MASTER'S THESIS

Simultaneous Binding Proxy Mobile IPv6

Author:
K. Idserda

Graduation committee:
Telematica Instituut
Dr.ir. Mortaza S. Bargh
Dr.ir. Henk Eertink
University of Twente
Dr.ir. Geert Heijenk
Dr.ir. Georgios Karagiannis

Telematica
Instituut

University of Twente
Enschede - The Netherlands

Simultaneous Binding Proxy Mobile IPv6
K. Idserda

Master's Thesis

Telematica Instituut
PO Box 589
7500 AN Enschede
The Netherlands

Design and Analysis of Communication Systems
Faculty of EEMCS
University of Twente
PO Box 217
7500 AE Enschede
The Netherlands

Enschede, The Netherlands, December 2008

Abstract

Today's communication networks are evolving towards an all-IP architecture. IP-level mobility protocols will have to support seamless handovers from one access technology to another. Existing IP-level mobility protocols have difficulties meeting these stringent handover delay requirements. Also, they do not give sufficient control to the network to optimize the handover process and they do not deal well with slow connection setups of some wireless technologies. This thesis presents a networked controlled IP-level mobility protocol called Simultaneous Binding Proxy Mobile IPv6 (SPMIPv6), based on the existing Proxy Mobile IPv6 protocol. The protocol deals with large layer 2 setup times by taking a proactive approach towards the handover. It is allows some parts of the handover process to be carried out before the actual handover. In the networks itself, packets for the MN are already sent to the MN's new point of attachment before the actual handover. This way, the handover latency can be limited to one RTT between the mobile node and the new access router and the packet loss due to handover can be decreased and eliminated by appropriately buffering packets at the new access router. The SPMIPv6 protocol was implemented in testbed, which showed that both UDP and TCP do not suffer from any performance degradation during the actual handover. Two important characteristics of SPMIPv6 are its robustness to incorrect handover predictions and its built-in features to accommodate large network attachment latencies.

Acknowledgements

This thesis could not have been written without the support of the following people.

First, I would like to thank Mortaza Bargh from Telematica Instituut for his inexhaustible support and many suggestions. I'm also grateful for the feedback I received from the rest of the graduation committee: Henk Eertink from Telematica Instituut and Geert Heijenk and Georgios Karagiannis from the University of Twente.

I would also like to thank Julien Laganier, Alf Zugenmaier and Anand Prasad for their help with the testbed at DOCOMO Euro-Labs in Munich.

Finally, I thank my parents for supporting me throughout my studies.

Enschede, December 5th, 2008.

Jeroen (K.) Idserda

Contents

List of Figures

List of Tables

Abbreviation list

3GPP	3rd Generation Partnership Project
AR	Access Router
BBM	Break Before Make
BU	Binding Update
CoA	Care of address
DAD	Duplicate Address Detection
DNAv6	Detecting Network Attachment version 6
DSL	Digital Subscriber Line
FBA	Fast Binding Acknowledgment
FBU	Fast Binding Update
FMIPv6	Fast Mobile Internet Protocol v6
FNA	Fast Neighbor Advertisement
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HA	Home Agent
HAck	Handover Acknowledgment
HI	Handover Initiate
HMIPv6	Hierarchical Mobile Internet Protocol version 6
HoA	Home Address
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
LCoA	Local Care of Address
LLA	Link Local Address
LMA	Local Mobility Anchor
LTE	Long Term Evolution
MAC	Media Access Control
MAG	Mobility Access Gateway
MAP	Mobility Anchor Point
MBB	Make Before Brake
MIPv6	Mobile IPv6
MN	Mobile Node
MnAP	Mobile-node Attachment Point
MNID	Mobile Node Identifier
NA	Neighbor Advertisement
NAI	Network Access Identifier
NAR	New Access Router
NCoA	New Care of Address

ND	Neighbor Discovery
nMAG	new MAG
NS	Neighbor Solicitation
oMAG	old MAG
PAR	Previous Access Router
PBUA	Proxy Binding Update Acknowledgment
PBUR	Proxy Binding Update Request
PDA	Personal Digital Assistant
PDP	Packet Data Protocol
PMIPv6	Proxy Mobile Internet Protocol version 6
PrRtAdv	Proxy Router Advertisement
QoS	Quality of Service
RA	Router Advertisement
RCoA	Regional Care of Address
RFC	Request for Comments
RIP	Regional Information Point
RS	Router Solicitation
RtSolPr	Router Solicitation for Proxy Advertisement
SAE	System Architecture Evolution
SPBUA	Simultaneous Binding Update Acknowledgment
SPBUR	Simultaneous Binding Update Request
SPMIPv6	Simultaneous binding Proxy Mobile Internet Protocol version 6
SSID	Service Set Identifier
UMTS	Universal Mobile Telecommunications System
VoIP	Voice over IP

1. Introduction

1.1. Research context

A wide range of wireless communication systems is available nowadays. Wireless network technologies like Global System for Mobile communications (GSM) and Universal Mobile Telecommunications System (UMTS) [1] offer a large coverage area, but have limited bandwidth and can be expensive to use. On the other side we have Wi-Fi technology (based on IEEE802.11 standard [2]), which offers a large bandwidth in a small coverage area, for a relatively cheap price. There is also the emerging Mobile WiMax technology [3] (based on the IEEE802.16e [4] standard), sitting in between when looking at both bandwidth and coverage area.

New communication devices like Personal Digital Assistants (PDAs) and smart-phones are able to connect to more than one of such wireless technologies. When these connections all offer the same capabilities (e.g., Internet Protocol (IP) data transport), the most appropriate (and available) wireless technology can be used. It would be possible to use Wi-Fi if the user is within range of a hotspot and needs a lot of bandwidth at that time. But, as said, Wi-Fi does not have a large coverage area. When moving out of range, the ongoing communication sessions should be transferred to, for example, UMTS. This connection transfer is called a handover, which typically disrupts the connection for some period called “handover latency”.

Figure 1: PDA playing streaming music

There are a number of applications like Voice over IP (VoIP) calls where a large handover latency is not permitted, since this would cause packet loss during the handover period or disrupt the call due to excessive jitter. A user could also be watching streaming audio or

video, both of which may stutter or even stop during a handover from one access technology to another.

To be able to support applications like these future networks must be able to support handover with the smallest amount of handover latency possible.

1.2. Problem statement and objectives

Today's communication networks are evolving towards an all-IP architecture [5]. This means that these communications networks will merge towards a heterogeneous network where different access technologies are all using IP as a common denominator. Development of such evolved communication networks is done by the 3rd Generation Partnership Project (3GPP), an organization that standardized GSM and UMTS in the past. They are working on a new mobile access technology standard called Long Term Evolution [6] (LTE). This new standard aims for increased data rates and a higher Quality of Service (QoS). The architecture of the IP-based core network for this will evolve towards the System Architecture Evolution [7] (SAE), which will provide support for more flexible handover towards other (fixed) networks, including IEEE802-type networks. A clear specification that will meet all requirements for a seamless handover is however still in development.

In an all-IP network, a lot of different TCP/IP layer-2 (data link layer, see Figure 2) [8] protocols such as Wi-Fi, UMTS or a wired technology such as any form of DSL (Digital Subscriber Line) will all be operating under this all-IP network. The IP layer sits on top of all these access technologies, which means that the protocol that supports mobility in the network is also assumed to be IP based. Internet Protocol version 6 (IPv6) [9] is the next generation of the currently used IPv4 [10] protocol, which is expected to be used widely in the near future. IPv6 includes a protocol to handle mobility issues, called Mobile IPv6 (MIPv6) [11]. This protocol hides movement of a node in the network, which means changing its point of attachment, for the applications running on the Mobile Node (MN). It operates on top of any other layer 2 technology. However, the different characteristics of these layer 2 technologies make a seamless handover difficult for Mobile IPv6.

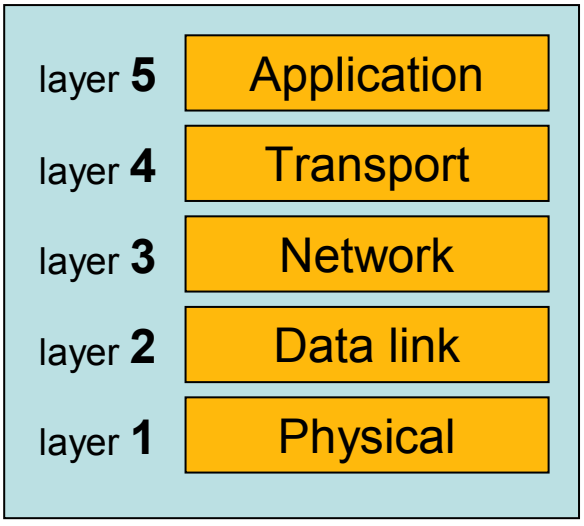

Figure 2: TCP/IP Model

We can distinguish two categories of handover: vertical handover and horizontal handover. The term vertical handover is used when a handover occurs from one technology type (for example Wi-Fi) to another (i.e. UMTS). When a handover occurs within a network technology type, e.g., from one Wi-Fi access point to another, we use the term 'horizontal handover'.

Handovers can be on layer 2, requiring the MN to connect to a new access point without changing the IP-subnet, or also on layer 3, when connecting to a link that is on a different IP-subnet. Furthermore, a handover can be within the same administrative domain (intra-domain handover) or between two different administrative domains (inter-domain handover).

Considering the types of wireless technologies involved in a handover, timing of the actual handover is crucial for a seamless continuation of ongoing data transfers on the new connection. On layer 2, it might be possible to connect to a new access network while still maintaining a connection to the old one. This is called Make Before Break (MBB). The opposite happens when for example a device is not capable of using two network interfaces at the same time or when the device is capable of using two network interfaces at the same time, but has not anticipated a sudden handover. This is called Break Before Make (BBM). This means that the old connection is lost before a new connection to the second network is fully activated. Note that we are talking about layer 2 here. On layer 3, both MBB and BBM are possible. In the case of MBB, the MN will get a second IP address that is active on the new connection. In layer 3 BBM, the MN will configure its IP address when the old connection is lost.

As a preliminary to this research we looked at the connection setup process in UMTS. This roughly consists of three parts: establishing a radio connection between the MN and the network, the General Packet Radio Service (GPRS) attach process (authentication and registering in the MN in the network) and finally the PDP (Packet Data Protocol) context activation, which gives the MN a valid IP address. This whole process takes about 1250 ms. When we look at a BBM handover (both on layer 2 and layer 3) towards UMTS this would mean that the handover latency would be this same amount, which is much too large to continue for example a VoIP call without any disruptions. We tried to see if certain parts of the UMTS connection process could be carried out in advance, resulting in a smaller handover latency. The minimum delay we could achieve, which was by only having to set up a radio channel between MN and the network, was around 180 ms, which is still too large. From this we concluded that a MBB handover towards UTMS with the total connection in place is the only way to achieve a seamless handover.

Mobile IPv6, the mobility protocol of IPv6, has no support for such a proactive MBB handover approach, which is necessary for a seamless handover in the scenario outlined above. Another issue is that it will take a while before data destined for the MN is available at its new access point. The routing in the network has to be altered to get this data to the new location. In a proactive handover solution, this data can already be sent to the new access point before the MN arrives there. This option is also not present in current mobility protocols.

The goal of this thesis is to design and validate a mobility protocol that:

- is IP based, so it can be used on the future all-IP SAE/LTE network. For this, mobility protocols available in IPv6 can be used as a starting point.
- supports both vertical and horizontal handovers towards different layer 2 technologies.
- provides seamless handovers with a maximum handover latency of 50 ms, to support applications such as VoIP.
- is proactive, to support large layer 2 setup delays.
- also does preparations for a handover in the network itself to minimizes packet loss

How this will be accomplished is explained in the next section.

1.3. Approach

Considering all the factors mentioned above, we aim at finding solutions that minimize the handover latency period caused by a handover from one access point to another. We consider a scenario where the MN experiences a handover from one access network to another. The handover can be vertical (from Wi-Fi towards UMTS) or horizontal (from one Wi-Fi access point to another where the access points have a different SSID). We require the coverage areas of the current and the new network to overlap during the handover period. Further, we require that the MN is not capable of having layer 3 connectivity to both networks at the same time during the handover. The MN, however, is capable of having layer 2 connections to two different access networks simultaneously for the handover duration. We mainly focus on Wi-Fi (802.11) and UMTS as layer 2 wireless network technologies.

The solution for the handover process presented in this report is based on a network controlled layer 3 mobility protocol. It is based on Proxy Mobile IPv6 (PMIPv6), the network controlled version of MIPv6, and is aimed at reducing the handover latency. We use a proactive approach towards the handover and propose an enhancement of PMIPv6 with simultaneous binding, called Simultaneous binding Proxy Mobile IPv6 (SPMIPv6). In SPMIPv6 the network is in charge of coordinating and predicting the handover. The handover prediction process instructs the MN and the network to prepare for a handover. With the use of two triggers different stages of the handover process are activated. The first trigger activates the preparation of the handover, both on the MN as well as in the network. In the MN, a layer 2 connection to the target link is set up via the new access point. This is necessary, since with some access techniques like UMTS, this setup delay can be quite large (as shown in Appendix A). Figure 3 further illustrates this approach. Here we see that trigger 2 marks the moment that the current layer 3 connection is lost, after which the layer 3 attachment process to the new attachment point has to be carried out. Normally, this would take quite some time because the MN would have to request an IP address to use, verify its uniqueness, etc. To prevent this delay, DNAv6 (Detecting Network Attachment version 6) will be used to ensure a fast layer 3 attachment. This is further explained in chapter 2. In the network, a copy of the traffic for the MN is already delivered to the new access network following the first trigger. After the second trigger that executes the actual handover, this approach allows the data destined for the MN to be already available at the new location.

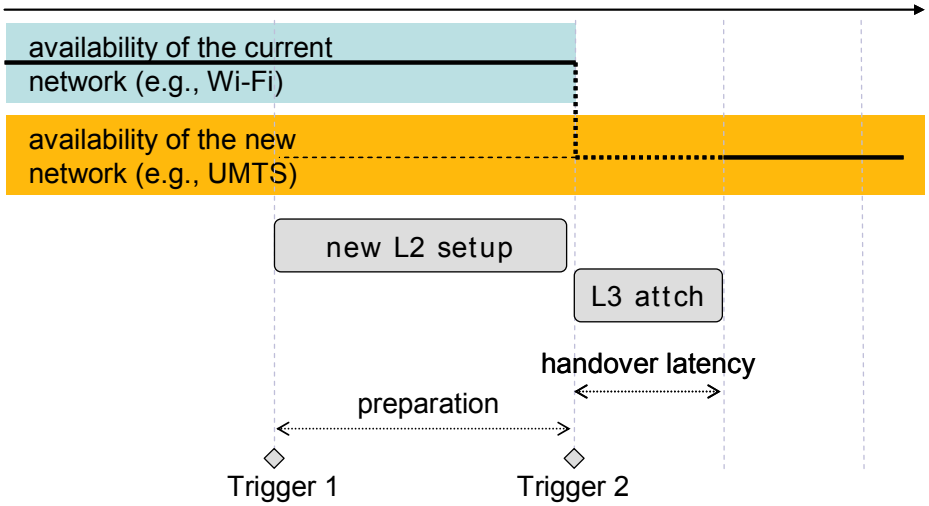

Figure 3: Handover approach

The objective is to attain the maximum handover latency of 50 ms, as required in [12],[13]. In this, it states that in order to support real-time streams such as VoIP in a handover between different radio technologies the packet transmission delay fluctuations should be less then this 50 ms. In order to determine if the protocol lives up to this, we will implement SPMIPv6 in a testbed and carry out tests to measure the handover latency and to determine the impact of the handover latency on both UDP and TCP protocols when transporting upstream and downstream data flows.

1.4. Outline

The rest of this report is structured as follows: Chapter 2 describes the relevant network (mobility) protocols that are used in combination with or offer similar functionality as SPMIPv6. Chapter 3 presents the design and operational aspects of the SPMIPv6 protocol. Chapter 4 shows the performance results of the protocol from runs in a testbed implementation. The last chapter gives our conclusion and depicts possible directions for future work. In Appendix A the results of a study of the UMTS connection setup and its corresponding delays is presented.

2. Background

This chapter describes all relevant mobility aspects and protocols used in the rest of this report. The main focus is on the mobility features of IPv6 that are present in the network. Those features that are embedded in IPv6 for mobility detection and configuration are described in section 2.1. Specific IP-level signaling protocols that deal with IP-level mobility are explained in section 2.2. In section 2.3, related work with respect to our solution, the SPMIPv6 protocol, is presented.

2.1. IPv6 mobility-related features

IPv6 [9] is the successor of the widely used IPv4. It has many improvements over the old IPv4. We will use IPv6 as the base of the mobility protocol instead of IPv6 since it has more advanced mobility aspects than IPv4 and will probably be widely used in the near future.

IPv6's features include expanded addressing capabilities, as the size of the IP address is increased from 32 to 128 bits. The scalability of multicasting is improved and an anycast address type is introduced, which can be used to send packets to any one of a group of nodes. Address types are explained in section 2.1.1. It is also possible for a node to obtain a valid IPv6 address using stateless auto configuration. Stateful configuration using for example a DHCP server is also still supported. Address configuration is explained in section 2.1.3.

In the remaining of this section the various aspects of IPv6 which are of importance for mobility support will be explained.

2.1.1. Address types

An IPv6 node has multiple types of addresses [14],[15]. A node typically has a loop-back address, a Link-Local Address (LLA) per network interface and one or more global unicast addresses. A node may also have multiple multicast addresses, i.e., the MN may be a member of several multicast groups.

The standard way to represent an IPv6 address is by showing it as eight 16-bit hexadecimal words, separated by colons. For example: FEDC:BA98:0000:0000:0000:000C:2154:7313. Since a large number of IPv6 addresses contain multiple fields of only zeros, the notation can be compressed. These fields with only zeroes can be replaced by a double colon (::). Also, leading zeros may be omitted. The address in the previous example can thus be written as FEDC:BA98::C:2154:7313. An IPv6 (sub)net can be denoted by the starting address of the subnet and the size in bits of the address prefix. So, FEDC:BA98::/32 is the subnet in which the previous example lies.

With IPv6 there are different types of addresses. These are shown in Table 1.

Address type	Range
Link-local	FE80::/8
Global unicast	2000::/3
Multicast	FF00::/8
Loopback	::1/128

Table 1: IPv6 address types

The link-local address is only used for communication with nodes that are on the same link, for example the Access Router (AR). This address is formed by appending the interface's identifier to the link-local prefix, which is well-known (FE80::0). This identifier can be determined in several ways. The first is to use the EUI-64 (Extended Unified Identifier) identifier, as defined by the IEEE (Institute of Electrical and Electronics Engineers) [16]. If this is not available, it is also possible to use a link layer identifier, such as the MAC (Media Access Control) address in Ethernet. The 48-bit MAC address is then converted into a modified EUI-64 identifier by flipping the 7th bit and inserting 'fffe' between the 3rd and 4th byte of the MAC address.

The global unicast address is an address type that is routable throughout the whole Internet. There are several special multicast addresses. FF02::1 is used to address nodes on the link (link multicast address). The prefix FF02::1:FF00:0/104 followed by 24 bits of the MN's unicast address is the solicited node address. This address is used by the Neighbor Discovery (ND) protocol.

The loopback interface is only used on the host itself. It is the equivalent to 127.0.0.1 (or 'localhost') in IPv4. The unspecified address type has all bits set to zero.

2.1.2. Neighbor discovery

The neighbor discovery protocol [17] is used

1. to determine the link-layer addresses for neighbors on the same link (address resolution).
2. to find neighbor routers that can forward packets.
3. to determine which neighbors are still reachable and which are not. Changed link layer addresses are also detected this way.

Address resolution is done by multicasting a Neighbor Solicitation (NS) packet, asking the node with the particular IPv6 address to return its link layer address. This message is multicasted to the destination's solicited-node address. The recipient node then replies with a unicast Neighbor Advertisement message.

Sometimes sending traffic to a certain host through the default AR is not the most efficient way. To solve this, routers can send redirect messages to inform hosts that a better first-hop router is available for a specific destination. Or, in case the destination host is on the same link, to let the host know that the destination is in fact a neighbor, reachable on the local link.

It is possible for any IPv6 node to broadcast an unsolicited Neighbor Advertisement (NA) message, when for example its layer 2 address has changed. This causes all other nodes on the same link to update their address resolution cache.

2.1.3. Address configuration

With IPv6, it is possible to automatically configure an interface [18]. This process consists of creating a link-local address, verifying its uniqueness and determining what settings, e.g., global IP addresses and default AR, should be auto configured. Global addresses can be configured using a stateless or stateful mechanism. The stateless procedure can be used when the network is not concerned with exact addresses handed out to a host, as long as the address is unique and properly routable. When tighter control is required the stateful method can be used. Both methods can be used simultaneously. An address can be acquired using the stateless mechanism, while the stateful method provides hosts with other information about for example domain name servers.

Routers periodically send out Router Advertisement (RA) messages. These messages contain various link and Internet parameters, such as prefixes that can be used for address configuration. This prefix is used by hosts to configure a global address. If a host whishes to receive this information direct, rather than having to wait for a periodical message, it can send a Router Solicitation (RS) message to the all-routers multicast address. A router will reply to this by directly sending a RA to the originator of the RS message. The link-local address of the sender is used a destination for this RA. Both RA and RS are a specific type of Internet Control Message Protocol (ICMP) message.

Before any address can be used, the host has to verify that there is no other node using this address. This is called Duplicate Address Detection (DAD), which is explained further in the next section. The assigned IPv6 address has a fixed (possibly infinite) lifetime. When just obtained and its uniqueness verified, an address is in the ‘preferred’ state. Its use is then unrestricted. An address can become ‘deprecated’ later, meaning that using the address is discouraged. Ongoing connections can still use it, but new connections should use a ‘preferred’ address.

All this has to be done before a node can start sending or receiving any data. In section 2.5.1, DNAv6 is described, which speeds up the whole address configuration process.

2.1.4. Duplicate address detection

An important and time consuming part of the address configuration process is duplicate address detection. All unicast addresses a host has should be tested for uniqueness. Until DAD is performed successfully on an address, the address is in the ‘tentative state’. If the address is found to be unique, its state is changed to ‘preferred’.

When using stateless autoconfiguration, the uniqueness of an address is determined by the uniqueness of the interface identifier, which is appended to the subnet prefix to form a complete address. If we assume that these prefixes are assigned correctly and that a host uses the same identifier for all its addresses, we only have to perform DAD on one address. So, if the link-local address is found to be unique, we can skip DAD on all other addresses.

The basic operation of the DAD protocol is that the IPv6 node sends a NS message to the link-local address it wishes to use. If it is in use, the host using it will return a NA message. Other host attempting to use this address will also send out a NS message. If no reply is received within a given amount of time, a timer expires and the address is assumed to be unique. Because a node has to wait for this timer to expire, a large delay is introduced here before the node can actually start sending packets using this address.

In case of a handover, the host will have to perform DAD on the new link before it can resume its ongoing data transfer. So, the handover latency may be increased by having to perform DAD, which is not desirable. The rest of this subsection explains several improvements to the original DAD algorithm which were considered as alternatives to be used with SPMIPv6 to speed up the layer 3 association process. All these changes to the original DAD process have the same goal: minimize the DAD period.

Optimistic DAD

With normal DAD, an address is in the ‘tentative’ state when the uniqueness of this address is being verified. In this state it is not assigned to an interface in the usual sense. No IP packets can be sent using this address and packets with the tentative address as destination are simply discarded by the interface on the host that receives them. Optimistic DAD, which is described in RFC4429 [19], introduces a new address state called ‘optimistic’. This state is assigned to addresses whose uniqueness is being verified. An address in this state should be treated the same as an address that is in the deprecated state. This means that its use is discouraged, but not forbidden. The address should not be used for new communications, but existing connections can be resumed immediately after the interface is activated. Packets sent to or from this address are delivered normally. When DAD is performed successfully on this address, its state changes to ‘preferred’.

To use optimistic DAD, several alternations have to be made to the ND process. Since the address in the optimistic state is in fact used before its uniqueness is checked, it is possible that the address is already in use. Several changes have to be made to the address resolution process to make sure that when the address is already in use impact to the rest of the network is minimal.

- In NA messages, the ‘Override’ flag is not set. If the address is already in use, this will not corrupt the neighbor cache of other nodes, since existing records will not be overwritten. Ongoing communication with the node that is already using the address is not disrupted.
- NS messages, used to acquire a link-layer address of other nodes on the same link, are never sent using the optimistic address as source. This means that there is no direct communication between the node with the optimistic address and other local nodes on the same link. Traffic for other local nodes must go via the AR.

- RS messages, multicast messages used to request address configuration information, cannot contain the link-layer address of the sending node. This means that RA replies from the AR have to be a sent to a multicast address.

When Optimistic DAD is used after a handover, the whole DAD delay period is reduced to zero, since data transfer can be resumed immediately. When DAD is complete, the address moves to the preferred state, also allowing new connections to make use of the address.

Advanced DAD

Another addition is Advanced DAD [20]. This is a proposal that cannot be found in any standard yet. Here, ARs maintain pool of unused addresses. Addresses from this pool are silently removed when a NS for this address is received.

The operation works as follows:

- Mobile nodes waiting for a prefix send RS messages with as an option a duplication-free CoA (Care of Address) request. This term (CoA) is used in different mobile IP variants to denote an (temporary) IP address that can be used by the MN on a foreign (non-home) link, see section 2.2.1 for more information on this. Including this option in the RS means that the MN wishes to receive an address that it can use directly, without having to perform DAD.
- The RS includes MN's previous CoA and LLA.
- Upon receiving this, the access router
    - Select a free address from the pool
    - Creates an entry in its neighbor cache with selected address and LLA
    - Also sets a route to MN's previous CoA/LLA
    - Sends a unicast RA (to old CoA) which contains the new duplicate-free CoA as an option

The DAD delay period is eliminated when Advanced DAD is used. There is no need to perform DAD, since the address is guaranteed to be unique.

Proactive DAD

Another proposal is Proactive DAD [21]. This DAD procedure assumes that there is a Regional Information Point (RIP) server on each network. This RIP maintains Mobile-node Attachment Point (MnAP) tables that store information about which access routers are connected to which access points in the serving domain. It also knows the prefix that is advertised by each AR. Each RIP periodically exchanges its MnAP table with neighbor RIPs. When connecting to the network, a MN will receive all information currently stored in the RIP. It then knows all prefixes used on all access points within its area. It can use this information to determine if the access router it will move to is advertising the same prefix as it is using now.

The MN has some algorithm to determine when a handover will occur, and to which access point this will be. From the information in the RIP it can determine if it will also move to a new access router. If this AR is advertising a different prefix than the one in use, the MN will have to perform DAD on the new address. Normally this should have been done after moving

to the new access point. But, with Proactive DAD, the MN can send a CoA_preAllocate Request message to the new AR, while still attached to the old AR. The AR will determine now if the address is unique. It sends a reply to the MN letting it know if the address is unique. If it is, the MN can now send a CoA_activation Request message to the new AR, which indicates that it is going to use the address. Now the MN can move to the new access point and can start using the new address immediately.

2.1.5. Detecting network attachment

The Detecting Network Attachment [22] addition to IPv6 (DNAv6) helps MNs to regain layer 3 connectivity quickly after connecting to a new access point. This protocol is currently being developed and is described in an Internet Draft. The protocol assumes that a change in layer 2 connectivity invokes a trigger, which is processed at layer 3. Aside from that, several other improvements are also part of DNAv6. Optimistic DAD (see section 2.1.4) is used to get a unique address and minimize the delay of duplication address detection.

After the layer 2 trigger that indicates a link change, layer 3 has to determine if it is still connected to the same link. This happens for example when multiple access points belong to the same network domain. To easily facilitate this, DNAv6 enabled routers know all the prefixes that are advertised on the link. They have learned these by listening to RA's from other routers. The base action is now to respond to a RS by including all these prefixes in the RA. If one of these prefixes match the prefix the host was previously using, it knows the link has not changed. The MN does not have to do any duplicate address detection, since it is on the same link. These RA's can get large when the number of prefixes is large. Because of this the MN can also include a Landmark option in the RS, which is the prefix it is currently using. If the router knows this prefix, it replies with a simple RA only containing this prefix. If it is not known, the full RA is sent.

The last improvement concerns fast router advertisements. Normally, a router should delay the sending of a RA for a random amount of time to avoid all routers on the link replying to a RS at the same time. DNAv6 solves this be letting the routers generate a ranking. The first router in this ranking sends a router advertisement with 0 seconds delay.

2.2. IP level mobility protocols

This section describes the IPv6 mobility protocols that currently exist. First, host based protocols are described. With these, the MN takes care of his own mobility management. In the second part, network based protocols are described, in which the network is the coordinator of the handover.

2.2.1. Host based

Mobile IPv6

Mobile IPv6 introduces mobility support into IPv6. This enables nodes to maintain connectivity while moving around between different access links. Mobile IPv6 makes it possible for nodes to use the same IP address on different connections.

The address that the MN gets when it boots up in its home domain is called the 'Home Address' (HoA). It is always reachable through this address. When the MN moves away from the home domain, a Care-of-Address (CoA) is assigned to it. This address is acquired in the normal IPv6 way using stateless or stateful auto configuration. The MN then registers itself with a router on its home link (called the home agent (HA)) by sending a Binding Update (BU) to this HA. If the registration is valid, the HA replies with a Binding ACK. The HA now uses proxy Neighbor Discovery to represent the MN in the home network. A bidirectional tunnel is set up between the home agent and the MN. Packets coming from the MN are sent to the HA and then routed on the internet in the normal way [23]. This is shown in Figure 4. Correspondent nodes (CN) sending data to the MN will also see these packets going through the HA.

It is also possible to communicate directly with a CN, if this node also supports MIPv6. This is shown in Figure 5. To start doing this, the MN sends a BU to the CN. Packets that the MN now sends to the CN have the CoA as the source address. The CN receives these packets and replaces the source address with the HoA of the MN. Packets that the CN wants to send to the MN directly have the CoA as destination and the HoA (in a special routing header) as the second hop. In the MN the routing header is removed so that the upper layers only see the HoA.

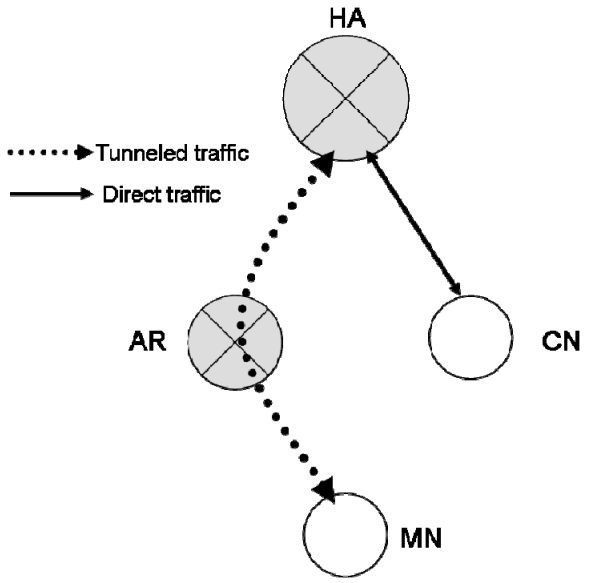

Figure 4: MIPv6: MN communicates with CN via HA

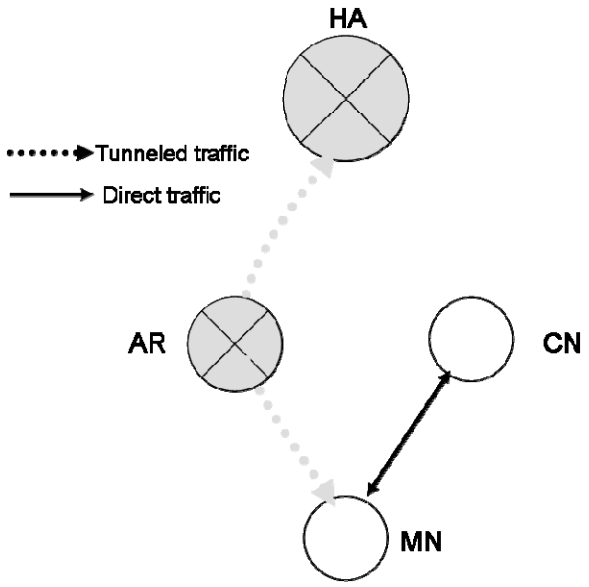

Figure 5: MIPv6: MN communicates directly with MIPv6 capable CN

Hierarchical mobile IPv6

Hierarchical Mobile IPv6 (HMIPv6) [24] was designed to reduce the amount of signaling needed between MN, CN and HA. It does this by separating global and local mobility. A Mobility Anchor Point (MAP) is introduced into the network. This MAP can exist at any level in a hierarchical network of routers. A MAP can span multiple subnets. A MN sends BU messages to the MAP, instead of to the home agent. The MN does not need to contact all CN's; all traffic is redirected after the one BU message is received by the MAP.

The MN has two care-of addresses in the HMIPv6 domain: the Local CoA (LCoA) and the Regional CoA (RCoA). The RCoA is used for communication with CN's and stays the same while connected to the same MAP. The LCoA (or on-link CoA) is used to communicate with the MAP. Whenever the MN moves to a different link in the domain of the MAP it obtains a new LCoA and has to register this address with the MAP. After a successful registration to the MAP by a MN, a bi-directional tunnel is set up. Packets sent by the MN have the LCoA as source in the outer (tunnel) header and the RCoA as source of the inner header. The MAP receives these packets from the MN, removes the outer header and forwards the packet towards the CN with the RCoA as the source address.

When the MN moves out of the domain of the MAP, it tries to find a new MAP on the new access link. If there is no MAP, normal Mobile IP is used. If there is a MAP, it registers itself and gets a new RCoA from the MAP. Its LCoA will also be updated.

Fast mobile IPv6

Fast Mobile IPv6 (FMIPv6) [25] tries to decrease the handover latency that is experienced when a MN moves from one access link to another. After the MN is 'IP-capable' on the new link, e.g. has a layer 2 connection and has a valid IP address, it can send a binding update to

the home agent and correspondent nodes. Packets only start arriving at the new CoA after a successful registration with home agent.

FMIP can work with both MN and network-initiated handovers. In the first mode, the MN can make use of layer 2 scanning techniques to identify other access point within its reach. While still connected to its current access router, it can already get information like other access routers L2 and IP address. This is done by sending a Router Solicitation for Proxy Advertisement (RtSolPr) message to its current access router, asking information about a certain access point identified by AP-ID. The reply to this is a Proxy Router Advertisement (PrRtAdv) message. With these messages it is also possible to already form a prospective new CoA (NCoA) that can be used when the MN moves to the new AR. This way, the latency due to the prefix discovery when connecting to a new AR can be eliminated.

When a handover occurs, the MN sends a Fast Binding Update (FBU) message; preferably to the AR it was previously connected to (the previous AR (PAR)). This is called 'predictive mode'. When the PAR receives this message, a Handover Initiate (HI) message is sent to the new AR (NAR). This is confirmed by a Handover Acknowledge (HAck) message sent by the NAR to the PAR. A tunnel is setup between NAR and PAR. Packets that arrive at the PAR after the MN disconnects there will now be sent using the tunnel towards the NAR. The NAR will deliver them to the MN when the MN will attach there. A Fast Binding Acknowledge (FBA) is sent by the PAR to the MN and the NAR to indicate a successful handover preparation. The MN can now connect to the NAR with minimal address configuration delay and packet loss.

The second mode is called 'reactive Fast Handover'. Here, the FBU cannot be sent on the old link. Instead, a FBU enclosed in a Fast Neighbor Advertisement (FNA) is sent to the NAR. The NAR now sends a FBU to the PAR, requesting the data for the MN to be forwarded. The PAR starts forwarding packets for the MN to the NAR and also sends an FBA.

Handover initiation can also come from the network itself. It is possible for AR to send an unsolicited PrRtAdv to the MN, when for example it knows a handover is imminent, or to inform the MN about adjacent networks.

Simultaneous bindings

Simultaneous Bindings [26] is an addition to FMIPv6. It tries to further minimize the packet loss the MN experiences. It does this by bicasting traffic for the MN to both its current location and the location it is supposed to move to in the near future. This feature was built into Mobile IPv4 [27] but is not present in Mobile IPv6. Simultaneous Bindings in FMIPv6 removes the timing ambiguity experienced with FMIPv6 regarding when to start forwarding traffic to the MN's new location. If this timing is off, data could be sent to the new location too late or too early, both resulting in extra packet loss.

When the MN receives a PrRtAdv message, it should immediately send a FBU message to its PAR with the S(simultaneous) flag enabled. This enables bicasting. The period the simultaneous binding should stay active is also specified in this message. The HA now sends traffic to both the PAR and the NAR. When the MN now moves to the NAR, packets for it are already available there.

2.2.2. Network based

Proxy mobile IPv6

Proxy Mobile IPv6 [28] tries to offer mobility to IPv6 hosts that do not have Mobile IPv6 in their stack. This is done by extending Mobile IPv6 signaling and also by reusing the home agent via a proxy mobility agent. With this approach it is not necessary for the MN to be part of layer 3 mobility signaling. The proxy mobility agent takes care of the MN's mobility management. This protocol can be used in networks that have both Mobile IPv6 enabled and non-Mobile IPv6 enabled nodes.

The following new nodes are introduced:

- Local Mobility Anchor (LMA)
  This entity is the home agent for the MN. The MN's prefix (HoA) is in the same network as the LMA.
- Mobility Access Gateway (MAG)
  The MAG takes care of the MN's mobility signaling. It tracks the MN on the access link. All traffic for MN's goes over a tunnel between MAG and LMA.

How these nodes connect is shown in Figure 6.

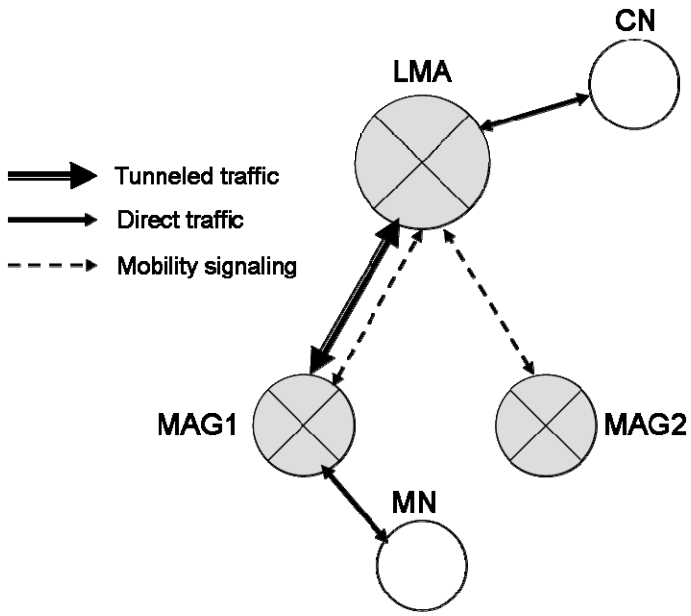

Figure 6: PMIPv6 network layout and communication

When a MN first attaches to a Proxy MIP domain, it contacts the MAG. It identifies itself with the MN ID. The MAG takes care of authenticating the MN. How this is done is not part of the specification. The MAG will now contact the LMA and register the MN. A tunnel is setup between the MAG and the LMA. The LMA will confirm the registration by sending an acknowledgment, which also contains the MN's home network prefix. The MN can now obtain an IP address using the normal router solicitation and router advertisement messages. The prefix advertised in the RA is the MN's own prefix which it also uses on its home link. The MN can send and receive traffic now. Periodic binding updates and router advertisements

are sent to make sure the MN is still connected and that the binding is still valid. All messages that are exchanged during this process are also shown in Figure 7.

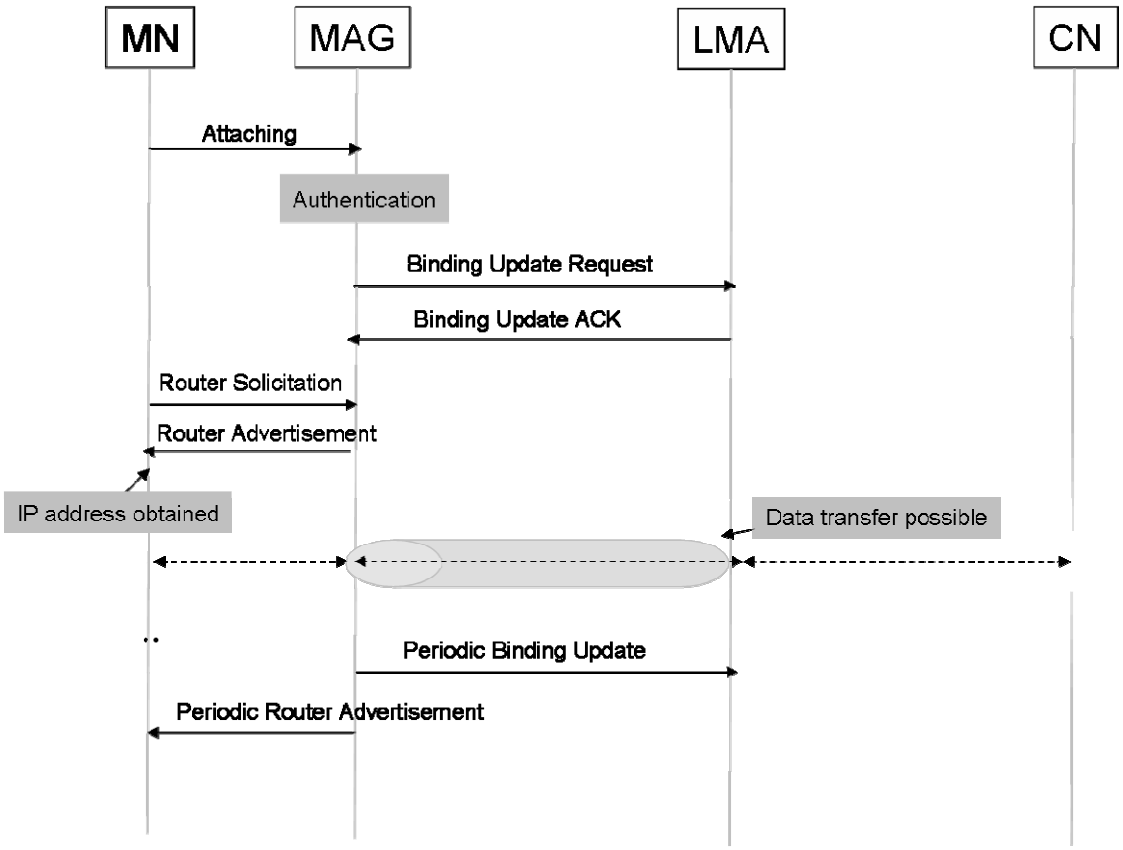

Figure 7: PMIPv6 boot up sequence

2.3. Related work

Mobile IPv6 offers mobility to IPv6 capable nodes. While this offers the capability of retaining connectivity while moving from one network to another, ongoing connections are still impacted by the actual handover between these two networks. Nodes involved in a handover experience a certain amount of handover latency: the period when there is no connectivity to neither of the two networks. During this period of time, packets destined for the MN are lost. This amount of packet loss is a direct consequence of the handover latency period. This handover latency is composed of several parts:

- Layer 2 setup delay. Before any packets can be received on the new connection, the lower level layers (such as 802.11 or UMTS) have to be connected. A study of the connection setup process of UMTS (see Appendix A) has shown that this delay is a big factor in the overall handover latency.
- Layer 3 setup delay. This delay consists of the delay caused by layer 3 address configuration.
- Registration delay. When in Mobile IP the MN moves to a new connection, after setting up layer 2 and 3, it has to register its new location with its home agent. This is done by sending a binding update message. This makes the home agent update its routing table in order to direct traffic for the MN to its new location. It is only after this registration is

complete that the MN can receive packets at its new location. This binding update message must also be sent to CN with which direct communication was established previously.

To make the handover process more seamless, the handover latency period has to be minimized. Several attempts have been done to do this. HMIPv6 tries to reduce the registration delay by introducing local and global mobility. If the MN moves within the local domain, registration messages do not have to be sent to the home agent, reducing this delay factor to 0. However if the MN moves out of the local domain, binding updates still have to be sent to the home agent. FMIPv6 reduces the layer 3 (address configuration) delay by determining the appropriate address settings on the new connection before the actual handover. It also has the possibility to let the old access router forward packets to the new access router. This forwarding does not reduce the handover latency, but fewer packets are lost this way. This forwarding is activated when the MN disconnects at its current access point (layer-2 handoff). It can be difficult in some situation to determine this exact moment. FMIPv6 and HMIPv6 can be combined, reducing both layer 3 setup and registration delay. By introducing simultaneous bindings, traffic can be forwarded to both the current and new access router. By using this the exact time of the layer-2 handoff does not have to be known for this mechanism to reduce packet loss.

Simulations have been done with several of these protocols. It was shown that a combination of HMIPv6 and FMIPv6 scores the best when overall handover latency is concerned [29]. This makes it the most suitable for real time traffic. Mobile IPv6 with simultaneous binding scores the best when a low packet loss is desired. Analytical results show that when the delay in the network is smaller than the delay on the wireless link, HMIPv6 performs better. This is because FMIPv6 uses more signaling between MN and AR, which traverse the slower wireless link [30]. If the delay in the network is larger than the delay on the air interface, a combination of HMIPv6 and FMIPv6 shows the smallest handover latency.

There is a proposal [31] that adds bicasting and buffering to FMIPv6 handovers. When a MN sends a fast binding update to its PAR, the PAR will start bicasting packets for this MN to the NAR. Packets are also delivered to the MN. The NAR buffers these packets for the MN until it moves over to this AR. Using this, no packets are lost when the MN moves to the NAR prematurely. A special header [32] is appended to bicasted packets. The MN can use this header to check for duplicate packets after it moves to the NAR and receives buffered packets there.

None of the protocols mentioned above attack all the three parts which make up the total handover latency. All protocols will perform poorly when the layer 2 setup delay is large. The next chapter will describe our proposal that is aimed at reducing all three parts of the handover latency period.

3. Protocol description

3.1. Introduction

This chapter will describe the SPMIPv6 protocol. It starts with a description of the protocol requirements and how they are met. Prerequisites and assumptions for the protocol to function are then explained. Messages formats and node operation conclude the specification.

3.1.1. Overview

Simultaneous binding Proxy Mobile IPv6 (SPMIPv6) is a network based mobility protocol. It is based on Proxy Mobile IPv6 (PMIPv6), which in turn is based on the Mobile IPv6 (MIPv6) specification. It enables hosts using IPv6 to change their point of attachment in the network with a minimized amount of service disruption. It does this by preparing for a handover before the actual disconnect at the hosts occurs. Both the network and the MN are part of the handover preparation process.

3.1.2. Objectives

The main objective of the protocol is to support seamless handover from the current point of attachment towards the future one. The handover latency, the period in which it is not possible for the MN to send or receive any data, should be no more than 50 ms. This way, applications like VoIP that are running on the MN can be continued after the handover without any noticeable service disruption.

The protocol supports both vertical and horizontal handovers, i.e. it is usable on top of any layer 2 technology. It takes a proactive approach towards the handover process to support long layer 2 setup times. It takes for example 1250 ms to set up a full UMTS connection (including layer 3 PDP context activation), which implies that a BBM strategy will be necessary. The proactive approach implies that a forecast should be made on the MN's movements. The protocol should be robust against faulty predictions, both when the MN moves differently than expected and when it does not move at all.

The protocol will be used with wireless access technologies. Since bandwidth can be scarce on air links, mobility signaling on these should be restricted to a minimum. Also, IP data packets should not be sent or received multiple times. This also means that the amount of buffering and resending packets to avoid packet loss in the handover latency period should be well tailored.

The protocol is based on existing protocols. This way, elements can be reused from existing protocol specifications and implementations.

3.1.3. Design

In order to achieve the objective of the maximum 50 ms handover delay, all factors that cause this delay have to be reduced. A large factor in this is the layer 2 setup delay. To be able to support even the largest layer 2 delay, we use a proactive MBB mechanism on layer 2 that prepares for an upcoming handover. This mechanism sends a message to the MN, called trigger 1, telling the MN to setup a layer 2 connection to the new access point. For this, the MN will have to be able to have 2 layer 2 connections active to 2 different access points at the same time. It also requires the MN to use additional processing power until the handover is completed. If it is a vertical handover, this means connecting to two different wireless technologies, something several devices might already be capable of. For example, some mobile phones have both UMTS and Wi-Fi interfaces. In a horizontal handover, the MN needs to connect to two access points of the same technology. This is more problematic, since for example a laptop may have only one Wi-Fi interface. Efforts have been made to create multiple virtual Wi-Fi interfaces on a single WLAN card [33]. If the MN is capable of this, it completely eliminates the layer 2 setup delay.

Trigger 2 (see Figure 3) marks the event of disconnection from the current access point. This could be in the form of a message sent from the current network to the MN or of a message produced within the MN when it observes disruption of the current connection. After a trigger 2 message, the MN disconnects from the old access point (if this has not occurred already) and starts the layer 3 attachment at the new access point, another factor of the total handover delay. We choose to use BBM on layer 3, as opposite to MBB on layer 2, because in this way the handover stays hidden from the application as the IP address does not change after a handover. If we used MBB on layer 3, the MN would have two different IP addresses active at the same time, between which the application had to switch after a handover. A solution would be to add some in-between IP layer in the MN, but this would only complicate the network stack on the MN. We choose to use BBM on layer 3 and speed up the association process instead to minimize the layer 3 break time. For this, we use DNAv6, which has been explained in section 2.1.5. Using this, the MN can continue sending and receiving data using the same IP address after 1 RTT between MN and access point.

After the trigger 1 message, the network also prepares for a handover. It already sends packets destined for the MN also to its future AR. This way, data is available for the MN immediately after the handover. This does however introduces extra load in the network, since this data is sent twice. This is not a direct problem since the bandwidth in the network is much larger than the bandwidth on the wireless link. However, if many MNs start using SPMIPv6 this might become a bit more problematic. More on this is in section 4.5.2. This preparation in the network also eliminates the third factor of the handover delay, the registration delay. Registration information for the MN can already be made present in the new AR to which the MN is about to connect, so the home network of the MN does not have to be contacted during the handover latency period.

When using normal MIPv6, the MN is responsible for its own mobility signaling. This means that a lot of mobility signaling messages traverses the air link. SPMIPv6 is a network controlled handover protocol, which means that mobility signaling on the air interface between the access point and the MN is minimized. All mobility signaling happens in the

network itself. Also, the MN does not have to spend processing power for processing any mobility signaling.

SPMIPv6 is based on PMIPv6, which in turn is based on the MIPv6. It reuses a lot of its terminology and can be implemented using PMIPv6 code as a starting point.

3.1.4. Protocol operation

SPMIPv6 reuses all the functional entities that are also present in PMIPv6. This means that these entities have the same behavior in SPMIPv6 and PMIPv6 when it comes to basic operations such as MN registration and data forwarding.

When a MN boots up in a SPMIPv6 domain, it register itself with the Mobile Access Gateway (MAG). The MAG now acts as a proxy on behalf of the MN while talking to the rest of the domain. The MAG functions as access router and is the MN's connection to the rest of the network. In its turn, the MAG then registers the MN with the Local Mobility Anchor (LMA). The LMA is the coordinator of the domain and has information about the MN in its database. This information includes the IPv6 address prefix the MN should use and other relevant data. When the registration is valid, the LMA sends this prefix to the MAG. The MAG then can forward this information in the form of a RA to the MN. The MN uses this prefix to configure a global address. The MN can now start sending data using this newly configured address.

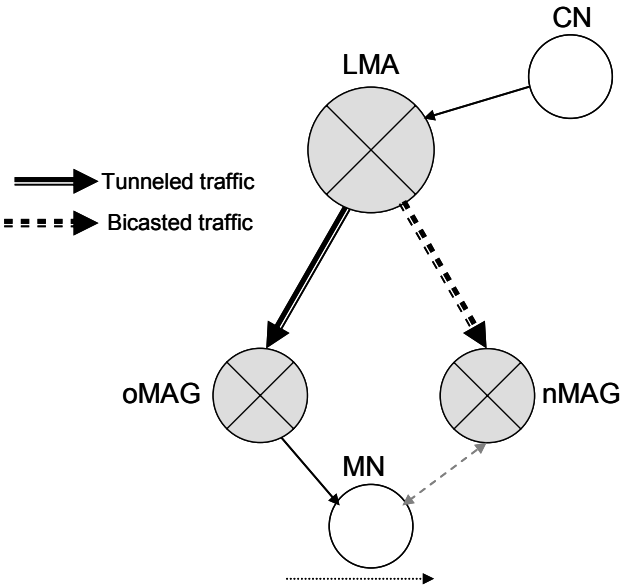

Figure 8: SPMIPv6 typical network layout

The main features that were added to PMIPv6 become clear when the MN is about to do a handover. We assume that the MAG to which the MN is currently connected (we will call this 'old MAG' (oMAG)) at some points knows that a MN will move away soon. How the MAG knows this is further described in section 3.2.2. The oMAG also knows where the MN will go to. It knows the address of the new MAG (nMAG) the MN will later on connect to.

When the handover is upcoming, the oMAG sends a message to the nMAG. This message contains information about the MN that is about to connect to the nMAG. The nMAG sends a

Simultaneous Binding Update Request (SPBUR) message to the LMA. This causes the LMA to start bicasting downstream data to both nMAG and oMAG for this particular MN. The nMAG starts buffering these packets for the MN now. These packets can be sent when the MN attaches there. This limits the amount of packet loss experienced by the MN during the handover latency period. All preparation for the handover is now in place. This is shown in Figure 8.

The actual handover starts when the MN is disconnected from the oMAG. The cause of this can be moving out of the coverage area of the oMAG. The oMAG can also instruct the MN to finalize the handover. After disconnecting, the MN directly sets up a connection to the nMAG. When this is done, the nMAG directly starts forwarding traffic for the MN. Optionally, the nMAG can also send buffered data to the MN. This further reduces the amount of data that is lost due to the handover.

3.1.5. Typical message exchange

In Figure 9 we see the typical message exchange for a handover in SPMIPv6. Initially, the CN is sending data to the MN. This data goes through the LMA, oMAG and then to the MN.

Trigger 1 comes from the oMAG and indicates to the MN that a handover to a certain nMAG will occur within a period of time. The MN is already in the coverage area of the nMAG and can thus set up a layer 2 connection to it. In parallel, oMAG sends a SPBUR to the nMAG. This inter-MAG signaling is a specific feature in SPMIPv6. The nMAG then sends a Proxy Binding Update Request (PBUR) to the LMA with the simultaneous binding flag set to 1, which causes the LMA to start bicasting traffic to both MAGs. This extra flag was added in SPMIPv6. Acknowledgements are sent to indicate the successful handover preparation. For this the LMA sends a Proxy Binding Update Acknowledgment (PBUA) to the nMAG. The nMAG sends a confirmation to the oMAG using a Simultaneous Proxy Binding Update Acknowledgment (SPBUA).

The actual handover is initiated by trigger 2. This trigger can come from the oMAG as shown in Figure 9, but the MN can also detect it on its own. Since the layer 2 connection to the nMAG is already in place, the MN sends a RS message to the AR, which in this case is the nMAG, for a prefix it can use. The reply to this, a RA message is then sent directly with (buffered) data for the MN. The data transfer is now resumed. Meanwhile the nMAG sends an update message (PBUR) to the LMA. The LMA now knows that the MN has connected to the nMAG. Data is no longer bicasted but is only sent to the nMAG. A PBUA is sent in reply.

The upstream data flow, that originates at the MN and has the CN as its destination, is also affected by the handover process. It is not possible for the MN to send data in the period between trigger 2 and receiving the RA from the nMAG. This data is not buffered in any point of the network, since the MN has no active connection in that period. Directly after layer 3 attachment the upstream data transfer is resumed. This data then goes from the nMAG to the LMA with the CN as its final destination.

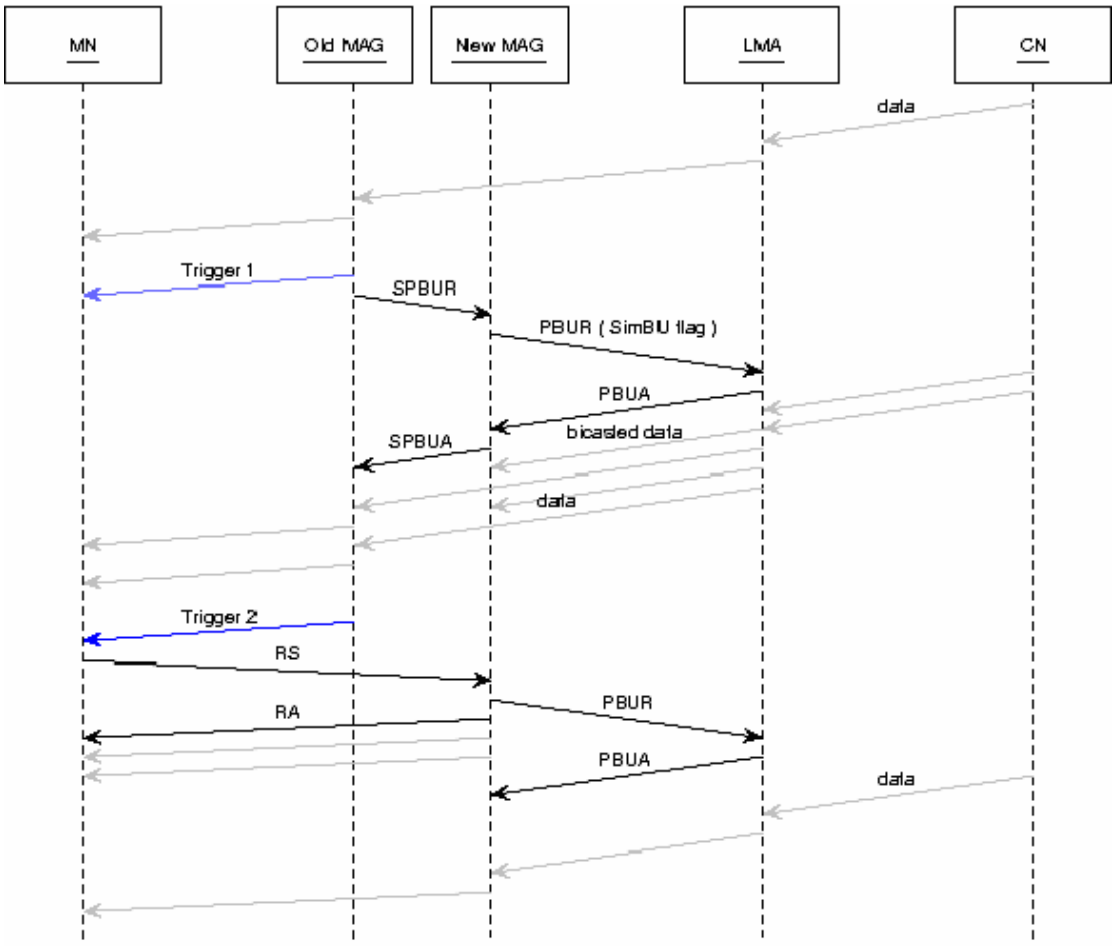

Figure 9: SPMIPv6 Handover message exchange

3.2. Preconditions

3.2.1. Triggers

The handover preparation part of the protocol is based on the existence of triggers: signals that indicate a change in network connectivity. These triggers are not directly associated with a certain layer 2 but are rather based on information available from any communication layer. These triggers are passed to other nodes in the network by the handover coordinator (the oMAG, see 3.2.2) via layer 3, for example using IEEE802.21 media independent handover services [34]. It is possible to define triggers for different kinds of events, such as link up, link down or for a drop a signal quality [35].

We use 2 triggers in the SPMIPv6 protocol:

- Trigger 1: This trigger indicates that the current connection might be lost somewhere in the near future. Preparations for a future handover should be done now, e.g., establishing L2 connectivity at the MN to the new access point, L3 handover signaling at the oMAG and nMAG, etc.

- Trigger 2: The actual handover is carried out now. This trigger, which sets off the MN for some actions, can come from the oMAG or can be detected at the MN itself. In most cases, the MN will detect a loss of connectivity at its current access point and will finalize the handover to the new access point. It is also possible that the coordinator of the handover, the oMAG, forces to MN to execute the handover. Reasons for this can be that the QoS offered at the current access point is not sufficient for the MN because of lower signal strength. The MN would only move to the new access point if the connection would be completely lost. This is because in this protocol the network, in the form of the oMAG, controls the handover process.

The time at which trigger 1 occurs relative to the actual handover (trigger 2) is very important to the execution of a seamless handover. The handover coordinator should be able to predict an upcoming handover. How this is done exactly is not part of this protocol description. A factor that could help to predict this is for example signal strength. If the signal strength drops under a certain level once or for some period of time, trigger 1 could be executed. The signal strength at other access point could also be known at the current access point or MN. The coordinator should also have an idea of:

- The time the MN needs to setup a layer 2 connection to the new access point.
- The time needed to enable bicasting in the network.

Using this, the coordinator can determine when a trigger 1 should be executed. There is a tradeoff however between executing trigger 1 too early, which means extra load in the network and extra processor load and battery usage on the MN, and too late, which means that the handover will not be seamless.

The point in time of executing trigger 1 also relates to certainty for which a handover will occur. If handover preparations are done far before the actual handover, the uncertainty in prediction of the handover is high. For example, the MN might just stay at the current access point, because signal strength remains high due to an unpredicted change of the MN's route. Trigger 1 should be executed enough in advance to allow all handover preparation to be done before trigger 2. These handover preparations include L2 setup to the new access point at the MN and enabling bicasting in the network. If this trigger 1 is issued way in advance, the probability of a false positive increases. A false positive in this situation means that handover preparations are done for a MN that does not move to another access point after all. The uncertainty of the prediction is high in this case. If it is indeed a false positive, a timeout will occur after an amount of time. This cancels the handover preparations. The SPMIPv6 protocol can handle these false positives gracefully. On the other hand, if the coordinator waits until it becomes almost certain that a MN will move to another access point, the time for handover preparations is probably small. This will result in a handover that is not seamless.

3.2.2. Handover coordination

Handover coordination is done by the MAG the MN is currently connected to, i.e., oMAG. This subsection describes roughly how oMAG can collect information about where the MN will move to and when. The end of this subsection describes what happens when a prediction is wrong.

In order to know the possible new points of attachments for the MN, the MAG somehow has to have a map of the physical network: it has to know what other access point are its neighbors. This map could be pre-defined, or built from some gathered data (for example recording MNs old and new access point). Technologies with both ubiquitous coverage, which can always function as a backup, and small coverage can all be part of this map. How this information is gathered is not described in the protocol specification. It is a necessity for the protocol to function however.

If the map is in place, it is still difficult to predict to which neighbor the MN will move. Another difficulty is to decide when the handover preparation must be executed. Signal quality measured at the access point could benefit both. The signal quality dropping below a certain level can indicate that the MN is at the edge of the coverage area. Trigger 1 can then be executed. Signal quality however is not a very reliable indicator, since it is possible that the signal is lost all of a sudden (moving into a radio signal shielded room), without time to do any preparations. The MN might also be able to receive beacon signals from other access points within its range. This could be communicated with their oMAG, which could use this information to predict the new access point the MN will connect to. This would transform the protocol into a network controlled but MN assisted mobility protocol.

A simultaneous binding is active for a certain amount of time. If the actual handover does not occur within that period, the LMA stops the bicasting. The nMAG also knows the lifetime of the binding, so after the binding times out, the nMAG removes all state information for the MN and empties the buffer. This might happen if the MN does not move out of the coverage area of the oMAG after all. If the MN moves to a different MAG than expected, the behavior of the protocol is equal to that of PMIPv6 (i.e. without any form of handover preparation). The nMAG will send a PBUR (without the simultaneous binding flag) to the LMA to register the MN. This will cancel all other bindings. An analysis of what happens when there is not enough time between trigger 1 and trigger 2 to setup the layer 2 connection to the new access point is given in section 4.4.

3.2.3. Mobile node capabilities

SPMIPv6 is a network based mobility protocol. One advantage of this protocol being network based is that the MN does not need to have any mobility code in its stack. However in order to make the handover more seamless it is in some cases desirable for the MN to also have some mobility enhancements in its stack. For example, the handover process can benefit from the MN having multiple interfaces. These interfaces can be multiple separated physical interfaces, for example for Wi-Fi and 3G (UMTS), or virtual interfaces mapped onto one physical interface. The interfaces can be distinguished by the LMA by their different virtual or hardware addresses. When the MN connects its second interface to the nMAG, it has to perform the same authentication as with its first interface. The MN ID is used here to identify the MN.

If the MN has two interfaces, handover preparations can be initiated by sending the trigger 1 message to the MN. After receiving this trigger the MN can already setup its second interface on the link layer level. This means delaying any layer 3 configuration until the actual handover. When the actual handover happens now, the router solicitation message can directly be sent without having the layer 2 connection setup delay.

There are some difficulties with implementing this however. The application running on the MN should not be aware of this interface switch, since reconnecting on a different interface would disrupt the whole data transfer for some time. So, if a node is capable of having two layer 2 connections active at the same time, there should be only one layer 3 visible to the higher layers. The MN should also use the same address on both the interfaces, but not at the same time. A way to solve this is to completely hide the existence of the multiple interfaces to the higher layers at the MN. An in-between layer could hide this and make sure that the MN only uses its designated IP address on one of its interfaces.

3.3. Node operation

This section describes the functionalities of the different nodes. Tasks marked with a 'minus' sign are also present in PMIPv6. Tasks with a 'plus' sign are added in SPMIPv6.

3.3.1. Local mobility anchor

The Local Mobility Anchor (LMA) takes care of the following tasks:

- MN registration: The LMA holds information for the MN's in its database.
- MAG registration: When MAGs boot up in the SPMIPv6 domain they register themselves with the LMA. The LMA should check if the MAG is part of its domain and save information about the MAG for later usage.
- Binding update processing: PBUR request messages coming from MAGs should be checked and processed if valid.
- Acknowledgment generation: In response to the PBUR message, acknowledgments should be generated (PBUA message) and sent.
- Forwarding MN traffic: The LMA is the router that is responsible for the MN's home address. All traffic for and towards the MN flows through this node.
+ Processing the Simultaneous Binding flag in the PBUR message.
+ Bicasting traffic to both oMAG and nMAG: As part of the handover preparation, the LMA should send traffic for a certain MN to both the MAG it is currently connected to (the oMAG) and the MAG it is about to move to (the nMAG).

The operation of the LMA is described in the form of a Finite State Machine (FSM) in Figure 10. All edges represent a state transition. Each edge is labeled with a fraction in which the nominator represents the input message (or action) that causes the transition. The denominator shows the message that is sent in reply. Some edges have multiple fractions, which means that multiple input events can cause a movement along this edge, with different output messages as a result.

In the idle state, no MN is connected to the SPMIPv6 domain. After a MN connects to a MAG, this MAG will send a PBUR message to the LMA. The LMA receives this message and sends an acknowledgments message (PBUA) back to the MAG. From now on, traffic for and from the MN goes through the LMA. The LMA is in the 'connected' state. From this state, three things can happen. First, the MN can disconnect from the SPMIPv6 domain. The LMA notices this by receiving a PBUR message from a MAG with the lifetime set to zero. This indicates that the binding can be deleted and thus the LMA returns to the idle state.

Second, if a MAG wishes to extend the lifetime of a binding or if a MN moves to another MAG directly, without any handover preparations, the LMA will receive a PBUR message. Data about the MN is updated and the LMA stays in the same state. The third possible transition in the connected state occurs when a PBUR message is received from a MAG with the simultaneous binding flag enabled (S=1). The LMA then moves to the bicasting state. A PBUA is sent back to the MAG in response. From this state it is possible to go back to the connected state if a PBUR message is received from a MAG, which indicates that the MN has moved to a new MAG. This PBUR can come from the MAG the MN was supposed to go to according to the handover preparation or a different one, in case of a faulty prediction. A PBUA message is again sent in reply. If the PBUR message is not received by the LMA within a certain period of time, a timeout can occur which also stops the bicasting.

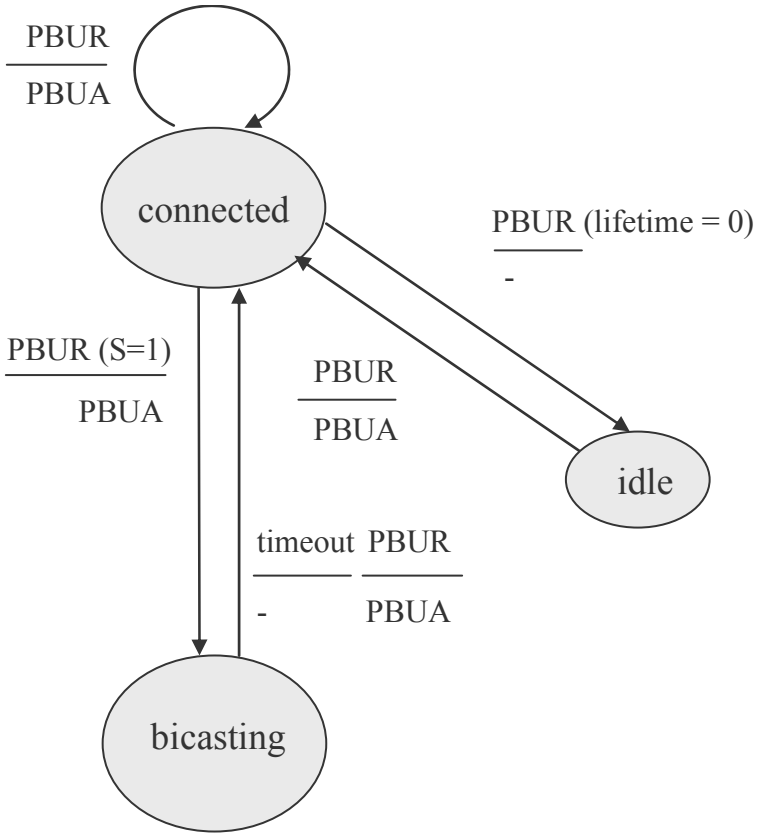

Figure 10: FSM of LMA operation

3.3.2. Mobile access gateway

The MAG has the following functionalities:

- Registration to LMA: When the MAG boots up it should register itself with the LMA.
- MN registrations: MN starting up will first register themselves with the MAG.
- Sending Binding Update Requests: Mobility signaling on behalf of the MN.
+ Handover coordination: In the protocol, the MAG is responsible for handover coordination.

+ Simultaneous Binding Update Requests processing: Part of the handover coordination is the sending and processing of Simultaneous Binding Update Requests, sent to and coming from other MAGs.
+ Receiving bicasted traffic for MN's that will connect to the MAG in the future. All this traffic should be buffered. The lifetime of the packets in the buffer determines how much data is sent to the MN when it attaches.

The behavior of the old and new MAG is different. They are described in two different FSM's in the following.

The operation of the oMAG is described in the FSM of Figure 11. The oMAG starts in the 'connected' state, since it already has a MN connected to it. The oMAG is the coordinator of future handovers. Certain events and/or info may indicate that a MN is about to do a handover. This causes the oMAG to send trigger 1 message to the MN. At the same time, a SPBUR message is sent to the nMAG. When it receives this, the nMAG will send a message to the LMA requesting bicasting of traffic for this specific MN. The oMAG is now in the 'HO initiated' state. When a trigger 2 occurs, it goes to the 'idle' state, since the MN is no longer connected.

This trigger 2 event can resemble both the sending of a trigger 2 message to the MN, in which case it is an output event, as well as the disconnecting of the MN at the current access point, in which case it is an input event. Both are displayed in the graph. The decision of sending a trigger 2 message is based on external events or info.

If the handover does not execute in time, a timeout makes the oMAG go back to the 'connected' state.

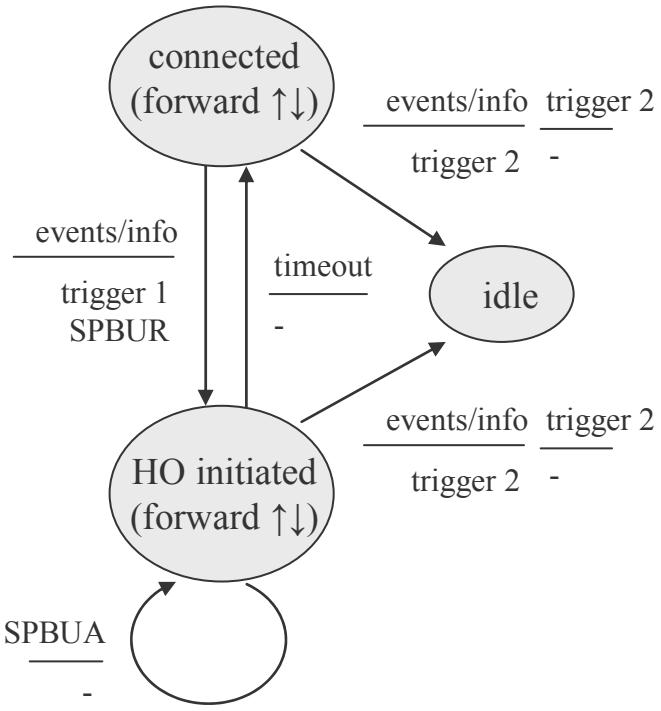

Figure 11: FSM of oMAG operation

The FSM of the nMAG is shown in Figure 12. It starts in the 'idle' state. Two things can happen here. If a MN suddenly connects to this MAG, without any preparation, the nMAG receives a RS from it and goes to the 'waiting for PBUA' state. A PBUR is sent by the nMAG to the LMA to register the MN. After this PBUA message is received from the LMA, the MN's home network prefix is known and a Router Advertisement (RA) can be sent to the MN. The nMAG is now in the 'connected' state, which is equal to the 'connected' state in the FSM of the oMAG. The other possible transition from the 'idle' state is that towards the 'handover initiated' state. This happens when the nMAG receives a SPBUR message from another MAG, which indicates that bicasting should be enabled for a certain MN. To achieve this, a PBUR, with the simultaneous binding flag enabled, is sent to the LMA. After receiving a PBUA from the LMA the bicasting is active and the nMAG moves to the 'buffering' state, in which the packets destined for the MN will be buffered. Also, a SPBUA is sent back to the oMAG. If the MN moves to the nMAG, it sends a RS message to it. In reply, the nMAG sends a RA message back to the MN. A PBUR message is also sent to the LMA to let it know the MN has moved to this MAG. If there is not enough preparation time (the time between trigger 1 and trigger 2), the MN can already move to the nMAG before the bicasting is enabled in the network. This is denoted by receiving a RS (from the MN) in the 'handover initiated' state. A PBUR is sent to the LMA to register the MN and a SPBUA is sent to the oMAG, in reply to the SPBUR message. The nMAG has to wait for a PBUA from the LMA again before it can send a RA to the MN and move to the 'connected' state.

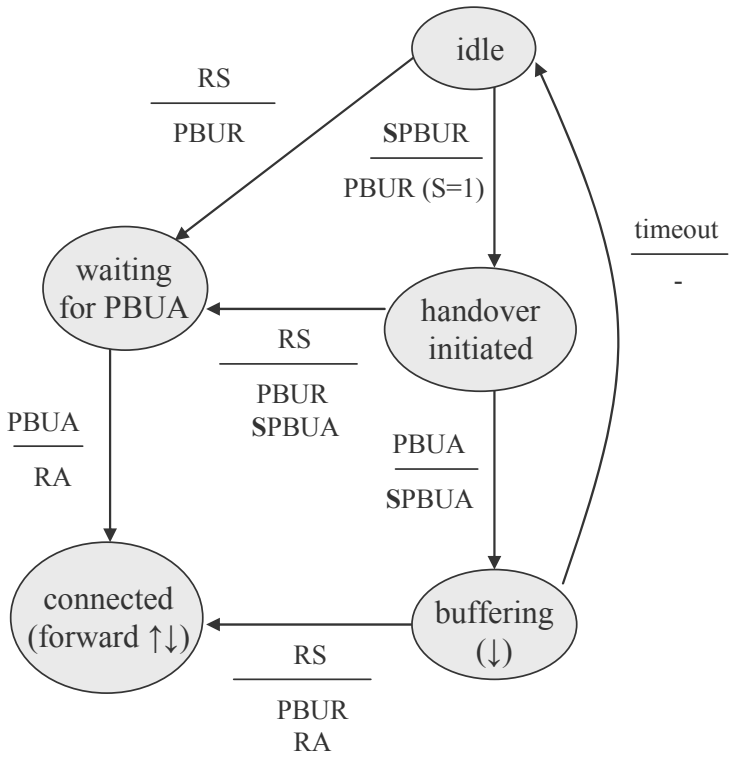

Figure 12: FSM of nMAG operation

3.3.3. Mobile node

The MN takes care of the following functions:

- Initial connection setup: Booting up in the mobile IPv6 domain, registering with the MAG.
+ Handling triggers: Receiving and processing trigger 1 and 2 messages received from the MAG. Processing a trigger 1 message means setting up a layer 2 connection to the new access point. After receiving a trigger 2 message the handover should be finalized. It is also possible to finalize a handover after loosing the connection with the current access point.

The operation of the MN is shown in the FSM of Figure 13. From the idle state it is possible to go to the 'connected' state by the 'Connect' input action, which denotes that the MN wishes to connect to the SPMIPv6 domain. As an output, the MN sets up a layer 2 connection to the MAG and sends a RS to the MAG to configure its layer 3 address. The MN is now in the connected state. A trigger 1 can now be received, which indicates that preparations for a future handover should be done. In this case this means that a layer 2 connection should be made to the nMAG. If there is enough time to do this, the MN reaches the 'Handover preparation' state. It can receive a trigger 2 now which means that it should move to the nMAG. A RS is sent after this to the nMAG to configure the MN's layer 3 address. If there is not enough preparation time or if there is no trigger 1 at all, the MN can receive trigger 2 in the 'connected' state. It does change MAGs then, but it remains in the 'connected' state. A RS is sent since its point of attachment has changed and the layer 3 address configuration has to be redone.

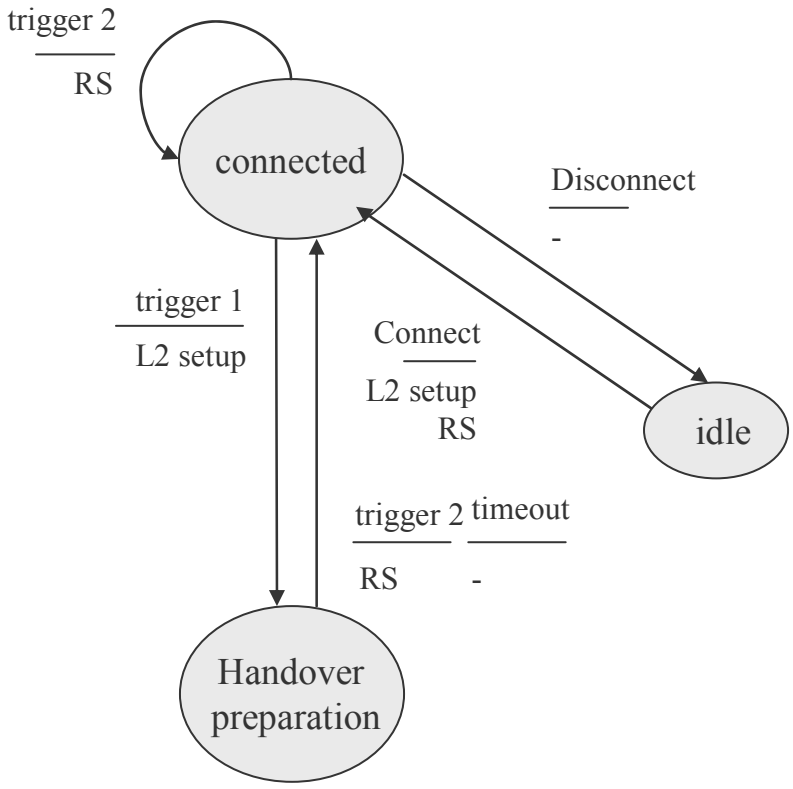

Figure 13: FSM of MN operation

3.4. Message formats

This section describes the messages that are sent in the SPMIPv6 protocol. The PBUR and PBUA messages have the same format as in PMIPv6. One flag is added to the PBUR and PBUA messages. The SPBUR and SPBUA message for inter-MAG signaling are new in SPMIPv6.

Each message can have a different number of options. These options are also described.

3.4.1. Proxy binding update request

This section explains the format of the Proxy Binding Update Request (PBUR) message. In Figure 14 the exact format is shown. The first row shows the byte number, the second shows the bit number. Each next row specifies the rest of message. This display method is used for all messages.

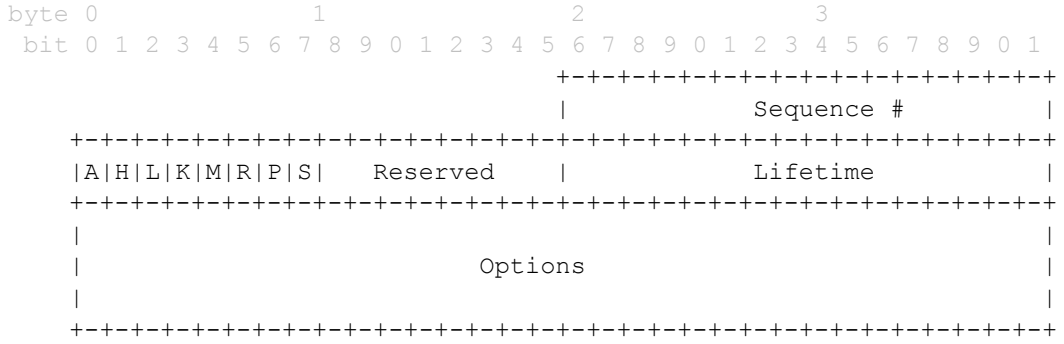

Figure 14: PBUR message format

This message is the same as the one in the PMIPv6 specification, with the flag ('S') added to it. The PMIPv6 specification is in turn based on the Mobile IPv6 specification, RFC 3775 [11].

The following fields are in this message:

- Sequence #: The sequence number is used to identify the transaction. An acknowledgment sent in response to this message will have the same sequence number.
- Flags:
  - The P flag indicates that this binding update originates from a proxy mobility agent, and not from a MN itself.
  - The S flag is the addition to the PMIPv6 message format. This header indicates that the requesting MAG wants the LMA to setup a simultaneous binding. The new binding can coexist with other existing bindings.
  - The other flags are described in RFC-3775.
- Reserved: Reserved space.
- Lifetime: Indicates the lifetime of the binding. If the S flag is set, this value indicates the time the actual bicasting should be activated (i.e. the timeout parameter in the FSMs of Figures 12, 13 and 14).
- Options: Home network prefix, Mobile Node Identifier, Timestamp. These options are described in section 3.4.5.

3.4.2. Proxy binding update acknowledgement

The format of the Proxy Binding Update Acknowledgement (PBUA) message is shown in the next figure. Again, the S flag is added in the SPMIPv6 protocol with respect to the original message used in PMIPv6. The prefix the MAG should advertise for this MN is attached as an option.

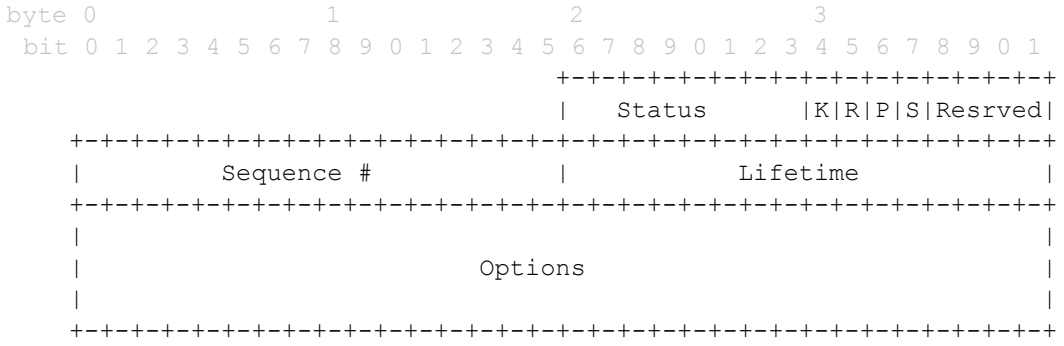

Figure 15: PBUA message format

The following fields are in this message:

- Status: The return code of the binding update. Status values less than 128 indicate that the BU was processed successfully. Values of 128 and higher indicate an error.
- Flags:
  - Again, the P flag indicates that this binding update originates from a proxy mobility agent, and not from a MN itself.
  - The 'S' flag indicates it is a reply to an update request asking for simultaneous binding. This is copied from the request.
  - The rest is according to RFC3775.
- Sequence #. Equal to that of the binding update request message that triggered sending this acknowledgement.
- Lifetime: The lifetime for which the MAG should keep entry for this MN in its Binding Cache.
- Options: Home network prefix, Timestamp.

3.4.3. Simultaneous proxy binding update request

The Simultaneous Proxy Binding Update Request (SPBUR) message is exchanged between MAGs. This inter-MAG signaling is new in SPMIPv6. The format is shown in Figure 16.

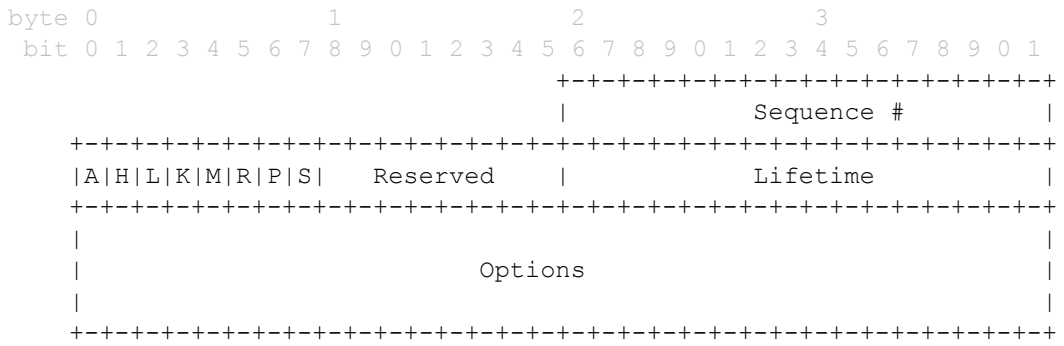

Figure 16: SPBUR message format

This message is equal to a normal PBUR (Proxy Binding Update Request) with the S flag enabled. The functions of the Lifetime field differs slightly however. It defines the lifetime of the *simultaneous* binding (a.k.a. the actual bicasting). After this lifetime a timeout will expire which cancels the bicasting. This value is determined by the oMAG.

The normal PBUR message is sent from MAG to LMA, but this one is for inter-MAG communication. It has a different set of options.

Options: Mobile Node Identifier, Timestamp.

3.4.4. Simultaneous proxy binding update acknowledgement

The Simultaneous PBU Acknowledgement (SPBUA) format is shown in the next figure. The format of this message is the same as the PBU Acknowledgement sent from LMA to MAG. The only option in this message is the timestamp option.

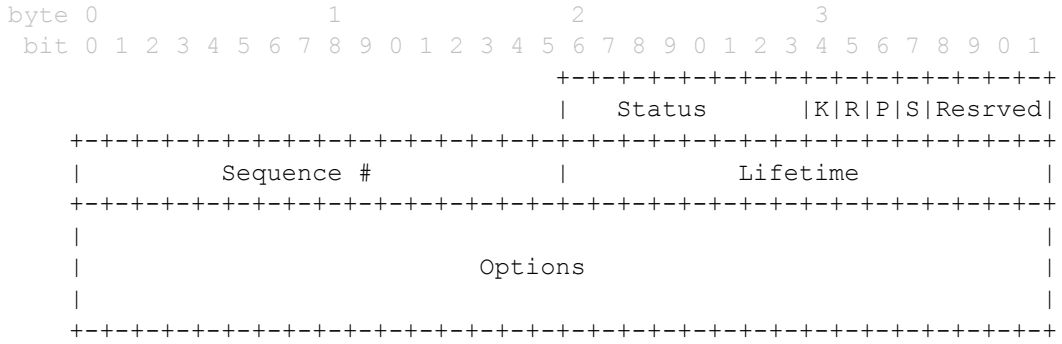

Figure 17: SPBUA message format

Options: Timestamp

3.4.5. Options

This section show the format of the options that can be appended to the four messages described earlier. The type field in the option fields are IANA defined [36]. The actual message format is copied from the PMIPv6 specification.

Mobile node identifier option

The MN needs a way to identify itself, other than with its default home prefix. The Mobile Node Identifier (MNID) [37] option is used for this. It can be used for authentication and authorization of the MN. The format of this message is shown in Figure 18.

The MNID can hold different kinds of identifiers. This is indicated in the 'subtype' field of the option. One subtype is that of the Network Access Identifier (NAI) (subtype 1). It uses an identifier of the form user@realm [38].

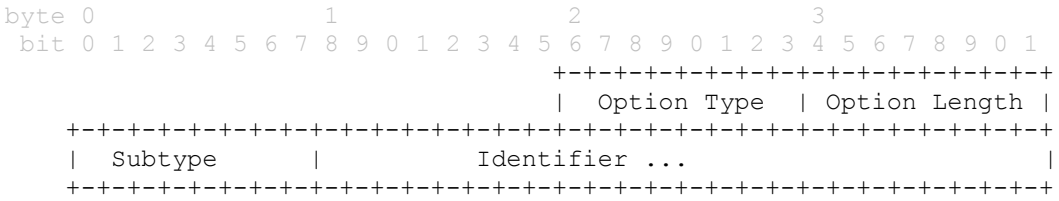

Figure 18: NAI option format

- Option type: As defined by IANA (8)
- Option length: The length of this option.
- Subtype: Type of identifier used.
- Identifier: The actual MN identifier.

Home network prefix

The home network prefix option is used to include the MN's prefix in messages exchanged between MAGs and MAGs and LMA.

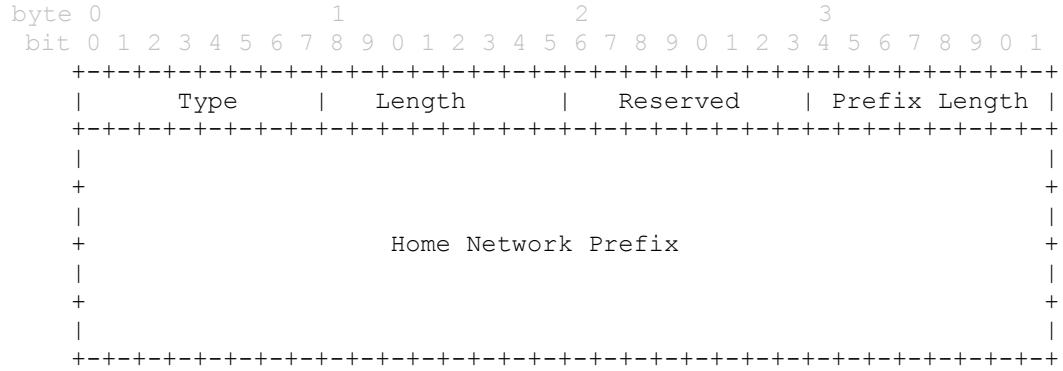

Figure 19: Home network prefix option format

- Type: As defined by IANA (6)
- Length: Option length. Must be set 18.
- Reserved: Unused. Must be initialized at the sender to 0.
- Prefix length: Indicates the length of the prefix as specified in the home network prefix field.
- Home network Prefix: contains the home network prefix.

Timestamp option

The original Mobile IPv6 specification uses the sequence number field to process binding updates in the order they were sent. These were normally sent by the MN. The LMA (called Home Agent in MIPv6) and MN manage this counter over the lifetime of a certain binding. In PMIPv6 however, the MAG handles the mobility signaling for the MN. When the MN moves to a nMAG, there is no mechanism to do a context transfer from the oMAG to the nMAG. So, the sequence number is not usable with PMIPv6.

With SPMIPv6 it would be possible to do context transfer, since MAGs already communicate. There are some issues however. When a handover does not go 'as planned', for example when the MN suddenly moves out of the coverage area of the MAG without any preparations or when the MN moves to a different MAG than expected, it is difficult to do a

context transfer. The nMAG does not know where the MN came from, so the address of the oMAG is not known. The oMAG does not know where the MN went in these cases either. The sequence number is used in the first message (PBUR) that the nMAG sends to the LMA, so the context transfer has to be executed before the MN can get any connectivity. Some of these issues could be solved, but for now the timestamp option can be used with SPMIPv6.

In order to use the timestamp option, it is necessary that hosts synchronize their clocks, for example using NTP [39]. If the timestamp option is not present nodes processing SPMIPv6 messages must fall back to using the sequence numbers.

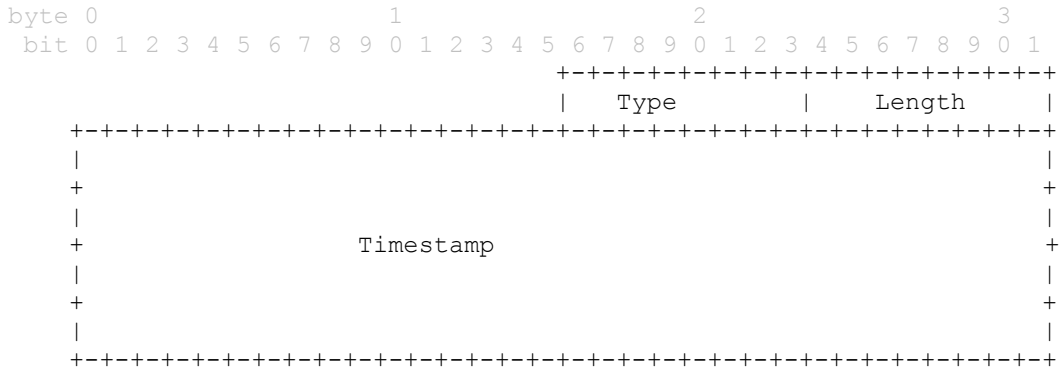

Figure 20: Timestamp option format

- Type: As defined by IANA (not defined yet).
- Length: Option length, must be set to 8.
- Timestamp: The actual UNIX timestamp (seconds since 01-01-1970 00:00).

4. Performance evaluation

In order to validate the operation and evaluate the performance of the SPMIPv6 protocol, we implemented it in a testbed. This way, the execution of the protocol could be closely watched in an almost real life scenario.

This chapter first explains the structure of the testbed, describing both the hardware and the software. Then the actual implementation of the SPMIPv6 protocol is described. The results of the performance evaluation test runs are then presented. Finally, an analysis of possible scenarios which could not be executed in the testbed is given. This analysis also includes a description of the extra load that is created in the wired links of the network because of the SPMIPv6 protocol.

4.1. Testbed

We used an existing testbed to show the performance of the SPMIPv6 protocol. This testbed was originally built at DOCOMO Euro-Labs [40] in Munich to show the performance of the NETLMM [41] protocol, which is almost similar to the normal Proxy MIPv6. The software already running on the testbed was used as a basis to which the SPMIPv6 specific functions were added. The rest of this section describes both the hardware and software of the testbed in detail.

4.1.1. Hardware

The testbed consists of 7 laptops (see Figure 21). All laptops are IBM Thinkpads and have integrated WLAN (802.11g, 54mbit) and gigabit Ethernet. One of these acts as the central server; this one is called NLSM, an acronym used for unknown legacy reasons. This laptop is not part of the SPMIPv6 domain, which means that it does not use the SPMIPv6 protocol to communicate with other nodes. It has Internet connectivity through the experimental company network. The 6 other clients, labeled as nlsm0 to nlsm5, are diskless and all start up from a boot image that is stored on the central server. For this, all laptops are connected using gigabit Ethernet.

Figure 21: Picture of the testbed

There are two other laptops outside the SPMIPv6 domain: the monitoring node (nlms5) and the corresponding node (CN, nlsm0). The first one logs all traffic that is being sent in the SPMIPv6 domain, both on the wired and wireless links. TCPdump [42] is used for this. The monitor node has two WLAN cards that are both in promiscuous mode, so they receive all (raw) packets being sent by all the WLAN nodes, even if they are not destined for the monitor node itself. The first WLAN card is on the same Wi-Fi channel as the MAG1. When the MN is connected to this MAG, all data is logged by this WLAN card. If the MN connects to MAG2, the second WLAN card in the monitor node logs all data transfers, since it is on the same channel as the MAG2 access point is. All packets being sent are thus recorded and will be decrypted later on to allow further inspection. The monitoring node is also physically connected to the CN using a hub. This way, it is also possible to see what data the CN node is sending, since it is not using the Wi-Fi interface as the other nodes are. For this, on IP level, the monitoring node is configured to be in the Virtual Local Area Network (VLAN) 30 between CN and MN. From all the data that this monitor collects it is possible to determine the exact time a layer 2 Wi-Fi connection is (re)established or when an RS message is sent by a MN and received by a MAG.

The other laptop outside the SPMIPv6 domain (i.e., the one acting as the CN) is a node that is communicating with the MN and is unaware of the whole SPMIPv6 domain. It is in a VLAN (VLAN 30) with the LMA and the monitoring node. All traffic the CN has for the MN enters the SPMIPv6 domain via the LMA. There is an IPv4 tunnel between the CN and the MN because in a previous setup of the testbed there was an IP camera attached to the network which would only operate using IPv4. This tunnel was still in place, although it was not strictly necessary in our experiments with the testbed.

The other 4 laptops have the function of LMA, MAG (2 nodes) and MN. Those laptops all run the Debian Linux distribution with kernel version 2.6.18.2. Although all laptops are

connected using Ethernet for configuration purposes, inside the SPMIPv6 domain the MN and MAG's are communicating using Wi-Fi (802.11g). The LMA and MAGs use a VLAN on the Ethernet connection (VLAN10), which simulates the backbone network. The whole setup is displayed in Figure 22.

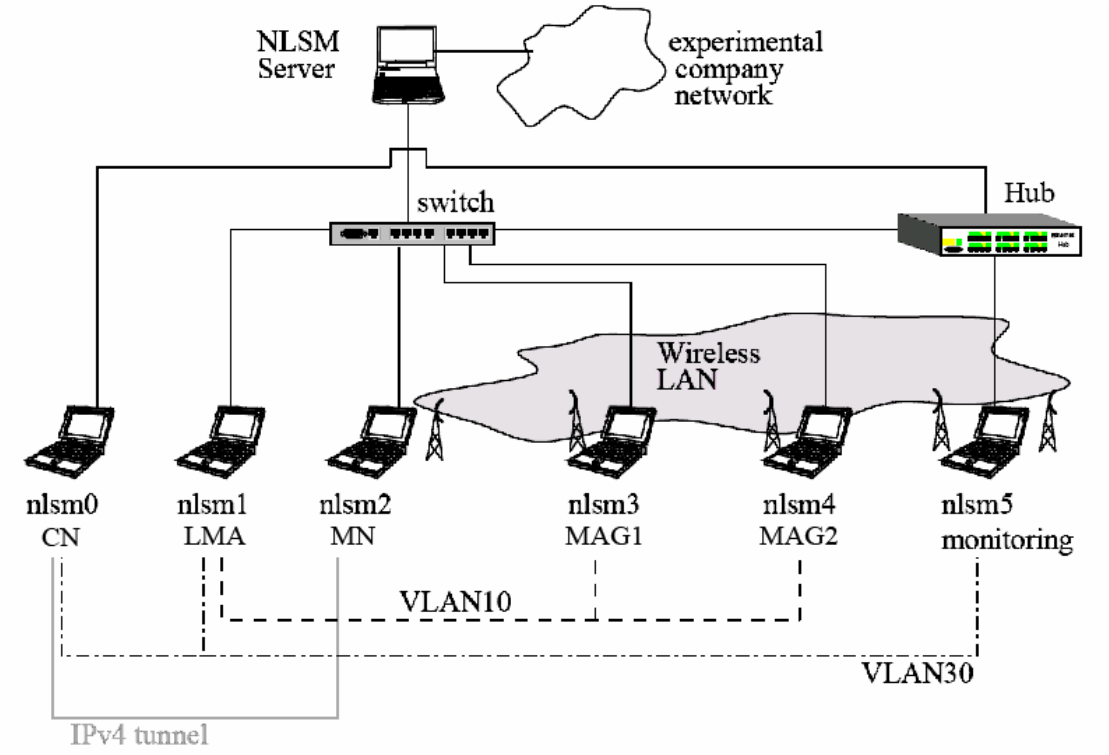

Figure 22: Testbed setup

4.1.2. Software

The existing testbed was running an implementation of the NETLMM protocol, based on an early draft. This protocol was developed at the same time as Proxy Mobile IPv6 and differs mainly in terminology. The protocol messages are the same. The implementation consists of user-space programs (daemons) implementing most of the protocol, some patches for the Linux kernel and a few patches for system administration tools that control networking in Linux. Everything is based on IPv6. The wireless interface is managed by the Madwifi driver [43] for the Atheros chipsets. This driver can run in both master and managed mode. The first is used in the MAGs, letting them act as an access point, where the second is used on the clients. The MAG and the access point (the Wi-Fi interface) are collocated on the same laptop. SCTP [44] is used as the transfer protocol for sending NETLMM/PMIPv6 messages. This protocol offers reliable data transfer and is used because it simplified implementing the messages exchanged in the NETLMM/PMIPv6 protocol. MAGs and the LMA are listening on specific IP ports for protocol messages.

Since all nodes are connected to the same switch using Ethernet, VLANs are used to separate traffic for different parts of the test network. A separate VLAN (id 10) is set up for the backbone of the test network, consisting of the LMA and the two MAGs. The LMA and CN are also connected in a separate VLAN (id 30). The third node in this VLAN is the monitor

node. A separate IPv4 tunnel is in place between CN and MN because of the legacy purposes mentioned. The LMA has a global IPv6 prefix which is divided into individual per MN subnet prefixes. Each MAG will announce the specific prefix to the MN using a Router Advertisement (RA).

After a handover to the other MAG, the MN will use the DNAv6 protocol. It will send a Router Solicitation (RS) to the All-routers multicast address, which is `ff02::2`. This RS is enhanced with the landmark option, which means that it includes the prefix the MN was using on its previous link. The MAG assumes that this is a correct prefix and already sets up routing for this prefix. This means that the MN can already receive and send packets on the new link before it receives the Router Acknowledgement (RA) which confirms its prefix. In the testbed, the nMAG first has to request the MN's prefix at the LMA. In the SPMIPv6 protocol, the nMAG already knows the prefix after a successful handover preparation with oMAG and LMA. In the testbed, after the MAG receives the home network prefix from the LMA, it can send a RA announcing the MN's home link prefix. This prefix is also already in use before the handover, so the MN will conclude it did not change link and will not reconfigure its IP address.

The MN, moving between different access points, is using a standard IPv6 stack, since all mobility signaling is done in the network. The MN is using the standard neighbor discovery and stateless address auto configuration procedures. The LMA maintains a route for the MN's prefixes, pointing to the MAG it is currently connected to. Binding updates are sent by MAGs to the LMA when a new MN connects there. The prefix the MAG should advertise for the MN is included in the reply acknowledgment message. When a MAG detects a MN is disconnected, it deregisters the MN with the LMA.

There are some implementation issues present. We are using per-MN subnet prefixes. Because the MN is moving, these subnets can span multiple links. This is not desirable [45], since the IP model states that a subnet is associated with one link only. Several techniques rely on this assumption, such as multicasting. To solve this, the wireless link between the MAG and MN is treated as a point-to-point link. This way there are only two nodes present on the link. This is achieved by setting the Atheros wireless driver in non-bridge mode. Packets coming from the wireless link with a destination on the same wireless link are not directly sent to the destination but are forwarded to the default access router instead.

This point-to-point link also prevents MNs from receiving NS (Neighbor Solicitation) or NA (Neighbor Acknowledgement) messages from other nodes connected to the same access point. From these messages the MN could extract the link local address of the other MN in the same link and use this to communicate with the other MN directly. Packets destined for the other MN would then have the link local address as their destination. This is not a problem as long as both mobile nodes stay connected to the same access point. If one of the nodes would move away from the access point and connect somewhere else, ongoing connections between these two nodes would not be resumed immediately after a handover. This is because the two nodes can only communicate using their link local addresses if they are on the same access point. This issued was not studied during the design of the SPMIPv6 protocol but turned up during the development of the testbed. This "hack" in the testbed solved this issue, but it is not a perfect solution. Another solution would be to instruct MN's using SPMIPv6 to not to use link local addresses to communicate with other nodes than its default access router.

Both MAGs use the same MAC-address on their wireless adapters. This is a hack and not permitted normally, but this solves the problem that packets sent by the MN directly after a handover are lost. These packets are sent to the MAC address of the oMAG. By letting both the MAGs having the same MAC address, these packets are received by the nMAG on the new link. These packets are then delivered to the CN. Without this, it would take one RTT from the MN to the LMA before the MN could resume sending packets.

The TCP implementation used was TPC Reno. This means that packet loss is detected by the reception of three duplicate acknowledgements. When this happens, fast retransmit and fast recovery are deployed to recover from it. Selective ACK (SACK) was also enabled by default.

4.1.3. SPMIPv6 implementation

Support for simultaneous bindings was added to the existing testbed. The rest of this subsection describes what was added to the code of the different nodes.

MAG

Two new startup options were added to the MAG daemon. The -C option forces the MAG to create a buffer in which packets for mobile nodes which will move there soon will be stored. It also sets up a SCTP socket listening at port EMP_AR_PORT (used: 4343) for inter-MAG signaling. A TCP socket is also set up at port EMP_TRIGGER_PORT (used: 4444) to listen for handover initiation triggers (trigger 1). After the second new option, -c, the address of the other MAG is specified. This address is used to signal the other MAG to request bicasting at the LMA. This address is now supplied at runtime, but could also be included in the Perl script that sends the Trigger 1 message. This Perl script is used to enable bicasting for a MN. Running this script will send a message to EMP_TRIGGER_PORT on the oMAG to which the MN, identified by MN ID, is currently connected. In the implementation, layer 2 (i.e., MAC) addresses are used as MN ID. The oMAG will then send a SPBUR message to the nMAG. The nMAG, upon receiving this, will setup a ‘black hole’ route for the MN’s prefix. This is necessary since packets for the MN would otherwise be bounced back to the LMA, because no route exists for that destination prefix. Packets received for this MN are buffered at the nMAG from now on. The buffer size (time-wise) can be set on the console of the daemon.

LMA

When the LMA daemon is started with the -C option, bicasting is enabled. After receiving a request to start bicasting, packets for this certain MN are sent twice. The first copy is sent using the normal way by looking up the destination address in the routing table and then forwarding it. The second copy is sent to the nMAG. This is done by creating a packet socket, which receives all packets passing through the LMA. If there are packets for this certain MN, these packets are forwarded to the address of the nMAG.

The trigger 2 message (the actual handover) is executed by letting the MN change wireless SSID.

The messages sent are summarized in Table 2. Since SCTP is used as reliable transport protocol, acknowledgment message were not implemented.

Source	Destination	Message (PARAMETERS)
anywhere	oMAG	Trigger 1 (MN_ID)
oMAG	nMAG	Simultaneous Proxy Binding Update Request (MN_ID, LLA, PREFIXES, SIMBUFLAG)
nMAG	LMA	Proxy Binding Update Request (PREFIX, MN_ID, ID_LEN, SIMBUFLAG)

Table 2: Message exchange in the testbed to activating bicasting

4.2. SPMIPv6 results

The previously described testbed and implementation is used to execute test runs. The goal of tests is to determine the impact of packet loss, caused by a handover, on the overall throughput and how buffering of data for the MN at the nMAG affects this. The amount of buffering is a parameter which will vary in the different test runs. We will use both UDP and TCP as the transfer protocol. Each test run will have one dataflow, which is the sending of a large binary file. All tests will be terminated long before the whole file is sent. In half of the test runs, the CN will be sending data to the MN (downstream). In the other half, the MN is sending data towards the CN (upstream). The topology of the network is shown in Figure 22.

Before a test begins, the MN is already connected to one of the MAGs. A script is then executed at the central server that takes care of the following:

- Bicasting is enabled by sending a trigger 1 message to the oMAG. This is done by running the Perl script with as parameters the address of the oMAG and the MN ID of the MN.
- oMAG, nMAG and LMA now start signaling to enable a simultaneous binding.
- The CN (downstream) or MN (upstream) initiates a data transfer to the other node.
- Handover is executed at a certain point in time. This is done by letting the MN switch to the SSID of the nMAG.
- Data transfer is resumed when the MN has associated to the nMAG.

The time on all nodes is synchronized. The monitor laptop in the testbed receives all signaling and data messages and saves them, together with a timestamp. Afterwards the results can be presented graphically.

There are several distinctive events in the handover process. These are important to determine the overall handover latency. The results of the test runs will be presented in graphs where the following abbreviations are used:

- **HO:** Start of handover: MN disconnects at the old access point.
- **Asc**: MN Layer 2 association at the new access point
- Layer 3 association
    - **RS:** MN sends Router Solicitation
    - **RA:** MN receives Router Advertisement

The receiving of new data by the MN from the new access point denotes the end of the handover process. As stated earlier, the MN can already receive and send traffic at the new access point before it receives the RA because it is using the landmark option in the RS, which is part of the DNAv6 protocol.

Unlike specified in the SPMIPv6 specification, a layer 2 connection to the new access point is not made after a trigger 1 signal. This was not possible in the current testbed. However, since we are using inter-WiFi handovers it is not necessary to do extensive authentication (e.g. fetching keys etc.) during layer 2 attachment. Several other improvements have been made to make the Wi-Fi attachment as fast as possible. In the testbed this has been reduced to 5 ms, which makes a BBM approach usable in this case.

The RTT between the LMA and the MAGs was set to 30 ms. The RTT between the two MAGs is smaller than 1 ms. Layer 2 association to the new access point takes 5 ms. One RTT from MN to the access point (MAG) is about 2 ms. The minimal handover latency is equal the time it takes to connect to the new access point (layer 2 association), plus one RTT from MN to the MAG, the time during which a RS is sent and new data is received. In this setup, this is equal to 7 ms.

4.2.1. UDP

Downstream

We first have a look at downstream UDP traffic. In this case, the CN is sending data towards the MN. There is no buffering at the nMAG. In Figure 23 the sequence number of the received UDP packets is set out against the time.

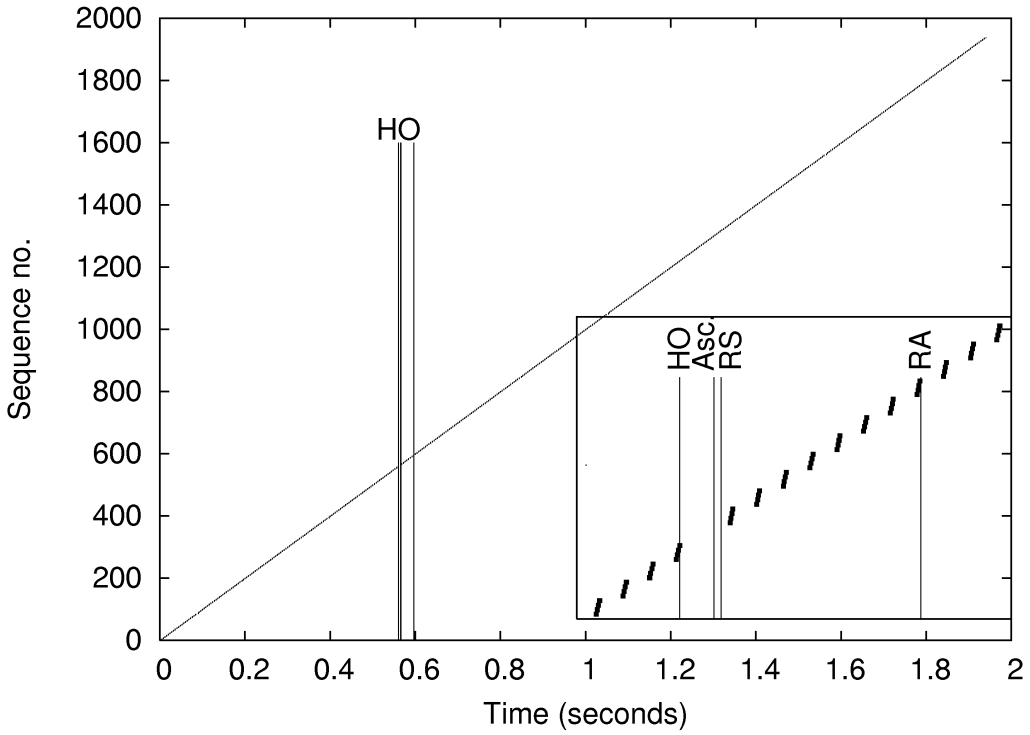

Figure 23: SPMIPv6 UDP downstream traffic without buffering

In the enlargement we see the data flow around the actual handover. Note that the scale on the x-axe (the time in seconds) does not apply to the enlargement. Here we see that there is a period where no data is received. This period is equal to the handover latency (7 ms). What we also see is that some packets are lost. There are some packets missing between the last one received at the oMAG and the first one received at the nMAG.

The next figure shows the same scenario, but now with buffering enabled in the nMAG. Buffering is set to 7 ms, which is equal to the handover latency. So, packets received by the oMAG for the MN while the MN is not connected there are not lost now but buffered at the nMAG.

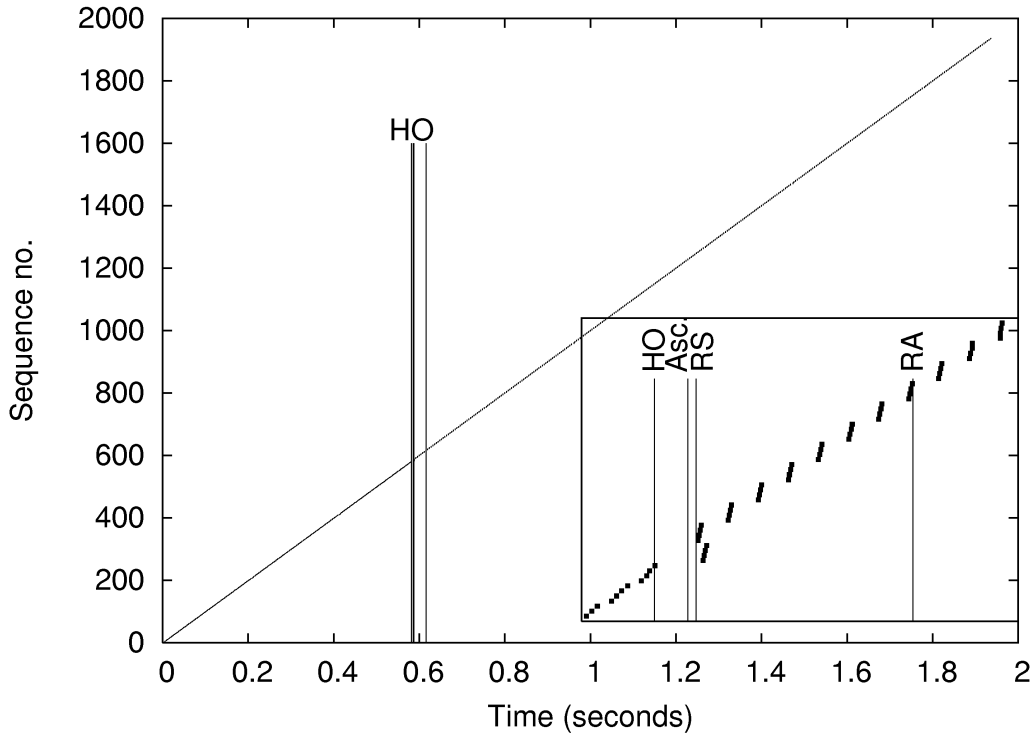

Figure 24: SPMIPv6 UDP downstream traffic with 7 ms buffering

In Figure 24 we see the result of the buffering. No packets are lost anymore. Some packets, however, arrive out-of-order.

If the nMAG would buffer packets for a longer period than the actual handover latency, it would send the MN packets after a handover it already received at the oMAG. With UDP, it now depends on the higher layers how these duplicate packets are treated. Some VoIP applications use the RTP [46] protocol on top of UDP. This way, each packet has a sequence number, by which the duplicate can be recognized and discarded. If the higher layers have no way of recognizing duplicate packets, hiccups may occur when for example a video stream is transported during the handover.

Upstream

There is no advantage for UDP upstream traffic when using SPMIPv6 instead of PMIPv6 in the experiment setting. Since there is only traffic flowing towards the MN there is no data that can be bicasted by the LMA or buffered at the nMAG.

4.2.2. TCP

Downstream

The behavior of a TCP data stream is a bit more complex. In Figure 25 we see the result of the CN sending data using TCP to the MN. The CN is sending data towards the MN, but since TCP is used, the MN has to send acknowledgements back (upstream) to the CN. This is

required by TCP to keep the data transfer going. This means that during the handover period packets are lost in both directions.

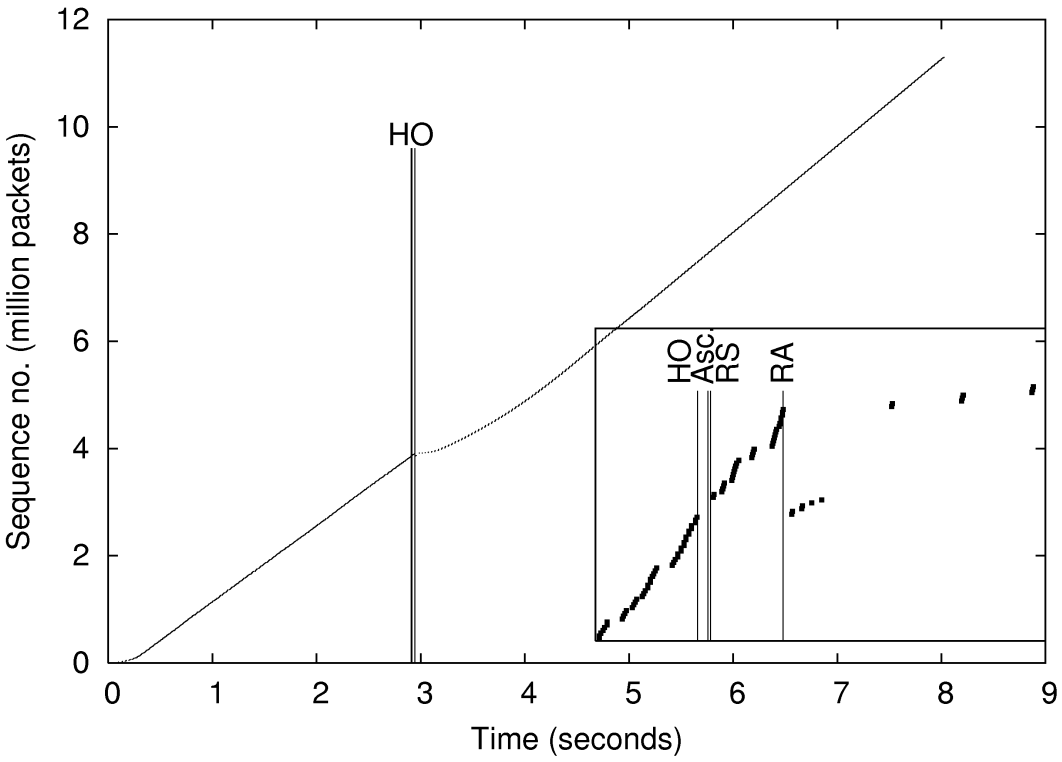

Figure 25: SPMIPv6 TCP downstream traffic without buffering

In Figure 25 we see the sequence numbers of received packages at the MN set out against the time. From this, we can see the behavior of TCP [47]. This is what happens:

- The MN has no connectivity in the period between disconnection at the oMAG (marked by HO) and association at the new one (marked by ASc).
- The MN does not receive any packets during this interval. Packets for the MN sent by the oMAG in this period are lost.
- When the MN is attached to the nMAG, new packets are received. Some packets were never received however. This can be seen by the 'vertical cap' in dots in the graph between the HO and ASc line.
- These packets were not acknowledged on time, so the retransmission timer in the sender expired. The packets are retransmitted. These are received just after receiving the RA message.
- Because of this timer expiring, the sender slows down sending of new packets. The new packets arrive slower than before the handover.
- There is again a 'vertical gap' after the retransmitted packets that were received after the RA message. This can be seen in the enlargement. This is because after having received the old retransmitted packets (directly after the RA message), new packets start arriving. Packets in between those were already received between Asc and RA events.

The previous graph only showed the receiving of data packets at the MN. The next graph, Figure 26, shows the sending of packets and the receiving of acknowledgement packets at the CN. The first diagonal line, marked with an X in a light color, shows the time at which a packet with a certain sequence number is sent by the CN. The second diagonal line, marked

with a + but in a darker color, shows when a packet with a certain sequence number is received by the MN. The third diagonal line, marked with squares, shows the time at which an acknowledgement is received by the CN for a packet with a certain sequence number.

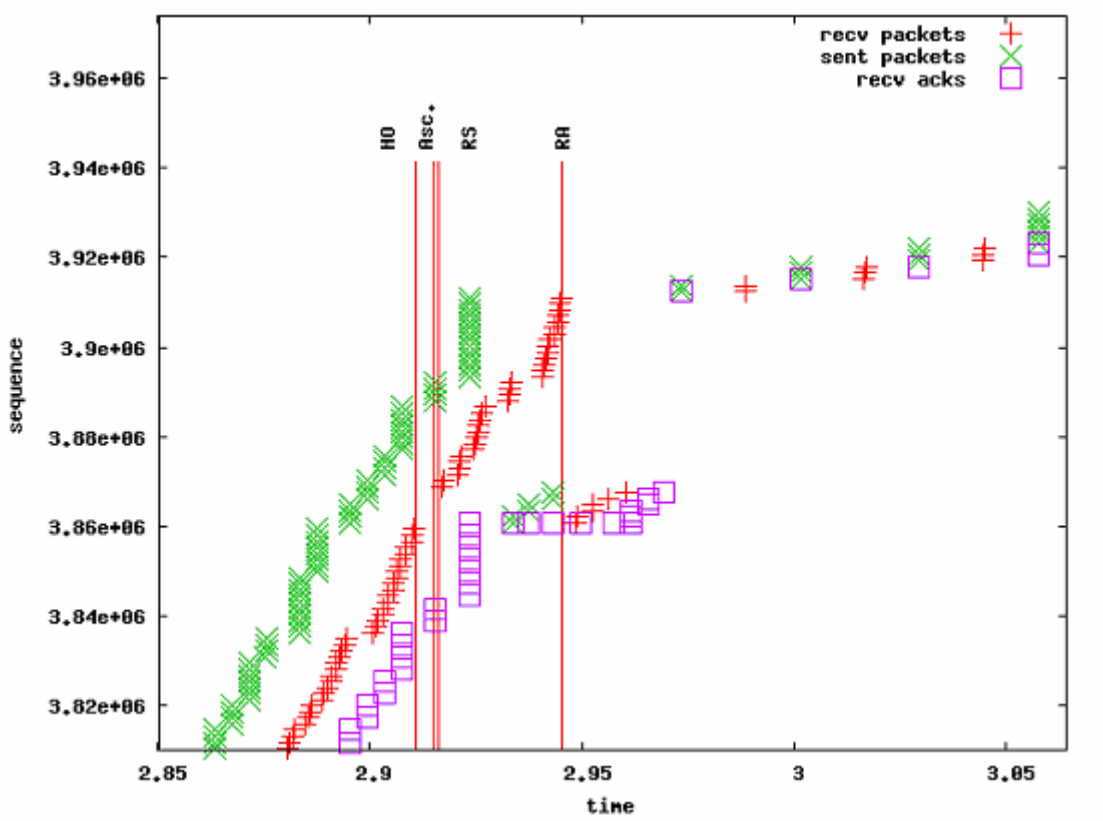

Figure 26: SPMIPv6 TCP downstream traffic without buffering, with ACKs

Figure 26 shows in more detail what happens. The CN starts retransmitting packets that were not acknowledged on time by the MN because of the handover delay. These retransmitted packets are indicated by the few light X'es just above the ACK line. This retransmitting is not triggered by receiving three duplicate ACKs (four ACKs in total with the same sequence number), which can be seen if we zoom into the graph a bit more. Only 2 duplicate ACKs were received when the retransmission occurred. If 3 duplicates were received this would cause the sender to start using fast retransmit and fast recovery, which we do not see in the graph. Because of the retransmission, slow start was initiated. The congestion window (cwnd) was set to 2 packets. This means that these two packets first have to be acknowledged before any new data can be sent. For each ACK that is received that acknowledges new data, the cwnd grows with 1 segment. This means that the cwnd grows exponentially each RTT. When half of the original cwnd is reached (before the loss occurred), the cwnd starts to grow linearly.

After the retransmission, acknowledgments are received, but they do not acknowledge new data. The cwnd does not grow. After the initial retransmission, two times two more lost packets are retransmitted, because of a timer expiring. All lost data is now retransmitted. Finally, an ACK is received for new data. The cwnd at the CN was still set to 2 segments, but starts growing when the 'normal' data transfer can be resumed after the handover.

We ran the same scenario, now with buffering enabled. Packets were again saved for 7 ms, which is equal to the total handover latency. The result of this is shown in Figure 27. Here we see that TCP does not notice the handover latency, since data transfer carries on as normal after the handover.

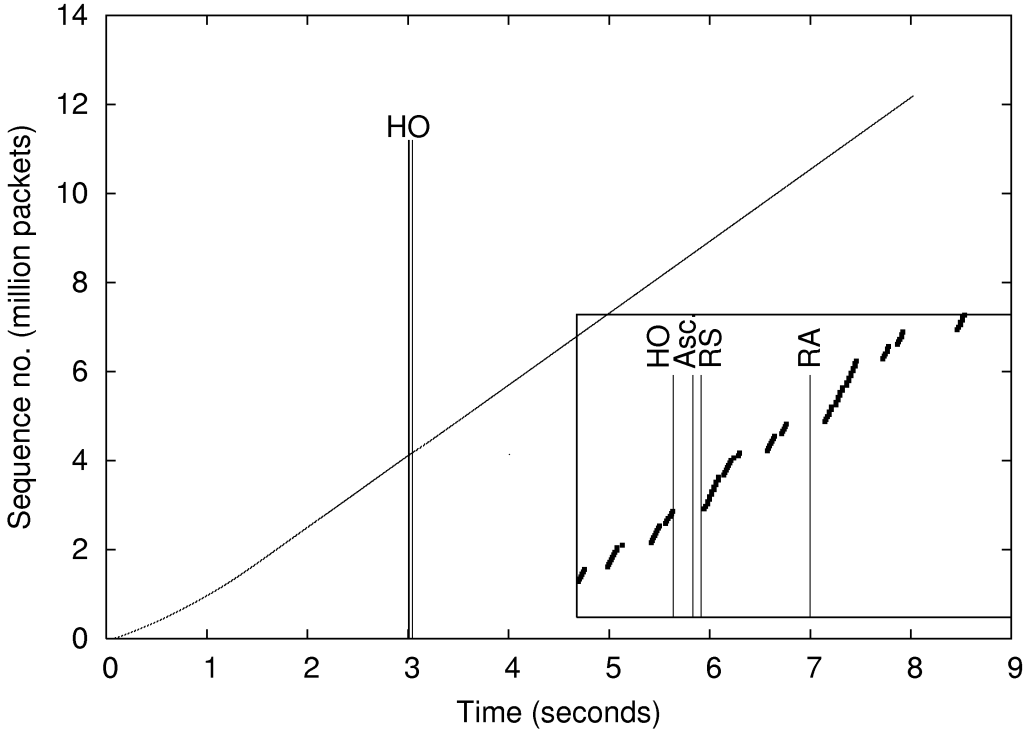

Figure 27: SPMIPv6 TCP downstream traffic with 7 ms buffering

The next few graphs show the impact of a buffer that is too large, e.g. much larger than the total handover latency. The MN will receive a lot of duplicate (old) packets after the handover. This will be handled by TCP and will not be passed through to higher layers. These packets are normally just dropped. The impact can be seen in Figure 28. After association all buffered packet are sent to the MN by the nMAG. After this, the speed at which packets are received slows down, as can be seen by the line having a slight dip after the handover.

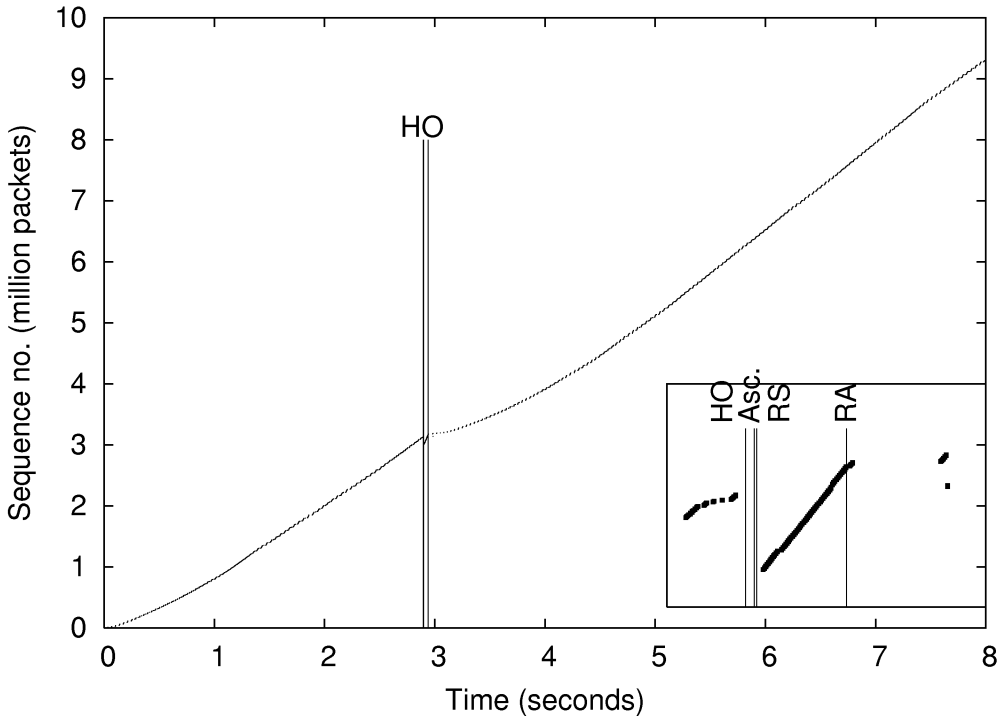

Figure 28: SPMIPv6 TCP downstream traffic with 100 ms buffering

In Figure 29 the packets that are sent and received after the handover are shown in more detail. The packets that are sent from the nMAG's buffer are visualized by the steep vertical line, marked with +, just after the association. Here we can see that there are two packets missing in the burst of traffic sent from the nMAG's buffer. This can be seen by the small gap in the burst. Looking into the actual packet logs from tcpdump we can see that the CN initially did sent these packets to the LMA. These two packets also appear on the link between oMAG and MN, but the MN does not receive them, since it is already disconnected there at that time. It is not possible to see what packets are exchanged between LMA and the two MAG's, but we may assume that the nMAG also receives all packets for the MN, since the link has enough capacity. Bicasting is enabled at the start of the test run, so it is not a timing issue.

A possible explanation for the two packets missing in the burst might be that the link between the MN and the nMAG is congested by the sudden burst of data. From analyzing the graph we can see that in 0,04 seconds 168780 bytes are sent. This equals about 4 megabyte per second (32 mbit/second). The average throughput of the test runs is about 1,4 megabyte/second. It is fairly possible that this burst of traffic temporarily congested the 802.11g wireless link.

The TCP data transfer recovers from the lost packets in the following way. The packets that were lost are retransmitted later on after receiving 3 (even more) duplicate ACKs. Because of this, fast transmit and fast recovery are executed by the sender. This means that the congestion window is set to halve its current value, slowing down the sending of new packets. After this, normal transmission is resumed.

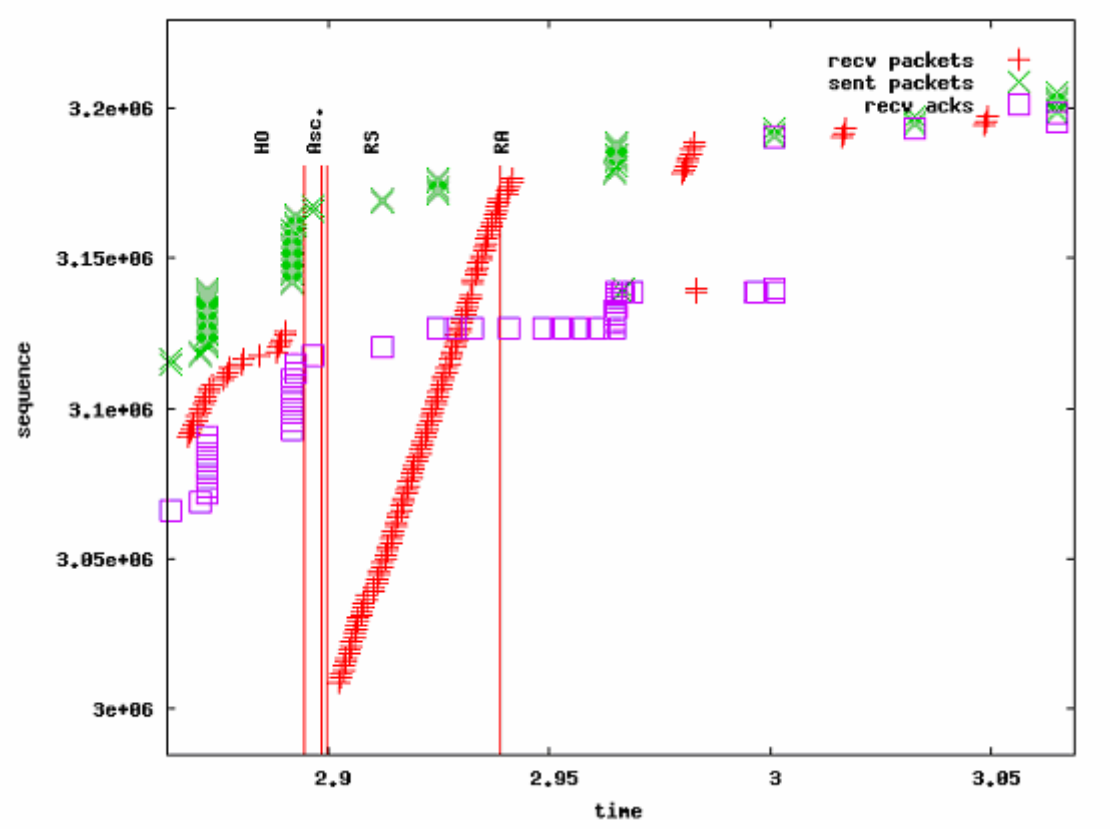

Figure 29: SPMIPv6 TCP downstream traffic with 100 ms buffering, with ACKs

Upstream

TCP upstream data, in contrast to UDP, does benefit from SPMIPv6. In the next scenario, the MN is sending data to the CN. Acknowledgments that have the MN as destination are now bicasted at the LMA. With bicasting enabled and no buffering the handover is already almost seamless, which can be seen in Figure 30. There is a small pause in the data transfer, as can be seen in the enlargement, but this does not have impact on the overall transfer speed.

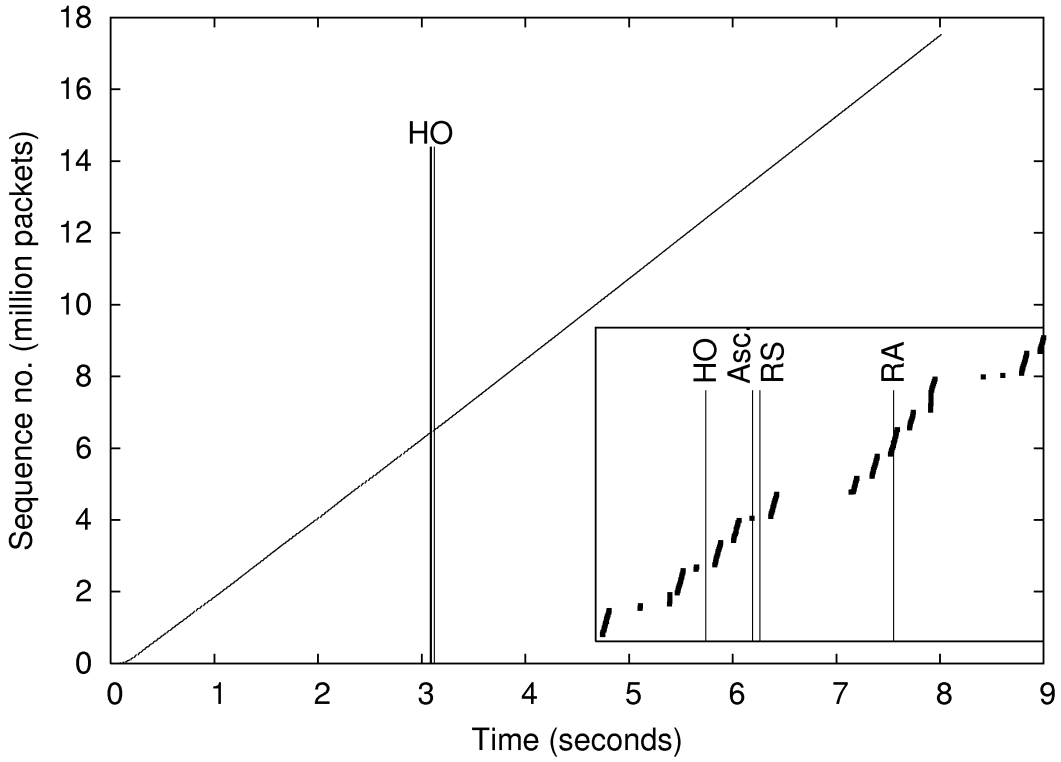

Figure 30: SPMIPv6 TCP upstream traffic without buffering

TCP upstream data would further benefit from buffering. With bicasting enabled, acknowledgment packets for the MN are delivered to both the new and old MAG. It could happen that a certain ACK packet is not delivered at both locations right after the MN disconnected from the oMAG. It does not receive this packet from the oMAG, since it is disconnected there. The nMAG tries to deliver the packet, but the MN is not connected there yet, since it takes some time to setup the new connection. So, this certain ACK never reaches the MN. If the MN has already sent the maximum amount of packets it is allowed to send without having them acknowledged[1] then data transfer halts if the MN does not receive this acknowledgment. The MN sending data would then reduce its sending rate, slowing down the whole data transfer. If this ACK would have been buffered at the nMAG this would not have happened. It is difficult to estimate how large this buffer should be, since it is difficult to determine how much handover latency a MN has experienced. However, Figure 31 shows that TCP is not affected by a buffer that is too large. Here, we set the buffer size to 100 ms. After a handover, all the buffered ACKs are sent. These ACKs are visualized in Figure 31 by the line marked with squares emerging right after the handover. This does not affect TCP, since these ACKs are considered 'old' and are just discarded.

[1] As specified by the congestion window

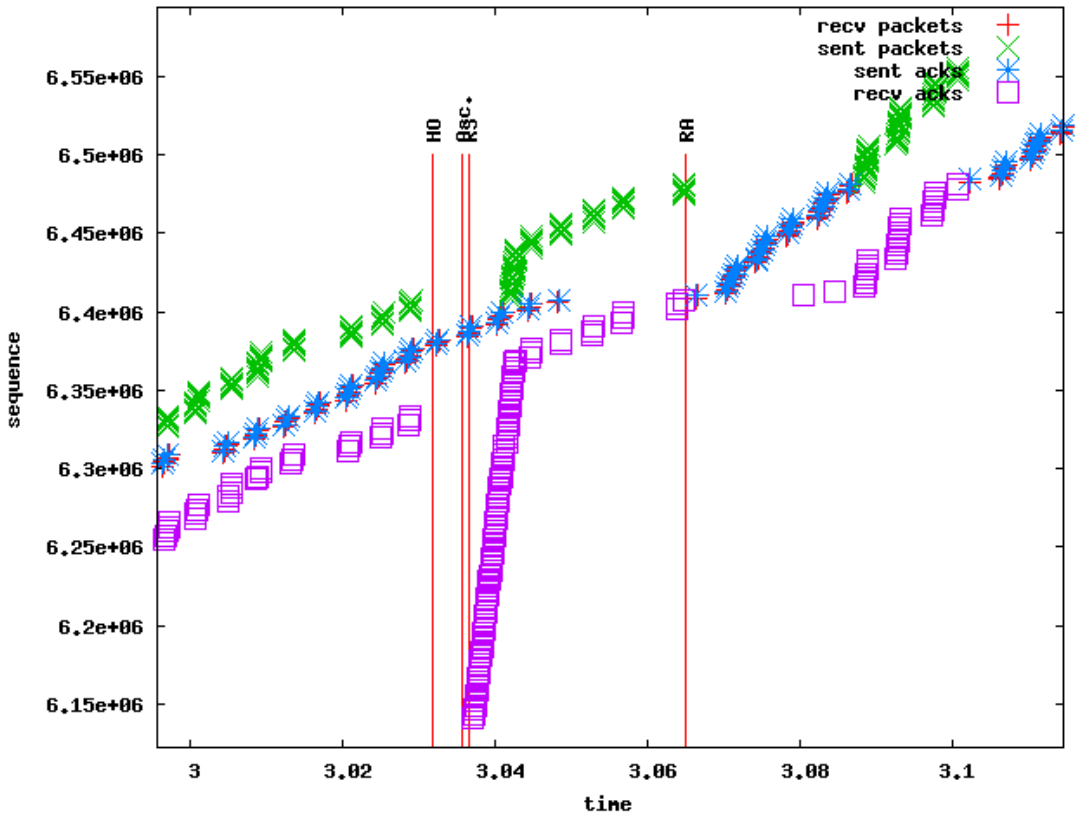

Figure 31: SPMIPv6 TCP upstream traffic with 100 ms buffering

4.3. Buffer size impact

The previous section proved that the amount of buffering at the nMAG is very important with respect to the total handover delay. The next table shows a summary of these findings.

Protocol	Upstream	Downstream
TCP		
Buffer too large	Too many acknowledgements are buffered, but this is not a problem since they acknowledge old data and are just discarded.	Duplicate packets are not a problem, but if the buffer is too large the link between the nMAG and the MN may become congested. This way, packets are lost and transfer slows down.
Buffer too small	Acknowledgements may be lost and slow down the upstream data flow.	Packets may be lost and this slows down the rate the CN is sending packets.
UDP		
Buffer too large	No impact.	If the higher layers (above UDP) have no way of discarding duplicate packets, hiccups in multimedia streams may occur because of this. Also, the link between MAG and MN might become congested if the buffer is much too large.
Buffer too small	No impact.	Packets will be lost. This will cause hiccups in for example audio and video streams.

Table 3: Buffer sizes impact

4.4. SPMIPv6 and PMIPv6 performance comparison

In this section we will compare SPMIPv6 with the results from the existing PMIPv6 protocol.

After a handover with PMIPv6, signaling messages go all the way up to the LMA and back before the MN can receive any data. In our test network, the delay between MAGs and LMA was already 15 ms one-way. With additional processing delay it takes 36 ms before the MN can receive any data at the nMAG. This can be seen in Figure 32. In this figure, the propagation delays between the different entities is displayed. With SPMIPv6, data is already available at the nMAG before handover. In this case, the handover delay is 5 ms layer 2 attachment delay plus 2 ms processing and propagation delay over the air interface. This means 7 ms total.

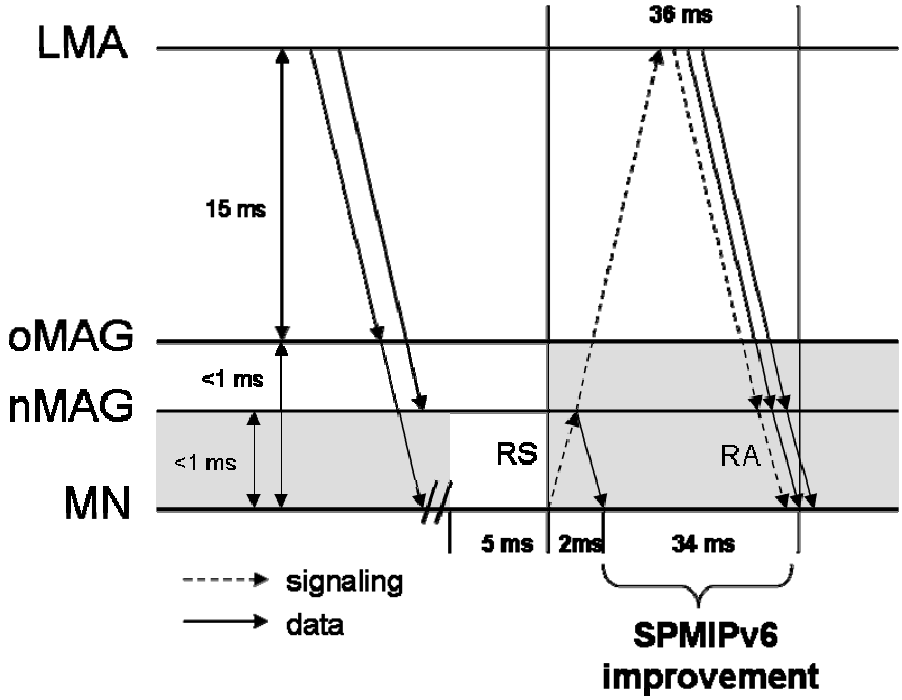

Figure 32: SPMIPv6 traffic flow after a handover.

In Figure 33 we have the test results for UDP downstream traffic using normal PMIPv6. This can be compared with Figure 23 (SPMIPv6). With PMIPv6 data is received around the same time the RA arrives. In Figure 23 data is received much earlier.

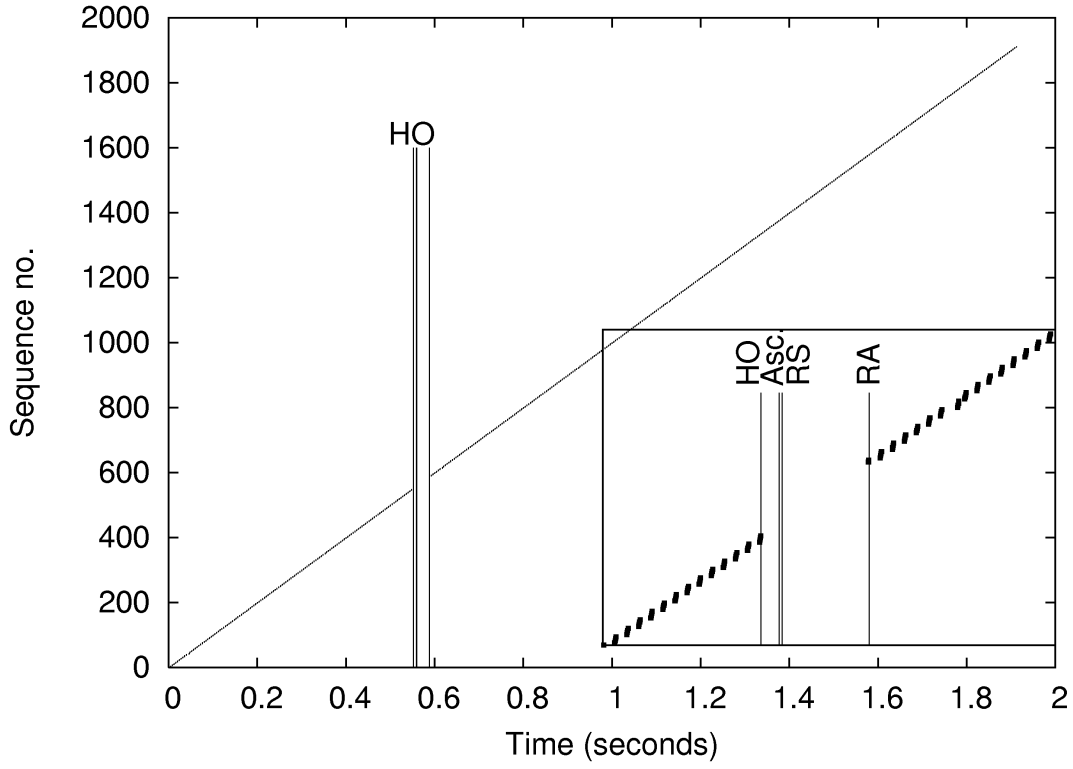

Figure 33: PMIPv6 UDP downstream traffic

The improvement for TCP flows is even more visible. In Figure 34 we see the results for TCP downstream traffic using PMIP. The whole transfer halts after the handover. This is because a lot of TCP segments were lost. To the sender this indicates congestion. The slow start algorithm is then enabled, after which it takes a while until the transfer is at its full speed again. Compare this graph to Figure 25.

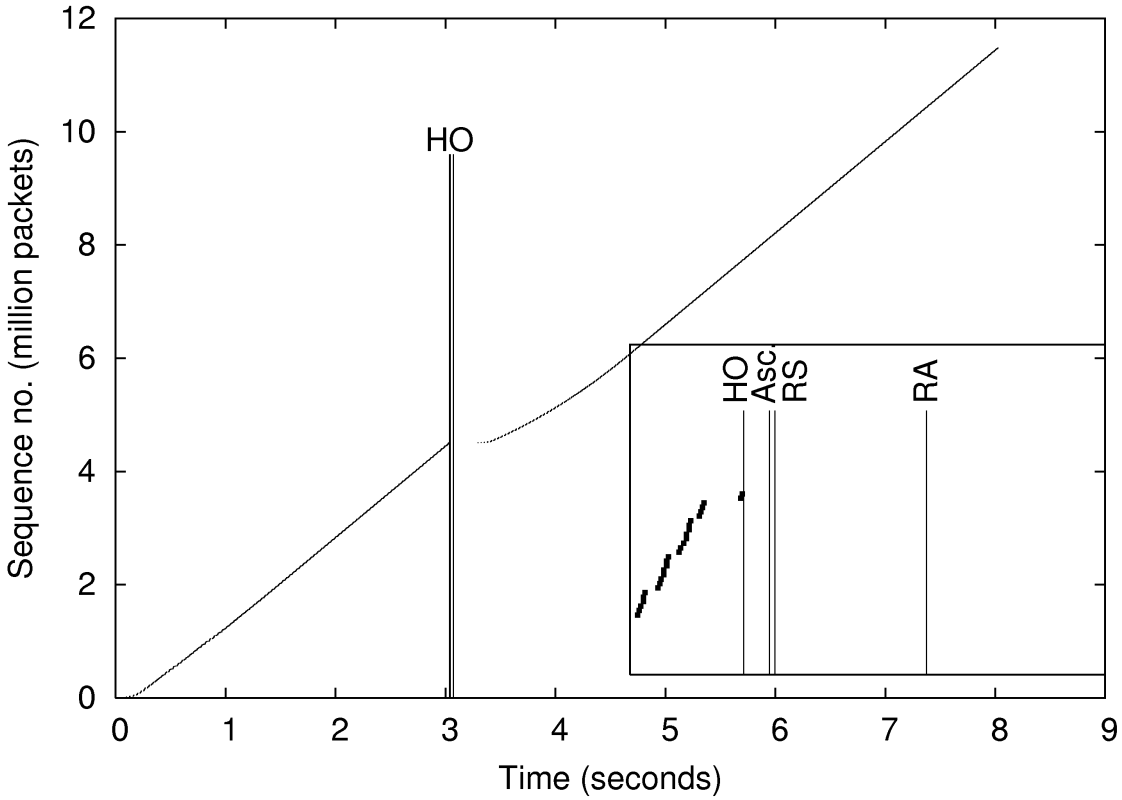

Figure 34: PMIPv6 TCP downstream traffic

4.5. Analysis

So far, we have only seen a perfect execution of a handover using the SPMIPv6 protocol. The prediction of a handover was always accurate: there was enough preparation time and the MN always moved to the MAG as expected. Several things could go wrong however in a real life scenario. It could happen that after trigger 1, trigger 2 does not occur. The MN just stays connected to its current access point. After some period of time, a timeout can then occur with which the simultaneous binding will be cancelled. The MN could also move to a different access point than expected. In this case, the handover process will be equal to that of PMIPv6. SPMIPv6 is robust in both of those cases. There is a third possibility. It could be that the handover is not predicted on time. This means that there is not enough time between trigger 1 and trigger 2. In the testbed, it was not possible to execute such scenarios. In this section we will analyze what the maximum handover latency will be in this case. Using the protocol also has an impact on the wired part of the network. This issue will also be analyzed in this section.

4.5.1. Timing of prediction

The main variable we are interested in is the handover latency period (D_{ho}). This is the period during which the MN is not able to receive any new IP packets. We assume trigger 1 occurs at a fixed point in time, denoted by $T_{trigger1}$. The time the second trigger occurs (denoted by $T_{trigger2}$) will now vary.

In the protocol, the MN can, upon receiving the trigger 1 message, already establish a layer 2 connection with the new access point. In the testbed however, it was not possible to make this part of the handover preparation process, but we will include this in our analysis now.

We will use the variables that are displayed in Table 4.

Variable	Description
$T_{trigger1}$	Moment in time at which trigger 1 occurs.
$T_{trigger2}$	Moment in time at which trigger 2 occurs.
T_{prep}	Avaiable preparation time.
D_{layer2}	Time it takes to setup a layer 2 connection to a new access point.
$D_{network}$	Period of time necessary to establish a simultaneous binding in the network.
$RTT_{MAG-MAG}$	RTT between a MAG and another MAG.
$RTT_{MAG-LMA}$	RTT between a MAG and the LMA.
RTT_{MN-MAG}	RTT between a MN and the MAG it is currently connected to.
D_{ho}	Handover latency
t_1	A point in time relative to the reception of trigger 1.
t_2	Another point in time relative to the reception of trigger 1.

Table 4: Variables used in analysis of handover latency

T_{prep} is the available amount of time to do handover preparations, as defined in eq. (1). $D_{network}$ is the time it takes to setup the simultaneous binding in the network after issuing trigger 1 at oMAG, as defined in eq. (2). First we assume that $D_{network}$ is much larger than D_{layer2}. We can now determine the maximum handover latency for a given preparation time period T_{prep}. The

moments t_1 and t_2 defined in eq. (3) and eq. (4) are auxiliary parameters that are used in our analysis. Table 5 presents eq (1)–eq. (4).

Variable	Description	
T_{prep}	$T_{trigger2} - T_{trigger1}$	(1)
$D_{network}$	$½ RTT_{MAG-MAG} + RTT_{MAG-LMA}$	(2)
t_1	$½ RTT_{MAG-MAG} - ½ RTT_{MN-MAG}$	(3)
t_2	$D_{network} - ½ RTT_{MN-MAG}$	(4)

Table 5: Definition of variables used in analysis of handover latency

The messages exchanged in the network after trigger 1, as well as the auxiliary parameters t_1 and t_2, are shown in Figure 35. First, the layer 2 setup is made to the new network. After this it is possible to send a RS message. The minimum handover latency is experienced when this RS sent by the MN is received at the nMAG when the bicasted traffic for the MN is already there. The first bicasted traffic comes at the same time as the PBUA message.

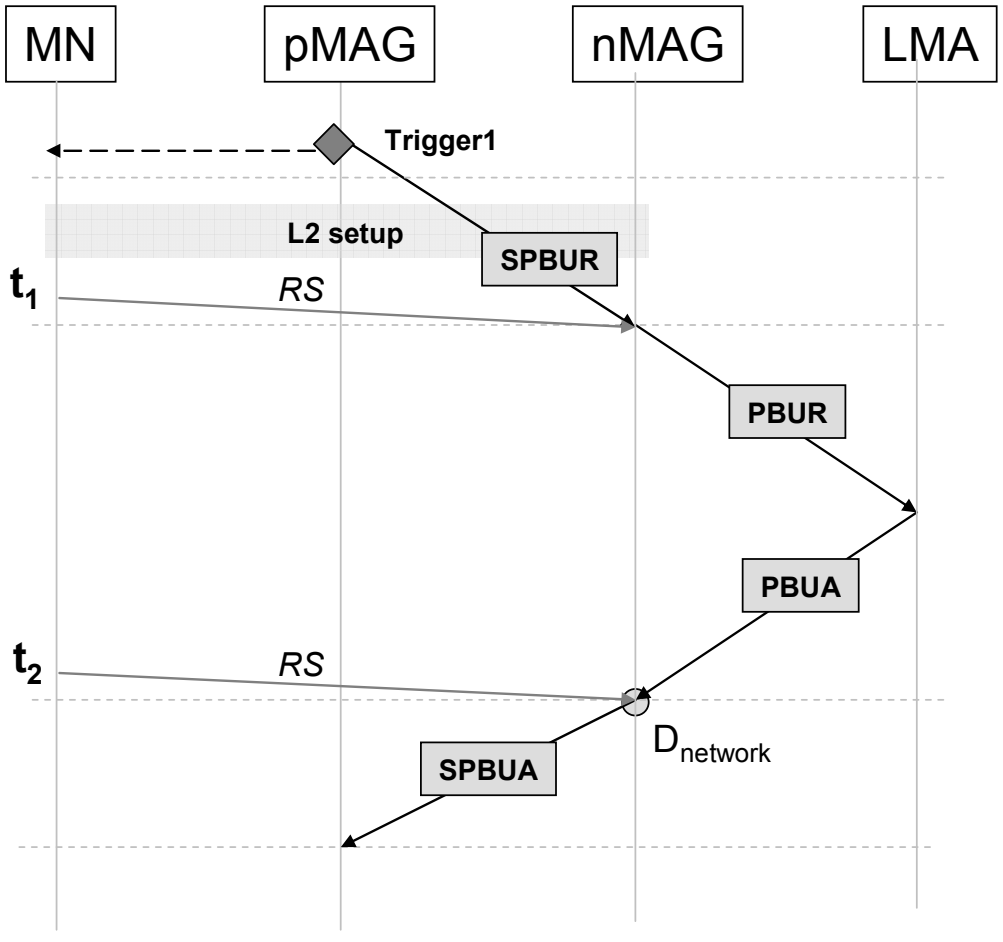

Figure 35: Message exchange after trigger 1

The minimum preparation time needed for the smallest possible handover latency is equal to $D_{network}$ minus half a RTT from MN to MAG. This is represented by the point in time t_2 as defined in (4). When there is at least this amount of time between trigger 1 and trigger 2, data for the MN has reached the nMAG when the RS from the MN is received. This RS has to be sent on or after t_2 (see Figure 35). Handover latency D_{ho} is then equal to one RTT_{MN-MAG}. So, if a RS is received on or after $D_{network}$, the minimum handover latency is experienced. If the MN has moved to the nMAG and sends a RS message on or before t_1 (3), the maximum handover latency is experienced. This is equal to one RTT_{MN-MAG}, which is the time it takes to send the RS and receive the RA and data, plus one $RTT_{MAG-LMA}$, the time it takes to update the binding with the LMA and wait for the PBUA and data to be received by the nMAG. This is equal to the handover latency experienced with PMIPv6. If T_{prep} is larger than t_1 but smaller than t_2, the handover latency is determined by the remaining signaling time in the network, which is equal to ($D_{network} - T_{prep}$) minus ½ RTT_{MN-MAG} .

	T_{prep}	Handover latency D_{ho}
Min.	$T_{prep} >= t_2$	RTT_{MN-MAG}
	$t1 <= T_{prep} < t_2$	$RTT_{MN-MAG} + (D_{network} - ½\ RTT_{MN-MAG} - T_{prep})$
Max.	$T_{prep} < t_1$	$RTT_{MN-MAG} + RTT_{MAG-LMA}$

Table 6: Maximum handover latency when $D_{network} >> D_{layer2}$

Table 6 shows the formulas for the maximum handover latency. These formulas are visualized in the graph in Figure 36 .

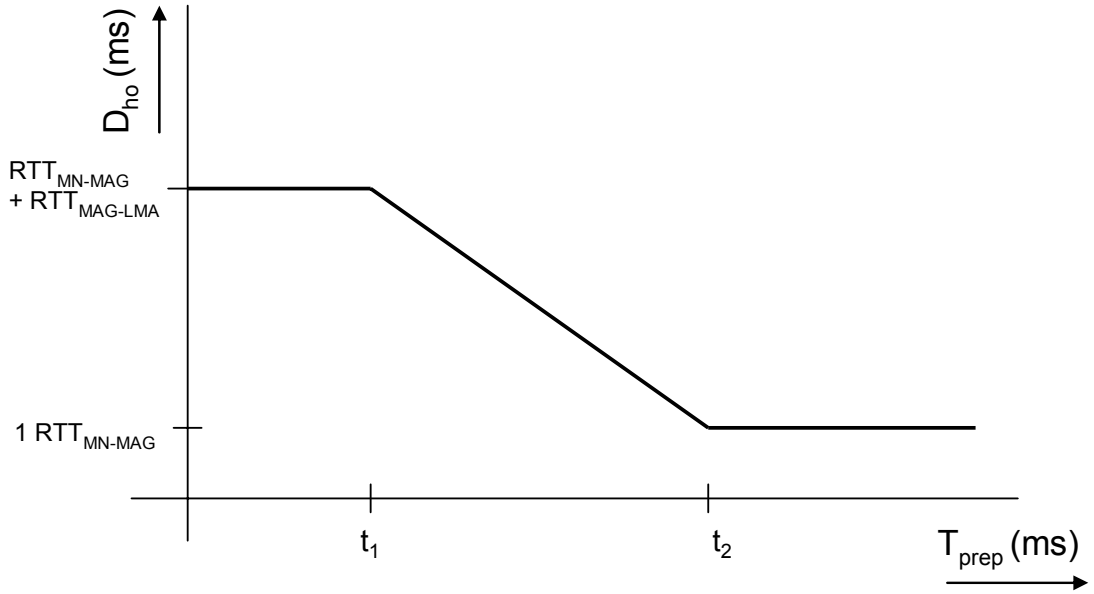

Figure 36: Handover latency $D_{network} > D_{layer2}$

In the previous part we have assumed that $D_{network}$ is much larger than D_{layer2}. We will now show what happens when this is not the case. For this, we have to incorporate the layer 2 setup delay, $D_{layer2,}$ into the equations. If T_{prep} is larger than or equal to t_2, the maximum handover latency is equal to one RTT_{MN-MAG} plus the remaining time necessary for the layer 2 connection to become activated (D_{layer2}-T_{prep}).

If T_{prep} is smaller than that, but larger than t_1, the handover latency is equal to the maximum of the remaining time of $D_{network}$ minus ½ RTT_{MN-MAG}, which is equal to $D_{network}$- ½ RTT_{MN-MAG}-T_{prep}, and the remaining layer 2 setup time (D_{layer2}-T_{prep}), plus one RTT_{MN-MAG}.

If T_{prep} is smaller than t_1, the maximum value of D_{ho} depends on the duration of one $RTT_{MAG-LMA}$ and the duration of D_{layer2}, whichever takes the longest. If D_{layer2} is smaller or equal to $RTT_{MAG-LMA}$, the handover curve will be equal to the curve drawn in Figure 36, since it still dependents on $RTT_{MAG-LMA}$. If D_{layer2} is bigger than $RTT_{MAG-LMA,}$ the maximum handover latency depends on how much of D_{layer2} has already been done in T_{prep}.

	T_{prep}	**Maximum handover latency D_{ho}**
Min.	$T_{prep} >= t_2$	$\max(0, D_{layer2}-T_{prep}) + RTT_{MN-MAG}$
	$t_1 <= T_{prep} < t_2$	$\max(D_{network}- ½ RTT_{MN-MAG}-T_{prep}, D_{layer2}-T_{prep})+ RTT_{MN-MAG}$
Max.	$T_{prep} < t_1$	$\max(RTT_{MAG-LMA}-T_{prep}, D_{layer2}-T_{prep})+ RTT_{MN-MAG}$

Table 7: D_{ho} with layer 2 delays

We visualize this in Figure 37 and Figure 38. The bare line shows the handover latency from the previous equations, without taking layer 2 setup delay into consideration, because $D_{network}$ is much larger than D_{layer2}. The dotted line shows the handover latency with the layer 2 setup delay (D_{layer2}) being equal or greater than $D_{network}$. In Figure 37, we see that D_{layer2} is only slightly bigger than $RTT_{MAG-LMA}$. In the graph, D_{layer2} is equal to 20 ms, $RTT_{MAG-LMA}$ is equal to 15 ms. There is only a difference in handover latency when T_{prep} is quite small. In the second figure, Figure 38, D_{layer2} is much larger. D_{layer2} is 30 ms there, twice the amount of $RTT_{MAG-LMA}$. The difference in handover latency increases when T_{prep} gets larger. Also, the minimum preparation period to get the minimal handover latency increases. This was $D_{network}$ - ½ $RTT_{MN-MAG,}$ but if we take D_{layer2} into consideration it will be the maximum of that and D_{layer2}.

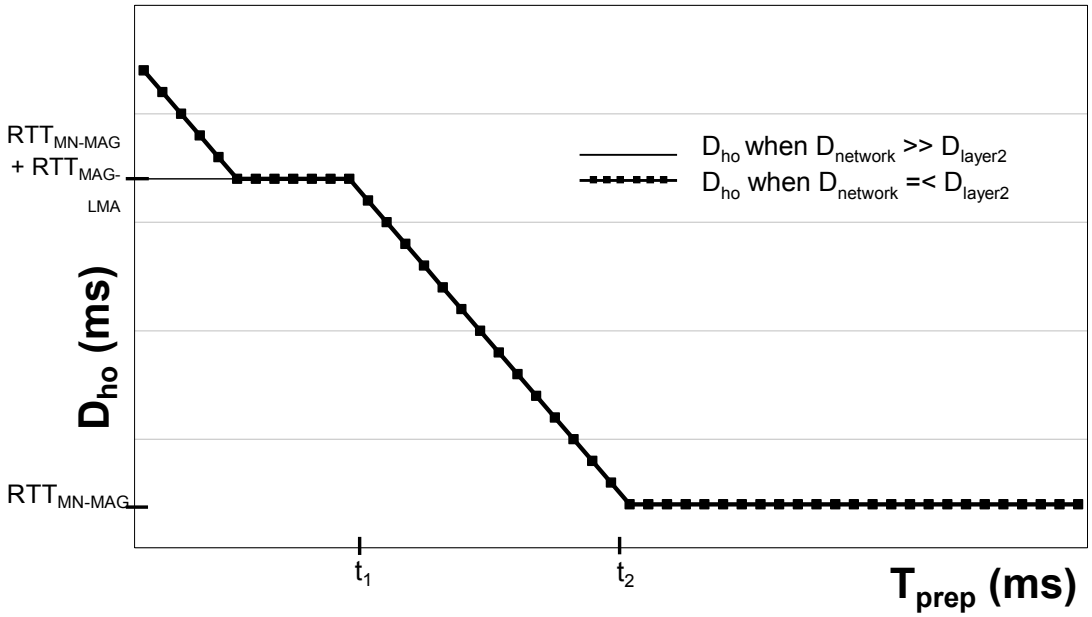

Figure 37: Handover Latency $D_{network} \cong D_{layer2}$

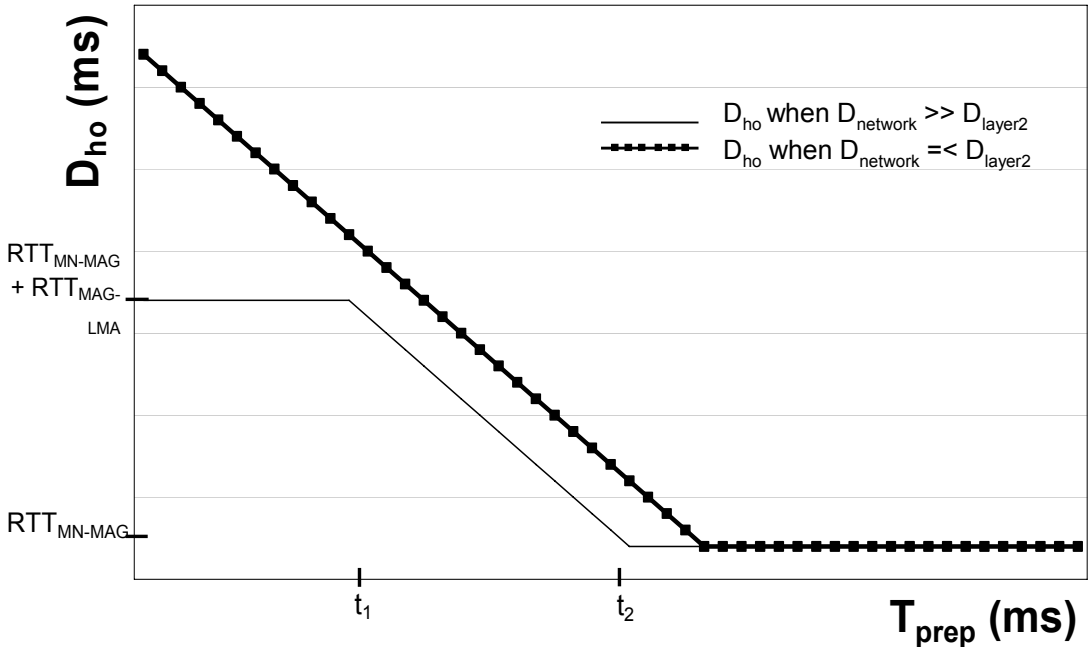

Figure 38: Handover Latency $D_{network} < D_{layer2}$

4.5.2. Bandwidth usage in the network

So far we have only looked at the impact of SPMIPv6 on the connectivity of the MN. Using the protocol also has an impact on the rest of the network however. The model which is given next shows the extra load in the network caused by using SPMIPv6.

SPMIPv6 uses bicasting to minimize the packet loss experienced by mobile nodes when doing a handover from one access network to another. Because of this bicasting, extra load is introduced in the network. The LMA sends data to both old and new MAGs simultaneously.

While the MN is still connected to the oMAG, the link between the LMA and the nMAG will have a small amount of unnecessary extra load.

The extra load in the network depends on the time the bicasting is active. We can distinguish two scenarios why bicasting stops: (1) the bicasting times out because the MN stays connected to the oMAG and (2) the MN actually moves to the nMAG. The first scenario is caused by a wrong handover prediction. The oMAG predicted that the MN would move to a new MAG, but it does not change access point, so after a while the bicasting stops because of a timeout. Note that in this model we only call it a wrong prediction if the MN does not move after a prediction done by the oMAG. A MN that moves to another MAG than the predicted one will fall under the first scenario. The extra load caused by this is equal to the load introduced in the network when the MN would have moved to the MAG that was predicted. In both cases, bicasting stops after the LMA receives a PBUR message from the nMAG. This can be the MAG the MN was predicted to move to or a different one.

If a MN moves to another MAG without any prediction, SPMIPv6 performs like PMIPv6 for which the extra load is equal to zero. We do not consider such cases in our calculation below; therefore, our extra load estimate is slightly higher than reality.

We can now determine the total extra load in the network, denoted by L_{extra}, for both scenarios separately. For the first scenario, we can express the extra load in the network ($L_{extra\text{-}incorrect}$) in the following way. We define a variable P_e (prediction error) which expresses the percentage of all handover predictions that will timeout because of faulty prediction. The variable $t_{timeout}$ represents the time a bicasting will stay active before it times out. R_{tr} expresses the average transmission rate at which mobile nodes connected to the SPMIPv6 domain are sending and receiving data. For clarity, all variables used in the model are summarized displayed in Table 8.

Variable	Description
L_{extra}	Extra load in the network due to SPMIPv6 usage
$L_{extra\text{-}correct}$	Extra load in the network caused by a correct handover in SPMIPv6
$L_{extra\text{-}incorrect}$	Extra load in the network caused by a incorrect handover in SPMIPv6
P_e	Predication error percentage
$(1\text{-}P_e)$	Percentage of correct predictions
R_{tr}	Average transmission rate of mobile nodes
$t_{timeout}$	Amount of time after which bicasting will timeout.

$T_{trigger1}$	Moment in time at which trigger 1 occurs.
$T_{trigger2}$	Moment in time at which trigger 2 occurs.
T_{prep}	Preparation time. Definition: $T_{prep} = T_{trigger2} - T_{trigger1}$
$RTT_{MAG-MAG}$	RTT between a MAG and another MAG.
$RTT_{MAG-LMA}$	RTT between a MAG and the LMA.
RTT_{MN-MAG}	RTT between a MN and the MAG it is currently connected to.

Table 8: Variables used in bandwidth usage model

The extra bandwidth usage on the link LMA-oMAG in the first scenario is now equal to the prediction error percentage (P_e) times the transmission rate (R_{tr}) times the time it takes for the bicasting to timeout. This is given in expression 1.

(1) $$L_{extra-incorrect} = P_e * R_{tr} * t_{timeout}$$

For the second scenario (correct prediction) the extra load is equal to the time the bicasting is active times the average transmission rate. This correct prediction happens with the percentage 1-P_e.We assume a prediction to be incorrect only if it times out. If a MN moves to another MAG than the predicted one, the extra load in the network is still equal to the same amount as the case where the MN moves to the predicted MAG.

We have to determine how long a bicasting is actually active. As can be seen in Figure 39, the actual bicasting starts at the time trigger 1 is activated plus the time it takes to activate the bicasting in the network, which is done by the SPBUR and PBUR messages. These take respectively 0.5 RTT MAG-MAG and 0.5 RTT MAG-LMA. This is stated in expression 2.

(2) $$T_{bicast-start} = T_{trigger1} + 0.5(RTT_{MAG-MAG} + RTT_{MAG-LMA})$$

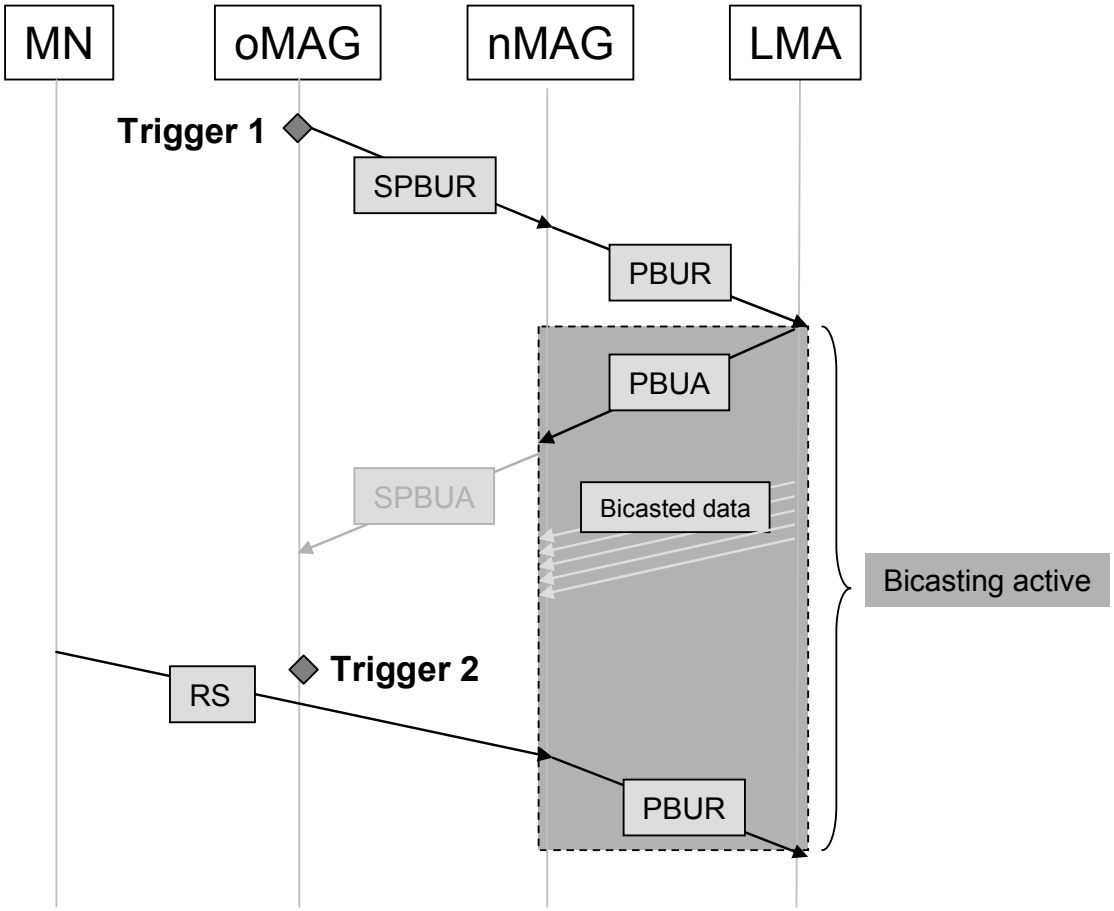

Figure 39: Bicast start and stop timing

The bicasting ends at the moment that Trigger 2 is activated, plus 0.5 RTT MN-MAG (RS message) plus 0.5 RTT MAG-LMA (PBUR message).

(3) $$T_{bicast-end} = T_{trigger\,2} + 0.5(\mathrm{RTT}_{\mathrm{MN\text{-}MAG}} + \mathrm{RTT}_{\mathrm{MAG\text{-}LMA}})$$

We can now express the time between the start and the end of the bicasting. This is done in expression 4.

(4) $$T_{bicast-active} = T_{bicast-end} - T_{bicast-start} = T_{trigger\,2} + 0.5RTT_{MN-MAG} - T_{trigger\,1} - 0.5RTT_{MAG-MAG}$$

In our earlier analysis, we used the variable T_{prep} to express the total preparation time, which is equal to $T_{trigger1}$ minus $T_{trigger2}$. We used this in expression 5 to simplify the previous expression.

(5) $$T_{bicast-active} = T_{prep} + 0.5RTT_{MN-MAG} - 0.5RTT_{MAG-MAG}$$

The expression for the extra load for an incorrect prediction is now given in expression 6, assuming that $t_{timeout} > T_{bicast\text{-}active}$.

(6) $$L_{extra-correct} = (1-P_e) * R_{tr} * (T_{prep} + 0.5\mathrm{RTT}_{\mathrm{MN\text{-}MAG}} - 0.5\mathrm{RTT}_{\mathrm{MAG\text{-}MAG}})$$

The total extra load is given in expression 7 which incorporates both scenarios.

(7) $$L_{extra} = P_e * R_{tr} * t_{timeout} + (1 - P_e) * R_{tr} * (T_{prep} + 0.5\text{RTT}_{\text{MN-MAG}} - 0.5\text{RTT}_{\text{MAG-MAG}})$$

We now illustrate this, see Figure 40. On the y-axe, we have the extra load induced in the network. On the x-axe, the preparation time T_{prep} is displayed. The timer for the bicasting lifetime is started in the LMA upon receiving the PBUR message. Because of this, bicasting times out if the T_{prep} is equal to $t_{timeout}$ plus 0.5 $RTT_{MAG-MAG}$ plus 0.5 $RTT_{MAG-LMA}$. It takes these two half RTT's to start the timer after a Trigger 1. From the graph we can see that how longer the bicasting is active, the higher the extra load in the network is.

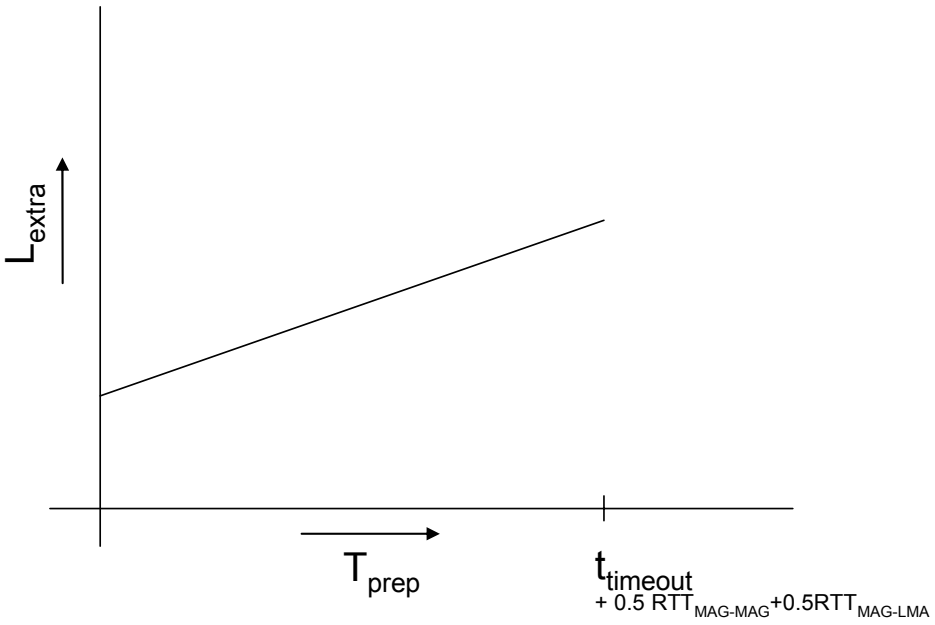

Figure 40: Bicast active versus extra load in network

5.Conclusion and future directions

This report described the design, analysis and test results for the SPMIPv6 network controlled handover protocol. This final section summarizes and draws conclusions from the previous chapters and outlines several topics for future research.

5.1. Conclusion

The goal of this project was to design an IP-based mobility protocol with a maximum handover latency of 50 ms. To achieve this in SAE/LTE environments that integrate heterogeneous access network technologies, it is required to take a proactive approach towards the actual handover. This way, it became possible to support handovers towards an access technology with a large setup delay and also to make sure that the data for the mobile node was available before the MN actually connected to the new network, ensuring a low amount of packet loss. This report described the design, analysis and test results of the SPMIPv6 protocol realizing these objectives. The SPMIPv6 protocol was based on PMIPv6, extending PMIPv6 with simultaneous binding.

When there is enough preparation time before the actual handover, the handover latency in SPMIPv6 is one RTT between the MN and the MAG. When comparing it to PMIPv6, the handover latency is reduced by 1 RTT between the MAG and the LMA. Handovers towards any layer 2 protocol are possible, since the proactive approach deals with slow layer 2 setups, as for example with UMTS. This report showed that an approach that uses layer 2 MBB combined with layer 3 BBM can handle this. The goal to support both vertical and horizontal handovers towards different layer 2 technologies is thus met.

Using buffering, it is possible to reduce the amount of packet loss during the handover latency period, another goal that was met. The implementation of the protocol in the testbed showed that the amount of buffering is important when TCP is used as the transport protocol. With not enough buffering, TCP may reduce its sending rate due to lost packets. With too much buffering, the wireless link can become congested.

The analysis section showed that when there is not enough time for a full handover preparation, the handover latency gradually increases. In the worst case, the performance of SPMIPv6 is equal to that of PMIPv6. The protocol is capable of handling those imperfect handovers, in which the MN moves to a different access point than expected or may not move at all. Also cases where the handover preparation time is too short are handled gracefully.

The extra bandwidth usage in the network due to mobile nodes using SPMIPv6 was also analyzed. Both correct and incorrect handovers contribute to the overall extra load. The load increases when the preparation time increases to deal with the large layer 2 setup delay of a certain access technology type. The maximum of the extra load is experienced when a MN does not move but stays connected to its current MAG.

Concluding we can say that SPMIPv6 does indeed meet the objectives under some preconditions. These preconditions are: overlapping coverage areas of old and new access network and the MN having two layer 2 connections active at the same time.

5.2. Future work

Future work in the area of the solution presented includes the topics described in this section.

The SPMIPv6 protocol relies heavily on the ability to predict an upcoming handover. How this can be done exactly is not described in this thesis. Future research can thus be done on how to exactly predict this. Research already has been done on how to predict this for horizontal handover, for example between different Wi-Fi access points [48]. This could also be extended towards vertical handovers, for example between Wi-Fi and UMTS. A part of this is gathering information from neighboring networks.

After the actual handover, the nMAG must decide how much buffered data it should send to the newly connected MN. In the current protocol there is no way to determine this. Maybe the MN could tell the nMAG how long it was disconnected, or maybe the nMAG can deduce it based on the information available, e.g., oMAG ID and its network and access technology type.

The implementation in the testbed was not production ready. This implementation could be further developed. Also, the MN implementation was running on PC's. This could also be ported towards PDA's and mobile phones, to actually be able to 'move around'.

Further, simulations could be done with the protocol to get insights in the effect of large scale use of the protocol, for example with many users and/or congested wired and wireless links.

Appendix A: UMTS Connection setup

1. Introduction

1.1 Background

A wide range of wireless communication systems is available nowadays. Technologies like GSM and UMTS offer a large coverage area, but have limited bandwidth and can be expensive to use. On the other side we have Wi-Fi (802.11), which offers a large bandwidth in a small coverage area, for a relatively cheap price. There is also the emerging WiMax technology, sitting in between when looking at both bandwidth and coverage area.

New communication devices like PDAs and smart-phones are able to connect to more than one of the mentioned wireless technologies. When these connections all offer the same capabilities (e.g., IP data transport), the most appropriate (and available) wireless technology can be used alternatively. Thus it would be possible to use Wi-Fi, if the user is within range of a hotspot and needs a lot of bandwidth at that time. But, as said, Wi-Fi does not have a large coverage area. When moving out of range, the ongoing communication sessions should be transferred to, for example, UMTS. This connection transfer is called handover, which typically disrupts the connection for some period called “handover latency”. If one of these sessions is a VoIP call, a large handover delay is not permitted, since this would disrupt the call.

1.2 Handover types

Two types of handover can be distinguished: vertical and horizontal handover. The term vertical handover is used when indicating a handover from one technology (for example Wi-Fi) to another (i.e. UMTS), on which the main focus of this report is. When doing a handover from one Wi-Fi access point to another, we use the term ‘horizontal handover’.

It might be possible to connect to a new access network while still maintaining a connection to the old one. This is called Make Before Break (MBB). The opposite happens when for example a device is not capable of using two network interfaces at the same time or when the device is capable of using two network interfaces at the same time, but has not anticipated a sudden handover. This is called Break Before Make (BBM). This means that the old connection is lost before a new connection to the second network is activated.

1.3 Handover processes

The handover process can be divided into three phases: info collection, decision phase and execution phase. In the first phase, information about the current and possible other wireless links is collected. When the quality of the current link degrades or when a ‘better’ link, with

for example higher bandwidth or lower costs, is available, the decision can be made to do an actual handover (execution phase).

Handover means switching network connectivity at the OSI Data Link layer (Layer 2) and / or the OSI Network layer (Layer 3). In case that there are two connections at Layer 2 only the Layer 3 connectivity has to be switched in case of a handover.

For example, Mobile IP (MIP) tries to accommodate this handover on layer 3.. An addition to this is Fast Mobile IP, which makes fast(er) layer 3 handover possible.

1.4 Problem statement

This report focuses on the scenario described in the beginning of this chapter: handover towards UMTS. In order to do this in a seamless way, one particular issue of this handover is investigated: the way the UMTS connection process works. When this is known, this process can be activated in a way the handover delay is minimized. If we can do certain parts of this process some time before an eminent handover this can positively change the handover delay.

2. UMTS Connection setup

2.1 Delays

In order to show the delays involved in different parts of the UMTS interface activation process, we first divide the messages that are exchanged into different classes according to their endpoints. In Figure 41 these classes are shown. All delays are expressed for connections starting at the UE and terminating at different points in the network, expect for t_{inter}.

The different classes are shows in Table 9.

variable		Description
t_{small}	Small delay	The delay introduced for transferring a message between the UE and the base station (Node B).
$t_{mod\text{-}small}$	Moderately small delay	Delay encountered by messages exchanged between the UE and nodes within the access network, for example an RNC.
$t_{mod\text{-}large}$	Moderately large delay	Message transmission Delay between UE and a node in the visited core network, such as SGSN and GGSN.
t_{large}	Large delay	When a UE needs to send a message to a node in the home core network such as the HLR or AuC, such a delay is encountered.
t_{inter}	Inter system delay	The delay in communication between nodes in the home and nodes in the visited core network.

Table 9: Delay classes

And in a graph in Figure 41.

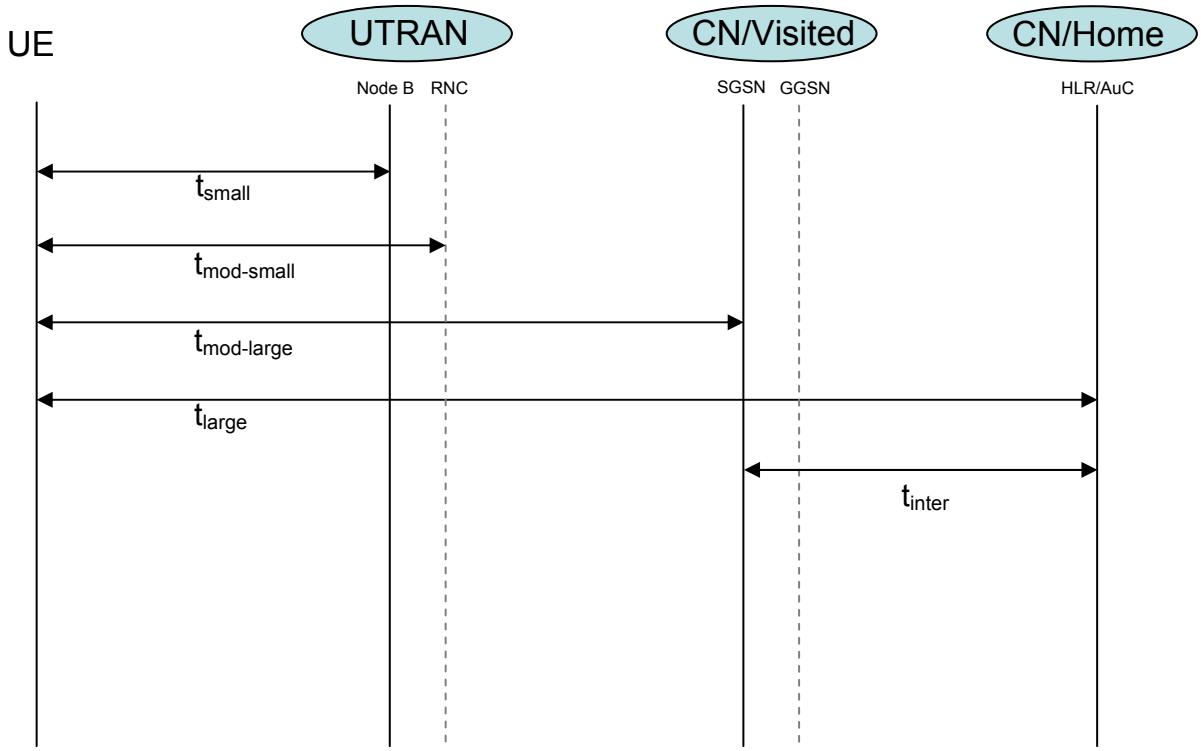

Figure 41: Delay classes

2.2 L2 connection process

Figure 42 and Figure 43 show the actual messages that are exchanged during the connection process [49], [50], [51] ,[52].

The connection process can be divided into three parts: RRC connection setup, GPRS attach and PDP context activation.

2.2.1 RRC connection setup

This process establishes a radio connection between the UE and the RNC. The UE sends an "RRC connection request" over a common control channel to the RNC. Reception of this message causes the RRC state to be changed from IDLE to Cell_FACH or Cell_DCH.

Three messages are exchanged during this part of the setup process. In between, several messages are exchanged between Node B and RNC, but since the delay on the Iub interface is quite small these messages are not displayed in this diagram.

The last message, RRC Connection setup complete, which is sent by the UE, contains three optional messages. If requested in the RRC Connection setup message, this reply by the UE contains info about its UTRAN-specific and its intersystem capabilities. The last optional attribute contains the START values, which are used for further ciphering and integrity protection. There are separated START values for the PS and CN domain. These values both have a certain lifetime [53], after which new START values have to be sent. This maximum

lifetime is set by the operator and stored in the USIM. If none of these messages have to be sent, this message could be omitted, which would speed up this step by one $t_{mod\text{-}small}$.

Total delay of this process: $3t_{mod\text{-}small}$

2.2.2 GPRS Attach

The next step is the GPRS attach. A large part of this process is about authentication. The other part consists registering the location of the UE with the HLR, in the home network. When this is all done an 'attach complete' message is sent to the UE.

An important part of this is the authentication procedure. For this, two messages are exchanged with the AuC. This causes a significant delay if this user is roaming, since the delay between home and visited network can be large, especially when both networks are geographically far apart. This process can be combined with an IMSI attach, which is the PS domain equivalent of GPRS attach.

Total delay of this process: $7t_{mod\text{-}large} + 6t_{inter}$

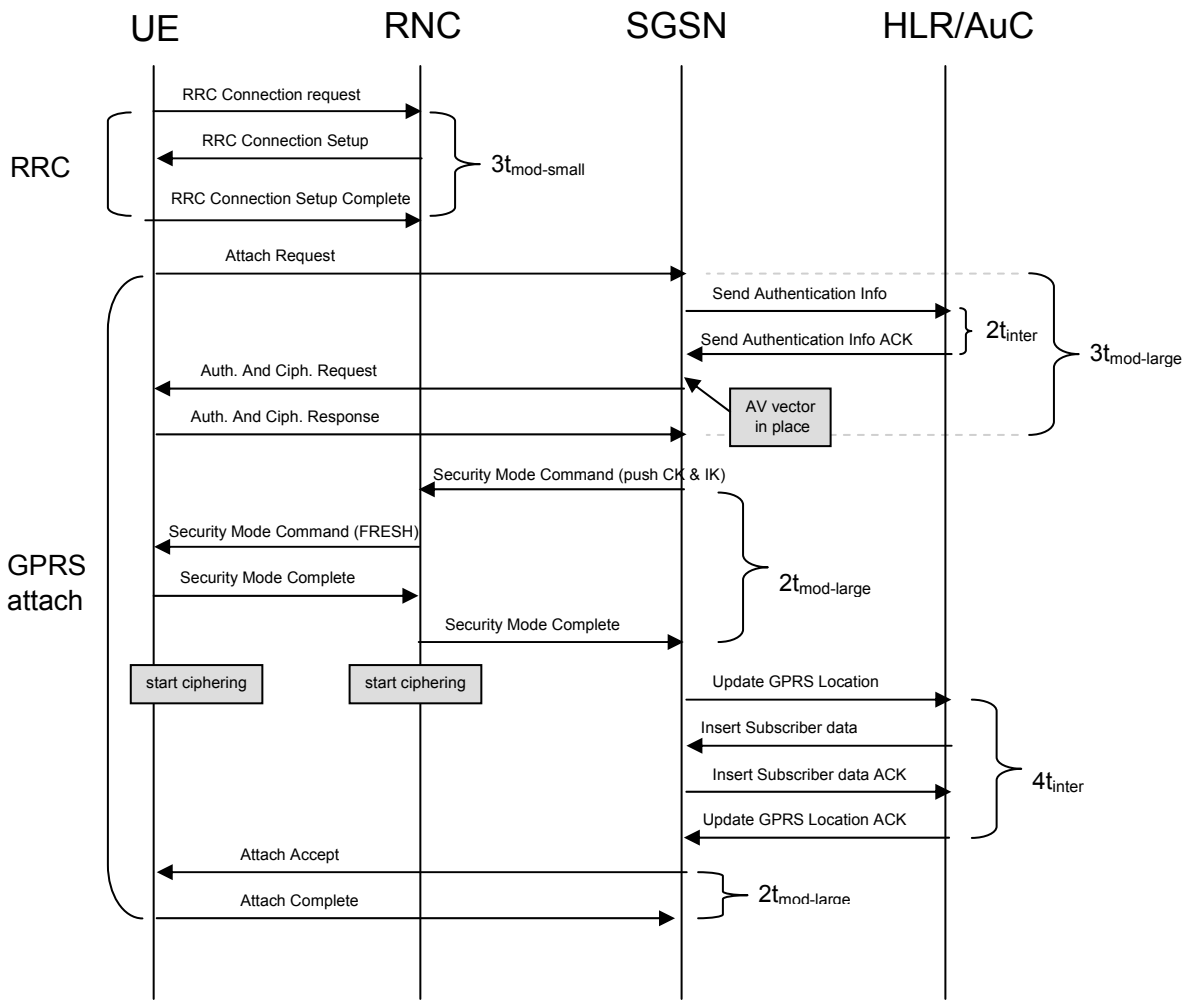

Figure 42: Connection setup messages (1)

2.2.3 PDP Context activation

The UE is now able to execute the last phase of the connection process: the activation of the PDP context. This is shown in Figure 43. At the end of this process the UE has obtained an IP address. The initial request is relayed to the SGSN and contains as a parameter the PS domain to which the UE whishes to attach. The SGSN forwards the message to the GGSN (not displayed in the diagram), which in turn communicates with the RADIUS server to authenticate the GPRS subscription and with the DHCP server, to obtain an actual IP address for the UE. In between, radio bearers (RABs) are also allocated.

Total delay of this process: $4t_{mod\text{-}large}$

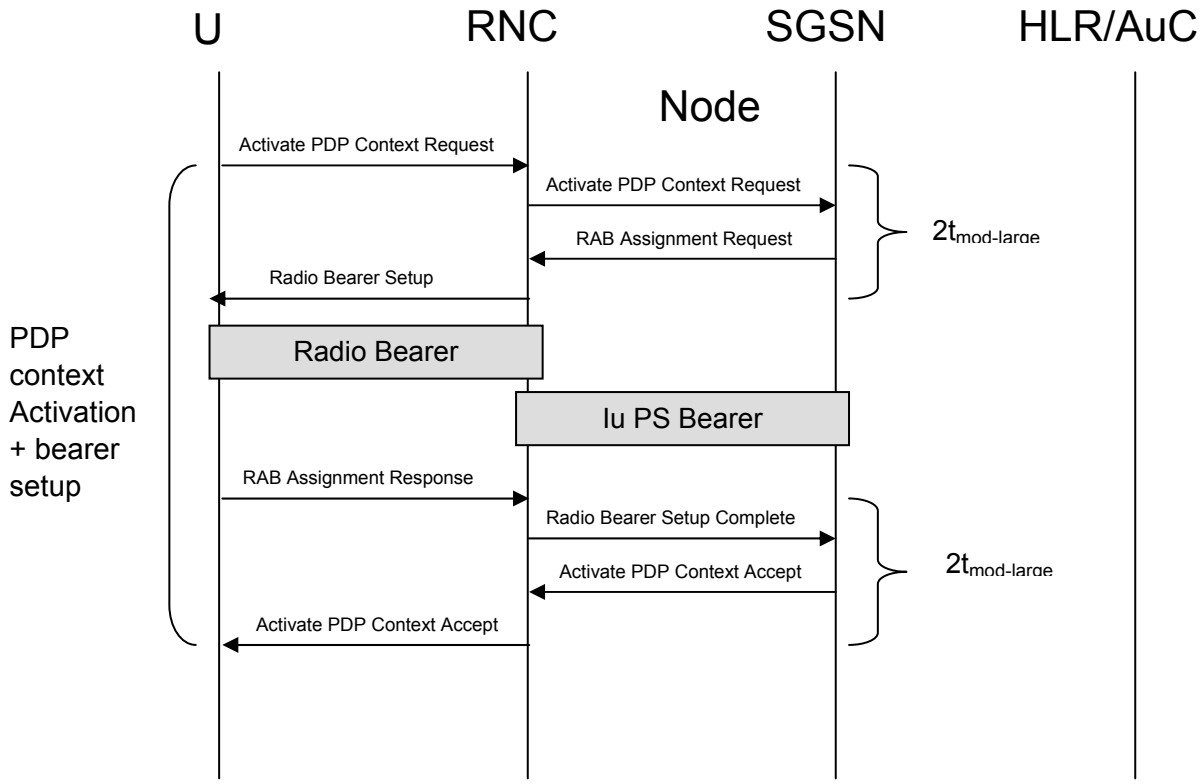

Figure 43: Connection setup messages (2)

2.3 Total delay

The total delay can now be captured in an expression:

Total delay = $3t_{mod\text{-}small} + 11t_{mod\text{-}large} + 6t_{inter}$

Numerical values given in [12] can be assigned to the delay classes defined earlier. These values are estimations, but there is no problem using them to give an indication of the typical delays encountered. The delays include propagation delay as well as processing time in the traversed nodes.

Variable	Description	Estimated value
t_{small}	Processing in the UE (15 ms) Interleaving (20ms) Radio Propagation (0.05 ms)	35 ms
$t_{mod\text{-}small}$	t_{small} Processing Node B (15 ms) (De)multiplexing (8 ms)	60 ms
$t_{mod\text{-}large}$	$t_{mod\text{-}small}$ Processing RNC (5 ms) Iu interface (5 ms)	70 ms
t_{inter}	Processing different nodes (10 ms) Propagation delay inter-domain link (40 ms)	50 ms
t_{large}	$t_{mod\text{-}large}$ t_{inter} Additional processing (20 ms)	140 ms

Table 10: Numerical values for delays

These values can be used in the formula for the total delay:

Total delay = 3*60 + 11*70+ 6*50=1250 ms

2.4 Connection setup components

Since the total UMTS connection delay is much larger than the target handover delay of 50 ms, we have to somehow split up the connection process. If we keep certain parts of the network in the active states, signaling for these parts, is no longer necessary.

The following parts are distinguished:

Component process #	state	description	costs	Est. value
1	RRC connection	The state of the radio connection between UE and RNC.	$3t_{mod\text{-}small}$	180 ms
2	Authentication (Home)	The SGSN contact the AuC in the home network to get the AV vector, used for authentication.	$2t_{inter}$	100 ms
3	Authentication (Local)	The AV vector is already in place, so authentication only requires signaling between UE and RNC/SGSN.	$4t_{mod\text{-}large}$ $+4t_{inter}$	480 ms
4	GPRS Attach	GPRS attachment, excluding authentication.	$3t_{mod\text{-}large}$	210 ms
5	PDP Context	Setting up the PDP context,	$2t_{mod\text{-}large}$	140 ms

		including RABs.		
6	RAB	Setting up RABs for the PDP context. The question is if this is possible	$2t_{mod\text{-}large}$	140 ms

Table 11: Connection process parts

2.5 Selective activation

Several scenarios can now be defined. The numbers behind the scenarios described below indicate which component processes have to be activated in that particular scenario.

2.5.1 Scenario 1 (1,2,3,4,5,6)

The whole connection process has to be executed.

2.5.2 Scenario 2 (1,3,5,6)

In this scenario, communication between home and visited network (in case of roaming) is avoided. This means that we assume that authentication home (2) and GPRS attach (4) are already active.

2.5.3 Scenario 3 (1,3,6)

When compared with the previous case, in this scenario the PDP context (5) is also already active. This means that the UE has a valid IP address, which can be used instantly after allocation of appropriate radio bearers.

2.5.4 Scenario 4 (1)

In the last scenario, only an RRC connection (6) has to be set up. Authentication local (3) has already been done. This is the scenario with the smallest delay.

As we have seen in 0, a speedup one $t_{mod\text{-}small}$ is possible if it is not mandatory that the last message from UE to RNC (RRC connection setup complete) is sent.

2.5.5. Overview

This table shows the estimated connection setup delays per scenario. As can be seen, the value for the ‘best scenario’ (4) is still larger than the target of 50 ms.

Scenario	Delay	Estimated value
1	$3t_{mod\text{-}small} + 11t_{mod\text{-}large} + 6t_{inter}$	1250 ms
2	$3t_{mod\text{-}small} + 8t_{mod\text{-}large}$	740 ms

3	$3t_{mod\text{-}small} + 4t_{mod\text{-}large}$	460 ms
4	$3t_{mod\text{-}small}$	180 ms

Table 12: Delay per scenario

3. Future work

3.1 Triggers

As for now, we have investigated how to do a partial connection setup, but not when. Somewhere in time the actual decision has to be made to start this connection process.

This decision can be made in two stages, denoted in Figure 44 with tr1 and tr2. When the signal quality degrades, the certainty of having to do a handover within some time period is getting higher. The MN can then at a certain point in time, i.e., at tr1, decide to start the connection setup with another wireless network, to cancel out the chance of a very big handover delay if the connection setup process has to be done as a whole.

In the figure, the grey part indicates a connection to both networks. When the certainty of loosing the current connection reaches a certain level and at a time appropriate to accommodate the rest of the handover signaling, the actual handover is executed, i.e., tr2.

The y-ax in the figure represents the certainty of an upcoming handover. This means the following for the two triggers:

At tr1, when there is enough time to activate the UMTS connection from the scratch, the certainty of having a handover is low.
At tr2, we are confident to have the handover AND to complete it in 50ms.

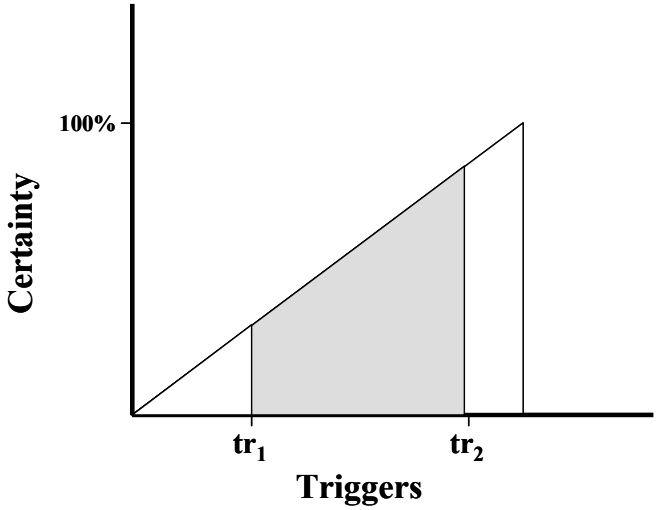

Figure 44: Handover triggers

3.2 UMTS-Wi-Fi integration

In release 7 [54] of the UMTS standard, inter working between WLAN (or a similar IP based packet data network) and UMTS is described. This standard describes two new procedures. The first procedure is the ability to authenticate and authorize connection to the WLAN system through the 3GPP System. The 3GPP AAA server is used for this. The other procedure makes it possible for the WLAN UE to establish IP connectivity to for example the Internet via the 3GPP system. The Packet Data Gateway (PDG) in the 3GPP network makes this possible (see Figure 45). When the WLAN UE whishes to use the 3GPP PS services to obtain Internet connectivity, a tunnel is setup, from the UE to the PDG.

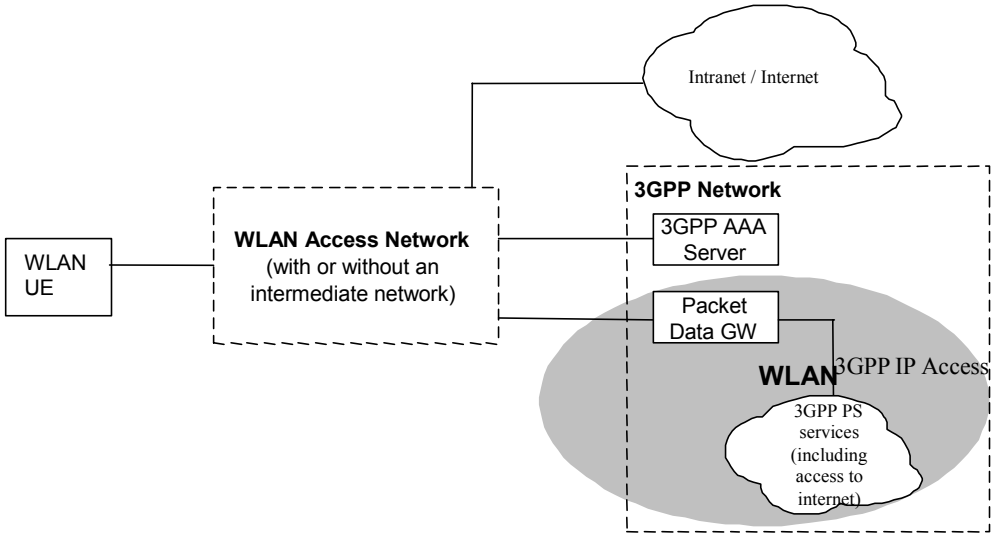

Figure 45: UMTS - UMTS interworking network layout

Actual handover from WLAN to UMTS is not yet described in this standard. This interworking has some advantages however for the handover process. The UE is already authenticated to the 3GPP system, so we can assume this does not have to be repeated when connection to 3GPP system using the UMTS interface. This possibility is supported by the fact that for authentication, the user can use a temporal identity called 'fast re-authentication username', with which a fast re-attach is possible.

3.3 HSDPA

The communication between the UE and the Node-B (first hop in the network) already causes a significant delay. This delay consists of two large components: processing in the UE (15 ms) and interleaving (20 ms). The last delay is in fact caused by the fact the frame size is quite large (10 ms).

This report is based in R99, but since R5 a technique called High Speed Download Packet Access [55] (HSDPA) can be used. The upload variant HSUPA also exists. Using this, the

Transmission Time Interval (caused by the large frame size and interleaving) is fixed to 2 ms, instead of 10-20 ms in earlier releases.

In total, 20 messages are exchanged during connection setup, all traversing the UE-NodeB link. This would mean a speedup of 360 ms for the total process if HSDPA is used. The total delay would then be 890 ms. The variable $t_{mod\text{-}small}$ decreases to 42 ms. Unfortunately, the 'best' scenario still is more than 50 ms, namely 126 ms.

4. Conclusion

This document showed an overview of the UMTS connection setup process. The reason for doing this was to get insight into the delays involved when doing a handover towards UMTS. All the messages that are exchanged during this connection setup were shown and estimated delay values were assigned to them.

The setup process was divided into to different parts, to see what parts should be done in advance to minimize the setup delay to 50 ms, our goal for the total handover delay. Scenario's were introduced, each doing different parts of the connection process in advance.

This report showed that even the 'best' scenario, where as much as possible is done in advance, is still not good enough. The minimum delay, which is for only setting up radio resources between the UE and the RNC, is already around 180 ms.

There are several ways to go now. The HSDPA technique improves things, but it is still not good enough. Another option is to keep the RRC in connected mode, which means keeping radio resources (dedicated channels) available for the UE. We can then only afford to send one message to the Node B then to start using this resource. For this the RRC has to be kept in DHC (direct channel) mode. Which and how this should be done is something for further research.

Future research could also be aimed at the interworking of UMTS and WLAN. But, the large latency of the UMTS network remains a bottleneck in the handover process.

Bibliography

[1] "3GPP Specifications", http://www.3gpp.org/specs/numbering.htm, seen on 27-06-2008

[2] IEEE 802.11 Wireless Local Area Networks, http://www.ieee802.org/11/, seen on 27-11-2008

[3] "WiMAX forum", http://www.wimaxforum.org, seen on 27-06-2008

[4] IEEE 802.16e, "Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems" http://standards.ieee.org/getieee802/download/802.16e-2005.pdf

[5] L. Morand, S. Tessier, "Global mobility approach with Mobile IP in "All IP" networks," in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol.4, no., pp. 2075-2079 vol.4, 2002

[6] 3GPP, "Feasibility Study for Evolved UTRA and UTRAN", TR 25.912, V7.2.0, release 7, June 2007.

[7] 3GPP, "3GPP System Architecture Evolution: Report on Technical Options and Conclusions", TR 23.882, V1.11.0, release 7, July 2007.

[8] "TCP/IP model", http://en.wikipedia.org/wiki/TCP/IP_model, seen on 27-06-2008

[9] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC2460, IETF, December 1998

[10] Information Sciences Institute, University of Southern California , "Internet Protocol, Darpa Internet Program, Protocol Specification", RFC791, September 1981

[11] D. Johnson, C. Perkins and J. Arkko, "Mobility Support in IPv6", RFC3775, IETF, June 2004

[12] 3GPP, "Delay Budget within the Access Stratum", TS 25.853, V4.0.0, Release 4, March 2001

[13] Mobile IT Forum, "4G mobile system requirements document", version 1.1, http://www.mitf.org/public_e/archives/4G_req_v110E.pdf, seen on 04/12/2008

[14] Charles M. Kozierok, "The TCP/IP Guide", http://www.tcpipguide.com/free/, seen on 20-11-2007

[15] The 6NET Consortium, *6net - An IPv6 deployment guide*, September 2005

[16] IEEE, "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority", http://standards.ieee.org/regauth/oui/tutorials/EUI64.html, seen on 25-06-2008

[17] T. Narten, E. Nordmark and W. Simpson, "Neighbor Discovery for IP Version 6", RFC2461, IETF, December 1998

[18] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC2462, IETF, December 1998

[19] N. Moore, "Optimistic DAD", RFC4429, IETF, April 2006

[20] Youn-Hee Han, Seung-Hee Hwang and HeeJin Jang, "Design and evaluation of an address configuration and confirmation scheme for IPv6 mobility support," in *Wireless Communications and Networking Conference*, 2004. WCNC. 2004 IEEE , vol.2, no., pp. 1270-1275 Vol.2, 21-25 March 2004

[21] Chien-Chao Tseng, Yung-Chang Wong, Li-Hsing Yen, Kai-Cheng Hsu, "Proactive DAD: A Fast Address-Acquisition Strategy for Mobile IPv6 Networks," *Internet Computing*, IEEE , vol.10, no.6, pp.50-55, Nov.-Dec. 2006

[22] S. Narayanan *et al*., "Detecting Network Attachment in IPv6 Networks (DNAv6)", Internet Draft, IETF, Feb 24, 2008

[23] H. Zuleger, "Mobile Internet Protocol v6: a short introduction", http://www.hznet.de/ipv6/mipv6-intro.pdf, seen on 28-06-2008

[24] H. Soliman, C. Castelluccia, K. El Malki and L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)", RFC4140, IETF, August 2005

[25] R. Koodli *et al*, "Fast Handovers for Mobile IPv6", RFC4068, IETF, July 2005

[26] Karim El Malki and Hesham Soliman, "Simultaneous Bindings for Mobile IPv6 Fast Handovers", Internet Draft, IETF, July 2005

[27] C. Perkins, "IP Mobility Support for IPv4", RFC3220, IETF, Jan 2002

[28] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury and B. Patil, "Proxy Mobile IPv6", Internet Draft, IETF, May 30, 2008

[29] Shariq Haseeb and Ahmad Faris Ismail, "Comparative Performance Analysis for Mobile IPv6 Protocols: Special Reference to Simultaneous Bindings", *Journal of Computer Science*, vol.2, no.2, pp. 154-159, Februari 2006

[30] XP Costa, R Schmitz, H Hartenstein, M Liebsch, "A MIPv6, FMIPv6 and HMIPv6 handover latency study: analytical approach", in *IST Mobile & Wireless Telecommunications Summit 2002*, Thessaloniki, Greece, June 17-19, 2002, pp. 100-105.

[31] H. Petander, "Bicasting with Buffering and Selective Delivery for Fast Handovers for Mobile IPv6", Internet Draft, IETF, October 16, 2006

[32] H. Petander, "Bicast packet identification header", Internet Draft, IETF, October 16, 2006

[33] Virtual WiFi (MultiNet), http://research.microsoft.com/netres/projects/virtualwifi/default.htm, seen on 7-11-2008

[34] IEEE802.21 Working Group, Media Independent Handover Services, http://www.ieee802.org/21/, seen on 24-10-2008

[35] Alper E. Yegin *et al*, "Supporting Optimized Handover for IP Mobility - Requirements for Underlying Systems", Internet Draft, IETF, June 2002

[36] IANA, "Mobile IPv6 parameters", http://www.iana.org/assignments/mobility-parameters, seen on 7-12-2008

[37] Patel, A., Leung, K., Khalil M., Akhtar, H., Chowdhury, K., "Mobile Node Identifier Option for Mobile IPv6 (MIPv6)", RFC4283, IETF, November 2005

[38] Aboba, B., Beadles, M., Arkko, J., Eronen, P., "The Network Access Identifier", RFC4282, IETF, December 2005

[39] Mills, D., "Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI", RFC4330, IETF, January 2006

[40] Matthis Flege, Internship report, internal report, DoCoMo Communications Laboratories Europe GmbH, June 22, 2007

[41] Network-based Localized Mobility Managemen WG, http://tools.ietf.org/wg/netlmm/, seen on 7-2-2008

[42] Tcpdump, http://www.tcpdump.org/, seen on 26-06-2008

[43] Madwifi, www.madwifi.org, soon on 25-06-2008

[44] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor *et* al, "Stream Control Transmission Protocol", RFC2960, IETF, October 2000
[45] D. Thaler, "Multilink Subnet Issues", RFC4903, IETF, June 2007
[46] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC3550, IETF, July 2003
[47] Stevens, W. R., *TCP/IP Illustrated (Vol. 1): the Protocols*. Addison-Wesley Longman Publishing Co., Inc., 1993.
[48] P. Bellavista, A. Corradi, C. Giannelli, "Adaptive Buffering-Based on Handoff Prediction for Wireless Internet Continuous Services", in *High Performance Computing and Communications,* Volume 3726/2005, p. 1021-1032
[49] 3GPP, "UTRAN functions, examples on signaling procedures", TS 25.931, V3.7.1, Release 1999, June 2006
[50] "GPRS Attach and PDP Context Activation", http://www.eventhelix.com/RealtimeMantra/Telecom/gprs_attach_pdp_sequence_diagram.pdf, seen on 15-03-2007
[51] R. Kreher, T. Rüdenbusch, *UMTS Signaling*, Wiley, 2005
[52] H. Kaaranen, A. Ahtiainen, L. Laitinen, S. Naghian and V. Niemi, *UMTS Networks: Architecture, Mobility and Services*, Wiley, 2005
[53] 3GPP, "3G security; Security architecture", TS 33.102, V3.13.0, Release 1999, December 2002
[54] 3GPP, "3GPP system to Wireless Local Area Network (WLAN) interworking; System description", TS 23.234, V7.5.0, Release 7, March 2007
[55] 3GPP, "High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2", TS 25.308, V7.1.0, Release 7, December 2006