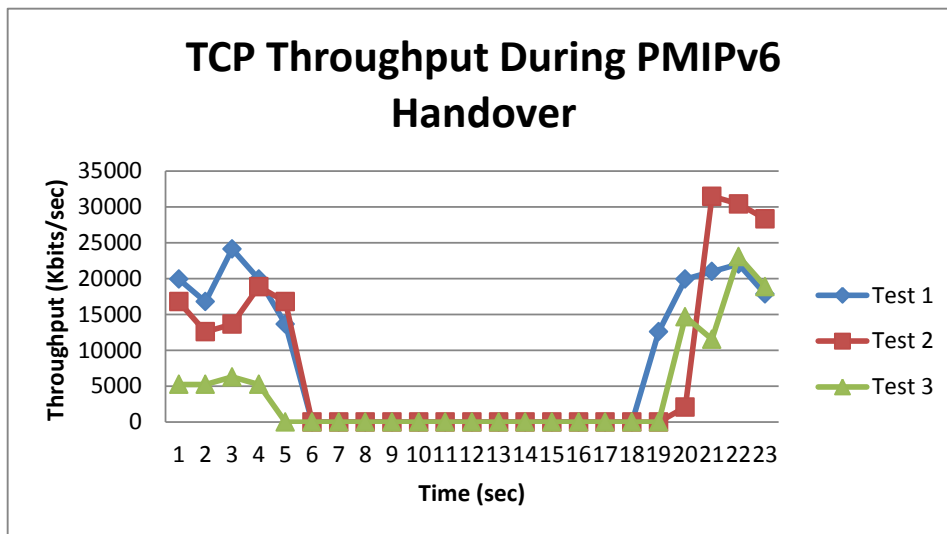


Figure 4.16: TCP Throughput in PMIPv6

Figure 4.17 shows the TCP throughput achieved by PMIPv6 when using an IP camera to generate real-time traffic on the testbed. The results also indicate a handover latency of about 13 s when the MN is not reachable. After the handover had been completed, the throughput was not as consistent as before the handover. The lower throughput is caused by various factors, namely hardware performance, wireless link quality and overhead produced by the PMIPv6 mobility messages. Figure 4.18 shows the average packet throughput for PMIPv6.

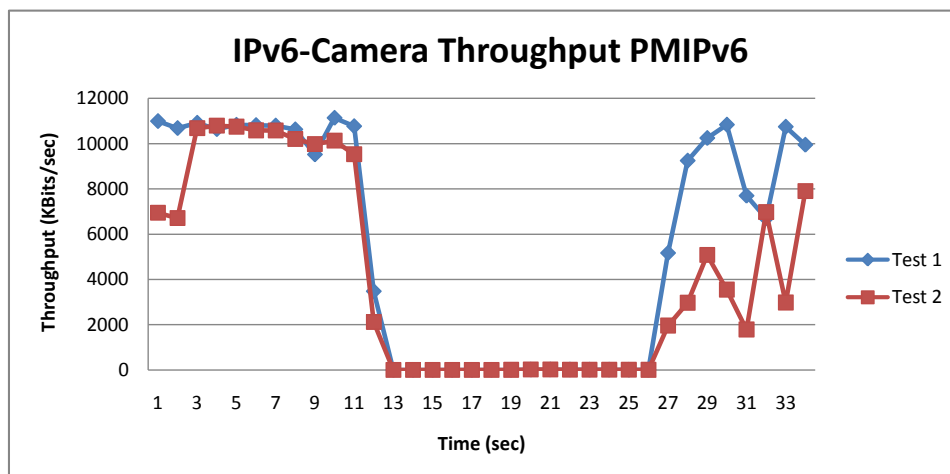


Figure 4.17: IP Camera TCP Throughput in PMIPv6

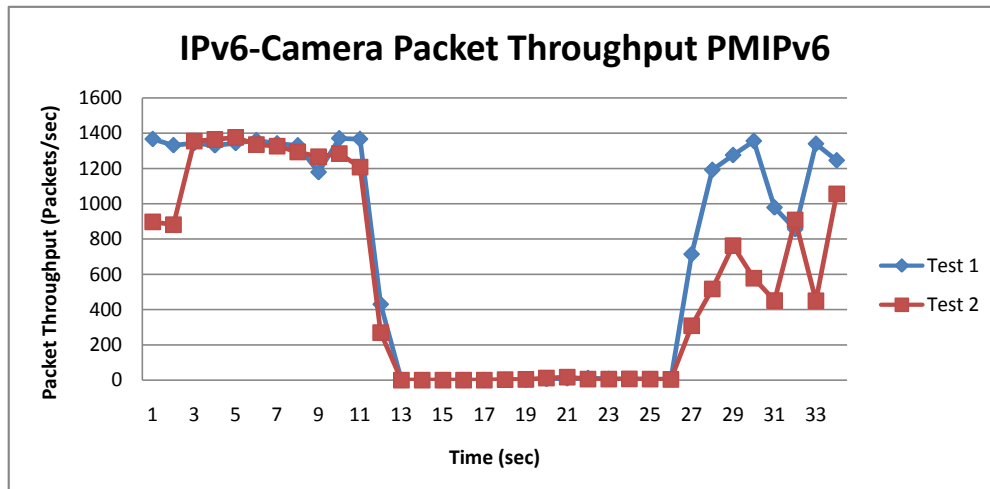


Figure 4.18: IP Camera TCP Packet Throughput in PMIPv6

4.7 Comparing Protocol Results

In this section the various test results shown in the previous sections are compared with each other. The UDP and TCP results from the MIPv6 OMNeT++ simulation model will be compared to the results obtained from the MIPv6 testbed. The section will also look at UDP, TCP, UDP jitter and Packet loss comparisons for MIPv6, FMIPv6 and PMIPv6. The handover latency of the protocols will also be compared.

4.7.1 MIPv6 Comparison - OMNeT++ Simulation and Testbed

UDP throughput comparison between the MIPv6 simulation and the MIPv6 testbed can be seen in Figure 4.19. It is clear that the total handover latency from the simulation is inline with the results obtained from the testbed.

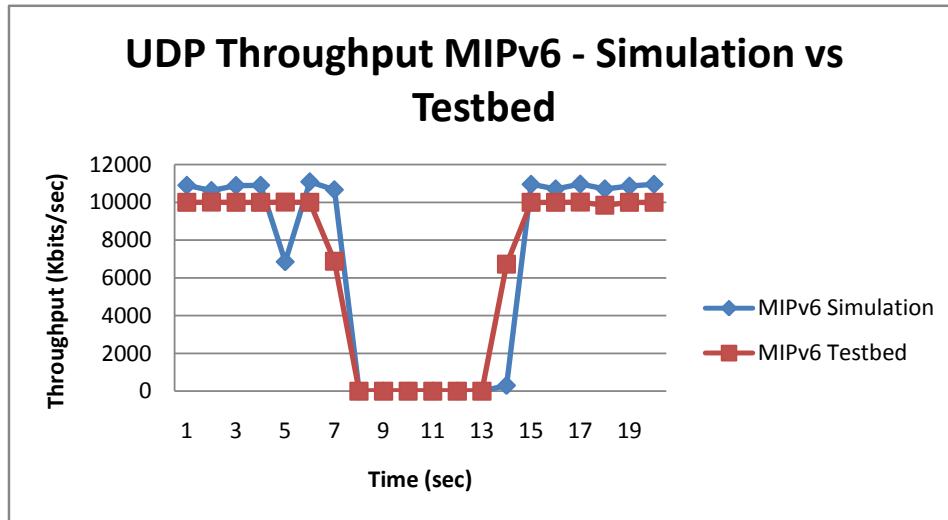


Figure 4.19: MIPv6 UDP Throughput - OMNeT++ Simulation vs Testbed

TCP throughput comparison between the MIPv6 simulation and the MIPv6 testbed can be seen in Figure 4.20. The handover latency varies a bit. The handover latency for the simulation is around 8s, where handover latency for the testbed is around 14s. A possible reason for this variation can be because of differences in the TCP/IP stack of OMNeT++ INET package and the testbed.

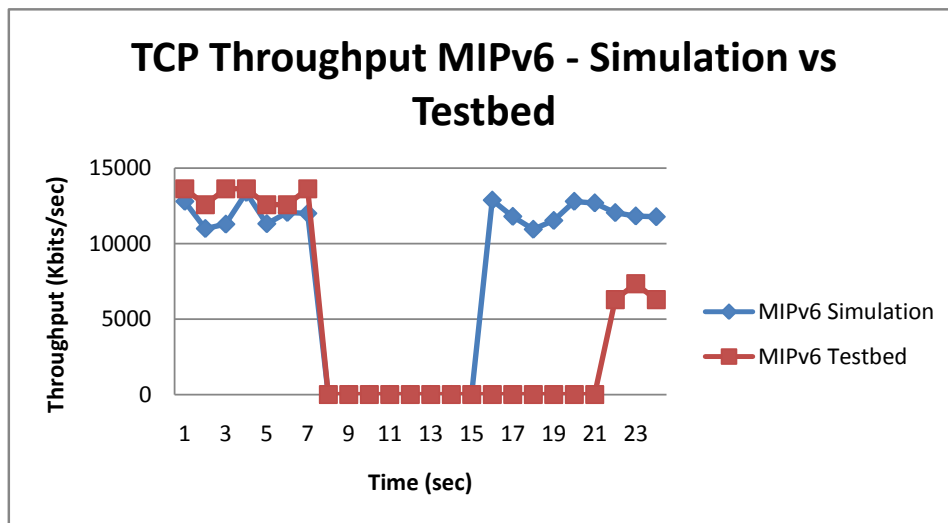


Figure 4.20: MIPv6 TCP Throughput - OMNeT++ Simulation vs Testbed

4.7.2 UDP Throughput Comparison

UDP throughput comparison can be seen in Figure 4.21. It is clearly visible that FMIPv6 outperforms the other protocols in the time it takes to re-establish connection with the corresponding node. The average time taken by FMIPv6 is around 2 seconds, compared to 7 seconds for MIPv6 and 6.5 seconds for PMIPv6. The average throughput for protocols can be seen in Table 4.1 column 6, named 'Throughput HO (Kbits/sec)'. The average throughput for FMIPv6 is better, compared with the other protocols over the same period of time. This is because of the decrease in handover latency for FMIPv6. MIPv6 and PMIPv6 are fairly equal in throughput comparison.

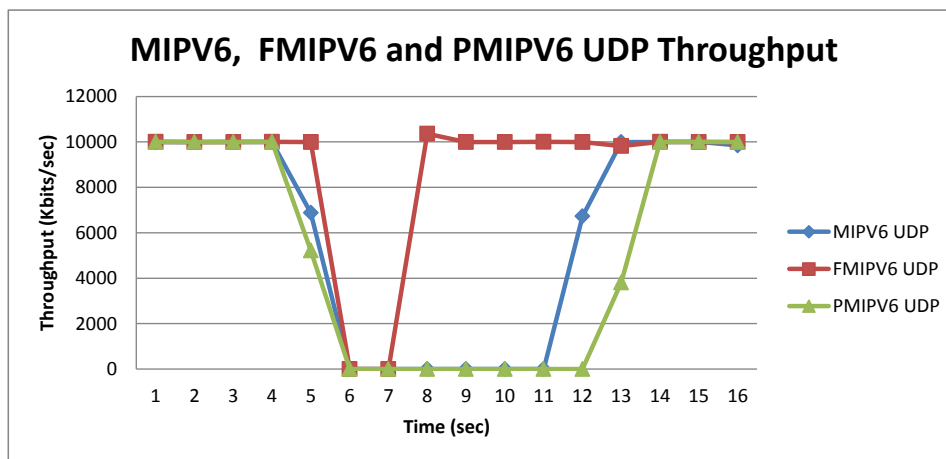


Figure 4.21: UDP Throughput in MIPv6, FMIPv6 and PMIPv6 during Handover

UDP throughput comparison during handover and when no handover is triggered can be seen in Figure 4.22. Figure 4.22 is used to illustrate the average throughput of protocols as seen in Table 4.1 column 5 (Throughput (Kbit-s/sec)) and column 6 (Throughput HO (Kbits/sec)). The average throughput for FMIPv6 is better compared to that of the other protocols over the same period of time when a handover is performed. MIPv6 and PMIPv6 are fairly equal in throughput comparison during handover and the average throughput when no handover is performed is very similar for all three protocols.

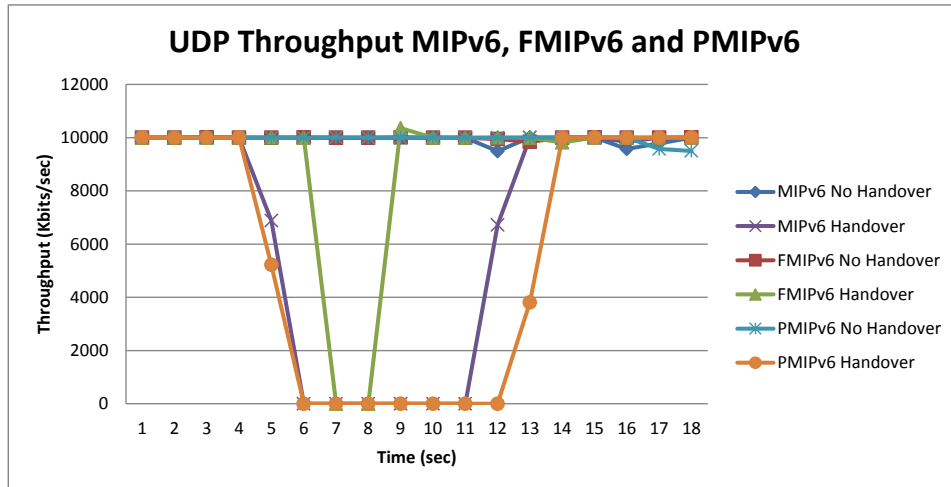


Figure 4.22: UDP Throughput MIPv6, FMIPv6 and PMIPv6

4.7.3 TCP Comparison Throughput

The average TCP throughput can be seen in Figure 4.23.

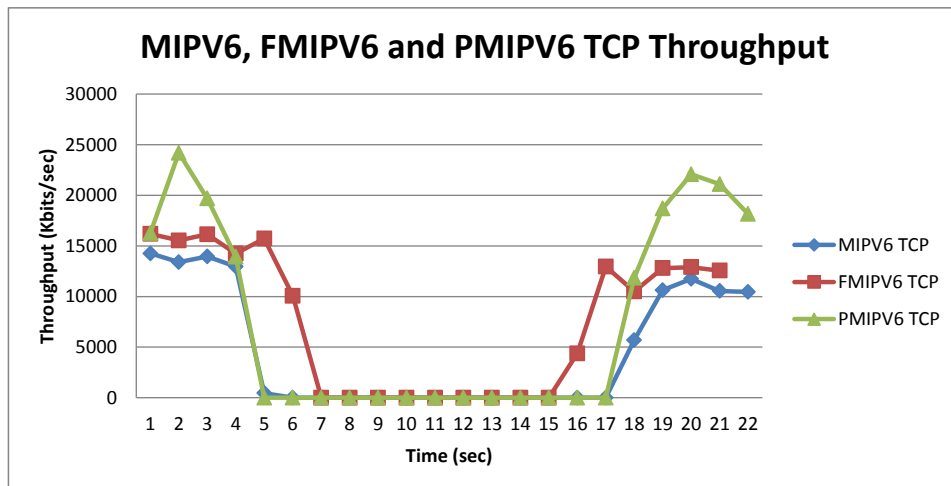


Figure 4.23: TCP Throughput MIPv6, FMIPv6 and PMIPv6

4.7.4 Handover Latency Comparison

Figure 4.24 shows the handover latency in seconds for MIPv6, FMIPv6 and PMIPv6 on three separate test sets. This figure shows that FMIPv6 has the smallest handover latency and PMIPv6 the largest handover latency for UDP

and TCP protocols. FMIPv6 performs better because of various advantages it possesses regarding to duplicate address detection and IPv6 configuration during handover.

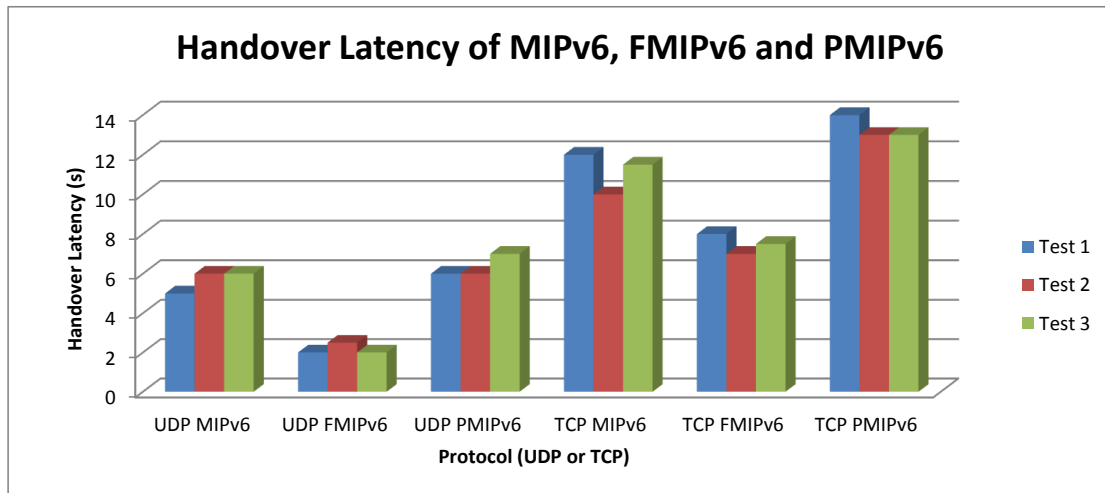


Figure 4.24: Handover Latency of MIPv6, FMIPv6 and PMIPv6

4.7.5 UDP Packet Loss Comparison

Figure 4.25 shows the comparison in packet loss for the various protocols. The total number of packets transmitted by the server for the test period can be seen compared with the total number of packets lost. A handover to a new access point took place during this time. The results of two tests for each protocol under similar network conditions are presented.

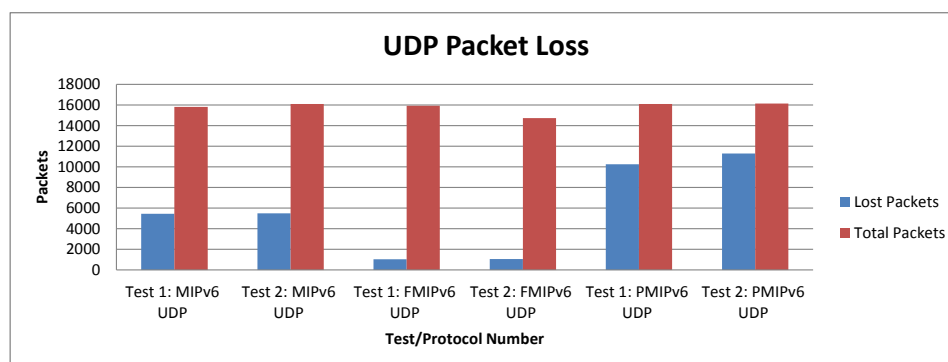


Figure 4.25: UDP Packet Loss MIPv6, FMIPv6 and PMIPv6

4.7.6 UDP Jitter Comparison

UDP jitter comparison can be seen in Figure 4.26. This shows the average jitter over the test period. During this period handovers were performed to a new access router.

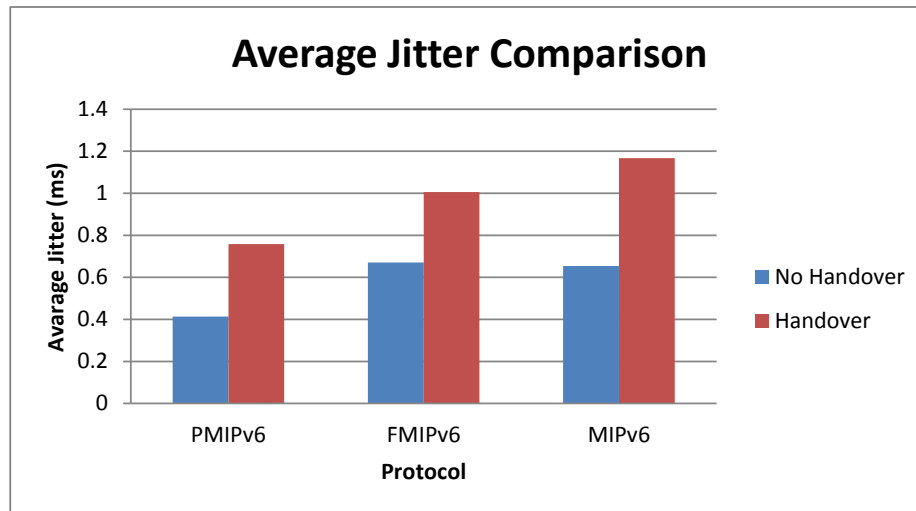


Figure 4.26: UDP Jitter Comparison

Table 4.1 shows a quick comparison of various test results. These results were used mainly to determine the QoS and the overall performance of each protocol. The results give the comparison of Handover time in seconds, jitter and throughput comparison.

Table 4.1: Protocol Results

Protocol	TCP HO(s)	UDP HO(s)	Jitter (ms)	Throughput (Kbits/sec)	Throughput HO (Kbits/sec)
MIPv6	12	7	1.166	9960	6301
FMIPv6	8	2	1.006	9914	8899
PMIPv6	13	6.5	0.758	9971	5502

4.8 MIPv6 for 6LoWPAN

In section 2.5.6 an implementation of MIPv6 was introduced for 6LoWPAN [5]. This section will look at the results obtained from a 6LoWPAN MIPv6 simulation module in the Cooja Network simulator for Contiki. Figure 4.27

shows the average handover latency for a mobile sensor node with the modified movement detection adaptation layer. The large handover intervals are related to the movement detection mechanism. Once the mobile sensor is in the new IPv6 network, the mobile sensor has to wait for the expiration of the default router lifetime before sending new router solicitations [5].

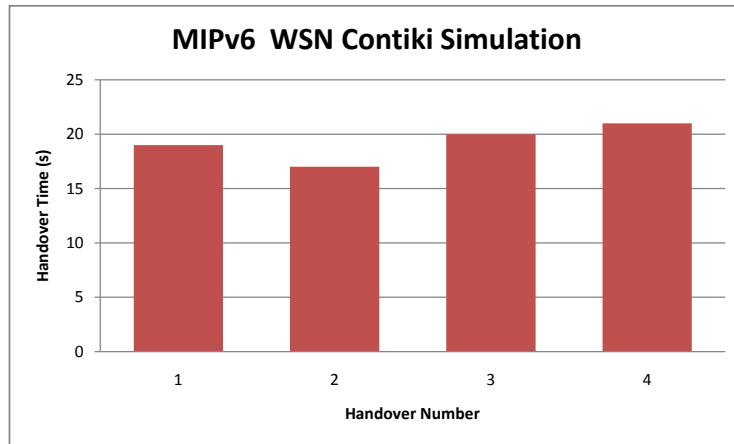


Figure 4.27: MIPv6 WSN Handover Latency

4.9 Conclusion

This chapter has summarised handover and network performances of the various protocols discussed and the results of the simulation model for MIPv6. The main focus of this thesis is to implement and compare the network performance of MIPv6, FMIPv6 and PMIPv6. From the results presented it is clear that the protocols improve the QoS of the network because of session continuity. This ensures the user of being able to hop between different access points without losing connection or session. The handover latency shown in this section is not as good as might be desired, but other Link-layer technologies can also be used to help improve the latency.

Chapter 5

Conclusion, Outcomes and Recommendations

The previous chapter presented all the results obtained from the various experiments. The results consisted of TCP, UDP and handover performances for MIPv6, FMIPv6 and PMIPv6. A further result was the evaluation of the effects on the performance of the application when using the mobility protocols. The implementation of MIPv6 for 6LoWPAN networks was also investigated. This chapter presents the outcomes of the completed work and recommendations for future work.

5.1 Conclusion

This thesis presents the evaluation of mobility protocols to determine the effectiveness and efficiency of these protocols. As the number of mobile devices grows rapidly and IPv4 addresses are being exhausted, making MIPv6 the future option as a network mobility protocol for IP Networks. The main goal of the mobility protocol is to enable network applications to operate continuously at the required QoS for both wired and wireless networks.

The protocols were tested on a real testbed environment to ensure reliable and real world results. MIPv6 was simulated under the OMNeT++ simulation framework. The results obtained from the simulation were used as a comparison for the testbed results. If one looks at the results that were obtained from the various experiments, it is clear that many external factors need to be considered to before any conclusion can be reached on the performance of the protocols.

From the results obtained in the simulation module it is clearly visible that the largest contribution of handover latency is Duplicate Address Detection. The simulation module serves as a very good platform to develop and expand the basic MIPv6 protocol and to introduce possible enhancements. These enhancements can be used to build FMIPv6 and PMIPv6 on top of the existing

MIPv6 implementation.

- The functional characteristics and performance of the simulation model are very similar to those of the testbed.
- Router advertisements interval delay mainly affects the T_{RDD} component of the overall delay and has a direct impact on the time it takes for the MN to determine whether it has moved or not.
- Router advertisements do not affect the MIPv6 protocol specific delay incurring components (T_{HRD} , T_{RRD} , T_{CRD}) and hence remain constant through all the test runs in both the test bed and simulation model.

The various protocols are implemented in a Linux OS environment and is strongly dependent on the hardware support of the network. The testbed setup was done in such a way that no network element other than those being used in the testbed can generate traffic on the network. The network was isolated from the outside network and each protocol was tested with the same network parameters to ensure consistent and reliable results.

- All protocols allow the mobile node always to be reachable at its home IP address and ensuring session continuity when changing access points.
- The implementation of layer 3 mobility protocols is transparent to all higher applications. The application layer is unaware of the changes when moving around in the network.
- The MIPv6 testbed behaved as expected and introduced session continuity for the MN when moving around in the network.
- The main contribution to handover delay was DAD and IP addresses configuration during handover.

The implementation of FMIPv6 on the testbed was a huge challenge due the fact that FMIPv6 is very dependent on the hardware/firmware of the network components. The FMIPv6 testbed behaved as expected and introduced lower handover latency for the MN when moving around in the network. The results showed that FMIPv6 outperformed the other protocols when it came to handover latency and packet loss. The reason for this is that FMIPv6 allows packet forwarding and buffering when the mobile node is changing access points. FMIPv6 also improves the DAD delay when a handover is performed by pre-configuration of the NCoA for the mobile node before the handover takes place. On the other hand, FMIPv6 had a few drawbacks that make it impractical to implement, the main reasons being:

- FMIPv6 is implemented and configured for a specific network only.

- For FMIPv6 implementation all the MAC addresses of the access routers need to be known and must be configured individually, making it very difficult and impractical to deploy on large scale networks.
- FMIPv6 does not provide a way for MN to discover a candidate access router. Sending an RtSolPr to PAR only requests the MAC address of candidate access router. FMIPv6 does not tell the MN whether it is in range of any of the candidate access routes. The only way for MN to discover a suitable AR is to perform periodic scanning and this can also take a long time to complete and is dependent on the type of hardware used.
- On the other hand, FMIPv6 also has the option to monitor the signal strength of the surrounding AP and can use this to make a decision on when to switch to a new AP.

The PMIPv6 testbed behaved as expected and introduced lower handover latency than MIPv6 for the MN when moving around in the network. PMIPv6 introduces a huge advantage when compared to MIPv6 and FMIPv6. The mobile node is independent of the network mobility and no enhancement is needed on the mobile node for mobility control messaging. The PMIPv6 network makes use of a Radius server to allow authentication of mobile nodes and this allows for easy management of mobile nodes in the PMIPv6 network domain. PMIPv6 has been chosen by 3rd Generation Partnership Project (3GPP) and Long Term Evolution (LTE) technologies as best mobility solution due to the fact that MN needs no stack modification.

The thesis has presented a brief overview of 6LoWPAN and the adaptation layer used by 6LoWPAN to reduce the length of the multiple headers used in IPv6 packets. Results obtained from a simulation for MIPv6 over 6LoWPAN [5] indicates that MIPv6 could be a practical solution for layer 3 mobility support in low power devices. The main focus of improving the handover efficiency is to look at optimizing the movement detection of the sensor. Possible solutions are presented in [66] and make use of neighbor detection through the process of overhearing.

5.2 Outcomes of Completed Work

The outcomes of this thesis in terms of the functioning of mobile IP networks, specifically Mobile IPv6, Fast Handovers for Mobile IPv6 and Proxy Mobile IPv6, the applicability of this protocols and simulation thereof are as follows:

- Investigation and explanation of the fundamental changes in IPv6 over IPv4 and the improvements made on IPv6 to enable network layer mobility in IPv6. Explanation of how IPv6 headers are implemented to enable mobility in the network layer and IP encapsulation.

- In depth explanation of the functioning of mobility protocols namely Mobile IPv6, Fast Handovers for Mobile IPv6 and Proxy Mobile IPv6. The thesis also explains the differences between the various protocols mentioned, how they can improve the handover latency and QoS of the network.
- Provision of insight into Network Mobility (NEMO) and the Internet-of-Things and how this correlates to the network protocols implemented in this thesis.
- Performance analysis of MIPv6 using OMNeT++ as a simulation environment (Section 3.2).
- Development of an IPv6 network (Section 3.3) to serve as a platform for evaluation of mobility protocols and handover scenarios applicable to a wireless network environment.
- Implementation and evaluation of MIPv6, FMIPv6 and PMIPv6 on the testbed platform.
- Explanation of IPv6 compression techniques to allow functioning of IPv6 in low power networks/devices.
- Investigation into the use of MIPv6 in a 6LoWPAN architecture.
- Explanation and illustration of the functioning of protocols. This virtually illustrates the mobility feature imposed by these networks and how it compares to a standard IPv6 network.
- Presentation of results obtained from the various tests conducted on the testbed and simulation environment. These results consisted of the handover latency produced and general network performance measurements such as throughput, packet loss and jitter.
- Discussion of various protocol advantages and drawbacks. This stipulates the practical implementation readiness of protocols in a large area network.
- Establishment of a realistic and comprehensive testbed to facilitate further mobility testing and evaluation into future research activities.

This thesis has attempted to present a detailed explanation, understanding and evaluation of a proposed mobility solution in an IPv6 network. The results produced in the thesis consist of in depth testing and simulation of mobility protocols. In particular, results was produced applicable to the handover latency of MIPv6, FMIPv6 and PMIPv6, as well as general network performance of the various protocols mentioned. This thesis can be used to determine a base point in protocol performances and as reference for deciding on a specific

mobility protocol or future enhancement of protocols.

In terms of the objectives initially defined, these have been achieved. The results suggest some interesting avenues for further research.

5.3 Recommendations for Future Work

The work reported in this thesis could form part of structured way to implement mobility protocols and to set up valuable testbeds to perform mobility testing and future enhancement testing. The work illustrates detailed evaluation of all the protocols and this can be used as a baseline for protocol performance and functioning. From the results obtained, PMIPv6 is a better candidate for future mobility, but still has some drawbacks to overcome before it can be considered a perfect solution for mobility. The development of simulation models for FMIPv6 and PMIPv6 in OMNeT++ can be seen as future work. This will contribute to the evaluation and improvement of the respected protocols. Future work will follow up on the drawbacks identified and will be aimed at possible solutions to improve mobility protocols and to overcome the drawbacks. Possible focus points are to investigate link layer handover techniques to minimize the delays produced by this, research can be based on work done in [67] as well as optimizing movement detection of mobile nodes. The establishment of a dual evaluation vehicle can also be seen as important in terms of more detailed application driven future work. Further work can also be focused on the Internet of Things [30]. The basic idea is that IoT will connect objects around us to provide seamless communication and contextual services provided by them.

Appendices

Appendix A

A.1 xMIPv6 Runtime Parameters

All the parameters used for the MIPv6 simulation as well as a description of the parameters are included in this section. These parameters are used for enabling certain simulation modules like TCP, UDP etc. Table A.1 shows all the simulation parameters for the MIPv6 simulation.

Table A.1: MIPv6 OMNeT++ Simulation Parameters

Parameter Name	Value	Description
isHomeAgent	boolean	Specify if the node is a HA or not
isMobileNode	boolean	Specify if the node is MN or not
num-rngs	2	Number of RNG used for the simulation
debug-on-errors	false	Enable debugging after simulation errors
sim-time-limit	308	Specify simulation time in (s)
**debug	true	Enable debug mode
*.total_mn	1	Number of MN's in the network
*.total_cn	2	Number of CN's in the network
**CNAddress	"CN[0]"	Used by the MN to directly access the CN address.
**CNAddress1	"CN[1]"	Used by the MN to directly access the CN address.
**gen[*].rng-0	1	Map physical RNG 1 to General Test
**neighbourDiscovery.minIntervalBetweenRAs	0.03s	Minimum RA interval specified by (RFC 6275)

Continued on next page

Table A.1 – continued from previous page

Parameter Name	Value	Description
** .neighbourDiscovery.maxIntervalBetweenRAs	0.07s	Maximum RA interval specified by (RFC 6275)
*.configurator.useTentativeAddrs	false	Set to true for testing DAD
*.playgroundSizeX	850	Channel control - testing ground size
*.playgroundSizeY	850	Channel control - testing ground size
*.channelcontrol.carrierFrequency	2.4GHz	Set carrier frequency
*.channelcontrol.pMax	2.0mW	Maximum transmission power possible
*.channelcontrol.sat	-82dBm	Signal attenuation threshold
*.channelcontrol.alpha	2	Path loss coefficient
*.channelcontrol.numChannels	3	Number of wireless channels used
** .AP*.wlan.mgmt.beaconInterval	0.1s	Set beacon interval timing
** .MN*.** .mgmt.accessPointAddress	"10:AA:00:00:00:01"	MN MAC address
** .wlan.mgmt.numAuthSteps	4	Number of authentication steps to AP
** .mgmt.frameCapacity	10	Frame capacity - Drop-TailQueue
** .AP_Home.wlan.mgmt.ssid	"HOME"	SSID name of AP_Home
** .AP_Home.wlan.mac.address	"10:AA:00:00:00:01"	MAC address AP_Home WLAN
** .AP_Home.eth[0].address	"10:AE:00:00:00:02"	MAC address AP_Home Ethernet
** .AP_Home.eth[0].txrate	100Mbps	Transmit rate
** .AP_Home.eth[0].duplexEnabled	true	Enable duplexing
** .AP_Home.eth[0].*.scalar-recording	false	Enable stats recording
** .AP_1.wlan.mgmt.ssid	"AP1"	SSID name of AP_1
** .AP_1.wlan.mac.address	"10:AA:00:00:A1:01"	A1:01 specifies AP_1:interface 1
** .AP_1.eth[0].address	"10:AE:00:00:A1:02"	A1:02 specifies AP_1:interface 2
** .AP_1.eth[0].txrate	100Mbps	Transmit rate
** .AP_1.eth[0].duplexEnabled	true	Enable duplexing
** .AP_1.eth[0].*.scalar-recording	false	Enable stats recording
** .MN[0].mobilityType	"RectangleMobility"	Select type of mobility to use
** .MN[0].mobility.debug	false	Enable mobility debugging
** .MN[0].mobility.x1	180	Start position x axis
** .MN[0].mobility.y1	100	Start position y axis
** .MN[0].mobility.x2	530	End position x axis
** .MN[0].mobility.y2	110	End position y axis

Continued on next page

Table A.1 – continued from previous page

Parameter Name	Value	Description
** .MN[0].mobility.startPos	0	Used to determine start position
** .MN[0].mobility.speed	1mps	Speed of the MN
** .MN[0].mobility.updateInterval	0.1s	Interval of updating mobility
** .numUdpApps	0	Set to 1 to enable UDP
** .udpAppType	"UDPBasicApp"	Select UDP app to use
** .MN[*].numTcpApps	0	To enable change to 1
** .MN[*].tcpAppType	"TelnetApp"	
** .MN[0].tcpApp[0].address	"aaaa:b::aaa:ff:fe00:7"	Source addr of MN[0] TCP app
** .MN[1].tcpApp[0].address	"aaaa:b::aaa:ff:fe00:8"	Source addr of MN[1] TCP app
** .MN[0].tcpApp[0].port	-1	Select prot to listen on, -1 is all ports
** .MN[1].tcpApp[0].port	-1	See above
** .MN[*].tcpApp[0].connectAddress	"CN"	Address to connect to
** .MN[0].tcpApp[0].connectPort	1000	Same destination port numbers
** .MN[1].tcpApp[0].connectPort	1000	Same destination port numbers
** .MN[*].tcpApp[0].numCommands	exponential(1)	Exponential distribution between 1
** .MN[*].tcpApp[0].commandLength	exponential(1)	Exponential distribution between 1
** .MN[*].tcpApp[0].keyPressDelay	exponential(0.1)	Exponential distribution between 0.1
** .MN[*].tcpApp[0].commandOutputLength	exponential(40)	Exponential distribution between 40
** .MN[*].tcpApp[0].thinkTime	truncnormal(2,3)	Truncnormal distribution
** .MN[*].tcpApp[0].idleInterval	truncnormal(3600,1200)	Idle interval truncnormal distribution
** .MN[*].tcpApp[0].reconnectInterval	30s	Reconnect interval
** .CN*.numTcpApps	0	To enable change to 1
** .CN*.tcpAppType	"TCPGenericSrvApp"	Select server type
** .CN*.tcpApp[0].address	" "	
** .CN*.tcpApp[0].port	1000	
** .CN*.tcpApp[0].replyDelay	0	
** .tcp.mss	1024	
** .tcp.advertisedWindow	14336	14*mss
** .tcp.sendQueueClass	"TCPMsgBasedSendQueue"	
** .tcp.receiveQueueClass	"TCPMsgBasedRcvQueue"	
** .tcp.tcpAlgorithmClass	"TCPReno"	
** .tcp.recordStats	true	

Continued on next page

Table A.1 – continued from previous page

Parameter Name	Value	Description
** .MN[0].pingApp.destAddr	""	"CN[0]"
** .MN*.pingApp.destAddr	""	"CN[1]"
** .MN*.pingApp.srcAddr	""	
** .MN*.pingApp.packetSize	56B	
** .MN*.pingApp.interval	0.01s	
** .MN*.pingApp.hopLimit	32	
** .MN*.pingApp.count	0	
** .MN*.pingApp.startTime	200s	Choose start time for the Ping Application
** .MN*.pingApp.stopTime	0	
** .MN*.pingApp.printPing	true	
** .ppp[*].queueType	"DropTailQueue"	In routers
** .ppp[*].queue.frameCapacity	10	Frame capacity in routers - DropTailQueue
** .eth[*].queueType	"DropTailQueue"	Select queue type in routers
** .eth[*].queue.frameCapacity	10	In routers
** .eth[*].encap.*.scalar-recording	false	
** .eth[*].mac.promiscuous	false	
** .eth[*].mac.address	"auto"	
** .eth*.mac.txrate	100Mbps	
** .eth*.mac.duplexEnabled	true	
** .eth*.mac.*.scalar-recording	false	
** .ap.*.scalar-recording	false	
** .hub.*.scalar-recording	false	
** .AP_Home.wlan.radio.channelNumber	1	
** .AP_1.wlan.radio.channelNumber	2	
** .MN*.wlan.radio.channelNumber	0	Initial value - The MN will scan
** .wlan.agent.activeScan	true	
** .wlan.agent.channelsToScan	"1 2"	"" means all
** .wlan.agent.probeDelay	0.1s	
** .wlan.agent.minChannelTime	0.15s	
** .wlan.agent.maxChannelTime	0.3s	
** .wlan.agent.authenticationTimeout	5s	
** .wlan.agent.associationTimeout	5s	
** .mac.address	"auto"	
** .mac.maxQueueSize	14	
** .mac.rtsThresholdBytes	4000B	
** .mac.bitrate	2Mbps	
** .wlan.mac.retryLimit	7	
** .wlan.mac.cwMinData	7	
** .wlan.mac.cwMinBroadcast	31	
** .radio.bitrate	2Mbps	

Continued on next page

Table A.1 – continued from previous page

Parameter Name	Value	Description
**radio.transmitterPower	2.0mW	
**radio.carrierFrequency	2.4GHz	
**radio.thermalNoise	-110dBm	
**radio.sensitivity	-82mW	
**radio.pathLossAlpha	2	
**radio.snirThreshold	4dB	
**relayUnitType	"MACRelayUnitNP"	
**relayUnit.addressTableSize	100	
**relayUnit.agingTime	120s	
**relayUnit.bufferSize	1MB	
**relayUnit.highWatermark	512KB	
**relayUnit.pauseUnits	300	pause for 300*512 bit (19200 byte) time
**relayUnit.addressTableFile	""	
**relayUnit.numCPUs	2	
**relayUnit.processingTime	2us	
**relayUnit.*.scalar-recording	false	

Appendix B

B.1 Router Advertisement Test Bed Configuration

The configuration for the Router Advertisement of the home network can be seen below. It is important to notice the `MinRtrAdvInterval` and `MaxRtrAdvInterval`. These parameters are specified in [8] but are continuously changed for evaluation purpose. The `AdvHomeAgentFlag` is used by the Home Agent to indicate to the network that it serves as a Home Agent on the network.

```
interface wlan0 {
    AdvSendAdvert on;
    AdvManagedFlag off;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;
    AdvHomeAgentFlag on;
    AdvHomeAgentInfo on;
    HomeAgentLifetime 1800;
    HomeAgentPreference 10;
    prefix 2001:db8:abcd:14::/64 {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

B.2 MIPv6 Test Bed Configuration

The MIPv6 network setup configuration used for the `mip6d` daemon running on the network to enable handling of mobility messages is described in this section. For the MIPv6 setup the configuration is split up into three sections, Home Agent configuration, Mobile Node configuration and Correspondent Node configuration.

B.2.1 MIP6D MN Configuration

All the configuration for the mobile in the MIPv6 network can be seen in Table B.1.

Table B.1: MIP6D MN Parameters

Parameter Name	Value	Description
NodeConfig	MN	Specify if the node is a MN, HA or CN
DebugLevel	10	If set to > 0, will not detach from tty
DoRouteOptimizationCN	enabled	Support route optimization with other MN's
DoRouteOptimizationMN	enabled	Use route optimization with CN's
UseCnBuAck	disabled	Indicates if the Ack bit should be set in BU sent to CN
OptimisticHandoff	disabled	Begin RO before the MN received a BACk
MnMaxHaBindingLife	420	Specify the maximum HA binding lifetime
MnMaxCnBindingLife	420	Specify the maximum CN binding lifetime
MnDiscardHaParamProb	disabled	Toggles if the MN should discard ICMPv6 Parameter Problem messages from HA.
Interface	"wlan0"	Choose wireless interface for MN
MnRouterProbes	1	Indicates how many times the MN should send NUD probes to its old router after receiving a RA from a new one.
MnHomeLink { IsMobRtr HomeAddress HomeAgentAddress MnRoPolicy }	"wlan0" enabled 2001:db8:abcd:14:217:c4ff:fe39:7306/64 2001:db8:abcd:14::1 enabled	Setup for the Homelink structure, multiple HomeAddress and MN.
UseMnHaIPsec	enabled	Enable IPsec for MIPv6
KeyMngMobCapability	enabled	Enable Key Management Mobility Capability

Continued on next page

Table B.1 – continued from previous page

Parameter Name	Value	Description
IPsecPolicySet { HomeAgentAddress HomeAddress IPsecPolicy Mh IPsecPolicy ICMP IPsecPolicy TunnelMh }	2001:db8:abcd:14::1 2001:db8:abcd:14:217:c4ff:fe39:7306/64 UseESP 1 2 UseESP 5 UseESP 3 4	IPsec settings for security in Binding messages.

B.2.2 MIP6D HA Configuration

Table B.2 shows the configuration for the home agent in the MIPv6 network.

Table B.2: MIP6D HA Parameters

Parameter Name	Value	Description
NodeConfig	HA	Specify if the node is a MN, HA or CN
DebugLevel	10	If set to > 0, will not detach from tty
Interface	"eth1"	Choose wireless interface for MN
DoRouteOptimizationMN	enabled	Support route optimization with other MN's
DoRouteOptimizationCN	enabled	Use route optimization with CN's
UseMnHaIPsec	enabled	Enable IPsec for MIPv6

B.2.3 MIP6D CN Configuration

Table B.3 shows the configuration for the correspondent node in the MIPv6 network.

Table B.3: MIP6D CN Parameters

Parameter Name	Value	Description
NodeConfig	CN	Specify if the node is a MN, HA or CN
DoRouteOptimizationCN	enabled	Support route optimization with other MN's.

B.3 FMIPv6 Test Bed Configuration

In this section the FMIPv6 network setup configuration used for the `fmip6d` daemon running on the network to enable handling of mobility messages for FMIPv6 is described. For the FMIPv6 setup, the configuration is split up into three sections, Previous Access Router configuration, New Access Router configuration and Mobile Node configuration. The FMIPv6 daemon is run on top of the MIPv6 protocol and this means that the configuration described in Section B.2 is also applicable in this network setup.

B.3.1 FMIPv6 MN Configuration

Table B.4 shows the configuration for a Mobile Node in a FMIPv6 network setup.

Table B.4: FMIPv6 MN Parameters

Parameter Name	Value	Description
DetachFromTTY	true	Specifies if FMIPv6 runs as daemon
DebugLevel	1	Specify debug level for daemon
PrimaryInterface { IfName }	wlan0	Specifies a primary interface for use.
LinkQuality{ UseLinkQualityTriggers Threshold }	false -50	Link quality and threshold for Link-Layer handover trigger.
ScanningInterface { IfName }	wlan1	Specifies an alternate interface used for scanning
OptimizeL2onlyHovers	true	Indicates whether the MN optimizes L2-only handovers or not
ScanNotificationWorkaround	false	Indicates whether the driver for the scanning interface is broken or not.

B.3.2 FMIPv6 PAR Configuration

In Table B.5 the configuration for a previous access router can be seen.

Table B.5: FMIPv6 PAR Parameters

Parameter Name	Value	Description
DetachFromTTY	true	Specify if FMIPv6 runs as daemon
DebugLevel	1	Specify debug level for daemon
AskSupportForBuffering	true	Ask NAR to buffer incoming packets until MN finishes its Handover.
BufferLength	5	Number of packets buffered on NAR until the MN attaches.
Interface { IfName Preference}	eth1 1	Specifies an interface and options associated with it.
NIHover{ do_ni_hover do_ni_hover }	true 00:0C:02:00:AD:19	Specifies whereas the Access Router incites the Mobile Nodes to perform a handover. When do_ni_hover is set to true, every RtSolPr acts as a trigger for a Network Initiated Handover.
nap MIP4-AP { nap_lla nr_lla nr_addr nr_pfx }	00:0C:02:00:AD:19 00:19:d1:75:b1:e1 2001:db8:abcd:15::1/64 2001:db8:abcd:15::/64	Known access points configuration.

B.3.3 FMIPv6 NAR Configuration

Table B.6 shows the configuration for the new access router in a FMIPv6 network.

Table B.6: FMIPv6 NAR Parameters

Parameter Name	Value	Description
DetachFromTTY	true	Specify if FMIPv6 runs as daemon
Continued on next page		

Table B.6 – continued from previous page

Parameter Name	Value	Description
DebugLevel	1	Specify debug level for daemon
AskSupportForBuffering	true	Ask NAR to buffer incoming packets until MN finishes its Handover.
BufferLength	5	Number of packets buffered on NAR until the MN attaches.
Interface { IfName Preference}	eth1 1	Specifies an interface and options associated with it.
NIHover{ do_ni_hover do_ni_hover }	true 80:1F:02:4E:AD:17	Specifies whereas the Access Router incites the Mobile Nodes to perform a handover. When do_ni_hover is set to true, every RtSolPr acts as a trigger for a Network Initiated Handover.
nap MIP4-AP { nap_lla nr_lla nr_addr nr_pfx }	80:1F:02:4E:AD:17 00:13:72:7a:09:f7 2001:db8:abcd:17::1/64 2001:db8:abcd:17::/64	Known access points configuration.

B.4 PMIPv6 Test Bed Configuration

To enable the handling of mobility messages, the pmip6d daemon must be able to run thus the PMIPv6 network was set up as described in this section. For the PMIPv6 setup the configuration is split up into three section, Local Mobility Anchor configuration, Mobile Access Router 1 configuration and Mobile Access Router 2 configuration. The PMIPv6 network setup does not rely on configuration mentioned in Section B.2 and Section B.3.

B.4.1 PMIPv6 LMA Configuration

Table B.7 shows the configuration for the local mobility anchor in a PMIPv6 network.

Table B.7: PMIPv6 LMA Parameters

Parameter Name	Value	Description
NodeConfig	LMA	Specify if the node is a MN, HA or CN

Continued on next page

Table B.7 – continued from previous page

Parameter Name	Value	Description
DebugLevel	10	Specify debug level for daemon
DoRouteOptimizationCN	disabled	Specifies a primary interface for use.
DoRouteOptimizationMN	disabled	Link quality and threshold for Link-Layer handover trigger.
UseMnHaIPsec	disabled	Specifies an alternate interface used for scanning
KeyMngMobCapability	disabled	Indicates whether the MN optimizes L2-only handovers or not
ProxyMipLma "LMA testbed n1" {		Indicates whether the driver for the scanning interface is broken or not.
Timest ampBasedApproachInUse	enabled	
MobileNodeGeneratedTimest ampInUse	disabled	
FixedMAGLinkLocalAddressOnAllAccessLinks	fe80::211:22ff:fe33:4455	
FixedMAGLinkLayerAddressOnAllAccessLinks	00:11:22:33:44:55	
MinDelayBeforeBCEDelete	10000	
MaxDelayBeforeNewBCEAssign	1500	
Timest ampValidityWindow	300	
LmaAddress		
LmaPmipNetworkDevice	eth0	
LmaCoreNetworkAddress		
LmaCoreNetworkAddress	eth0	
RetransmissionTimeOut	500	
MaxMessageRetransmissions	5	
TunnelingEnabled	enabled	
DynamicTunnelingEnabled	enabled	
Mag1AddressIngress		
Mag1AddressEgress		
Mag2AddressIngress		
Mag2AddressEgress		
}		

B.4.2 PMIPv6 MAG1 Configuration

Table B.8 shows the configuration for mobile access gateway 1 as seen in Figure 3.8 in a PMIPv6 network.

Table B.8: PMIPv6 MAG1 Parameters

Parameter Name	Value	Description
NodeConfig	MAG	Specify if the node is a MN, HA or CN
DebugLevel	10	Specify debug level for daemon
DoRouteOptimizationCN	disabled	Specifies a primary interface for use.
DoRouteOptimizationMN	disabled	Link quality and threshold for Link-Layer handover trigger.
UseMnHaIPsec	disabled	Specifies an alternate interface used for scanning
KeyMngMobCapability	disabled	Indicates whether the MN optimizes L2-only handovers or not

Continued on next page

Table B.8 – continued from previous page

Parameter Name	Value	Description
ProxyMipMag "MAG2 testbed n1" {		Indicates whether the driver for the scanning interface is broken or not.
TimeStamptBasedApproachInUse	enabled	
MobileNodeGeneratedTimeStamptInUse	disabled	
FixedMAGLinkLocalAddressOnAllAccessLinks	00:11:22:33:44:55	
FixedMAGLinkLayerAddressOnAllAccessLinks	enabled	
EnableMAGLocalRouting		
LmaAddress		
MagAddressIngress		
MagAddressEgress		
MagDeviceIngress	"eth1"	
MagDeviceEgress	"eth0"	
PBULifeTime	40000	
RetransmissionTimeOut	500	
MaxMessageRetransmissions	5	
TunnelingEnabled	enabled	
DynamicTunnelingEnabled	enabled	
RadiusClientConfigFile	/usr/local/etc/radiusclient/radiusclient.conf	
RadiusPassword	password	
PcapSyslogAssociationGrepString	ReAssociation Success:STA	
PcapSyslogDeAssociationGrepString	Received Deauth:STA	
}		

B.4.3 PMIPv6 MAG2 Configuration

Table B.9 shows the configuration for mobile access gateway 2 as seen in Figure 3.8 in a PMIPv6 network.

Table B.9: PMIPv6 MAG2 Parameters

Parameter Name	Value	Description
NodeConfig	MAG	Specify if the node is a MN, HA or CN
DebugLevel	10	Specify debug level for daemon
DoRouteOptimizationCN	disabled	Specifies a primary interface for use.
DoRouteOptimizationMN	disabled	Link quality and threshold for Link-Layer handover trigger.
UseMnHaIPsec	disabled	Specifies an alternate interface used for scanning
KeyMngMobCapability	disabled	Indicates whether the MN optimizes L2-only handovers or not
Continued on next page		

Table B.9 – continued from previous page

Parameter Name	Value	Description
ProxyMipMag "MAG2 testbed n1"		Indicates whether the driver for the scanning interface is broken or not.
TimestampBasedApproachInUse	enabled	
MobileNodeGeneratedTimestampInUse	disabled	
FixedMAGLinkLocalAddressOnAllAccessLinks	00:11:22:33:44:55	
FixedMAGLinkLayerAddressOnAllAccessLinks	enabled	
EnableMAGLocalRouting		
LmaAddress		
MagAddressIngress		
MagAddressEgress		
MagDeviceIngress	"eth1"	
MagDeviceEgress	"eth0"	
PBULifeTime	40000	
RetransmissionTimeOut	500	
MaxMessageRetransmissions	5	
TunnelingEnabled	enabled	
DynamicTunnelingEnabled	enabled	
RadiusClientConfigFile	/usr/local/etc/radiusclient/radiusclient.conf	
RadiusPassword	password	
PcapSyslogAssociationGrepString	ReAssociation Success:STA	
PcapSyslogDeAssociationGrepString	Received Deauth:STA	

B.4.4 PMIPv6 FreeRadius Server

The following text is an example for the FreeRadius server configuration. This specifies the MN's that will have to access the LMA for control messages. This configuration will only be used to authorize the MN.

```
0000001c232fe989
Auth-Type := Accept, User-Password == "linux"
Service-Type = Authenticate-Only,
Framed-Interface-Id = 0000:0000:0000:0000,
Framed-IPv6-Prefix = 2001:0100:0005:6000::/64
```

B.4.5 PMIPv6 FreeRadius Client

The FreeRadius client was updated to accommodate IPv6. The following configuration is used to identify the clients.

```
client 2001:100::3
secret = testing123
shortname = mag2
nastype = other
password = linux
```

Appendix C

C.1 MIPv6 Contiki Runtime Parameters

Table C.1 shows all the simulation parameters for the MIPv6 WSN simulation in Contiki. The parameters consist of node parameters, 802.15.4 parameters and 6 LoWPAN network parameters.

Table C.1: WSN MIPv6 Contiki Simulation Parameters

Parameter Name	Value
Number of nodes	1 deployed MN
Network setup	2 Visited networks
Network dimension	Spacing between the two network is 10m.
Application model	Time-driven
Packet size	Between 20 to 35 bytes
Router lifetime	30s
Unicast router solicitation timeout	1s
Device frequency	2.4 GHz
Queue length	10

List of References

- [1] R. Hinden and S. Deering, “IP Version 6 Addressing Architecture,” <http://tools.ietf.org/html/rfc4291>, February 2006.
- [2] J. Kovács, L. Bokor, Z. Kanizsai, and S. Imre, *Review of Advanced Mobility Solutions for Multimedia Networking in IPv6*. June 2013.
- [3] S. Khan, K. Loo, and A. Kiani, “Tested Evaluation of Fast and Secure Handover in FMIPv6,” *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 2, April 2010.
- [4] EUROCOM, “OpenAirInterface Proxy Mobile IPv6,” <http://www.openairinterface.org/openairinterface-proxy-mobile-ipv6-oai-pmipv6>.
- [5] D. Roth, J. Montavont, and T. Noel, “Performance evaluation of mobile IPv6 over 6LoWPAN,” in *Proceedings of the 9th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, PE-WASUN '12*, (New York, NY, USA), pp. 77–84, ACM, 2012.
- [6] P. Ngamtura, “Performance Comparison Of Mipv6 And Fmipv6 Over WLAN,” Master’s thesis, Faculty Of Graduate Studies Mahidol University, March 2010.
- [7] C. Systems, “IPv6 Extension Headers Review and Considerations,” http://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8 pp. 1,12, October 2006.
- [8] C.Perkins, D. Johnson, and J. Arkko, “Mobility Support in IPv6,” <http://tools.ietf.org/html/rfc6275>, July 2011.
- [9] A. Sekercioglu and A. Varga, “OMNeT++ Comparison,” Tech. Rep. 1.6, March 2013.
- [10] R. Prasad and S. Dixit, *Wireless IP and Building the Internet*. The Artech House Universal Personal Communications Series, Artech House, 2002.
- [11] G. Xie, J. Chen, H. Zheng, J. Yang, and Y. Zhang, “Handover Latency of MIPv6 Implementation in Linux,” in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pp. 1780–1785, 2007.

- [12] J. Chandrasekaran, “Mobile IP: Issues, Challenges and Solutions,” in *CS 552 Computer Networks*, Rutgers University, 2009.
- [13] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, “Hierarchical Mobile IPv6 (HMIPv6) Mobility Management,” <http://tools.ietf.org/html/rfc5380>, October 2008.
- [14] T. Ernst, “Network Mobility Support Goals and Requirements,” <http://tools.ietf.org/html/rfc4886#page-2>, July 2007.
- [15] J. Hui and P. Thubert, “Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks,” <http://tools.ietf.org/html/rfc6282>, September 2011.
- [16] “Internet Protocol, Version 4 (IPv4) Specification,” Tech. Rep. RFC 791, IETF Secretariat, September 1981.
- [17] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” Tech. Rep. RFC 2460, IETF Secretariat, December 1998.
- [18] “World IPv6 Launch,” June 2012.
- [19] R. Koodli, “Fast Handovers for Mobile IPv6,” <http://tools.ietf.org/html/rfc5568>, July 2009.
- [20] K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, “Proxy Mobile IPv6,” <http://tools.ietf.org/html/rfc5213>, August 2008.
- [21] S. Gundavelli, “Extensions to Proxy Mobile IPv6 - Motivation,” <http://tools.ietf.org/html/draft-gundavelli-netext-extensions-motivation-00>, May 2009.
- [22] J. Pieterse and R. Wolhuter, “Investigation of handover techniques in a IPv6 mobile network,” in *Antennas and Propagation in Wireless Communications (APWC), 2012 IEEE-APS Topical Conference on*, pp. 1020–1023, 2012.
- [23] J. Pieterse, R. Wolhuter, and N. Mitton, “Implementation and Analysis of FMIPv6, an Enhancement of MIPv6,” in *Ad Hoc Networks* (J. Zheng, N. Mitton, J. Li, and P. Lorenz, eds.), vol. 111 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 351–364, Springer Berlin Heidelberg.
- [24] S. Ziegler, C. Crettaz, L. Ladid, S. Krco, A. Skarmeta, A. Jara, and W. Kastner, “IoT6 – Moving to an IPv6-based Future IoT report,” Tech. Rep. 20, 2013.

- [25] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," <http://tools.ietf.org/html/rfc4944>, September 2007.
- [26] M. Grayson, K. Shatzkamer, and K. Wierenga, *Building the Mobile Internet*. Cisco Press, 1 ed., February 2011.
- [27] "Understanding IPv6 Link Local Address," http://www.cisco.com/en/US/tech/tk872/technologies_configuration_example09186a0080ba, November 2011.
- [28] C. Systems, *IPv6 Configuration Guide, Cisco IOS Release*, ch. IPv6 Stateless Autoconfiguration. Cisco Systems, July 2012.
- [29] M. Kamichoff, "Stateful Autoconfiguration (DHCPv6)," April 2004.
- [30] D. MINOLI, *Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications*, vol. The Evolving World of The Evolving World. John Wiley & Sons, wol online book ed., July 2013.
- [31] "IEEE Standard 802.15.4-2009: Wireless medium access control and physical layer specifications for low-rate wireless personal area network ," <http://standards.ieee.org/about/get/802/802.15.html>, April 2009.
- [32] Z. Shelby and C. Bormann, "The Wireless Embedded Internet," p. 244, December 2009.
- [33] G. M. Lee, J. Park, N. Kong, N. Crespi, and I. Chong, "The Internet of Things - Concept and Problem Statement," <http://tools.ietf.org/html/draft-lee-iot-problem-statement-00>, July 2012.
- [34] F. Nazir and A. Seneviratne, "Towards Mobility Enabled Protocol Stack For Future Wireless Networks," *Ubiquitous Computing and Communication Journal*, vol. 2, no. 4, 2010.
- [35] Y. Chen and L. Li, "A fair packet dropping algorithm considering channel condition in diff-serv wireless networks," in *Computer and Information Technology, 2004. CIT '04. The Fourth International Conference on*, pp. 554–559, 2004.
- [36] M. Atiquzzaman and A. Reaz, "Survey and classification of transport layer mobility management schemes," in *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, vol. 4, pp. 2109–2115 Vol. 4, 2005.
- [37] X. Perez-Costa, M. Torrent-Morenoab, and H. Hartenstein, "A Performance Comparison of Mobile IPv6, Hierarchical Mobile IPv6, Fast Handovers for Mobile IPv6 and their Combination," *Mobile Computing and Communications Review*, vol. 7, no. 4, 2004.

- [38] A. Cabellos-Aparicio and H. Julian-Bertomeu, "Measurement-Based Comparison of IPv4/IPv6 Mobility Protocols on a WLAN Scenario," *Universitat Politècnica de Catalunya*, 2010.
- [39] V. Vassiliou and Z. Zinonos, "An Analysis of the Handover Latency Components in Mobile IPv6," *JOURNAL OF INTERNET ENGINEERING*, vol. 3, December 2009.
- [40] J. Darwich, "Comparative study of Mobile IPv4 and Mobile IPv6," p. 26, 2011.
- [41] R. Atkinson, "Security Architecture for the Internet Protocol," <http://www.ietf.org/rfc/rfc2401.txt>, November 1998.
- [42] M. SKOREPA and K. MOLNAR, "Time analysis of Route Optimization in MIPv6," in *Recent Advances in Computers* (N. E. Mastorakis, V. Mladenov, Z. Bojkovic, S. Kartalopoulos, and A. Varonides, eds.), p. 659, December 2012.
- [43] J. Arkko, C. Vogt, and W. Haddad, "Enhanced Route Optimization for Mobile IPv6," <http://www.ietf.org/rfc/rfc4866.txt>, May 2007.
- [44] S. Sudanthi, "SANS Institute InfoSec Reading Room - Mobile IPv6," SANS Institute, January 2003.
- [45] R. Silva and J. S. Silva, "An Adaptation Model for Mobile IPv6 support in lowPANs," <https://www.cisuc.uc.pt/publication/show/2300>, 2009.
- [46] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," <http://tools.ietf.org/html/rfc3963>, January 2005.
- [47] J. Xie, I. Howitt, and I. Shibeika, "IEEE 802.11-Based Mobile IP Fast Handoff Latency Analysis," in *Communications, 2007. ICC '07. IEEE International Conference on*, pp. 6055–6060, 2007.
- [48] Y. Bernet, S. Blake, D. Grossman, and A. Smith, "An Informal Management Model for Diffserv Routers," <http://tools.ietf.org/html/rfc3290>, May 2002.
- [49] H. N. Nguyen and C. Bonnet, "Scalable proxy mobile IPv6 for heterogeneous wireless networks," in *Mobiworld 2008, 2008 International Workshop on Mobile IPv6 and Network-based Localized Mobility Management, in conjunction with Mobility Conference 2008, September 10-12, I-Lan, Taiwan*, 09 2008.

- [50] Y. Kim and Y. Mun, “Adaptive Selection of MIPv6 and Hierarchical MIPv6 for Minimizing Signaling Cost,” in *Computational Science and Its Applications - ICCSA 2006* (M. Gavrilova, O. Gervasi, V. Kumar, C. Tan, D. Taniar, A. Laganá, Y. Mun, and H. Choo, eds.), vol. 3981 of *Lecture Notes in Computer Science*, pp. 611–620, Springer Berlin Heidelberg.
- [51] M. C. M. Albrecht, “Introduction to Discrete Event Simulation.” January 2010.
- [52] L. Leemis and S. Park, *Discrete-Event Simulation: A First Course* . No. 6–2004, December 2004.
- [53] L. Leemis and S. Park, *Random Number Generation*. No. 6–2004, December 2004.
- [54] *INET Framework for OMNeT++*, June 2012.
- [55] M. Matsumoto and T. Nishimura, “Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-RNG,” *ACM Transactions on Modeling and Computer Simulation*, vol. 8, pp. 3–30, January 1998.
- [56] S. Mustafa, F. Sardar, S. Shabbir, and A. Maqsood, “Analysis Of Mobile IP Handover Latency And Packet Loss Issues Using Simulation,” Master’s thesis, Department of Electrical Engineering, Feb 2011.
- [57] T. Henderson, *LBNL’s Network Simulator 2*, November 2011.
- [58] A. Alignan, “Introduction to Cooja,” January 2013.
- [59] F. Z. Yousaf, C. Bauer, and C. Wietfeld, “An accurate and extensible mobile IPv6 (xMIPv6) simulation model for OMNeT++,” in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, Simutools ’08, pp. 88:1–88:8, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.
- [60] V. Nuorvala and A. J. Tuominen, “UMIP - Mobile IPv6 and NEMO for Linux,” April 2013.
- [61] “Iperf: Network Performance,” tech. rep., July 2011.
- [62] “FMIPv6.org: Fast Handovers for Mobile IPv6,” September 2007.
- [63] A. Dunkels, “Full TCP/IP for 8-bit architectures,” in *Proceedings of the 1st international conference on Mobile systems, applications and services*, MobiSys ’03, (New York, NY, USA), pp. 85–98, ACM, 2003.

- [64] J. Polastre, R. Szewczyk, and D. Culler, “Telos: enabling ultra-low power wireless research,” in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*.
- [65] J. Postel, “The User Datagram Protocol (UDP),” <http://www.erg.abdn.ac.uk/~gorry/course/inet-pages/udp.html>, May 1996.
- [66] D. Roth, J. Montavont, and T. Noel, “MOBINET: Mobility management across different wireless sensor networks,” in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pp. 351–356, 2011.
- [67] E. Iovov and T. Noel, “Soft Handovers over 802.11b with Multiple Interfaces,” in *Wireless Communication Systems, 2005. 2nd International Symposium on*, pp. 549–554, 2005.