February 18, 2015

Master thesis

Lossless Multicast Handovers in Proxy Fast Mobile IPv6 Networks

B.J. Meijerink
Master Telematics

Graduation committee:
dr.ir. G.J. Heijenk
prof.dr.ir. A. Pras
dr.ir. P.T. de Boer

UNIVERSITY OF TWENTE

Abstract

There is a demand in the Public Protection and Disaster Relief (PPDR) community for high bandwidth services on mobile devices, including group communications. The two technologies that are currently available and can meet the bandwidth requirements of the PPDR community are LTE and IEEE 802.11. Both technologies are mostly used to transmit IP traffic in which multicasting is the efficient way to transmit data to more than one receiver. It is important PPDR users can switch seamlessly between wireless networks.

In this thesis an architecture for a future PPDR networks that is fully IP based and uses Proxy Fast Mobile IPv6 (PFMIPv6) to provide seamless mobility to its users is presented. It also describes the problematic nature of multicast handovers in PFMIPv6 networks and how to overcome them. An implementation for such a multicast capable PFMIPv6 system is described and this implementation is evaluated. It is shown the approach can reduce packet loss for multicast packets during a handover to almost 0.

Table of contents

Acronyms

IP Internet Protocol

MIP Mobile IP

PMIP Proxy Mobile IP

PMIPv6 Proxy Mobile IPv6

PFMIPv6 Proxy Fast Mobile IPv6

LMA Local Mobility Anchor

MAG Mobile Access Gateway

pMAG previous MAG

nMAG new MAG

MN Mobile Node

CN Corresponding Node

PPDR Public Protection and Disaster Relief

MIH Media Independent Handover

VoIP Voice over IP

LTE Long Term Evolution

MLDv2 Multicast Listener Discovery Version 2

Chapter 1

Introduction

There is an increasing demand within the Public Protection and Disaster Relief (PPDR) community for high bandwidth mobile connections for services like streaming video and audio. PPDR systems are communication systems for public services like the police and fire department. Requirements for PPDR systems are different from the requirements for consumer systems in that they have to be highly resilient and provide strong security. They differ from the more well-known mobile communications systems such as 3G and Long Term Evolution (LTE) by providing group call functionalities and operating (in a reduced manner) without support from a base station. While consumer demand for high bandwidth is driver by services like Youtube, Netflix and Spotify the demand in the PPDR community is based on live video from emergency sites and voice communication with groups of other users.

Current PPDR implementations used in Europe are mostly based on the TETRA system. While the system provides reliable voice communication among users there is very limited support for applications that require high bandwidth such as video streaming. In contrast to mobile networks used by consumers which are increasingly, and in the case of LTE completely Internet Protocol (IP) based, TETRA uses a circuit switching mechanism closely related to the system used by the original GSM standard.

For the future there is a desire from the PPDR community for more bandwidth and more services [2]. The logical step is to switch to newer technologies such as LTE and use existing infrastructure in the form of IEEE 802.11 networks (WiFi) in buildings. To allow data services on the network such as video and image sharing a switch to a fully IP based network is also useful as this allows better resilience due to packet switching, easier development of services and applications. An other large benefit of a switch to IP is the general trend in mobile network technologies to switch to IP

leading to cost benefits for purchasing equipment. Adopting existing mobile platforms for PPDR use also increases the availability in areas where it might not be efficient to deploy a private PPDR platform.

The main problems for PPDR users on consumer networks is that there are no guarantees that the system will always function. PPDR users require maximum reliability to do their work. On the mobility side the main problems in current mobile networks are that seamless handovers between different technologies like WiFi and LTE are difficult and lead to packet loss. Packetloss is problematic as it can lead to users missing important data, for example critical frames in a live video of a robbery. Group call functionality is also a problem in the current IP based wireless networks because there is no or very limited support for multicast traffic. The lack of multicast support in wireless networks requires each user to send data separately to each other user in the group call, leading to inefficiencies.

1.1 Motivation

Seamless switching between different wireless networks can greatly improve the user experience and functionality of mobile devices. In the specific use-case of PPDR networks this can for example allow a police officer to switch from the IEEE 802.11 network at the police station to a 4G mobile network without losing any IP connections. The result is that services such as Voice over IP (VoIP) and video conferencing will not disconnect leading to a seamless experience for the user.

Users of PPDR systems would like to be able to roam seamlessly between networks across borders [2]. Seamless handover in IP networks can be achieved using Mobile IP (MIP) technologies with fast handovers. Traditional MIP requires participation of the Mobile Node (MN) in the mobility process, this is not desirable as this is not standard functionality and would require non standard devices. In Proxy Mobile IP (PMIP) the network is responsible for the mobility management allowing unmodified MNs to roam with IP mobility.

PPDR work is characterized by working in groups leading to a requirement for group communication on the network [2]. In traditional mobile networks this is problematic as a separate connection needs to be set up from some central point to all users in the group call. In IP based networks multicast functionality allows data to flow from a sender to multiple receivers, bypassing the need for a central server that connects all participants in group communication. Multicasting data in mobile networks can also greatly improve the network performance as devices receiving the same data will only cause the network to sent the data once instead of one time per device.

1.2 Problem statement

Mobility in IP networks can be achieved with the PMIP standard [8]. There are also extensions to the standard such as Proxy Fast Mobile IPv6 (PFMIPv6) that allow seamless handovers between two different networks (technology independent) [36]. Other extensions cover multicast support for these network architectures through the use of multicast proxies at the PMIP network devices.

Currently there are no implementations of the PFMIPv6 protocol or Proxy Mobile IPv6 (PMIPv6) with multicast support integrated available for testing. Supporting seamless handovers for multicast flows puts special demands on the handover processes, as these flows are typically destined to multiple users at the same time. The goal of this thesis is to provide fully functioning multicast support in a system for seamless handovers for IEEE 802.11 networks.

1.3 Research questions

The lack of available PFMIPv6 implementations with mutlicast support leads to the following main research question for this thesis:

How can we design and implement reliable and efficient multicasting with IEEE 802.11 and PMIPv6?

This question will be answered with the help of several sub questions:

- Can multicasting be efficiently integrated with PMIPv6?
- Can multicast handovers be optimized to allow minimal packet loss?
- Can we efficiently store multicast data during handovers?
- Can the load on the wireless interface be minimized during multicast handovers?
- How should multicast traffic be secured in PMIPv6 networks?

1.4 Approach

To start answering the research questions a literature study was done on the available work done on Proxy (Fast handover) mobile IP, multicasting and

security. This will give a basis for identifying problems and designing the implementation. A design for a future IP based PPDR network that supports seamless multicast handovers was made. An implementation of the PFMIPv6 protocol was made that supports multicast listeners and fast handovers for them. Finally the performance of the implementations was evaluated by deploying it with actual IEEE 802.11 interfaces and connecting a Mobile Node to do measurements during a handover.

1.5 Contribution

(i) The main contribution of this work is the implementation of the PFMIPv6 software with multicast proxies and fast multicast handovers. (ii) The implementation is evaluated on performance aspects based on measurements taken in a test environment. (iii) The thesis contains a design for a PFMIPv6 network architecture tailored for use in PPDR networks. (iv) The current state of security of multicast traffic in mobile networks is evaluated with a focus on specifics related to the PPDR scenario.

1.6 Outline

The following chapter will provide background information on the technologies related to PFMIPv6, multicasting and security. Its parts can be safely skipped by anyone who feels confident in their knowledge on these areas and related work. Chapter 3 describes a design for a future PPDR networks. Chapter 4 describes the timing problems surrounding multicast fast handovers and a solution to this problem. Chapter 5 describes the actual implementation of the software. In Chapter 6 the implementation is validated by measurements in real wireless environment and finally we draw conclusion in the final chapter.

Chapter 2

Background and Related Work

This chapter will explain the technologies used and built upon for this project. The first section will talk about PPDR networks and the special conditions that apply to them. The next section will explain PFMIPv6, followed by a section on Media Independent Handover. IP Multicasting is the subject of the following section.

2.1 PPDR networks

While the research into seamless multicast handovers for PMIPv6 networks is not directly related to PPDR networks this research was done with PPDR networks in mind. Decisions on the design and implementation of the PFMIPv6 system are made to fully support PPDR users and their specific requirements.

PPDR networks are systems mostly used by police, fire fighter and ambulance personnel. Their main purpose is to offer reliable communications. Losing communication in a crisis situation can lead to dangerous situations and in the Netherlands prompted some fire departments to replace part of the C2000 system (based on TETRA) with analog equipment [1].

Switching PPDR networks from old TETRA based to newer fully IP based technologies will allow emergency services a large amount of bandwidth for data intensive services such as video calls and the viewing of video surveillance on mobile devices. In the PPDR context it is important that data should not be interrupted when the mobile device switches attachment point (for example roaming between a 802.11 and a LTE network). Where for everyday situations a few lost video frames are not important in a crisis even a few lost video frames could lead emergency personnel to miss important information.

PPDR users have different requirements than other mobile users. This section will describe the application requirements as taken from a recent

survey of PPDR users in Europe [23]. The requirements relevant to the network were taken and expanded upon.

2.1.1 Voice communication

There are three different types of voice communications that users would like to have available (and are in current systems): Groups calls, One-to-one calls and Priority voice communications.

Group calls

Group calls allow users in a specific group to talk to all other users in that group at once. Users expect the following functionality:

- Drop in, drop out functionality: Users can join and leave groups at any time without affecting the communication between the other group members.
- Dynamic reassignment to new / other groups: Users can be remotely assigned to other groups (including new groups).
- Fast call set-up: Groups can be set-up and usable in a short time period (several seconds, comparable to a normal telephone call).
- Priority speakers: It should be possible to give priority to specific users. When these users are talking, other users in the group cannot interrupt them. Their voice overrides other voices.
- Only one person can speak at a time: It should be possible to limit the number of users that can speak at the same time. This will prevent chaos in the group.

One-to-one calls

Voice calls between two users. Users expect the following functionality:

- In system calls: Calls between two users of the system.
- Telephone calls: Support for calls to the PSTN.

Priority voice communications

It should be possible to make priority voice calls within the system. The following requirements have been identified:

- Instantly be hearable on other nearby devices (push to talk functionality).
- An open microphone system when a user toggles the emergency option: User will be hearable on speakers of nearby devices and in the command and control centre.
- Drop existing communications from the network to free bandwidth for the emergency call.

2.1.2 Video communication

Video communications are important in the new network as it will enable emergency services to share live images of what is happening. The following requirements have been identified:

- Send and receive video within a group.
- Send and receive video between two users (one-to-one communications).
- Drop in, drop out functionality. Users can join and leave the video group at any time without affecting the video stream for other users.
- Dynamic reassignment. Users can be assigned to a group and reassigned to another group on the fly.
- Priority within a group. Users can have priority within a group. when this user is broadcasting others have to view this stream. Other streams in the group are dropped in favour of this stream when there are bandwidth limitations.

2.1.3 Data access

Users require data access (this seems to be mostly IP based) on their devices.

- It should be possible to access organization specific databases, for example police databases.
- Another requirement is to be able to work on the go like in the office (VPN to office).

- Internet access should be possible.
- Instant messaging within the network, picture messaging and other data applications.

2.1.4 Air to ground communication

Communications between users in an aircraft and users on the ground are possible.

2.1.5 Interoperability

Communications between users using different technologies or networks are possible. For example: Users on an 802.11 network should be able to communicate with users on an LTE network for example.

2.1.6 Support for multiple networks and mobility

Devices should be able to connect to multiple technologies. Users should also not notice any communications problems while moving from one cell to another or when handing over to a different access technology.

2.1.7 Scalability

The network should be scalable. It should be easily extendible with more access points or new access technologies. Sudden extensions might me needed in case of emergency. An example would be an emergency in a remote location with limit coverage that suddenly sees an influx of emergency equipment users.

2.1.8 Security

The network should be secure. Unauthorised terminals should not be granted access to the network. Communication on the network should also be encrypted. Authorized terminals should not be able to intercept data that is not meant for them.

2.1.9 Performance and quality

The network should be able to guarantee a connection between to terminals connected to it. If a part of the network where to fail the remaining part should

keep functioning. Disconnected terminals should also be able to continue operation with neighbouring terminals without the network infrastructure.

2.2 The Internet Protocol

The Internet Protocol is used to route information in the internet. It is required to transmit a data packet from one device to another. Any device connected to the internet requires an unique IP address. When a system on the internet wants to send data to another system this data is put inside an IP packet and send on its way. Routers where the packet arrive can determine the route the packet needs to take based on the destination IP address it has.

Currently most addresses are in the IPv4 format. An IPv4 address is a 32 bit address and thus has a total of 4.294.967.296 addresses available [27]. Currently there is a shortage of these addresses as the amount of devices connected to the internet keeps increasing on a yearly basis [29]. This is a huge problem as once the IPv4 address space is exhausted no new devices can be added to the IPv4 internet.

A replacement addressing method exists to solve the address shortage called IPv6. IPv6 addresses are 128 bits long [4], resulting in a total of 2^128 addresses, which is a number consisting of 39 digits. This new addressing format should effectively solve the address shortage problem. IPv6 also contains improvements to multicast functionality (See section 2.5 for more information). Because of the available address space and multicast improvements IPv6 is the logical choice for a future PPDR network.

2.3 PFMIPv6

Fast Proxy Mobile IPv6 (FPMIPv6) is an IP mobility technology. It allows mobile devices to move through a network and switch attachment point without the device noticing this on the IP layer. To explain the way this system works we will start with explaining basic Mobile IP and build up to the more complex systems. It is interesting to note that Proxy Mobile IP is also an optional part of the 3GPP LTE specification, in which it replaces the GPRS Tunnelling Protocol (GTP) when used.

2.3.1 MIP

MIP works by having a fixed point in the (mobile) network that acts as a mobility anchor for mobile devices. In MIP this entity is called the Home Agent and it will tunnel packets for a MN to the network that node is

currently on. When a Mobile Node enters another network a router local to that network called the Foreign Agent will be the endpoint for the data tunnel and send the packets on to the mobile node. For packets coming from the mobile node the FA will act as default router [25].

In MIP the MN knows about the mobility and actively helps maintain it. This is not desirable in most circumstances as it is more useful to be able to use any device in a mobile context without a special IP stack. For this reason PMIPv6 was developed. PMIPv6 is a technology that allows devices to keep the same IP address while moving through a so called PMIP domain. The main benefit of PMIP is that the mobile device has no knowledge of ip mobility, all mobility details are handled in the network. This technology can also help with faster handovers and less packet loss during handovers by buffering packets in the next access point during a handover.

2.3.2 PMIPv6

PMIPv6 works by adding a Local Mobility Anchor (LMA) to the provider core network [8]. This LMA is the anchor point for the MNs in the domain (All IP routes for the MNs lead to the LMA). The LMA tunnels the data for the MN to the Mobile Access Gateway (MAG) that serves the MNs point of attachment. When a handover is performed the serving MAG lets the LMA know the MN has disconnected. The new MAG (nMAG) lets the LMA know the MN has connected to it and a tunnel is set up between the LMA and the nMAG. Because the nMAG will have the same link layer address as seen from the node it does not know that is it actually on a different link and keeps the same IP address (If the MN requests a new IP address it would also receive the same one it had before).

2.3.3 Fast handovers

Proxy mobile IP does not reduce the handover times by itself. An extension called "Fast Handovers for Proxy Mobile IPv6" (PFMIPv6) enables faster handovers by changing the handover procedure in the MAG (The LMA is not changed) [36]. This protocol covers two handover situations:

1. Predictive handover.
2. Reactive handover.

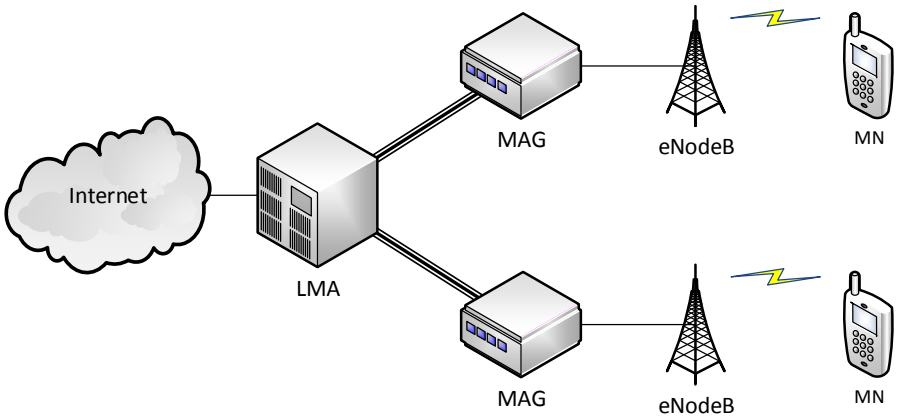


Figure 2.1: Proxy Mobile IPv6 architecture

Predictive Handover

With a predictive handover the network predicts the movement of the MN. The MN needs to report link layer information to the network for this, possible by using Media Independent Handover (MIH) (See Section 2.4). The network then initiates a handover when it thinks the MN is moving towards a new network with a nMAG. The previous MAG (pMAG) sends a Handover Initiation message to the nMAG. A tunnel is set up between the pMAG and nMAG to send data for the MN that arrives at the pMAG to the nMAG. The nMAG buffers this data until the MN is connected. The pMAG now orders the MN to perform the handover. Once the MN connects to the nMAG the data that was buffered is send to the MN. Data from the MN that is received at the nMAG before the LMA is updated is tunnelled back towards the pMAG to be send to the LMA. The nMAG finally updates the LMA with the MNs new location information. This system allows the MN to handover quickly without packet loss as these are all buffered during the time the MN is disconnected from the network. Handover delays with predictive PFMIPv6 are between 200 ms and 1 second depending on the amount of transmission errors that occur during handover [16].

Reactive Handover

In the reactive handover scenario the MN performs a handover without the network being aware of this beforehand. This situation can occur when the signal strength of the access point the MN is currently using suddenly drops. When the MN connects to the nMAG the nMAG will send a Handover Initiation message to the pMAG (It is left up to the implementation to decide

how the pMAG learns about the nMAG). The tunnel can then be set up to allow messages still arriving at the pMAG to be forwarded to the nMAG. The nMAG then updates the LMA and the tunnel is discarded. Handover delays for a reactive handover take between 1.8 and 2.2 seconds depending on the amount of transmission errors during handover [16].

2.4 Media Independent Handover

Media Independent Handover (MIH), also known as IEEE 802.21, is a technology that can help perform vertical handovers (handovers between different technologies like LTE and IEEE 802.11). MIH does not actually perform the handover but rather facilitates communication related to the handover between different devices.

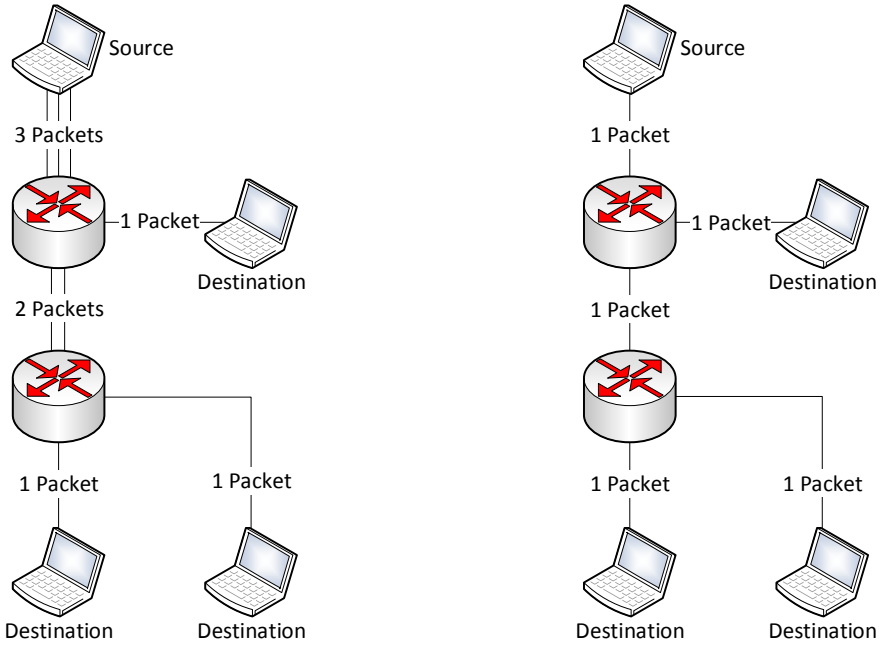
Communications are handles by what is called the Media Independent Handover Function (MIHF). This MIHF can communicate with the MIHF of other devices. The MIHF is located between layer 2 and 3. It communicates with the lower layers using so called Service Access Points (MIH_LINK_SAP). This SAP maps certain functions of the lower layers to general functions the MIHF understands. This method allows the MIHF to function regardless of what underlying layers there are, as long as an appropriate LINK_SAP exists. MIH entities in higher layers are called MIH users. Users are pieces of software that handle the handover data. For example: A user can keep track of neighbouring networks (frequency they are on, network type, maximum bandwidth etc...), this information can greatly help in making handover decisions. there could even be a user that waits for handover commands from the network and initiates a handover based on that.

Information about surrounding networks can be send from the Media Independent Information Service (MIIS). This entity collects data such as available bandwidth, operating frequency and access technology from available networks and transmits them to terminals near those networks.

While not directly used in the software at the moment it is import to note the existence of this technology as it the most likely candidate for a signalling system to make predictive handovers work in a PFMIPv6 network.

2.5 IP Multicast

Multicasting in IP networks is the transmission of data from one or more sources to one or more destinations. In normal IP unicast this would be achieved by having a connection from each source to each destination. With



(a) Unicast (b) Multicast

Figure 2.2: Unicast traffic vs multicast traffic

multicast a single IP packet destined for N sources only has to be transmitted once, instead of N times. Multicast routers duplicate the packet when the destinations are on different links. This leads to greatly reduces bandwidth usage for the network, especially in situation were there are a large number of destinations (Figure 2.2).

Multicast addressing works a bit different from normal IP addressing. Multicast packets have a multicast address in their destination field. This multicast address is in the reserved 224.0.0.0/4 range for IPv4 [9] and the ff00::/8 range for IPv6 [10]. These addresses are not used for end hosts. A system that wants to receive multicast traffic will indicate to a local router it would like to receive multicast packets from address x. The router will then know that it has a destination for address x on the link to that host. There are several routing protocols that enable this information to travel efficiently through the IP network, an example is PIM (Protocol Independent Multicast). Explaining how these protocols work is beyond the scope of this paper.

Challenges in multicasting exist mainly in the routing aspect. Routers need to keep track of which of its links require multicast traffic from a specific multicast address and it is difficult to ensure a multicast address is only used

for one purpose. Because of these problems multicast traffic is currently not routed over the global Internet, but is mostly used internally in networks for applications like video distribution (in IP-television systems for example).

Multicasting using radio technology instead of a wired communications is easy in principle but difficult in practice. When a packet is transmitted over the air any device in range of the transmission point can receive this packet. Sadly modern broadband wireless systems such as UMTS and LTE do not support multicast traffic in this way, but only provide point to point data links between a device and the base station.

2.6 IETF Multicast PMIP drafts

Implementing multicast functionality in Proxy Mobile IP networks is not straightforward. Simply enabling multicast routing on the MAGs and LMA would not lead to the desired results as multicast traffic would be routed over the normal network instead of the MAG-LMA tunnels. To solve this problem the IETF multicast mobility working group has released several experimental status RFCs regarding this issue. RFC 6224 [30] explains the deployment of MLD proxies at the MAG and LMA. These proxies query and listen for incoming MLD reports from mobile nodes. These reports are forwarded from MAG to the LMA, which in turn can subscribe to the multicast traffic on the greater network it is connected to. Multicast traffic is forwarded to the MN in a reverse fashion. When it arrives at the LMA it is forwarded to the MAGs which have subscribed MNs. The MAGs in turn forward it the the interfaces with these MNs.

For fast handovers RFC 7411 [15] contains an extra option for the Handover Initiation and Handover Acknowledgement messages. This option contains all the multicast groups the MN is subscribed to. The nMAG can already subscribe to these groups while the handover is happening. As with unicast traffic it is also possible to buffer multicast traffic being tunnelled from the pMAG, this prevents packetloss during handovers. The operation is similar to unicast handover apart from already subscribing.

2.7 IEEE 802.11 characteristics

IEEE 802.11 (WiFi) is a wireless network technology widely supported in most mobile devices today. Multicast is supported using the broadcast functionality in which any connected node is able to receive the broadcast data.

Newer standards such as 802.11n support power saving modes in which

broadcast traffic is buffered to be transmitted on fixed intervals such as every 100 or 200 ms [11]. This allows connected nodes that are idle to only wake up for those specific times, saving power. The result is an added delay to multicast traffic.

Encryption is an important part of any mobile network as we do not want our communication to be visible to third parties, this is even more true in the PPDR context. IEEE 802.11 has relatively strong encryption with the wpa2 standard. While in the case of wpa2-personal there is still a possibility of data being decrypted if the password is known by an attacker, this is not possible with wpa2-enterpise encryption. Broadcast traffic is however always encrypted with the shared group key [5], that any node connected to an access point knows. Multicast traffic is as a consequence not secure from eavesdropping from devices connected to the same access point.

2.8 Optimistic Duplicate Address Detection

Duplicate address detection is a technology that allows a device to keep listing on the same IP address while switching networks. It configures a device with a temporary IP address that is used to send traffic on. Incoming data for the old address is still received as normally [22]. The device will switch to using the old address once is has confirmed this can be used on the new network. This method saves time as establishing the old address can be used can take up to several seconds.

In situation in which the IP address allocation over several networks is controlled and known by the network having Optimistic DAD enabled allows us to send traffic to a device the moment it attaches to our network without worrying if we know the new IP address.

2.9 Related Work

As mentioned before the IETF has a working group on multicast mobility [12]. Several informational and experimental RFCs have been released that deal with multicast handovers. None however deal with the problem of synchronizing multicast flows after the handover. There are also no reference implementations available of multicast fast handover systems.

One of the members of the IETF multicast mobility working group has published a multicast proxy implementation that can be used in PMIP networks [31]. The system is mainly build as a demonstration for peering between proxies for improving efficiency in situation were the MNs are also

multicast senders. Enabling better support for multicast senders is not covered in this thesis as the focus is multicast handovers. The application from [31] does however not integrate with fast handovers as is the goal in this thesis. Integration with the PFMIPv6 system is vital as multicast subscriptions need to be transferred to the nMAG.

There is a large amount of work available on optimizing handovers for MIP standardised in [26]. The main downside of MIP is that is requires a modified MN that needs to communicate with the network for mobility management. In [21] the performance of fast handovers in MIP enabled IEEE 802.11 networks is evaluated. It is shown that handover times can still be several seconds with up to 10 seconds in busy networks.

The authors of [14] show that PMIP provides significant improvements over MIP in that it improves handover performance, is link layer agnostic and reduces the usage of wireless resources. While the paper does not look specifically at multicast all these improvements are also relevant to multicast traffic.

Work has been done on PMIP handovers that lead to the RFC 5949 standard for Fast Handovers for Proxy Mobile IPv6 [36]. This standard describes the handover procedure and how messages between the PFMIPv6 entities should be formatted. This standard will be used to implement fast handovers.

To make a handover appear seamless to a user it is very important that handovers occur as fast as possible. To reach this goal the handover delay needs to be minimized. In [19] it is shown through analytical modelling that using MIH can greatly improve the handover delay of PMIPv6 networks. The paper does however not cover fast handover situations with multicasting. It is shown in [17] that multicast extensions for PFMIPv6 will greatly reduce handover delay for multicast traffic, but the authors did not test an actual implementation and all results were obtained analytically. In [28] it is shown that packets reordering during the buffer flush at the nMAG can be prevented by tagging the last packet sent from the LMA to the pMAG. While this method is shown to work well for unicast traffic, there is no potential end to multicast traffic streaming towards the pMAG, making this method unsuitable for multicast handovers.

Chapter 3

Future PPDR Network Architecture

This chapter describes an architecture for a IPv6 based PPDR network including IEEE 802.11 and LTE networks. These networks can be both private (operated by the PPDR provider) or public (operated by a telecommunications provider).

3.1 IP Mobility

A PPDR user needs to be reachable at all times preferably on a fixed address so they can be reached easily. PMIPv6 enabled networks allow a device to keep using the same IP address on all networks (be it fixed or wireless). When a MN switches from one network to another the PFMIPv6 system makes sure the MN is configured wit the same acip address.

To enable this functionality the PPDR network needs to be equipped with at least one LMA. Any network connected to the PPDR network would need at least one MAG to provide mobility to its connected devices.

3.2 Group communications

As described in previous chapters group communications is an important part of PPDR networks. Communicating in a one-to-many or many-to-many style is possible in IPv6 networks using multicast packets. Supporting multicast in a PMIPv6 environment is possible by deploying multicast proxies on the MAGs and LMAs. The LMA can also act as multicast router or simply be connected to a multicast capable router on its upstream interface.

3.3 The need for fast handovers

Fast handovers are crucial in a PPDR context. When a users device switches from one network to another the IP packets being sent to the device need to take a different route. Without fast handovers the packets already travelling to the old MN position before it has reattached at a new location are lost. With fast handovers these packets are transmitted to the new point of attachment from the old, resulting in no packetloss caused by the handover. Packets that the device wants to sent have to be buffered on the device during the handover, and sent to the network when the handover is completed.

Most common application for multicast traffic like streaming video are relatively resilient to packet loss. The applications envisioned for PPDR networks however can benefit from lossless data traffic as a few lost frames can for example lead a police officer to miss vital information.

3.4 Core Network

The PPDR core network connects all the access networks such as a IEEE 802.11 or LTE network together. The core should be largely IP based as this will be the main access method to the network due to its general availability and built in resilience to connection failure. PFMIPv6 could be used to provide mobility and fast handover in access network connected to the PPDR core.

Radio access devices in the network such as eNodeBs and IEEE 802.11 APs could be placed behind PFMIPv6 MAGs. LMAs in the network could set up tunnels to the MAGs when needed (Triple black lines in Figure 3.1). All MAGs implement the multicast proxy functionality for efficient and low delay multicasting. When multicast traffic is flowing between MAGs there will be dedicated tunnels between the proxy instances on each MAG (Triple blue lines in Figure 3.1), allowing the traffic to skip the LMAs completely causing less delay in communications and less possibility of failure. Each LMA should be connected to a multicast router to allow multicast traffic between LMAs and other locations in the network like the control center. Special MAGs will be needed that connect the network to external networks. Examples are a non trusted cellular provider and the global Internet.

Because the network should be highly resilient to failures it is important there be no single point of failure. Providing multiple MAGs per access network would be a good way to prevent problems on a single line or device from leading to network failure. There should also be multiple LMAs in the network that can take over MNs from a failed LMA in the event of failure.

These duplicate devices are not shown in Figure 3.1.

3.5 Radio Access

The main access technology could be LTE Advanced and IEEE 802.11. User requirements show that there is a large bandwidth requirement for applications like video streaming. This requirement forces the use of LTE Advanced and 802.11 for the radio access as these technologies provide sufficient bandwidth for the proposed applications.

Multicasting over the LTE air interface will have to be handled by the evolved Multimedia Broadcast Multicast Service (eMBMS). This system will have to be integrated with the PFMIPv6 multicast solution. One possible way to do this is give each MAG its own eMBMS gateway. Benefits of making the solution MAG centric is that this allows the use of Single Frequency Network functionality that allows multiple LTE base stations to send broadcast traffic on a shared frequency to increase coverage. It also helps IEEE 802.11 networks to be seamlessly integrated into the multicast system.

The use of LTE femtocells can be considered in stead of IEEE 802.11 APs in some locations. Femtocells are small base stations with limited range (on the order of tens of meters) that are usually meant for indoor use. They allow communication without using the normal base station, thus greatly unloading the radio network. An added benefit is that vertical handovers are not needed when going outside of the building.

3.6 Public Networks

The future PPDR system terminals could also use public cellular networks and IEEE 802.11 access points to connect to the PPDR core network. This allows terminals to connect when out of range of the private infrastructure or in buildings that block cellular signals.

Tunnels could be made from a special MAG device to the terminal on the public network (Triple red lines in Figure 3.1). While the use of tunnels to the device will increase overhead it is the only way to guarantee that a terminal will receive an IP address in the PPDR network range and that security of its communication is ensured.

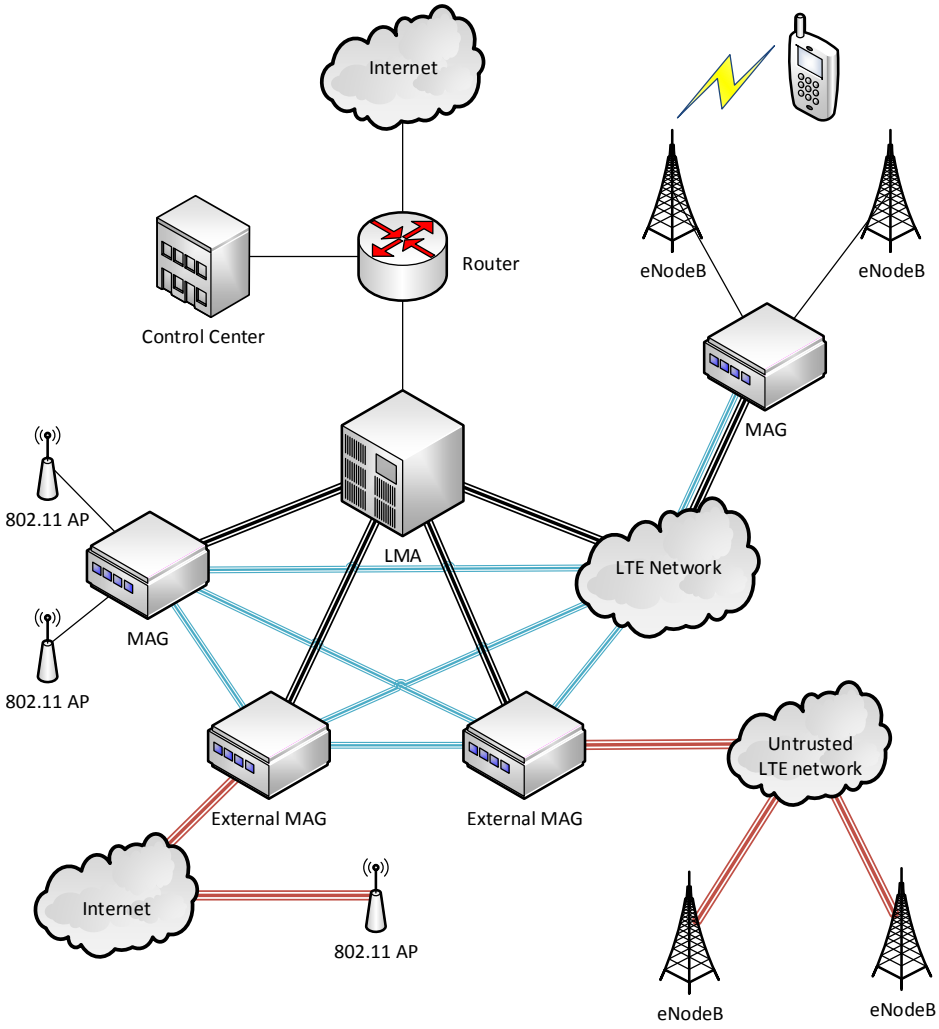


Figure 3.1: Overview of the proposed PPDR network

3.7 Protocol overview

Different devices in the network communicate with each other using several protocols. Figure 3.2 shows the devices important for PFMIPv6 and which protocols they use for communication.

To perform seamless handovers using the *predictive handover* mechanism the pMAG needs to be aware of two pieces of information:

1. That a MN is going to perform a handover,

2. The MAG that is responsible for the target access network.

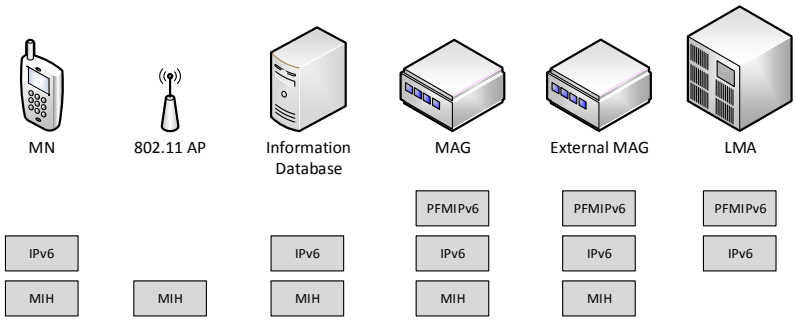


Figure 3.2: Protocol overview

How this information is obtained and signalled towards the PMAG is left to the PFMIPv6 implementation. There is also the possibility that the MN might need a trigger to perform the handover once PFMIPv6 has prepared the fast handover. The need for this functionality depends on the network and mobile device.

For IP based networks MIH is perfectly positioned to perform this signalling role. MIH can provide information to base the handover on and also signal the handover to the relevant PFMIPv6 devices. A MN or network device could signal the PMAG that the MN is about to perform a handover. The message would preferably also contain the IP address of the destination network MAG (the NMAG in this case). MIH could provide this mapping (Target network identifier - MAG IP address) as the system already contains information services for MIH devices.

Figure 3.3 shows a situation in which a MN wants to switch to a different 802.11 AP which is on a different MAG. The handover could also be initiated from the network in which case the MN can be replaced with a network device. The message flow is an example of how this could work using MIH. The system that wants to initiate the handover queries an information database for the IP address of the MAG that serves the target access network (1) and gets a response (2). The handover initiating system then signals the current MAG of the MN with the handover command containing the IP address of the NMAG (3). This message is where MIH and PFMIPv6 meet. The MIH function on the MAG will now give PFMIPv6 the command to prepare the handover (4). Both MAGs now prepares the handover using normal PFMIPv6 signalling (5).

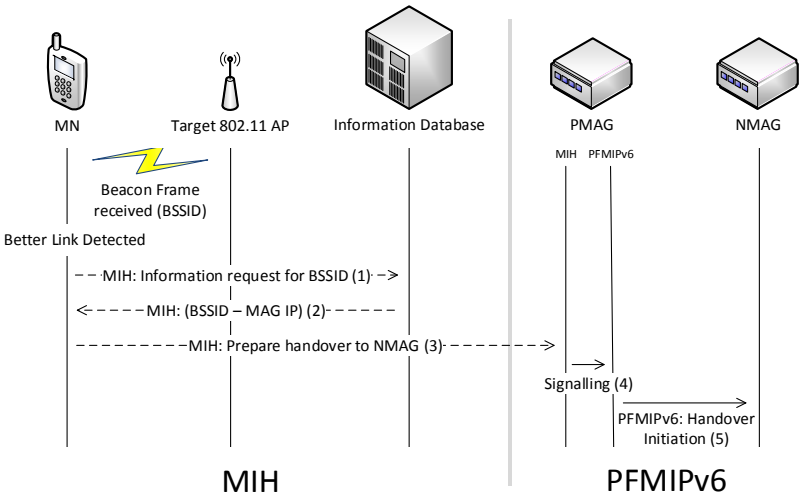


Figure 3.3: Example MIH signalling for a PFMIPv6 handover

3.8 Security

Secure communications are important in PPDR networks. Security within the PPDR core network can be provided by encrypting the tunnels between the different PMIP devices (As seen in Figure 3.1). Simply encrypting all tunnels does however not provide complete security as packets will have to leave the PPDR core network at some point. As explained in Section 2.7 IEEE 802.11 security can also not be relied on for secure communications. It is also difficult when using encryption in the network to prevent malicious users already on the network from overhearing sensitive data. To provide full security for PPDR communications the packets themselves will have to be encrypted using end-to-end encryption. End-to-end encryption will only allow devices that take part in a certain group to decrypt data, thus providing secure communications.

Not only the communications on the network but the network itself will need to be secured from outside threats. A form of authentication is needed that prevents users not allowed on the network to be granted access by the MAGs. A central authentication system controlled by the PPDR provider will need to authenticate new users, and authenticate them again when they handover to a different MAG.

3.9 Resilience

Resilience in the network is provided by several methods. The use of a packet switched infrastructure in itself helps, when a link goes down the traffic that was using this link will automatically be sent over a different link (if available). To further enhance the resilience of the network devices such as LMAs and MAGs will need to be redundant. When one of these devices goes down another device will need to be available to take over the task of the first.

Chapter 4

Multicast Handover Timing

This chapter will describe the main problem with different delays on the tunnels during a PFMIPv6 handover with multicast subscriptions that causes packet loss and inefficient use of the wireless link. A method to prevent this problem from occurring is presented after the problem description. Finally a way to measure the time difference between different interfaces during a seamless handover will be presented. The implementation of the system will be described in Chapter 5.

4.1 Multicast delay problem

When performing a handover using PFMIPv6 there are temporarily two interfaces that multicast traffic arrives on at the nMAG. Multicast traffic arrives from the pMAG - nMAG tunnel and the LMA - nMAG tunnel. Because the data on one tunnel takes a different route than traffic from the other tunnel, there can be a difference in arrival time of the same packet on both tunnels. This time difference can lead to problems during the handover. When the MN connects to the nMAG and the traffic buffer is flushed the traffic should continue flowing from the LMA tunnel. When fro example the LMA - nMAG tunnel has less delay than the pMAG - nMAG tunnel this might lead to packet loss as these packets have not yet arrived on the LMA - nMAG tunnel.

Due to the possible differences in delay between the two tunnels there can be 3 different situations at the nMAG:

1. The two tunnels are in sync (travel time is identical for both routes, see Figure 4.1),
2. The LMA - nMAG route is faster than the LMA - pMAG - nMAG route (see Figure 4.2),

3. The LMA - nMAG route is slower than the LMA - pMAG - nMAG route (see Figure 4.3).

The most problematic situation for PPDR applications is the third. Another important aspect of synchronization is not sending the same packet twice over the air interface when not needed. In crisis situations this can save valuable bandwidth that could be used to transmit useful data.

4.1.1 Tunnels in sync

When the tunnels are in sync no special action needs to be taken. The application can simply flush the multicast traffic buffer that has build up during the handover and switch to the data coming from the LMA - nMAG tunnel once it has arrived there (see Figure 4.1). In this figure the numbered boxes represent packets traveling to the nMAG. The green packets coming from the pMAG are replaced by the blue packets coming from the LMA. In this case the switch can occur at any moment because there is no difference in the delay between the two tunnels. While this situation is ideal it is not very

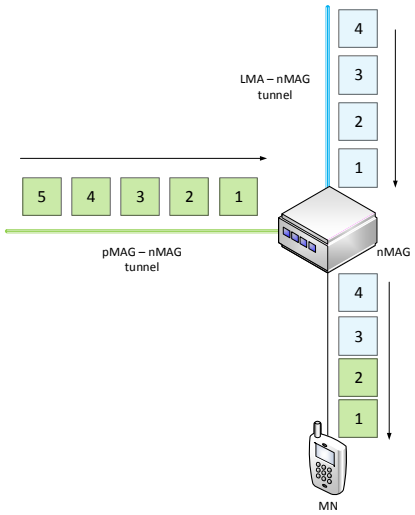


Figure 4.1: Multicast synchronization with no delay difference

likely to occur during normal operations as it requires the delay on the direct LMA-nMAG tunnel to be equal to the delay on the LMA-pMAG-nMAG route.

4.1.2 LMA-nMAG tunnel is faster

In this situation the multicast packets coming through the pMAG - nMAG tunnel arrive behind those from the LMA-nMAG tunnel. The result as can be seen in Figure 4.2 is that the MN is still receiving packets that would have already arrived over the LMA-nMAG tunnel. A useful solution in this scenario is to simply forward the packets when they arrive at the nMAG and stop the pMAG-nMAG tunnel forwarding once all the data that was still missing from the LMA - nMAG tunnel has arrived. The duration that data still needs to be forwarded from the pMAG - nMAG tunnel is the delay difference between the two tunnels. The applications on the MN should be able to handle the out of order delivery of packets (handover performance should be fast enough so the application buffer can handle this). See Figure 4.2 for an example of this switching procedure in which the green packets are 'mixed' with the blue packets until all packets that were missing on the LMA-nMAG tunnel have been transmitted to the MN.

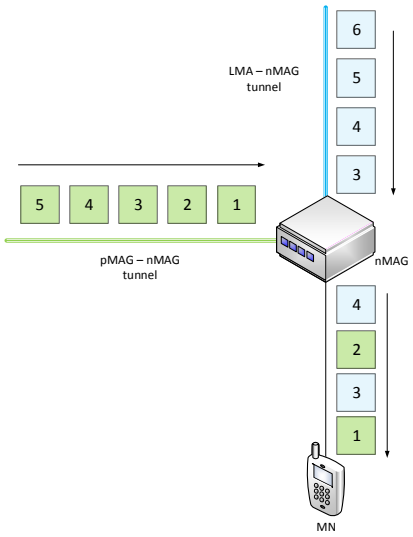


Figure 4.2: Multicast synchronization with a faster LMA - nMAG link

4.1.3 pMAG-nMAG tunnel is faster

In this situation the new LMA - nMAG tunnel contains older data than the LMA - pMAG - nMAG tunnel. It is possible to simply send all the packets from the LMA - nMAG tunnel onto the network, but this would waste wireless bandwidth. A solution in which the traffic is simply not forwarded until the LMA - nMAG tunnel is caught up with the LMA - pMAG - nMAG route

is most efficient in terms of network resources. This will cause a temporary gap in transmitted data during the switch but this would occur regardless when performing a handover in this situation. See Figure 4.3 for an example packet flow. In this example the green packets (pMAG - nMAG tunnel) are not forwarded once the LMA - nMAG tunnel (blue packets) is established and the blue packets start forwarding once there is new data for the MN on that tunnel.

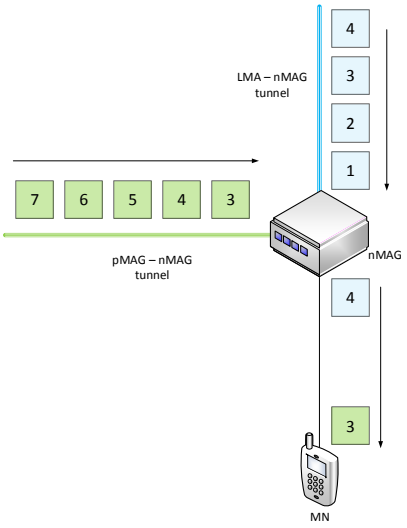


Figure 4.3: Multicast synchronization with a slower LMA - nMAG link

4.2 Delay difference estimation

To make the correct decision on how to perform the multicast handover the nMAG needs to know the difference in arrival time between the two tunnels. Several options to determine the time difference will be considered in this chapter. Traditional methods to determine one way delays are based on precise clock synchronization [3]. The delay estimation problem here is unique in that there is identical traffic arriving on two different interfaces and we are not interested in the delay of the link itself, but in the relative delay between the links.

4.2.1 ICMP echo requests

The most straightforward option to measure the one way delay is based on tunnel round trip times using an ICMP echo request. The time difference

between tunnels could be estimated based on the different round trip times of the tunnels. A problem with this method is that the pMAG would also need some way to reply on the different tunnels to the same IP address. A possible solution is measuring the LMA-pMAG delay time independently and adding it to the echo reply time on the pMAG-nMAG tunnel. The round trip time would need to be halved to get an estimation of the one way delay, but this will never be as accurate as a true one way delay measurement.

4.2.2 Modifying the tunnel

It might be possible to estimate the one way delay per link by adding information to the tunnel headers. An identifier can be added to a tunnelled packet that allows us to calculate a time difference between the two paths at the receiving end. An example protocol would be the GRE tunnelling protocol that has unused reserved space in the header [7] that could be used for such an identifier.

The delay difference between the tunnels can then be calculated by storing the time a certain identifier arrived at an interface and calculating the time difference when the corresponding packet arrives on the different interface.

There are several drawbacks to this option. The main one is that we are not dealing with two but three tunnels. The pMAG would need to copy data from its LMA tunnel headers to the nMAG tunnel headers. This would lead to some overhead and possible delay leading to inaccuracies in determining the time difference between tunnels. The second drawback is that this method requires modification of the tunnelling protocol, which would make it diverge from the standard, and assigning unique identifiers to specific multicast packets which could cause extra overhead.

4.2.3 Timing packets

This method send special timing packets from the LMA over all the LMA - MAG tunnels. The timing packets should be send on a fixed interval that needs to be known to the receiver. A method to transfer this interval is to simply send it in the timing packet. The timing packets are simple UDP datagrams containing a 16 bit sequence number and 32 bit timing interval that are sent using multicast down all tunnels.

When a MN want to perform a handover and the pMAG initiates the pMAG - nMAG tunnel it will also forward the incoming timing packets from the MNs LMA over this tunnel. The nMAG will now receive timing packets on the LMA - nMAG and pMAG - nMAG links. It can use the difference between the received time of a sequence number on each link to calculate

the difference in transmission times between both links. The delay difference between the two tunnels TD in ms is given by:

$$TD = (SeqN_{Tunnel1} - SeqN_{Tunnel2}) \times I - \left(T^{arrival}_{Tunnel1} - T^{arrival}_{Tunnel2}\right)$$

In this formula $SeqN_{tunnel}$ is the sequence number of the last received packet on that tunnel. I is the interval the packets are sent on in ms. $T^{arrival}_{tunnel}$ is the arrival time of the last packet for that tunnel in ms Unix time. Calculating the delay difference in this manner only requires us to store the last received sequence number and arrival time per link, greatly simplifying the system.

Once the nMAG knows the delay difference between the tunnel decisions on how to switch the two multicast tunnels can be made based on the difference between them. The data on the difference is at most I ms old, the accuracy of the data is thus dependant on how often the timing packets are transmitted. In very congestion networks with a high variation in delay the value for I should be reduced to compensate possible sudden changes in the delay.

4.3 Chosen solution

The solution that was chosen is the use of timing packets sent over the different tunnels from the LMA. While modifying the tunnel packets would most likely cause less network overhead the extra logic that would need to be put into the implementation would most likely also cause some overhead. Another downside is that the implementation becomes dependant upon modified tunnels. The timing packets method can use any tunnel technology, making it technology independent. The benefit of this solution over ICMP echo requests is that it does not take a round trip time to measure the delay. The timing packets are always arriving when a tunnel exists. Other active methods to find the one way delay require precise time synchronization between the sender and receiver. The maximum time that is required to measure the transmission delay between two tunnels is the transmission interval I with an average of $\frac{1}{2}I$.

Now that the system has a way to find the delay difference between tunnels it can be used to trigger one of the three scenarios described earlier in this chapter. The delay difference will be calculated per handover when the new MN has arrived at the nMAG so the most recent data is always used.

Chapter 5

Implementation

The implementation of the multicast for PFMIPv6 system was build on top of an existing MIPv6 implementation called UMIP [33] extended by open air interface to also provide PMIPv6 operations [24]. The decision to extend existing software was made to focus on the implementation of fast handovers and multicast support instead of building the entire system from the ground up.

The PFMIPv6 system with multicast support runs on top of a standard Linux operating system. Instructions for installing the system are included in Appendix A.

5.1 Available existing work

As mentioned before the Multicast for PFMIPv6 implementation is based on the UMIP platform as extended by Open Air Interface. This platform provides us with a fully functional PMIPv6 implementation that can perform handovers between different MAGs. As this is a PMIP implementation this handover is performed with packet loss during the time the handover takes place. The software provides all the basic functionality that is needed to set up a PMIPv6 environment:

- Manipulating routing tables,
- setting up tunnels,
- detecting new mobile nodes,
- authenticating mobile nodes.

The authentication is done by verifying the MAC address of the MN with a radius server. This allows any MN with a known MAC address to connect to the PMIPv6 network. This functionality was not further extended as it was decided to focus on implementing fast handovers and lossless multicast handovers. While security is an important aspect it is left to be considered on a later date due to time constraints.

5.2 Design choices

The implementation work was started by extending the existing PMIPv6 platform with fast handover support. This was done in a relatively straightforward manner by extending the existing methods of the software with fast handover functionality, adding new methods were needed.

5.2.1 Fast handovers

The Fast handover functionality was mostly implemented by extending existing C files and methods. The relevant message structures for fast handovers and ways to send, receive and parse them were added to the software. The MAG to MAG tunnels were implemented by using the existing methods to set up tunnels between the MAG and LMA. The MAG to MAG tunnel is build on demand when a Handover initiation request is received at the nMAG and its acknowledgement is received at the pMAG.

The logic for starting and stopping traffic buffers was written from scratch and put in a separate C file for readability. The traffic buffer is build around libpcap [32]. Captured packets are buffered in memory and flushed towards the wireless interface when a Router Solicitation is received from the MN performing the handover. The buffer is implemented in memory for speed. The first attempt was storing the buffer on disk but this proved to cause too much delay in reading and writing. The downside of this implementation is that the memory limit might be reached if a large number of mobile nodes perform handovers to the same MAG at the same time (although this limit does of course depend on the amount of memory installed in the MAG).

Performance considerations

A router acting as MAG for a large number of mobile devices (for example an entire network as described in Chapter 3) would need a large amount of memory to store data for incoming handovers. The actual amount needed depends on the number of devices handing over at the same time and the

amount of data they are currently receiving. A typical core router has a buffer of about 1 round trip time [35], which is estimated at 250ms. Assuming a 20Gbit/s interface on our PPDR router this would result in a $20Gbit/s \times 250ms = 5Gbit$ buffer size, which is 625 megabytes.

To calculate the resulting load of handovers we need to estimate network usage. Let us assume a MN is receiving a video stream of 2 Mbit/s when it want to handover. A PFMIPv6 handover between 3G networks takes between 200ms and 1 second [16]. Observed times between two IEEE 802.11 networks from Section 6.4 suggest between 2 and 3 seconds. Making a conservative estimate of 2 second for a handover this would result in a buffer of $2Mbit/s \times 2s = 4Mbit$. This would allow $5000/4 = 1250MNs$ to perform a handover simultaneously. Even if we increase the average receiving bandwidth this would still allow hundreds of MNs to perform a handover simultaneously. A standard buffer would thus allow a sizeable number of MNs to perform a handover at the same time.

5.2.2 Multicast support

To implement multicast functionality a completely new part was written for the PFMIPv6 software implementing multicast proxy functionality as standardised in RFC 6224 [30]. This allows multicast clients to connect to the PFMIPv6 network and subscribe to multicast streams from their home network. An overview of these extensions can be seen in Figure 5.1.

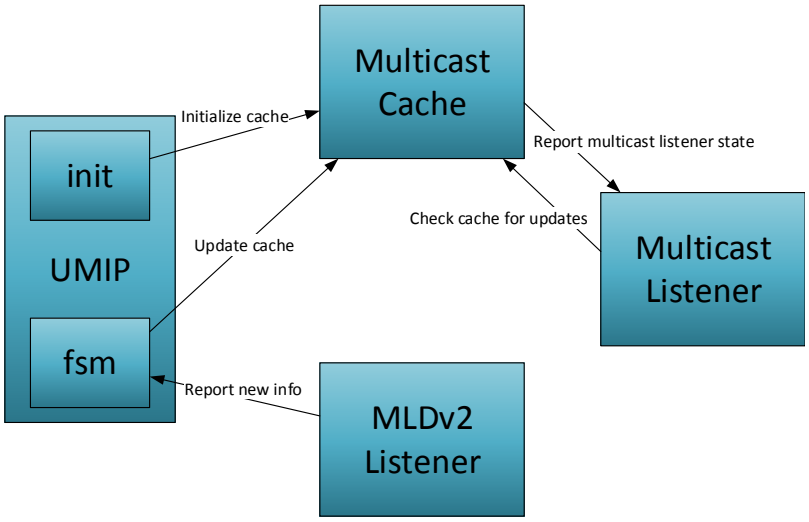


Figure 5.1: Multicast proxy implementation overview

The core of the multicast component is the *Multicast Cache* which contains

information on all the multicast subscriptions a MN (in the case of a MAG) or MAG (in case of a LMA) has on the systems downstream interfaces.

The *MLDv2 listener thread* listens for incoming Multicast Listener Discovery Version 2 (MLDv2) messages [34] on the systems downstream interfaces. When a group membership request is received this thread will update the *Multicast Cache* with the new information (adding of removing membership) and forward the request to the correct upstream interface (a tunnel to the MNs LMA in case of the MAG).

To set up the actual multicast forwarding the *Multicast listener* thread is used. This thread listens on all upstream interfaces (physical of tunnel) for incoming multicast traffic. It then checks if there are downstream devices with subscription for the specific multicast packet and updates the multicast forwarding table accordingly. This construction might seem more complicated than needed but is a result of the way multicast forwarding entries must be passed to the linux kernel. The multicast router will only accept entries as a (source, group) pair. This leads to the need to know the source of a multicast packet for the kernel to be able to forward it. The result is the need for checking all incoming multicast packets to see if we need to make a forwarding entry for it. An attempt at efficiency is made by filtering for multicast UDP packets on the kernel level and only then passing them to the program. The first step in the program is to check if we actually have downstream nodes with subscriptions to this group. If this is the case we check if we need to make or alter a forwarding entry for this packet.

To optimize the handovers for multicast the HI and HACK messages were extended with the multicast option. The nMAG takes these subscriptions and already subscribes to these groups before the new MN has arrived. Once the MN has performed the handover the multicast cache can be updated and the traffic is already incoming, leading to minimal delay in receiving this data.

5.2.3 Tunnel delay estimation

As mentioned in the previous chapter the choice for the method to find the time difference between the different tunnels is sending special packets down them. These packets are sent on a fixed interval currently defined in the configuration file of the LMA. The timer packet contains a 16 bit sequence number and a 32 bit integer with the sending interval in milliseconds.

The *Multicast timer* is a thread that listens for incoming timer packets. The LMA adds all new tunnels to its list of links to send the packets on while the MAGs subscribe to the group the packets are sent to on all new tunnels.

When a handover is performed the pMAG adds an entry to the multicast cache for the MN performing the handover that timer packets should also

be forwarded over the pMAG - nMAG link. This allows the forwarding to be specific for that MN and be automatically deleted once the handover is complete.

When a timer packet is received its sequence number and the time it was received in milliseconds are saved per link. The interval is saved globally as this value is the same for all timer packets coming from the same LMA. At the time of the handover the MN is fully connected to the nMAG and the multicast buffer can be flushed the current difference in transmission time for the tunnels TD is calculated in the following way:

$$TD = (SeqN_{Tunnel1} - SeqN_{Tunnel2}) \times I - \left(T^{arrival}_{Tunnel1} - T^{arrival}_{Tunnel2}\right)$$

In this formula $SeqN_{tunnel}$ is the sequence number of the last received packet on that tunnel. I is the interval the packets are sent on in ms. $T^{arrival}_{tunnel}$ is the arrival time of the last packet for that tunnel in ms Unix time. This allows a calculation of the time difference based on the latest data available to the nMAG. Based on this data the multicast buffer can be flushed immediately or after a certain time has passed. See Section 4.1 for the exact action to be taken based on the time difference.

5.3 Implementation challenges

The main challenge during the implementation was the implementation of the multicast proxy functionality. As the Linux kernel requires forwarding entries to have a (source, group) pair, these entries need to be entered based on incoming multicast traffic. The program must also keep track of all forwarding entries made for each MN so they can be deleted when no longer needed to save bandwidth on the air interface.

Another aspect are the PFMIPv6 handovers themselves. The traffic for a MN performing the handover needs to be forwarded from the pMAG to the nMAG. There is however no signalling from nMAG to pMAG for handover completion in the standard. The pMAG does not know when it can stop forwarding traffic. Normally this would not be a problem as the LMA will not forward traffic to the pMAG once the handover is completed and the forwarding state can be deleted any time after traffic is not being received any more. For multicast however the traffic will only stop arriving once the pMAG notifies the LMA the multicast data is no longer needed, and the forwarding will not stop in this scenario. Currently the solution used is to let the forwarding exists as long as the entry for the MN exists at the pMAG. This entry is automatically deleted when the MN is not responding to Router Solicitation messages for a certain amount of time (currently 30 seconds).

Flushing the traffic buffer to a newly arrived MN can put a large amount of data on the wireless network. When the buffer contains multicast traffic this can greatly affect the throughput of the network as these packets are transmitted on a lower rate in IEEE 802.11. To prevent this problem and also prevent MNs already on the network from possibly receiving a packet twice the buffered multicast packets are not broadcast on the wireless link. This is done by sending the multicast packet within a unicast Ethernet frame. The wireless interface then sends the packet to the MN with all the unicast benefits such as retransmission on loss and higher data rates. This allows a larger throughput of the buffered traffic without possibly impacting other MNs.

5.4 Security

In the current implementation the traffic within the system is not secure. While the security of the PMIP core network might be guaranteed up to a certain point it might be desirable to encrypt the tunnels between the LMA and MAGs. The current implementation uses simple IP-in-IP tunnels, but these can be replaced with a more secure alternative or encrypted with IPsec for example.

Securing traffic on the IEEE 802.11 part of the network is more difficult. While unicast traffic is secure in a wpa2 enterprise encrypted environment, it is possible to decrypt data in a wpa2 personal network (the variant which uses a shared password for authentication) if you can intercept the initial connection request of the MN. Multicast traffic that is broadcast can be decrypted by anyone on the same network as the broadcast key is shared with all connected MNs. The only option is this case is to use application level encryption.

Another problem for all access networks is that anyone on the network can in theory join any multicast group. Again application level security is needed to guarantee security. It can be concluded that application level security is a must for multicast traffic in any situation.

Resilience was not considered while building the prototype. An important improvement to the current system would be the possibility to use multiple LMAs. When one LMA is unreachable the MN should be automatically transferred to a different LMA. The LMA should also be able to transfer the MN to a different MAG on the same network (if available) to compensate for MAG failure.

Chapter 6

Validation

This chapter will present the validation of the PFMIPv6 implementation. The goal is to measure how effective the implemented system is in performing lossless multicast handovers. Packet loss is the most important characteristic here as the handover should be as seamless as possible. In an ideal scenario the MN will not have any packet loss and the system will not forward duplicate multicast packets onto the air interface. Less important for this thesis (as there is a focus on lossless multicast handovers) is the handover duration. These values will be determined by measurements in a PFMIPv6 network.

The first section of this chapter will explain the environment the measurements will take place in. In Section 6.2 the software used during the measurements will be presented. Section 6.3 explains how the measurements will be done followed by the results in Section 6.4. Finally Section 6.5 will discuss and draw conclusions based on the results.

6.1 Measurement network

The validation was done based on measurements taken from a PFMIPv6 network built for this purpose. Figure 6.1 shows the different devices in the network and how they are implemented on physical hardware. The LMA and server are virtual machines running in VMWare on a laptop (Blue boxes in Figure 6.1). The server will act as Corresponding Node (CN) for the MN that sends out multicast packets. The LMA VM and Server VM are connected using a virtual Ethernet device with a 1 gigabit per second link. Both MAGs are also VMs. They are also connected with the LMA VM using a virtual Ethernet network with a 1 gigabit per second network speed. These machines are virtualized as there is no direct benefit to running them on separate machines and it makes the measurement network easier to transport.

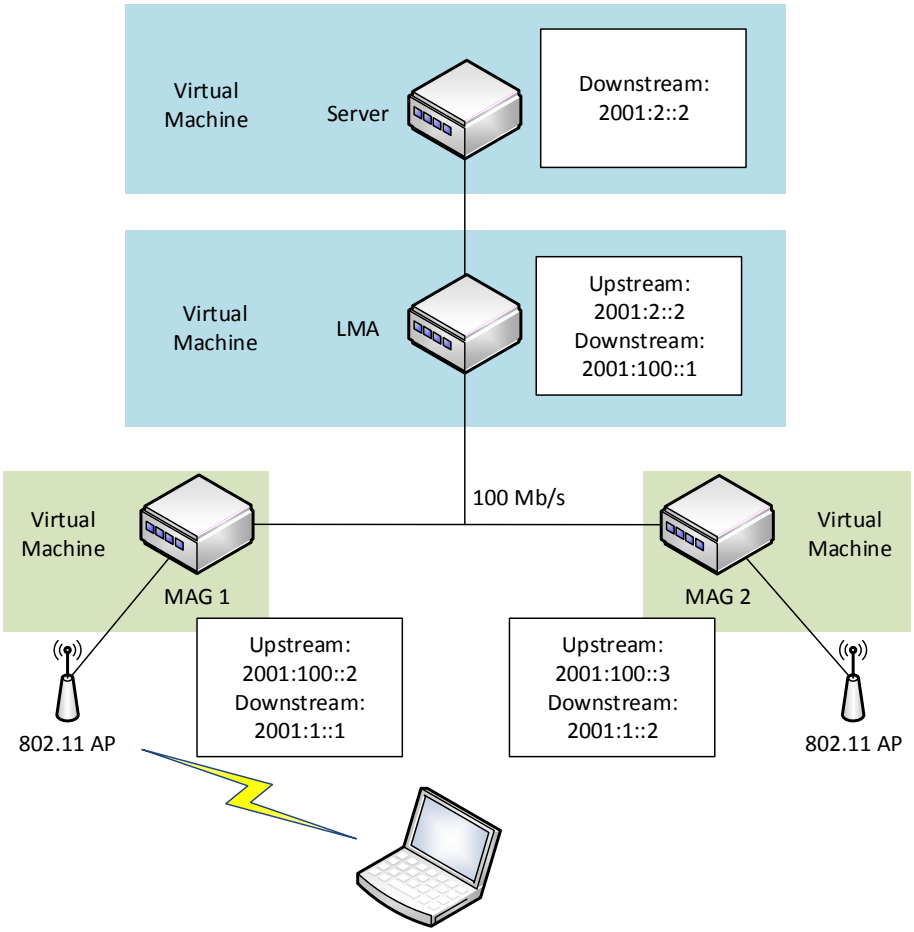


Figure 6.1: PFMIPv6 experiment network

The MAGs are each connected to physical IEEE 802.11 interfaces that are connected to the host computer using USB. The USB devices are shared with the MAG VMs to enable direct access to the IEEE 802.11 hardware.

The MN (A laptop with an IEEE 802.11 interface) will connect to the MAGs through the MAGs wireless interface. This laptop is running a recent Ubuntu Linux distribution with a kernel that was compiled with Optimistic Duplicate Address Detection. While Optimistic DAD decreases the time need for a layer 3 handover significantly (it is around 2 seconds faster due to going online before duplicate address detection is done), there is still a delay of around 2 seconds for the handover on the layer 2 level and a second before the layer 3 connection is fully functioning. It is outside the scope of this thesis to attempt to further reduce this delay.

6.2 Software used

Several software packages are used to perform the measurements required for the validation.

The LMA, MAGs and server will be running the latest testing release of Debian Linux with Linux kernel version 3.16.2. Compiling a kernel with IPv6 multicast forwarding options is not necessary when running this newer version, but is needed in older releases.

Forwarding will be done with the developed PFMIPv6 implementation which is deployed on the LMA and MAGs. The different devices will be configured with the IP addresses as shown in Figure 6.1. Installation instructions for the PFMIPv6 implementation can be found in Appendix A.

The MAGs are equipped with IEEE 802.11n interfaces that are turned into access points using the hostapd [20] software (configuration file is included in Appendix A.2.1). The 802.11 interface was set to 802.11g mode to prevent multicast delays due to power saving options in the newer versions of the standards. As both the PFMIPv6 implementation and hostapd attempt to change interface characteristics it is important to start them in the correct order. The PFMIPv6 implementation should be fully started before hostapd is.

Measurements will be done and results obtained using IPerf [13], a tool for measuring network performance. This tool has the ability to send TCP or UDP traffic from one device to the other and reports statistics like bandwidth, jitter and packet loss. Iperf is run at the CN with the following command: "iperf -c ff08::1:2:3 -V -u –ttl 5 -t 20 -l *packetSize* -b *bandwidth*" with *packetSize* the packet size in bytes up to 1200 bytes (larger packets would cause fragmentation on the tunnel link, affecting the reported results at the MN side) and *bandwidth*

the bandwidth used in bits per second. This lets Iperf sent multicast packets on group ff08::1:2:3 for 20 seconds. Iperf is started on the MN with "iperf -s -V -u -B ff08::1:2:3 -i 1" to capture the incoming packets and output statistics.

The NETEM functionality of the Linux kernel [18] is used to simulate network delays. NETEM is part of the network traffic control system and can be used to add delays and drops to a link. NETEM can add delay with the following command: "tc qdisc add dev *device* root netem delay *delay*ms *variance*ms distribution *distribution*", in which *device* is the name of the network device (or tunnel) the delay needs to be added to, *delay* is the delay in milliseconds, *variance* the variance in milliseconds and *distribution* the distribution used to choose the delay for packets.

6.3 Measurement method

Several experiments were done to measure the effectiveness of the handover system and how it copes with delays on the network. The measurements were done in the network environment described in Section 6.1. This section will describe the method used to perform the measurements.

Measurements are done using UDP multicast traffic flowing from the CN to the MN for 20 seconds. The handover trigger is sent exactly 5 seconds after the CN starts sending multicast traffic. Different packet sizes will be used to measure the effect of packet size on the performance of multicast handovers.

Report gathering was automated using several scripts. These scripts also partially provide the functionality MIH would provide, namely the triggering of the handover on the MN. For the measurements the pMAG controls the process. It gives a command to the CN to start sending multicast packets. It then waits 5 seconds to send a Handover Initiation packet to the nMAG. Once the Handover Acknowledgement is received the MAG sends a signal to the MN to perform the handover before changing the routes for that MN to the pMAG-nMAG tunnel. The MN performs a handover back to the network of the pMAG once all the measurement packets have been received. The script on the pMAG waits for this process to complete before starting a new measurement. For measurements without fast handovers the script on the pMAG sends a handover signal directly to the MN instead of sending a HI message and waiting for a HACK.

In a larger network it would be common te find differences in the one way delay between the connections of the PFMIPv6 devices due to delays in the routing and the distances involved. To simulate these delays on the measurements network delay was added to specific tunnels. For this experiment several delay values will be used with or without an added

variance in the delay. The delay between consecutive packets is distributed normally, it was shown in [6] that a normal distribution approximates real world packet delay.

Generally speaking the measurements can be divided into four groups:

1. Measurements without fast handovers,
2. Measurements with fast handovers and no added delay,
3. Measurements with fast handovers and added delay on the pMAG - nMAG tunnel,
4. Measurements with fast handovers and added delay on the LMA - nMAG tunnel.

The first set of measurements will focus on establishing a baseline by measuring the PMIPv6 handover performance in a situation with no other traffic (except PMIPv6 signalling) on the network. This will allow us to measure multicast handover performance without the fast handover improvements for comparison. The expectation is that all traffic that should have been received during the handover will be lost.

The second measurement set will measure the performance of the system without any delays on the tunnels. This will test if the implementation of the fast handover system provides improvements on the normal PMIPv6 system and establish a baseline for the measurements in a network with delays. The expectation is that all traffic during the handover will be buffered at the nMAG and transmitted to the MN when the handover is complete, resulting in close to 0 packet losses during the handover.

The third measurement set will introduce delay on the tunnel between the pMAG and nMAG. During the handover this delay will cause packets from the LMA - nMAG tunnel to arrive before the same packets from the pMAG - nMAG tunnel. Measurements are done with a delay value of 50, 100, 200 and 300 ms delay. The delay variance used was 0 and 40 ms.

Finally the fourth measurement set will introduce delay on the LMA - nMAG tunnel. This added delay should make the nMAG stop forwarding for the delay duration to compensate for this delay difference and not waste wireless resources. The values for delay and variance used are identical to the values of the third measurement set.

The goal of the measurements is to determine the effect of packet size, packets per second and different delay values on the performance of the implementation. The most important metric here is the packet loss but the jitter value perceived at the MN will also be taken into account.

For each measurement set the packet loss and jitter will be recorded on the MN. Because multicast traffic is transmitted as broadcast packets on IEEE 802.11 networks there is no retransmission mechanism as with unicast packets. The result of this unreliable delivery method is packet loss for multicast packets on the wireless interface. Because this background loss is not caused by the handover process itself the measurement results will have to compensate for this loss. The packet loss measured surrounding the handover will be subtracted from the total packet loss, the result being the total amount of packets lost caused by the handover. This compensation is done by subtracting the packet loss during the first 5 seconds of the measurement to compensate for losses before the handover. Compensating for losses after the handover is done by taking the number of packets lost from seconds 14 until 19 and multiplying them by 3 to compensate for losses that would otherwise have occurred during seconds 5 until 20. In an ideal scenario the packet loss will be close to 0. The jitter value is interesting as it will give an indication of how fast the handover was performed. Ideally the packets would not have been delayed much (the handover time was short). Each measurement will be run at least 10 times for each variable. Once a measurement set is done the average and 95% confidence interval of the set will be calculated.

Several of the measurements will also have a packet capture running using Wireshark to analyse the layer 2 and layer 3 handover delay. While not the focus of this thesis the handover delay plays a large part in the seamlessness of the handover.

6.4 Results

This section will describe the results of the measurements described in the previous section starting with the analyses of the layer 2 and 3 handover delay. The layer 2 handover took 2.41 seconds on average with a 95% confidence interval of (2.38, 2.43) seconds. This means that no layer 3 communication is possible for about 2.4 seconds in this specific IEEE 802.11 to IEEE 802.11 handover. The layer 3 communication started (IPv6 neighbour solicitation for own link local address) on average 2.51 seconds (95% confidence interval (2.47, 2.54)) after the handover was initiated. The layer 3 connection became active (IPv6 router solicitation sent) on average 3.51 seconds with a 95% confidence interval of (3.47, 3.55) after the handover started. Communications are still delayed 1 second more than might be possible due to the delay between the layer 3 connection starting and being fully active.

Figure 6.2 shows a graph of the throughput during the entire measurement process for PFMIPv6 handovers with different traffic loads with no added

delay. The first 5 seconds the MN is connected to the pMAG and the throughput is around the value the data is sent at. At the 5 second mark the handover is initiated and the throughput drops to 0 during the time the MN is disconnected from the network. After the handover is completed there is a visible spike in the throughput as the pMAG flushes the packet buffer to the MN. It seems that higher data rates are not correctly handled. Figure 6.2 shows the packets sent at 700 kbit/s or higher are not always transmitted over the wireless interface at that speed initially. The transmission speed is highly variable but does average out to the speed the packets are transmitted on at the CN. Because this behaviour only occurs on the wireless network and all packets are delivered in the end this variable transmission speed seems to be caused by the wireless hardware. It is interesting to note that the throughput does significantly increase during the buffer flush. This can be explained by the fact that the buffer is flushed using unicast traffic on the wireless interface, the throughput problem at higher data rates only appears to happen with multicast traffic. The basic rates for broadcasting set on the access points was 12 Mbit/s, which should not affect the throughput of data at rates less than 2 Mbit/s as were used during these measurements. It is important to note that despite the lower than expected data rates all packets were delivered at the MN. Another interesting observation that can be made is that with higher data rates the handover appears to happen later, notably the handovers with speeds of 700 kbit/s and higher where the handover occurs around the 6 second mark. In al cases the handover was triggered at exactly the 5 second mark. The cause of this delay might be that the packet signalling the MN to perform the handover is queued in the wireless interface while the large amount of multicast packets is handled, but further investigation is needed to find the source of this problem and the lower than expected initial throughput.

The results for the packet loss during the measurements with a fixed packet size can be seen in Figure 6.3 and 6.4. The packet loss seems to increase linearly with higher transmission speeds, with performance slightly worse for the 1200 byte packets compared to the 600 byte packets. Overall the packet loss during the fast handover measurements is greatly reduced compared to the normal PMIPv6 case. There are a few problems that prevent us from reaching close to zero drop rates for traffic. One is the way the handover is currently triggered in the testing network. The handover command is sent from the MAG the MN is currently connected to. This requires the handover command to be sent before the MAG actually starts forwarding traffic towards the nMAG because updating the routing tables would make us unable to signal the handover. This leads to a small period in which the MN is already switching network but the current MAG is still forwarding traffic for the

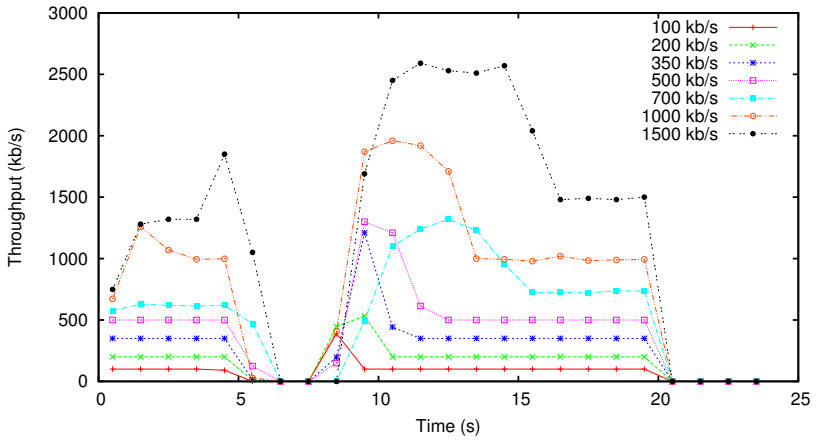


Figure 6.2: Throughput during 600 byte packets measurement

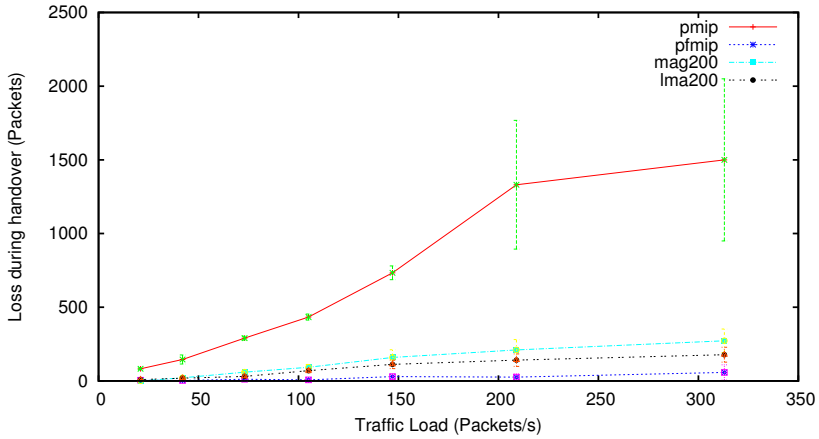


Figure 6.3: 600 byte packet loss

MN to the wireless interface it was on. Switching to actual MIH signalling should be able to help us as this can use layer 2 communication taking layer 3 routing for the signalling out of the picture entirely. Another factor causing packet loss is the switch from MAG-MAG tunnel to LMA-MAG tunnel as a source. In some cases this switch takes a very small amount of time leading to the loss of one or two packets. During the handovers with delays on the tunnels there is more packet loss than during the handovers without delay. The reason for this is again that the switching between tunnels does not happen instantaneously.

Figure 6.5 shows the packet loss during the handover for different delay values on the pMAG - nMAG and LMA - nMAG tunnels. An interesting observation here is the low packet loss on the measurement with 300 ms delay

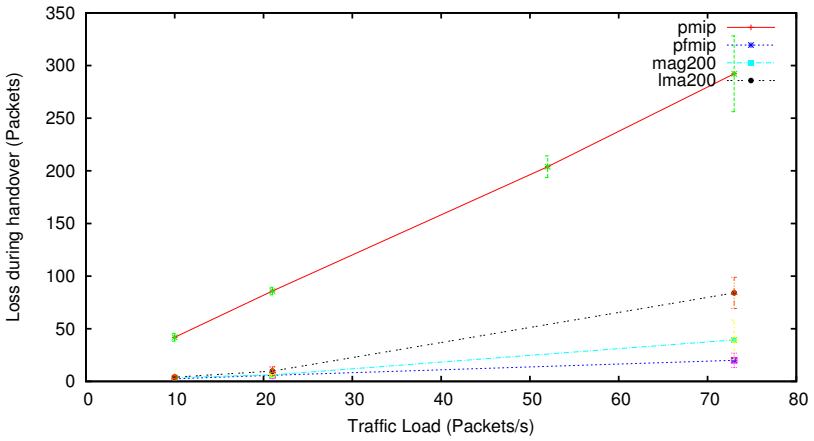


Figure 6.4: 1200 byte packet loss

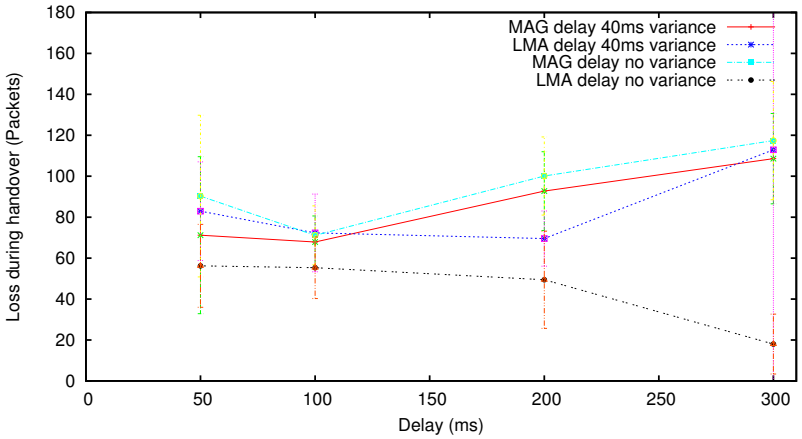


Figure 6.5: 600 byte packet loss / delay correlation

on the LMA - nMAG tunnel with no delay variance. In all other instances the delay variance does not seem to have a significant impact on the packet losses. Further measurements will be needed to determine why the performance is better in this high delay situation compared to the measurements with a lower delay. The variance added to the delay does not seem to cause a significant difference in packet loss but more testing is needed to determine the effect of delay variance on the handover process.

Jitter results are very low (around 1 ms) for the normal handover, fast handover and fast handover with added delay on the pMAG-nMAG tunnel. This means that the difference in arrival time between consecutive packets is relatively low. Only the measurements with delay on the LMA-nMAG tunnel show high jitter values as can be seen in Figure 6.6. The reason the jitter

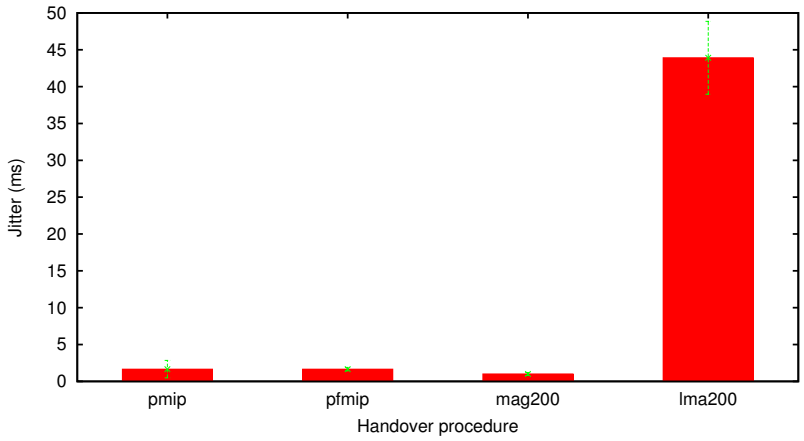


Figure 6.6: Jitter (ms) for 600 byte packets @ 200 kb/s

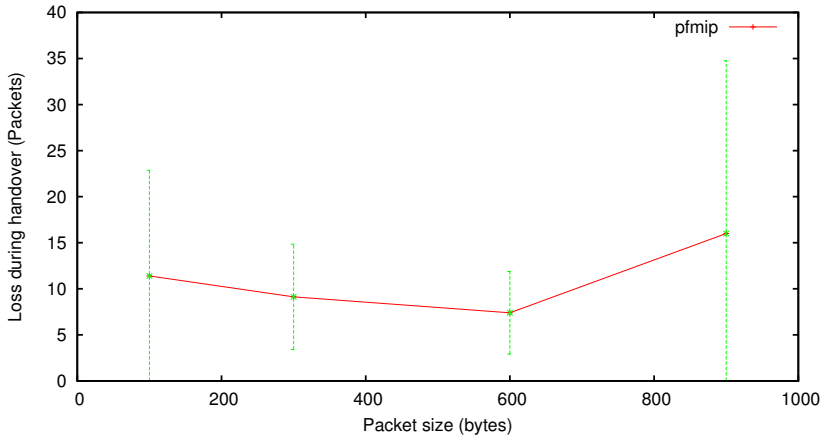


Figure 6.7: Packet loss dependent on packet size with 104 packets/s

values are relatively high in the case of fast handovers with added delay on the LMA-nMAG tunnel is that traffic is temporarily stopped to prevent the transmission of duplicate packets. This causes a temporary spike of packets arriving much later than would otherwise be expected.

Figure 6.7 shows the packet loss for different packet sizes during a PFMIPv6 handover with no relative delay on the pMAG-nMAG and LMA-nMAG tunnels. Performance is generally good but has large variations with very small (100 byte) and large (900 byte) packets.

Another notable difference between the handover with delay on the LMA-nMAG tunnel and the other handovers is the out of order delivery of packets. As can be seen in Figure 6.8 this value is very high for the measurements with added delay on this tunnel. This effect is caused by the switch between

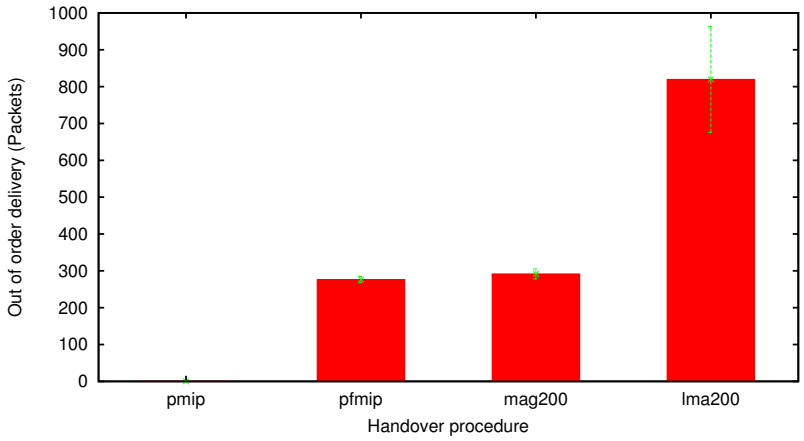


Figure 6.8: Out of order packet delivery, 600 byte packet size with 104 packets/s

the pMAG - nMAG and the LMA - nMAG tunnel. The packets from the pMAG - nMAG tunnel are forwarded longer than they should be, causing a high number of out of order packets.

6.5 Discussion

The most important contribution to the PFMIPv6 handover delay is the layer 2 delay of 2.41 seconds. Even with Optimistic DAD enabled the entire handover still takes 3.51 seconds to complete. There might be an optimization possible to further reduce this delay to 2.47 seconds by starting communication after the neighbour solicitation for the MNs link local address is sent. This delay might be specific for the IEEE 802.11 hardware used, but similar results in [28] suggest delays of this order are to be expected in IEEE 802.11 networks. Performance should be better on layer 2 technologies with less handover delay.

Fast handovers using the PFMIPv6 implementation show a much reduced loss rate compared to the normal PMIPv6 handovers. The packet loss is mostly close to 0 except for higher traffic loads where it increases slightly. The measurements show that the PFMIPv6 implementation performs mostly as it should, the buffering mechanism is working correctly. The slightly higher losses in the measurements with delays added to the tunnels are caused by small problems in the implementation related to the switching between the two tunnels. It should also be noted that even when the test network couldn't keep up with the amount of packets offered all the data was correctly delivered to the MN.

There are high jitter values in the measurements with a high delay on the LMA-nMAG tunnel. These results are related to the out of order delivery of packets in these cases. This effect could have consequences for real time applications such as streaming voice and video as they tend to discard packets arriving late. This is an area in which the implementation needs improvement.

It can be concluded that the PFMIPv6 implementation with multicast support works well. There are still a few minor problems related to the switching between different tunnels that carry multicast traffic that require some fine tuning to optimize the system for the PPDR use case but overall the performance compared to normal PMIPv6 is very good.

Chapter 7

Conclusion and Future Work

This thesis describes the design and implementation of a Proxy Fast Mobile IPv6 system with multicast support and optimizations to the multicast handover process. The implementation was validated with measurements done in an actual deployment with IEEE 802.11 networks and it was shown the implementation and multicast improvements perform as expected with the exception of the situation with high delay on the LMA-nMAG tunnel. The problem with high jitter and high out of order values is however mostly an optimization issue and does not have a large impact on packet loss ratios.

It was shown that multicast can be successfully integrated with PFMIPv6. It is important to integrate the two as the handover messages need to contain the multicast groups a MN is subscribed to. Multicast packets are efficiently buffered in the time sense. The buffer is fast, but the buffering needs to be done in memory. The downside is that when a large amount of traffic needs to be buffered the memory of the MAG device might not be enough, however this problem is not likely to occur on dedicated routing hardware. Multicast handover performance is relatively good with minimal packet loss during the handover and the possibility of even better performance with some tuning.

Transmission of duplicate packets on the wireless network during a handover is avoided by switching the interface the multicast packets are forwarded from at the right moment. The decision of when to switch between the pMAG - nMAG and LMA - nMAG tunnels is based on the difference in delay between the two tunnels. This prevents multicast packets from being transmitted twice, reducing the load on the wireless network.

Security in the PMIP network can be achieved by encrypting the tunnels between LMAs and MAGs. Security on the air interface and prevention of devices joining groups they do not belong to are more difficult. Application level security might be the best option for securing multicast traffic in a semi open environment.

The current implementation is still missing reactive handover support. Handovers that are made without initiating them first in the pMAG are currently still performed as normal PMIP handovers. It should be relatively straightforward to implement reactive handover support into the software, but this was not done due to time constraints.

Support for multicast senders needs to be added to the implementation. While it is already possible for a MN to act as a multicast sender improvements are possible to allow multicast traffic to flow from MAG to MAG directly, skipping the LMA and reducing latency.

Another possible improvement is tuning the interval the timing packets are send on. Increasing the time between timing packets puts less load on the network but makes the estimated delay between tunnels much less accurate. In a network with high amounts of congestion it could be beneficial to have the timing packet interval much lower than on a low congestion network.

There is still a possibility for large optimizations on the impact of multicast traffic on the air interface. 802.11 is relatively inefficient compared to techniques such as LTE (when eMBMS is implemented) so it might be useful to look into optimizing 802.11 multicasting.

Work needs to be done on the integration of MIH and the PFMIPv6 implementation. This could make prediction of handovers possible and improve the handover process by allowing the handover trigger to be sent on layer 2, allowing for layer 3 traffic to switch to the forwarding tunnel more efficiently.

Multiple MNs listening to the same group on a MAG present extra synchronization problems. These mainly occur on the air interface. If MN A performs a handover to a MAG that already has a MN B listening to the same multicast group it becomes difficult to deliver the same data on the air interface twice. In LTE configuration this is not a problem as (currently) these have no broadcast mode but simply have unicast connections to each MN. On IEEE 802.11 this is however a problem as the multicast packets from both sources (pMAG-nMAG tunnel and LMA-nMAG tunnel) might be broadcast at the same time. The same problem occurs when two nodes from different LMAs are connected that listen to the same multicast group. A possible solution for the second problem is simply blocking “double” groups on one of the incoming interfaces (introducing even more handover complexity for the ‘double’ MN).

More research needs to be done on securing multicast traffic in a PPDR context. There might be trade-offs between security and performance that need to be taken into account.

Bibliography

[1] Computable. (2013) Brandweer vervangt C2000-portofoons. [Online]. Available: http://www.computable.nl/artikel/nieuws/telecom/4923768/1276977/brandweer-vervangt-c2000portofoons.html

[2] Critical Communications Broadband Group, "The Strategic Case for Mission Critical Mobile Broadband," TCCA, Tech. Rep., 2013.

[3] L. De Vito, S. Rapuano, and L. Tomaciello, "One-way delay measurement: State of the art," *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, no. 12, pp. 2742–2750, 2008.

[4] S. Deering and R. Hinden, "Rfc 2460: Internet protocol, version 6 (ipv6) specification," 1998.

[5] dot11 info. (2011) 802.11 layer 2 dynamic encryption key generation. [Online]. Available: http://dot11.info/index.php?title=Chapter_5_:_802.11_layer_2_dynamic_encryption_key_generation

[6] T. Elteto and S. Molnar, "On the distribution of round-trip delays in tcp/ip networks," in *Local Computer Networks, 1999. LCN'99. Conference on*. IEEE, 1999, pp. 172–181.

[7] D. Farinacci, P. Traina, S. Hanks, and T. Li, "Generic routing encapsulation (gre)," 1994.

[8] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy mobile ipv6," RFC 5213, August, Tech. Rep., 2008.

[9] IANA. (2014) IPv4 Multicast Address Space Registry. [Online]. Available: http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml

[10] ——. (2014) IPv6 Multicast Address Space Registry. [Online]. Available: http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml

[11] IEEE Standards Association and others, "802.11-2012-IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 2012.

[12] IETF Multicast Mobility Working Group. (2014) Multimob status pages. [Online]. Available: http://tools.ietf.org/wg/multimob

[13] Iperf. (2014) Iperf: the TCP/UDP bandwidth measurement tool. [Online]. Available: https://iperf.fr/

[14] K.-S. Kong, W. Lee, Y.-H. Han, M.-K. Shin, and H. You, "Mobility management for all-ip mobile networks: mobile ipv6 vs. proxy mobile ipv6," *Wireless Communications, IEEE*, vol. 15, no. 2, pp. 36–45, April 2008.

[15] R. Koodli, M. Waehlisch, G. Fairhurst, T. Schmidt, and D. Liu, "Multicast Listener Extensions for Mobile IPv6 (MIPv6) and Proxy Mobile IPv6 (PMIPv6) Fast Handovers," 2014.

[16] J.-H. Lee, J.-M. Bonnin, I. You, and T.-M. Chung, "Comparative handover performance analysis of IPv6 mobility management protocols," *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 3, pp. 1077–1088, 2013.

[17] J.-H. Lee and T. Ernst, "Fast pmipv6 multicast handover procedure for mobility-unaware mobile nodes," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, May 2011, pp. 1–5.

[18] Linux Foundation. (2009) netem. [Online]. Available: http://www.linuxfoundation.org/collaborate/workgroups/networking/netem

[19] L. Magagula and H. Chan, "Ieee802.21 optimized handover delay for proxy mobile ipv6," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, Nov 2008, pp. 1–7.

[20] J. Malinen. (2013) hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator. [Online]. Available: http://w1.fi/hostapd/

[21] N. Montavont and T. Noel, "Handover management for mobile nodes in ipv6 networks," *Communications Magazine, IEEE*, vol. 40, no. 8, pp. 38–43, Aug 2002.

[22] N. Moore, "Rfc 4429 optimistic duplicate address detection (dad) for ipv6," *Internet Engineering Task Force*, 2006.

[23] Nyanyo, Marques, Rodriguez, Wickson, H. de Groot, Delmas, Blaha, K. Wolterink, Petrov, and Jelenc, "Salus: Deliverable 2.2 salus ppdr user requirements - intermediate," 2013.

[24] Open Air Interface. (2011) Open air interface proxy mobile ipv6 (OAI PMIPV6). [Online]. Available: http://www.openairinterface.org/openairinterface-proxy-mobile-ipv6-oai-pmipv6

[25] C. Perkins, "IP Mobility Support for IPv4, Revised," RFC 5944, November, Tech. Rep., 2010.

[26] C. Perkins *et al.*, "Rfc 3344: Ip mobility support for ipv4," *Network Working Group*, 2002.

[27] J. Postel, "Rfc 791: Internet protocol," 1981.

[28] A. K. Quoc, D. Kim, and H. Choo, "A novel scheme for preventing out-of-order packets in fast handover for proxy mobile ipv6," in *Information Networking (ICOIN), 2014 International Conference on*, Feb 2014, pp. 422–427.

[29] RIPE NCC. (2015) IPv4 Exhaustion. [Online]. Available: http://www.ripe.net/internet-coordination/ipv4-exhaustion

[30] T. Schmidt, M. Waehlisch, and S. Krishnan, "Base deployment for multicast listener support in Proxy Mobile IPv6 (PMIPv6) Domains," *IETF RFC6224, April*, 2011.

[31] T. C. Schmidt, S. Wölke, and M. Wählisch, "Peer my Proxy - A Performance Study of Peering Extensions for Multicast in Proxy Mobile IP Domains," in *Proc. of 7th IFIP Wireless and Mobile Networking Conference (WMNC 2014)*. Piscataway, NJ, USA: IEEEPress, May 2014, pp. 1–8.

[32] Tcpdump/Libpcap. (2015) TCPDUMP & LIBPCAP public repository. [Online]. Available: http://www.tcpdump.org/

[33] UMIP. (2013) UMIP - Mobile IPv6 and NEMO for Linux. [Online]. Available: http://www.umip.org/

[34] R. Vida and L. Costa, "Rfc 3810: Multicast listener discovery version 2 (mldv2) for ipv6," *Request for Comments, IETF*, 2004.

[35] D. Wischik and N. McKeown, "Part i: Buffer sizes for core routers," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 3, pp. 75–78, 2005.

[36] H. Yokota, K. Chowdhury, R. Koodli, B. Patil, and F. Xia, "Fast Handovers for Proxy Mobile IPv6," RFC 5949, September, Tech. Rep., 2010.

Appendix A

Installation instructions

Installation instructions in this section are specific to a x86 machine running Debian testing (will be released as "jessie" in 2015). These instructions should be followed on the LMA and MAGs. Instructions specific to MAG or LMA will be explained later. As the PFMIPv6 implementation was build on top of the UMIP PMIP system these instruction are based on those found on the UMIP website but adapted to the new situation.

The following non-standard software should be installed (can all be found in the Debian repositories): hostapd, libpcap-dev, indent, bison, flex, iproute-dev, libc6-dev, libssl-dev, autoconf, libtool, macchanger and python-netaddr.

```
# apt-get install hostapd libpcap-dev indent bison flex
iproute-dev libc6-dev libssl-dev autoconf libtool macchanger
python-netaddr
```

Before we can install the PFMIPv6 implementation we need to install a radius client for authentication of the MNs. We use the FreeRadius 1.1.6 client with patches from the UMIP website:

```
$ wget http://umip.org/contrib/others/freeradius-client-1.1.6
.tar.bz2
$ tar jxf freeradius-client-1.1.6.tar.bz2
$ cd freeradius-client-1.1.6/
$ wget http://umip.org/contrib/others/freeradius-client-1.1.6
-oai-pmipv6.patch
$ patch -p1 < freeradius-client-1.1.6-oai-pmipv6.patch
$ autoreconf -i
$ ./configure
$ make
# make install
```

Edit the "/etc/ld.so.conf" file to add a new line: "include /usr/local/lib/". Save the file and run "ldconfig".

The PFMIPv6 software can now be installed. The source can be retrieved from the GIT repository:

```
$ env GIT_SSL_NO_VERIFY=true git clone -b multicast
https://git.berndmeijerink.nl/bernd/pfmipv6.git
$ cd pfmipv6
$ autoreconf -i
$ CPPFLAGS='-isystem /usr/src/linux/usr/include/' ./configure
--enable-vt --with-pmip-use-radius
$ make
# make install
```

A.1 LMA specific

For MNs to be authenticated and given an IP address we need a radius server. This server was ran on the same machine as the LMA but it could be deployed on a different machine.

```
$ wget ftp://ftp.freeradius.org/pub/freeradius/freeradius
-server-2.2.6.tar.gz
$ tar xjf freeradius-server-2.1.12.tar.bz2
$ cd freeradius-server-2.1.12
$ ./configure
$ make
# make install
```

The patched freeRadius client contains configuration files for the server. We need to copy those files for the server:

```
$ cd /path/to/freeradiusclient-1.1.6/examples/
# cp users /usr/local/etc/raddb/
# cp radiusd.conf /usr/local/etc/raddb/
# cp clients.conf /usr/local/etc/raddb/
```

Finally the *users* file we just copied should be edited to include the MAC addresses of the MN(s) we want to use in the network.

A.2 MAG specific

We need to edit the "/etc/hosts" file to include "2001:100::1 radius6server" (assuming the radius server will be running on the LMA).

A.2.1 Hostapd configuration

Hostapd was run with the following configuration:

```
###### hostapd configuration file ##################
interface=wlan0

logger_syslog=-1
logger_syslog_level=2
logger_stdout=-1
logger_stdout_level=2

##### IEEE 802.11 related configuration ###########
ssid=PMIPAP
hw_mode=g
channel=1
ieee80211n=0
basic_rates=120
beacon_int=100
dtim_period=2
max_num_sta=255
rts_threshold=2347
fragm_threshold=2346
macaddr_acl=0
auth_algs=3
ignore_broadcast_ssid=0
wmm_enabled=1
wmm_ac_bk_cwmin=4
wmm_ac_bk_cwmax=10
wmm_ac_bk_aifs=7
wmm_ac_bk_txop_limit=0
wmm_ac_bk_acm=0
wmm_ac_be_aifs=3
wmm_ac_be_cwmin=4
wmm_ac_be_cwmax=10
wmm_ac_be_txop_limit=0
wmm_ac_be_acm=0
wmm_ac_vi_aifs=2
wmm_ac_vi_cwmin=3
wmm_ac_vi_cwmax=4
wmm_ac_vi_txop_limit=94
wmm_ac_vi_acm=0
wmm_ac_vo_aifs=2
wmm_ac_vo_cwmin=2
```

```
wmm_ac_vo_cwmax=3
wmm_ac_vo_txop_limit=47
wmm_ac_vo_acm=0

##### WPA/IEEE 802.11i configuration ############
wpa=1
wpa_passphrase=password
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP
wpa_group_rekey=600
wpa_strict_rekey=1
wpa_gmk_rekey=86400
wpa_ptk_rekey=600
```

A.3 Using the software

Configure the LMA for your system by editing "some/dir/pfmipv6/extras/example-ha-lma.conf" file. When this is done run:

```
# python some/dir/pfmipv6/extras/UMIP0.4_LMA_UBUNTU.10.04.py
```

When this is done we can start the radius server:

```
# radiusd
```

We also need to configure the MAG by editing "some/dir/pfmipv6/extras/example-mag1.conf" (use one file per MAG). Also edit the MAG startup script to point to the correct configuration file by editing "UMIP0.4_MAG_UBUNTU.10.04.py". The MAG can now be started:

```
# python some/dir/pfmipv6/extras/UMIP0.4_MAG_UBUNTU.10.04.py
```

After the MAG is started we can start hostapd:

```
# hostapd config.conf
```