Authentication, Authorization and Accounting within an Access Network using Mobile IPv6 and Diameter

ANTON LARSSON
Authentication, Authorization and Accounting within an Access Network using Mobile IPv6 and Diameter

Anton Larsson

Examiner and supervisor: Andreas Jonsson
External supervisor: Richard Nelson
Centre for Telecommunication and Information Engineering, Monash University, Australia.

Melbourne, December 2002

Master of Science Thesis at
Computer Science and Electrical engineering
Luleå University of Technology
Abstract

The large expansion of Internet requires a change of hierarchy, future users will be more mobile than today users and information that previously were static will become dynamic. To ensure further growth of Internet the development of a scalable Authentication, Authorization, Accounting (AAA) infrastructure is critical since AAA is a prerequisite for those who want to create new services. DIAMETER is proposed to be the next generation Internet AAA infrastructure and is currently under development by the Internet Engineering Task Force (IETF) AAA working group.

This thesis aims to give an insight into issues that arises when trying to combine Mobile IP version 6 (MIPv6) access networks with DIAMETER. As an initial part the thesis will provide the reader with fundamental information covering IP version 6 (IPv6), Mobile IPv6, AAA fundamentals and DIAMETER.

The investigation was performed through extensive literature studies, design cases and an lightweight test implementation.

The thesis shows that it is possible to combine Mobile IP version 6 and Diameter. But in order to make the combination easier the structure of the IPv6 stack and the Mobile IPv6 extension has to be slightly redesigned. The thesis also shows how Authentication can be performed without adding severe delay to the handover in an controlled access network.
Acknowledgements

This master thesis was performed at The Centre for Telecommunication and Information Engineering at Monash University, from July 2002 to December 2002.

The author would like to thank: Andreas Jonsson and Dr. Richard Nelson for being his supervisors, Greg Daley and Nick Moore for many discussions related to MIP version 6 and Sven Molin for starting the exchange between Luleä University of Technology and Monash University making this thesis possible.

*Melbourne, 2002-12-10*

*Anton Larsson*
# Contents

1 Introduction ............................................. 1  
   1.1 Motivation ........................................ 1  
   1.2 Previous work ..................................... 1  
   1.3 Research objectives ............................... 2  
   1.4 Method ............................................ 2  
   1.5 Limitations ....................................... 2  
   1.6 Thesis outline .................................... 3  

2 Technology overview ................................. 4  
   2.1 IPv6 ............................................... 4  
      2.1.1 Introduction ................................ 4  
      2.1.2 Headers .................................... 5  
      2.1.3 Addresses .................................... 6  
      2.1.4 Address configuration ....................... 7  
      2.1.5 Stateful configuration ....................... 8  
   2.2 ICMPv6 ............................................ 9  
   2.3 Mobile IPv6 ....................................... 9  
      2.3.1 Introduction ................................ 9  
      2.3.2 Security .................................... 11  
      2.3.3 Mobility Header ............................. 11  
      2.3.4 New ICMPv6 messages ....................... 13  
      2.3.5 Mobile IPv6 route optimization .......... 14  
   2.4 Diameter .......................................... 14  
      2.4.1 Introduction ................................ 15
5 Summary

5.1 Contribution ........................................... 38
5.2 Conclusions ............................................ 38
  5.2.1 Combining Mobile IPv6 and DIAMETER .......... 39
  5.2.2 About key distribution .......................... 39
  5.2.3 The development process ......................... 39
  5.2.4 Using UML ........................................... 39
  5.2.5 Extending Ethereal ............................... 40
  5.2.6 Opensource ......................................... 40
5.3 Future work ............................................ 40

A Acronyms ...................................................... 45

B MIPv6 Testbed configuration ............................ 48
  B.1 Introduction .......................................... 48
  B.2 Mobile IPv6 for Linux ............................... 48
  B.3 Kernel 2.4.18 ........................................... 49
  B.4 User Mode Linux ....................................... 49
  B.5 GBootRoot ............................................. 49
  B.6 Ethereal ............................................... 50
  B.7 Redhat 7.3 ............................................. 50
  B.8 Diameter libraries ................................... 50
  B.9 Router advertisement daemon ...................... 50
  B.10 List of software ..................................... 51

C Protocol specification .................................. 52
  C.1 AAA challenge option ................................. 52
  C.2 AAA protocol messages ............................... 53
    C.2.1 AAA Protocol Message Options ................. 54
    C.2.2 Client Identifier option ......................... 54
    C.2.3 Security Data option ............................. 55
    C.2.4 Challenge Option .................................. 55
    C.2.5 Generalized Key Reply Option ................. 56
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>IPv6 Header</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>IPv6 header with an extension header</td>
<td>5</td>
</tr>
<tr>
<td>2.3</td>
<td>Mobile IPv6 Header</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>Route optimization used in Mobile IPv6</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>Triangular routing used in Mobile IPv4</td>
<td>14</td>
</tr>
<tr>
<td>2.6</td>
<td>Infrastructure of a AAA network</td>
<td>17</td>
</tr>
<tr>
<td>2.7</td>
<td>Diameter application architecture</td>
<td>17</td>
</tr>
<tr>
<td>2.8</td>
<td>Diameter message format</td>
<td>20</td>
</tr>
<tr>
<td>2.9</td>
<td>Diameter AVP format</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Mobile node receives challenge and issues authentication request.</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>Operating in basic mode</td>
<td>27</td>
</tr>
<tr>
<td>3.3</td>
<td>Operating in extended mode</td>
<td>27</td>
</tr>
<tr>
<td>3.4</td>
<td>Security Associations</td>
<td>28</td>
</tr>
<tr>
<td>4.1</td>
<td>MIPL module architecture</td>
<td>31</td>
</tr>
<tr>
<td>4.2</td>
<td>Fundamental network</td>
<td>35</td>
</tr>
<tr>
<td>4.3</td>
<td>Diameter aware network</td>
<td>36</td>
</tr>
<tr>
<td>4.4</td>
<td>Handover capable network</td>
<td>37</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

As the Internet keep growing in number of services available and users online the need of mobility increases rapidly. Accurate authentication, authorization and accounting procedures are required to ensure further development and growth of Internet. IETF has put much effort into developing a replacement of the currently used AAA protocol, RADIUS [RW00]. The replacement is named DIAMETER [CA02]. DIAMETER borrows heavily from RADIUS but many of the drawbacks, as scalability and configuration issues, has been resolved.

MIP version 6 (MIPv6) [JP02a] is developed as an extension to IPv6 [DH98] in order to handle mobile computers. IPv6 itself is developed to solve issues as route ability, address space capacity and security flaws in previous protocol versions as IP version 4 (IPv4). The adaptation toward an AAA protocol is a natural step in the development cycle of MIPv6. With an adaptation to DIAMETER an increased usability is ensured as DIAMETER is supposed to be the next generation AAA protocol [Ven02], the replacement of RADIUS.

1.2 Previous work

There has been a lot of research effort put into developing MIPv6 into a solution better than MIP version 4 (MIPv4), especially as MIPv4 requires much network equipment and have several security issues. The research put into DIAMETER has been concentrated in creating a more unified solution than RADIUS, resulting in a solution containing a base protocol with add on appli-
cations. The development of combining these areas has been suffering from this focus and it is important to see what issues that arise when combining MIPv6 with DIAMETER. Also the deployment of DIAMETER relies heavily in its suitability as the AAA protocol to be used with MIPv6 since mobility probably will be the main service in future service provider networks. Work has been done to create a MIPv6 application for DIAMETER[FL02]. Work has also been put in to transferring AAA credentials from the MIPv6 node to the access router [PE02].

1.3 Research objectives

This thesis aims to give more insight into different issues that can arise when combining MIPv6 and DIAMETER. As the time of the thesis is limited the author has focused on the following questions.

- Perform a literature study within the field of DIAMETER and MIPv6 with focus in how to create an access network for MIPv6 using DIAMETER.

- Design an access network based on the literature study.

- Create a test implementation of the access network containing MIPv6 nodes, access routers and DIAMETERenteties.

1.4 Method

The questions raised in the Research objectives will be addressed through literature studies and discussion through the thesis. Case studies will mainly be performed on a virtual network which is further described in Appendix B. The reader of this thesis will also be given a crash course in MIPv6, DIAMETER and ICMP version 6 (ICMPv6) in order to easier follow the on-going discussion through out the thesis, a test implementation will finally be presented.

1.5 Limitations

This thesis does not aim to give a complete description of DIAMETER, IPv6, MIPv6. It is not supposed to be a reference of these above stated standards.
In case of more interest there are several documents at the IETF web pages that better suit that purpose of reference. The features mentioned in this thesis aims only to fulfil the goal of understanding how Diameter can be used with MIPv6 in order to grant a mobile node access to a network. The main target of the thesis is to give insight to the mechanism involved in a access network with MIPv6 nodes.

1.6 Thesis outline

The thesis is organized as follows:

- **Technology overview** - A description of MIP version 6, AAA basics and Diameter, how they work and their characteristics

- **The access network** - The author presents different options available when designing an access network and explains the use of different entities within an access network.

- **Implementation** - Description of an implementation of a access network including modifications to the MIPv6 kernel module for Linux, AAA-server and Access Router (AR).

- **Summary** - an analysis of the implemented access network as well a discussion about possible improvements and future work.
Chapter 2

Technology overview

2.1 IPv6

This section gives the reader a short introduction to IPv6 and aims to fulfill the requisite of knowledge so that the reader more easily will understand the rest of this thesis.

2.1.1 Introduction

The development of IPv6 \(^1\) started in 1994 by IETF, the reason for starting the development was mainly the limited address space but also security and routing issues. As IPv4 was developed in the early 1970’s, since then requirements of a global protocol with millions of users have changed radically, some of the improvements in IPv6 are:

- Expanded Addressing Capabilities
- Header Format Simplification
- Improved Support for Extensions and Options
- Flow Labeling Capability
- Authentication and Privacy Capabilities

\(^1\) Also known as IPng or Next Generation Internet Protocol
In order to reduce bandwidth overhead and the computational cost of processing large headers, IPv6 headers contain less options than its predecessor IPv4. Some of the old IPv4 header options have become optional and can be added through extension headers. Figure 2.1 shows an IPv6 header.

**Extension headers**

In order to add more optional functionality to the Internet layer, extension headers have been defined. The extension header is added in between the IPv6 header and higher level protocols, the technique is referred to as piggybacking, see Figure 2.2. Currently there are 6 different extension headers defined.
• **Hop-by-Hop options header** - Contains information to be processed by every intermediate node upon the packets delivery path.

• **Routing header** - Defines nodes to be visited by the packet before delivery to the destination interface.

• **Fragment header** - Used by the source node when the packet is too large to fit into the link MTU. Intermediate routers are not allowed to use this header, instead they are supposed to drop the packet and send an ICMPv6 packet with a PACKET_TOO_LARGE error code to the source address.

• **Authentication header** - Used by the receiver to authenticate the sender of a packet.

• **Encrypted security payload** - Contains additional information needed to decrypt the encrypted payload.

• **Destination option header** - Contains additional information that is only needed to be examined by the packets destination node(s).

### 2.1.3 Addresses

One of the main reasons of creating a new version of the Internet Protocol (IP) was the lack of address space. The new IPv6 address [HD98] is based upon 128 bits and offers several millions of addresses to each square metre of the area covering the earth. With this amount of addresses it is very unlikely that the addresses will be consumed.

#### Types

There are different types of addresses in IPv6, these are:

• **Unicast** - Is used to address a single interface.

• **Multicast** - Is used to address a group of interfaces.

• **Anycast** - Is used to address a group of interfaces, but the packet is only sent to one of them. The interface is selected using a policy, most commonly the closest.
Notation

The most basic notation of an IPv6 address is eight groups of four hexadecimal digits separated by semicolons.

**FE38:BE31:0000:0000:0000:1111:FFFF:2200**

While this notation might seem to be a bit time consuming there exist shorter notations, if there are consecutive zeros in a segment they can be replaced by '::' this means that the above address can be written like this:

**FE38:BE31::1111:FFFF:2200**

When presenting network prefix, this is done by adding a '/n' to the end of the address, where n is the number of digits in the prefix.

**FE38:BE31::1111:FFFF:2200/64**

The given prefix is:

**FE38:BE31::**

### 2.1.4 Address configuration

There are two ways of automatically assigning valid IPv6 addresses to an interface, they are called stateless and stateful auto configuration.

**Unspecified address**

Before the auto configuration is performed, the hosts source address contains of solely zero’s. By definition it is called the unspecified address.

0:0:0:0:0:0:0:0 or ::

**Link-local address**

The link-local address is only valid on the local network and is always constructed the same way. The link-local address is identified by the prefix:

**FE80::/64**

The other 64 bits, forming the rest of the address, is constructed out of the node’s Media Access Control Address (MAC) address. The MAC address is only 48 Bit in length so it has to be padded in order to be a valid IPv6 address. The padding is done by adding FFFE to the Organizationally Unique Identifier (OUI). The OUI is the 3 octets in the beginning of a MAC address so if the MAC address is:

**AB 34 12 ED AA F1**

The OUI is:
The final link-local IPv6 address should become:

**FE80:0:0:0:AB34:12FF:FEED:AAF1**

### Stateless configuration

The stateless autoconfiguration mechanisms [TN98] allows a host to create a valid global IPv6 address using information available through router solicitations and parameters at the host. The different steps involved are listed below:

1. The node creates a link-local address

2. The node sends a neighbour solicitations message to all nodes Multicast group asking if someone else is using the tentative link-local address, this is also referred to as Duplicate address detection (DAD)

3. If the address was not in use by any other interface, the tentative link-local address is assigned and the node has IP level connectivity. Otherwise manual configuration is required.

4. The node sends a router solicitation message to all routers Multicast group and asks for subnet prefix.

5. The node receives a subnet prefix and constructs an address using the prefix and the MAC-address.

6. The node performs DAD on the new address through all nodes Multicast group, and if it is unique the address is assigned to the nodes interface.

As the procedure of stateless autoconfiguration may take several seconds, step 1-3 and 4-6 can be performed in parallel. Also, the node is told via the router advertisements wether it is allowed to use stateless autoconfiguration or if it should deploy stateful configuration via DHCP version 6 (DHCPv6).

### 2.1.5 Stateful configuration

The stateful autoconfiguration [DB02] is also known as DHCPv6, and is an IPv6 version of the Dynamic Host Configuration Protocol (DHCP) that was used with IPv4. The protocol shortly performs the following steps:
1. The node joins the all DHCP servers and relay agents Multicast group and sends a request using its link-local address, the request is sent using Uniform Datagram Protocol (UDP).

2. If there is no DHCP servers available on the local subnet a relay agent will forward the message to the correct DHCP server.

3. A reply is formed and sent directly to the requesting host or via a relay agent, the Multicast group is not to be used.

In order to keep an address the node has to continuously update it toward the DHCP server during the lifetime of the address.

2.2 ICMPv6

ICMPv6 is the Internet Control Message Protocol developed to suit the needs of IPv6. ICMP version 4 (ICMPv4) was used together with IPv4 and the ICMPv6 has inherited many features from ICMPv4. One of the new features within ICMPv6 is the support of Internet Group Management, in IPv4 this was a separate protocol called Internet Group Management Protocol (IGMP). ICMPv6 is mainly used to submit messages and options between nodes participating in a exchange of data. As an example is ICMPv6 involved in stateful and stateless address configuration, IPv6, MIPv6 and the neighbour discovery scheme.

2.3 Mobile IPv6

This section explains how mobility can be obtained within extensions to the current IPv6 protocol, the section aims to give the reader an introduction to MIPv6 in according to draft number 15 [JP02a].

2.3.1 Introduction

In order to extend IPv6 with mobility support IETF introduced MIPv6. By the use of MIPv6 a node can change its point of attachment without affecting higher level applications. Primarily the solution is the use of multiple addresses, one home address combined with one or several care-of-addresses. The node is always reachable by its home address, even though its current
point of attachment is unknown to the sender. As the node changes its current point of attachment its care-of-address changes while its home address remains unchanged. When introducing MIPv6 there are several new terms.

- **Correspondent Node (CN)** - The other part in a peer communication, the node may either be mobile or stationary

- **Home Agent (HA)** - A router on the mobile nodes home link. While away the mobile node has registered its current care-of-address within the router. The router can then encapsulate and forward all packets destined to the home-address to the mobile nodes current primary care-of-address.

- **Mobile Node (MN)** - A node that can change it current point of attachment to the network whilst still being reachable via its home address.

- **Access Router (AR)** - A router on the visited network, the router can control ingress and egress traffic.

To achieve the higher level of mobility MIPv6 has to introduce new mechanisms into the IPv6 protocol.

- **Movement discovery** - The mobile node has to discover when it has changed its point of attachment in order to receive/construct a new care-of-address.

- **Binding update** - When the mobile node has received a new care-of-address it has to register it at the home agent. Otherwise it will be impossible to open new sessions. The node also has to tell its corresponding nodes in order to maintain current sessions.

- **Tunneling** - If someone tries to open a new connection to the mobile node while it is away from the home link, the home agent has to tunnel the packet to the primary care-of-address registered within the home agent.

The modification mentioned are the main changes, other extensions as Hierarchical MIPv6 mobility management (HMIPv6) [SC02] are continuously discussed and developed by the research community. Especially as MIPv6 has not reached Request For Comment (RFC) level yet, extensions to MIPv6 are explained in detail at the IETF web site [IET02].
2.3.2 Security

One of the major issues with MIPv4 was the security flaws inherited from IPv4. When handling mobile nodes the security becomes more important, otherwise we would never be able to tell who is changing the primary care-of-address at the home agent or performing a binding update at the correspondent node. As IPv6 has introduced functions as Encapsulated Security Payload (ESP) and Authentication Header (AH) also known as IP Security Protocol (IPsec), MIPv6 can use these functions to remove previous security issues.

Mobile node towards Home agent

When exchanging binding information between the mobile node and its assigned home agent ESP, AH can be used. The main problem of this is how to convey the keys to use, some techniques uses mechanisms as Internet Key Exchange (IKE) or the AAA backbone to distribute the keys.

Mobile node towards Correspondent node

When sending Binding update (BU)s between the Mobile Node (MN) and the Correspondent Node (CN), there is a need for authentication. MIPv6 draft 18 [JP02b] solves the issue with distribution of keys by the exchange of cookies. The exchange is performed throughout the conversation between the nodes therefore Evil Attacker (EvA) must control all network paths in order to interfere in the exchange of keys.

2.3.3 Mobility Header

In order to send necessary MIPv6 information between involved nodes, a mobility header has been developed. The mobility header is inserted into the IPv6 packet in the same way as an extension header and contains different options necessary for mobility support. The header format is illustrated by figure 2.3.

- **Proto** - Identifies the header immediately following the Mobility header, uses same notation as next header in IPv6.

- **Length** - 8-bit unsigned integer, length of entire header, should have a length with multiple of 8 octets.
Defined mobility options

There are several different message types, the reason why these exist is that additional information must be sent between the MN, the CN and the Home Agent (HA) in order to support the mobility mechanisms in MIPv6.

- Binding Request (BR) option - Sent by a CN interested sending packets directly to the MN. As a response the MN should respond with a BU message.

- Binding Update (BU) option - The BU is sent by the MN to inform the CN, the HA and eventually dynamically assigned HAs about its new Care-of-address (CoA).

- Binding Acknowledgement (BA) option - Used by nodes that received a BU, the purpose of sending a Binding Acknowledgement (BA) is to tell the MN that the BU was successful.

- Home address option - Sent by the MN when connected to a foreign network \(^2\) with the purpose of telling the CN its correct HA address.

\(^2\)Routers applying ingress filtering will not forward packets with source addresses from other networks, like the HA address.
2.3.4 New ICMPv6 messages

To add full mobility to MIPv6 some new ICMPv6 messages have been defined. They are mainly used to detect movement, receive network prefixes, locate dynamic HAs and receive information about what ARs to use.

- **Home Agent Address Discovery Request** - Sent by the MN when looking for a dynamically assignable HA.

- **Home Agent Address Discovery Reply** - Sent by a router supporting dynamic Home Agent assignment as a reply to the HA discovery request sent by the MN.

- **Mobile Prefix Solicitation** - Sent by the MN to gather information about the home network and the HA.

- **Mobile Prefix Advertisement** - Sent by the HA to the MN, the message contains information about the home network and what prefixes the MN can use when contacting the HA.

![Figure 2.4: Route optimization used in Mobile IPv6](image)
2.3.5 Mobile IPv6 route optimization

Due to changes in IPv6 mechanisms such as neighbor discovery [NN98] and address autoconfiguration [TN98] there is no longer any need of a Foreign Agent (FA) such as introduced in MIPv4. In order to minimize the delay introduced by triangular routing as used in MIPv4, MIPv6 has support for route optimization, see figure 2.4. The difference is that when a session has been setup the rest of the packets are sent between the MN and the CN directly, instead as in MIPv4 where the CNs packets are routed via the home agent and sent in a tunnel to the MN, see figure 2.5.

2.4 Diameter

In this chapter the AAA protocol Diameter [CA02] will be presented and discussed, the objective of the chapter is to give the reader an introduction to AAA, Diameter and the different Diameter applications available.
2.4.1 Introduction

**Diameter** is a AAA protocol developed by the IETF and is supposed to replace **Radius**. The development toward a new standard has been performed in close relation to the larger companies in the world and according to investigations done [Ven02], **Diameter** is the most suitable AAA solution to become the major AAA protocol of the future.

2.4.2 AAA

As internet grows the amount of administrative task is exploding. As an Internet Service Provider (ISP) you will need to keep track of every user, this is done by collecting accounting information. The information can consist of user statistics, billing information and access logs. In order to give this functionality to the ISP, Authentication, Authorization, Accounting protocols have been constructed. These are called AAA protocols.

**Authentication**

Authentication is the act of identifying who you are. One way of doing this could be to send a username / password pair to the authentication authority, other credentials could be key exchange, fingerprint, voice etc. The definition of authentication is given by the following sentence [CA02]:

"The act of verifying a claimed identity, in the form of a pre-existing label from a mutually known name space, as the originator of a message (message authentication) or as the end-point of a channel (entity authentication)"

Given the wrong premises the process of authentication results in an error, as a result access to the requested resource is denied.

**Authorization**

Authorization is the act of granting access to a requested resource. The standard definition yields [CA02]:

"The act of determining if a particular right, such as access to some resource, can be granted to the presenter of a particular credential.”

15
Accounting

Accounting is the act of keeping records of a particular user’s usage of a resource. The purpose of accounting could be future billing, resource planning, or monitoring of EvA. The definition of accounting is [CA02]:

"The act of collecting information on resource usage for the purpose of trend analysis, auditing, billing, or cost allocation."

Architecture

In order to increase scalability, usability and maintainability, the records of a certain network is stored centrally. To increase accessibility the entry points or service requestors are located at the edges, very often at the Access Router or Network Access Server (NAS). The procedure of using Diameter could be described as follows:

1. A user connects to the AR submitting info about requested service, and user credentials.

2. The AR forwards the received user information to the central AAA-server.

3. The AAA-server replies and tells the AR whether access is granted or denied to the requested service.

4. The reply is forwarded to the user.

This way of operating the AAA infrastructure makes it easy to control and maintain the records of the AAA-server, a single AAA-server can handle several AR and increases the accessibility to the services while not situated at the AR. If additional services is introduced, only the AAA-server has to be upgraded in order to support the new service.

2.4.3 Base protocol

Diameter is built up from a base protocol which can be extended in order to suit the needs of an ISP. The extensions are called applications.
Figure 2.6: Infrastructure of a AAA network

Figure 2.7: Diameter application architecture
2.4.4 Diameter Application

In order to maintain the concurrency of the DIAMETER base draft [CA02] new addons to the DIAMETER standard have to be created as substandards and incorporated as plugin modules to the main draft. The modular architecture of DIAMETER is illustrated in figure 2.7. Examples of different applications and their names are presented below.

NASREQ

The Network Access Server Requirements (NASREQ) application is used for AAA purposes in a PPP/SLIP Dial-Up and Terminal Server Access environment. The NASREQ application, combined with the DIAMETER base protocol, satisfies the requirements stated by the IETF. The application is further described within the DIAMETER NASREQ Application draft [CB02].

Mobile IPv4

The MIPv4 application allows a DIAMETER server to authenticate, authorize and collect accounting information for MIPv4 services rendered to a mobile node. Combined with the Inter-Realm capability of the base protocol, this application allows mobile nodes to receive service from foreign service providers. DIAMETER Accounting messages will be used by the foreign and home agents to transfer usage information to the DIAMETER servers. The application is further described within the DIAMETER MIPv4 Application [CJ02].

CMS Security

The DIAMETER base protocol supports either IPsec or Transport Layer Security (TLS) for integrity and confidentiality between two DIAMETER peers. DIAMETER also allows the peers to communicate through relay and proxy agents. Relay agents performs message routing and do not modify DIAMETER messages, except routing information. Proxy agents, on the other hand actively modifies DIAMETER messages. The Cryptographic Message Syntax (CMS) Security application describes how a security association is established by two peers through agents, and how authentication, integrity, confidentiality plus data origin authentication are achieved by using a mixture of symmetric and asymmetric transform. This is achieved by encapsulating CMS data within the DIAMETER Attribute Value Pair (AVP)s. The appli-
cation is further described within the DIAMETER CMS Security Application draft [CF02].

**ROAMOPS**

This application defines how roaming between networks should be performed and administrated. The document describes a broker to use between different domains in order to setup trust relations between different ISPs. By experience from previous roaming scenarios it was decided to create a broker that dynamically could set up relations between different administrative domains. Setting up those relations manually would be far to complex. The application and the Roaming Operations (ROAMOPS) requirements are defined within the RFC document Criteria for Evaluating Roaming Protocols [AZ99].

**Future applications**

Future applications can be added using the DIAMETER API [KC02] specified for programming using C. Sun Microsystems test implementation SunWaal follows the DIAMETER Application Programming Interface (API). There also exists another draft named DIAMETER C++ API, it is defined for use with C++. The opendiameter implementation is based upon the DIAMETER C++ API and is released under Lesser GNU Public License (LGPL), it is available through the opendiameter homepage [TAR02].

**2.4.5 Diameter messages**

A DIAMETER message consist out of two parts, a header and a list of suitable AVPs.

- **Version** - A field describing the version of the DIAMETER protocol used
- **Message Length** - Length of the diameter message, header length included.
- **Flags** - There are eight bits of different flags these are:
  - **R(equest)** - If set, the message is an request, otherwise it is a reply.
  - **P(roxiable)** - If set the message may be relayed or proxied, otherwise the message must be locally processed.
Figure 2.8: Diameter message format

- **E(error)** - If set the message contains a protocol error, and the message will not confirm the Augmented Backus-Naur Form (ABNF) format [CO97] described in the DIAMETER draft [CA02].

- **Command-code** - Contains the actual command sent within this message, the command codes are managed by Internet Assigned Numbers Authority (IANA).

- **Vendor-ID** - If IETF based command-codes are to be used the vendor-ID has to be set to zero, Vendors wishing to implement their own Command-codes has to set their vendor-ID to their IANA assigned "SMI Network Management Private Enterprise Codes" in order to avoid usage and collisions with other vendors command address space.

- **Hop-by-Hop Identifier** - An identifier used to match request with replies. A sender has to ensure the ID is unique on the link in order to avoid confusion.

- **End-to-End Identifier** - An identifier used to detect duplicate messages. Duplicate messages are silently discarded.

- **AVPs** - Can contain AAA, routing or security information as well as configuration details about the request and reply.
2.4.6 Diameter AVP

An Attribute Value Pair defines an option (eg. password, username, replay protectors). The AVP always consists of a header followed by its data.

Here is a short explanation of the option fields within a Diameter AVP.

- **AVP Code** - The AVP code combined with the Vendor ID identifies the command to be performed.

- **Flags** - There are eight bits that define different options. They are written on the form VMPrerrrr.

- **M(andatory)** - Indicates whether support is required for the AVP.

- **V(endor Specific)** - Indicates whether or not the optional Vendor ID field is available.

- **P** - Indicates the need of End-to-End encryption.

- **AVP Length** - Defines the number of octets in this AVP including the AVP code, AVP Length, AVP flags, Vendor-ID (if present) and the AVP data.

- **Vendor ID** - The vendor ID field is available if the P flag is set, the field contains the IANA assigned “SMI Network Management Private Enterprise Codes”.

- **Data** - This field contains information specific to the addressed attribute.
Chapter 3

The access network

In this chapter different parts of the access network will be discussed and thoroughly analyzed.

3.1 Introduction

As seen in the figure 2.6 the access network consist of different areas such as private, public, visited realm and home realm.

3.1.1 Access network

The access network could be wireless but this is not a prerequisite. An example of this is authentication within a wired company network with only stationary computers. In this scenario authentication could be used to reduce the risk of foreign computers attaching to the wired network, ie. personnel using the company network for private use or espionage.

3.1.2 Access router

The AR is positioned as a gatekeeper between the access network and the core network. But that is not its only task; the AR has to tell the attached nodes about its presence, otherwise the nodes will not be able to properly connect to the access network. This is done through sending advertisements to the attached nodes. The advertisements tells the node wether it has moved to a new network or not. The mechanism is based on comparison of the node’s current address prefix and the prefix advertised through the advertisements
sent out by the AR. Because of its role as a gatekeeper the AR is divided into two areas, controlled and uncontrolled. The Access Router is also called Network Access Server.

**Controlled**

The controlled part of the AR is toward the core network, it is within this part the AR performs access control. The controlled part should not let any traffic in nor out of the access network as long as it is not originated or meant for an authenticated node. Traffic that is used for authentication should always pass through and be addressed to the uncontrolled part of the AR. The ingress and egress filtering could be performed by netfilter [Tea02] or similar filtering software.

**Uncontrolled**

This is the part of the AR that is visible to the MN situated within the access network. Every node within the access network is allowed to send data to the uncontrolled part of the AR, but the AR will not handle the data unless the MN is authenticated and authorized toward its home AAA server.

### 3.1.3 Mobile nodes

The MN access the network through the AR, the procedure is triggered by the advertisements sent out by the AR, the information transferred to the AR is called the credentials and is used by the AR to authenticate the the MN towards the MN’s home AAA server.

### 3.1.4 AAA server

Is the entity in the network that handles Authentication, Authorization, Accounting. The AAA equipment within a network can have many different tasks, see figure 2.6.

**AAA server**

An AAA server that is located in the nodes home network. It only keeps Authentication and Authorization records of the nodes belonging to that
network. Accounting records of all visiting nodes can be kept due to billing reasons.

**AAAV server**

An AAA server in the visited network, the server contains authentication keys toward other ISP networks that it shares trust with. It can also work as a peer toward other AAA servers. This way the information of where to send requests from a certain node is kept centrally. This minimizes the amount of administration needed.

**AAA peer**

The peer servers are located at the border of a network. Peers are most commonly used if the ISP has many AAA entities within the network. The AAA peer distributes the AAA requests to the most suitable entity. The use of AAA peers lowers the amount of administrational information to be deployed by neighbor networks. The neighbor networks only need to care of the exact address to the peer instead of all the separate AAA entities. It also lowers the need of internal administration since the information about your neighbors becomes more static when the neighbors are using AAA peers as in figure 2.6

### 3.2 Requirements

This section will discuss the main requirements of the access network to be designed. It will also discuss previous known issues concerning construction of access networks.

#### 3.2.1 Reduced signaling

A very important requirement is to reduce the amount of signaling. Signaling affects the handover time severely since the MN usually can not do anything in parallel with the signaling. Signaling is most intensive when the MN is attaching to a network for the first time. Then the MN has to perform DAD, send user credentials to the AR and send BUs to the CNs and the HA. Eventually the MN also send information to its previous AR about its new point of attachment. As seen there are a lot of things going on while
Figure 3.1: Mobile node receives challenge and issues authentication request.

attaching to a new network and it is this time we aim to reduce. If the signaling time could be reduced the handover time will decrease as well.

3.2.2 Security

There are many known issues concerning access network and security. One well known issue is problems with authentication and authorization [AS01] within wireless networks that make use of Wired Equivalent Privacy protocol (WEP) [Bar01]. Basically the problems depend on all nodes using either a network identifier as a shared secret or an access control list based on the MAC addresses of allowed nodes. Future solution must include key exchange mechanisms that are safe and that changes keys during the access period, such as IKE [HC98] in combination with AH [KA98]. There already exists at least one solution based upon previous mentioned mechanism, it is called Extensible Authentication Protocol (EAP) [HZ02] and is an application for DIAMETER.

3.2.3 Modification

Modifications done to the different network entities such as MN, AR, DIAMETER server should be easy to follow and well documented. Otherwise it will be hard for others to look into the modifications made.
3.3 Mobile IPv6 integrated with Diameter

When a MN appears in the access network it receives a router advertisement, see figure 3.1, from the AR. The router advertisement contains the IPv6 address to the AR and a valid prefix used on the access network to form addresses using automatic address configuration. The prefix is also used by the MN to detect movement. When movement is detected, automatic address configuration and the AAA extension are triggered. The MN needs a valid global address before contacting the AR, this involves using DAD to ensure no one else on the access network is using the same address as the MN. When the MN has formed a valid global address it transfers its user credentials and replay protectors to the AR. The main reason for transferring the user credentials to the AR instead of directly to the Diameter server is because it has no network access until it is authenticated towards the AR. The AR itself does not contain any records about the MN, therefore the AR contacts the AAA server located in the same administrative domain as the AR. If the AAA server does not recognize the Network Access Identifier belonging to the MN it will forward the Authentication request to the AAA home server of the MN. Depending on what mode the MN is operating in, different scenarios take place. Commonly though, is the reply sent from the AAA home server telling the AR whether it should grant access to the MN or not.

3.3.1 Basic mode

When operating in basic mode (see figure 3.2). The AAA server only replies to the AR with an accept or deny. The reply also includes keys to use when communicating with the MN in future exchange of packets. In basic mode the MN sends the BU’s to CN’s and HA’s after access to the network has been granted by the AR. The basic mode contains one major drawback, the handover is delayed by the authentication process. This means that instead of one roundtrip the process takes two roundtrips.

3.3.2 Extended mode

Extended mode (see figure 3.3) offers a solution to the issue of double roundtrips. The solution is called Embedded data option (see appendix C.2.9). Using the Embedded data option the MN can transfer a valid BU within the data sent to the AR. When transferring the BU to the AR, the BU will be forwarded to the AAA server within the authentication request. The AAA server sends
the BU to the HA. The HA replies to the AAA server with a BA message which is included in the authentication reply sent back to the AR and later forwarded to the MN. Extended mode lowers the authentication delay from two roundtrips to one. The MN could include multiple packets in the embedded data option so that the AAA home server could send BU’s to all CN’s and all HA’s.

### 3.3.3 Security Associations

When data is transferred between the nodes within a network it has to be safe, as in privacy and authenticity. This is especially important in the case of transferring user credentials, passwords, cryptographic keys etc. When combining MIPv6 with Diameter one of the results is a backbone of Security Association (SA) servers. When the MN sends its authentication request to the AR. Neither the MN or the AR has a SA with each other. Therefore the
AR requests a session key to use from the MN’s AAA home server. The AR can be authenticated toward the MN’s AAA home server by the AAA server positioned within the same administrative domain as the AR. This is because of the fact that the AR’s AAA server and the MN’s AAA home server has an administrative relationship between each other, in other words they already share a SA. As the MN and the AAA home server share multiple keys, it is an easy task for the AR to receive a valid shared key to use when talking to the MN. As setup of SA’s are done by the Diameter CMS application. The key to use when sending data between the MN and the AR is transferred from the AR to the MN with the help of the Generalized key reply option (see appendix C.2.5). As the MN’s AAA home server and the MN maintains an index of shared keys the Key index has to be included in the option, the index is called Security Parameter Index (SPI). Figure 3.4 shows what SA’s are needed in an access network with MIPv6 nodes and Diameter aware equipment.

### 3.4 Security

Since the world is not totally honest there is a need of cryptographic protection of the data transmitted, as all the data exchanged is not secret we do not always need privacy, oftener we need authenticity. This can be obtained in many ways.
3.4.1 Authenticate signaling data

The data exchange in the AAA process has to be authenticated, the goal of authentication is obtained by a shared secret between the AAA home server and the MN. The authentication is performed by adding a credential message option to the data exchange. The data contained within the credentials section is a control messages computed using the MNs Network Access Identifier (NAI), the current timestamp and a client challenge as input. The output is simply a cryptographic checksum that is copied in to the credentials option. The authenticity is controlled by a shared secret key. The MN uses it to compute the checksum and the AAA home server is performing the computation once again to ensure its authenticity. The technique is called Keyed-Hashing for Message Authentication (HMAC) and Nested Keyed-Hashing for Message Authentication (NMAC) [BCK96]. The robustness of HMAC and NMAC is totally dependent on the hash function used [BCK96]. The robustness of Message Digest 5 (MD5) as underlying hash function is good when following certain prerequisites [Dob96].

3.4.2 Authenticate user traffic

When the authentication part is over and the MN is granted access to the access network the MN can start communicating with other nodes on the core network. One problem still exists, the AR has to authenticate the origin of the traffic so that it could block non authorized traffic. One way of doing this is to use AH [KA98]. When using AH in combination with the source address the issue of authentication of packets is solved. If we only relied on the source address of a packet it would be an easy task of EvA to assign the address of an already authenticated node to its own interface, and in such way gain access to the core network without authentication toward the AR.
Chapter 4

Implementation

The implementation performed during the thesis covers only a subset of the described protocol. This chapter aims to describe the implementation and ease further development of the current implementation. The chapter also shows examples of test scenarios used during the implementation.

4.1 Background

The reasons why to implement a subset of the protocol are many.

- To discover issues not obvious during planning phase.
- Create a test environment suitable for full scale test scenarios.
- Discover differences in user space and kernel space programming.

As a result of not implementing the entire protocol a stub is available. The stub can be used as a base and inspiration to future projects.

4.2 Design

The design is straight forward in the sense of being an extension to already existing software. What makes it tricky is that no software has the same design approach. Since the software exists in many different environments and under different requirements it is a challenge to understand the source code so that suitable changes can be made.
4.2.1 Client

The client is located at a system running Redhat with a Linux kernel. The kernel is patched to support MIPv6. The MIPv6 patch (see Appendix B.2) contains almost 15000 lines of code. Full understanding is not required but the most of its functionality should be understood before starting to add code to the MIPv6 module. The MIPv6 module is fitted together with the IPv6 stack using a data structure called mipglue, figure 4.1 shows the architectural principle of the MIPv6 module.

Modifications

The AAA submodule of the MIPv6 module introduces some changes to the IPv6 stack and to the MIPv6 module files. In IPv6 the files affected are:

- `addrconf.c` - The file handles DAD, since the process of authentication is triggered by the completion of DAD the trigger function is located within this file.

- `icmp.c` - The IPv6 module ICMPv6 handler, additions are made to reduce error messages in syslog since the AAA extension has its own ICMPv6 handler.

In Mobile IPv6 the files affected are:

- `mipglue.c` - The file is an interface for the kernel toward the MIPv6 module. The file contains some registered files so the kernel can communicate with the AAA extension.

- `mipv6.c` - The main file of the MIPv6 module, the AAA extension is loaded from within this file when started. The decision whether to load the module or not is done when compiling the kernel.

---

![Figure 4.1: MIPL module architecture.](image)

---
The file handles the MIPv6 modules movement detection. When movement is detected the AAA extension is told.

**Internal structure**

The internal structure of the AAA extension is quite straightforward. The extension could be divided into different categories.

- **Timer**
- **Callback**
- **Lock and data structure**
- **ICMPv6 handler**

**Timer** The extension contains a timer to keep track of the lifetime of the valid authentication. When the timer expires a function is executed, the function takes care of reauthentication towards the AR.

**Callback** A callback function is used to tell the extension when the MN has finished DAD. The completion of DAD is a sign of movement and should trigger the reauthorization process. Even if the MN performs DAD without moving it is a sign of changed care-of-address and is a reason to perform reauthorization since the AR performs filtering of packets based on the tuple, source address, associated key.

**Lock and data structure** The lock keeps track of the shared data structure, the data structure is shared in the sense of that callback routines and timers have to share the data, together with the ICMPv6 handler function.

**ICMPv6 handler** The ICMPv6 handler takes care of packet processing within the extension. It supports the packets defined in appendix C.2.

**Limitations**

The client only supports basic mode without key exchange. It is out of the scope of this thesis to find a suitable cryptographic solution to the problem of key exchange. Further it also out of the scope to modify all entities in
order to extended mode with BU’s to be sent as embedded data within the authentication request message.

4.2.2 Access Router

The AR software is designed to act as a gatekeeper towards the access network. It is supposed to control and authorize MIPv6 nodes backed by a network of Diameter servers. To fulfill this task three major modifications have to be introduced to the router advertisement daemon.

- AAA ICMPv6 support  
- NetFilter control  
- Diameter connectivity

AAA ICMPv6 support

To start with, the router advertisement daemon has to support the challenge option (see Appendix C.1) in order to include it within its router advertisements sent out on the access network. The AR also need to keep track of the last five valid local challenges sent out since the challenge is a replay protector. If an authentication request is received with erroneous local challenge, it should be dropped. An error message should be sent to the requesting node containing a valid local challenge and the code of NEW_CHALLENGE (see Appendix C.2).

NetFilter control

The AR should also contain a module through which it can control a network filter. The filter used is NetFilter [Tea02] as support is built into the kernel from the start. The NetFilter module needs data specifying the valid period of a certain address. Using this technique it is easy to flush the rules one time per hour or so or remove the rule when its presence is checked upon arrival of a packet.

Diameter connectivity

Diameter connectivity is achieved by using Sun Microsystems Diameter evaluation kit called SunWaal [mic01]. The functionality of the library
is based upon registration of callback routines. It is easy to use and example code is available for download. Important to mention though are that SunWaal adds some drawbacks to the AR software.

- SunWaal is closed software, no source code from the internals are available for download.
- The closed software makes it hard to debug reasons of failure.

There is at least one possible replacer of SunWaal on the opensource market today, it is called OpenDiameter [TAR02] and is a project driven by Toshiba America Research.

Limitations

The AR software developed supports basic operation without encryption. The development of suitable encryption techniques to use is out of the scope of this thesis. Today there are several techniques available, one of them is AH built into IPsec.

4.2.3 AAA server

Since no implementation of a Diameter application exist for MIPv6 today, the AAA server application had to be designed from start. The SunWaal library was used once again, this time as the base AAA server. The introduced Diameter packets used in the MIPv6 application are defined in Extensible Markup Language (XML) [Veil02] files residing both on the server and on the AR. Callbacks for the different packets are defined as well.

4.3 Test network

The test environment was created to validate the implementations as far as possible and to be used during debugging of the implementations. As the implementation covers both user space and kernel space plus extensive network debugging special cases applies that makes it harder to debug.

- Kernel space offers limited debugging options since kernel space is protected area and an error within the kernel often ends up in a total system crash.
Since errors end up in system crash, debugging kernel space software can be very time consuming.

Sniffing packets from real networks are difficult and the equipment is expensive, especially for wireless debugging.

The solution to these points is called User Mode Linux (UML) [mLkpp02].

4.3.1 User Mode Linux

UML is a kernel patch that allows us to run a linux system in user space of an already running linux system. The differences are very small and its very hard to tell the difference once inside a UML system. Another reason of running UML is that it allows us to connect multiple UML systems together into a network. This feature makes it possible to simulate an entire network consisting of several UML systems on one Linux system. Analysis of the network traffic is also possible since UML offer connectivity to the outer world through devices called taps. The setup of the test network and where to find suitable software is further described in Appendix B.

4.3.2 Network topologies

Mainly three topologies were used when debugging and verifying the functionality of the code.
Figure 4.3: Diameter aware network

**Fundamental network**

Figure 4.2 describes the most basic network, it was used to start up configuration without any diameter involved. It could also be used to develop the modified MIPv6 kernel module and the DIAMETER extension to the router advertisement daemon.

**Diameter aware network**

Figure 4.3 describes a network where the DIAMETER server has been added and acts as the main Authentication, Authorization, Accounting entity within the network. All access to the access network has to be granted by the DIAMETER server. This network was used to develop and test the DIAMETER related extensions within the router advertisement daemon and the DIAMETER server.

**Handover capable network**

Figure 4.4 describes the most complex network, the network contains an extra access network so that handovers between the two access networks can be performed. In this scenario the robustness of the test implementation is tested.
Figure 4.4: Handover capable network
Chapter 5

Summary

This chapter aims to analyze the work presented within the thesis, the author presents conclusions and suggests future work.

5.1 Contribution

The thesis gives the reader insight to the technology available today in the form of MIPv6 and DIA METER. The author also presents a proposition of an access network design solving the prerequisites stated in the Research objectives presented in chapter 1. Further the author presents a basic test implementation based upon the Linux Kernel, HUT’s Mobile IP for Linux (MIPL) and Fernbergh’s radvd.

5.2 Conclusions

Today the base technology needed to construct functional access networks exists. But in order to seamlessly combine MIPv6 and DIA METER further research has to take place, more standards has to be stated and more test implementations have to be performed. The different working groups at the IETF need to have closer cooperation and integration since the drafts ”Diameter Mobile IPv6 Application” [FL02] and ”AAA for IPV6 Network Access” [PE02] does not fit seamlessly together. One way of achieving seamlessness is the use of test implementations. Unfortunately does not the implementation included within this thesis provide a fully functional implementation. The implementation is solely created in the purpose of investigating issues arising within MIPL when combining MIPv6 and DIA METER in basic mode.
5.2.1 Combining Mobile IPv6 and Diameter

When implementing the AAA module for MIPv6 it became obvious that the current implementation of the IPv6 stack and MIPv6 extension module are still separated from each other and maintained by different teams, the code contains different coding styles and comments which my worsen readability. The AAA module requires information from both parts and deep understanding of the code is required. Hopefully the next Kernel 2.5 will become easier to modify since it is supposed to include MIPL from the start, the design should therefore be a bit different and hopefully will the MIPL extension be fully integrated into the IPv6 stack instead of a module as today. The main problem found when combining MIPv6 and Diameter was the problem of fetching information from the IPv6 stack and combining it with the MIPv6 extension. An example is that BUs should be included into the authentication request, this makes it necessary to modify the MIPv6 extension severely and may cause severe problems.

5.2.2 About key distribution

The thesis proposes the diameter backbone to be used as an authentication key distributor. The interesting question arising is how large the delay introduced by the key distribution is. The simples way of finding that out would be to continue the implementation given in the thesis, and to perform further testing.

5.2.3 The development process

The development process used during the thesis was iterative, all development have been performed through small steps where every functionality have been tested in the test network. At first a user space client was created, but since the userspace client lacks in the ability of sending embedded binding updates. A kernel space based client was created which better reflect issues that could arise when adding authentication to MIPL.

5.2.4 Using UML

In order to properly debug and monitor packet flows inside the kernel the author used User Mode Linux [mLkpp02]. UML is a very powerful development environment since it makes it easy to debug kernel threads. It is also
possible to set up automatic testing since different scripting languages such as Bash and Perl are supported. The use of UML’s Copy On Write (COW) feature makes it easy to setup multiple hosts sharing the same filesystem. One drawback is the long learning curve, since UML acts as a Linux system, even if you are quite used to working in an Unix environment it takes a while to learn how to get the most out of UML. The author had 3 weeks of debugging before discovering an error within the MIPL code based on that UML does not null the kernel memory by default as the common Linux kernel does.

5.2.5 Extending Ethereal

Ethereal is a packet sniffer that was used to monitor packet flows within the test networks, It had to be extended in order to support AAAv6 message types. The use of Ethereal is obvious since it makes it possible to faster find the source of an error, to detect packet building or parsing faults. Ethereal can also be used to measure arrival and departure times.

5.2.6 Opensource

Further should the aim of producing opensource software compatible with the standard be maintained, since this makes it possible for researchers to further develop already existing protocols. Also it is important to keep reference implementations open in order to make it easier for companies to test their implementation toward the given standard.

5.3 Future work

As stated in the Conclusions there is a need of further development in order to find suitable solutions and to create functional standards that are easy to understand and implement. As the implementation created within this thesis is far from complete suitable future projects could be:

- **NetFilter module** A suitable netFilter module performing access control combining AH headers and IPv6 addresses. The thought is to create time limited rules removing them selves when expiring, this way the access router does not need to keep track of what rules to remove and when.

- **Diameter module** An implementation of the DIAMETER application for Mobile IPv6. Supporting basic and extended mode.
• **Extended mode** Extend the current implementation in order to support Extended mode. It would be suitable to perform the implementation on the new 2.5 kernel as the MIPL release will be a part of the kernel by default.

• **Key distribution** Investigate how the AAA infrastructure could be used to distribute keys used for authentication of MN toward AR, CN and HA.

• **Diameter entities** Continue development of the DIAMETER infrastructure. As the DIAMETER server never were completed due to time limitations it could be a suitable entity to start with.
Bibliography


Appendix A

Acronyms

AAA  Authentication, Authorization, Accounting
ABNF Augmented Backus-Naur Form
AH Authentication Header
API Application Programming Interface
AR Access Router
AVP Attribute Value Pair
BA Binding Acknowledgement
BU Binding update
CMS Cryptographic Message Syntax
CN Correspondent Node
CoA Care-of-address
COW Copy On Write
DAD Duplicate address detection
DHCP Dynamic Host Configuration Protocol
DHCPv6 DHCP version 6
EAP Extensible Authentication Protocol
**ESP**  Encapsulated Security Payload

**EvA**  Evil Attacker

**FA**  Foreign Agent

**GDB**  GNU Debugger

**GNU**  GNU Not Unix

**HA**  Home Agent

**HMAC**  Keyed-Hashing for Message Authentication

**HMIPv6**  Hierarchical MIPv6 mobility management

**IANA**  Internet Assigned Numbers Authority

**ICMP**  Internet Control Message Protocol

**ICMPv4**  ICMP version 4

**ICMPv6**  ICMP version 6

**IETF**  Internet Engineering Task Force

**IGMP**  Internet Group Management Protocol

**IKE**  Internet Key Exchange

**IP**  Internet Protocol

**IPsec**  IP Security Protocol

**IPv4**  IP version 4

**IPv6**  IP version 6

**ISP**  Internet Service Provider

**LGPL**  Lesser GNU Public License

**MAC**  Media Access Control Address

**MAP**  Mobility Anchor Point

**MD5**  Message Digest 5

**MIP**  Mobile IP
MIPL Mobile IP for Linux
MIPv4 MIP version 4
MIPv6 MIP version 6
MN Mobile Node
MTU Minimal Transfer Unit
NAI Network Access Identifier
NAS Network Access Server
NASREQ Network Access Server Requirements
NMAC Nested Keyed-Hashing for Message Authentication
OUI Organizationally Unique Identifier
RADVD Router Advertisement Daemon
RFC Request For Comment
ROAMOPS Roaming Operations
SA Security Association
SPI Security Parameter Index
TCP Transport Control Protocol
TLS Transport Layer Security
UDP Uniform Datagram Protocol
UML User Mode Linux
WEP Wired Equivalent Privacy protocol
XML Extensible Markup Language
Appendix B

MIPv6 Testbed configuration

This appendix will introduce the reader to the environment used to implement the test implementation.

B.1 Introduction

All software used when developing and testing the implementation is available for download via Internet. If there are any questions about modifications or configurations please feel free to contact the author using the email address anton.larsson@home.se.

B.2 Mobile IPv6 for Linux

In order to create Mobile IPv6 nodes within the test network MIPL-0.9.3-2.4.18 was used. It is available through MIPL project homepage [fLp02]. The Mobile IPv6 kernel module was modified in order to support AAA within the access network, the modification is described in chapter 4. A diff file can be requested from the author. Worth to note is that the MIPL may contain serious bugs, those bugs are only visible and affects the system only when executed in UML. Those becomes visible since user space memory in Linux is not set to NULL by default as in the kernel space. And since UML is run under user space it is affected.
B.3 Kernel 2.4.18

Linux kernel 2.4.18 was used within the emulated Linux machines in the network. At the time of the thesis project, kernel 2.4.18 was the latest stable kernel available. MIPL-0.9.3-2.4.18 was also made especially for the 2.4.18 kernel. To support Mobile IPv6 the kernel had to be patched with MIPL-0.9.3-2.4.18.

B.4 User Mode Linux

UML was used to emulate the access network. Another reason why using UML is that the emulated kernel runs in user space which makes it easier to debug. In kernel 2.5 and further UML is supported by default but as kernel 2.4.18 was used, it had to be patched to support UML. A large variety of patches are available though the UML homepage [mLkpp02]. The UML homepage also contains a lot of tutorials covering things as patching kernels with UML, compiling the kernel, setting up networking and how to debug the kernel in user space using UML and GNU Debugger (GDB). In order to run UML you will need a file system, there are plenty of filesystems with different Linux distributions available for download at the UML homepage [mLkpp02]. If you want to create your own file system there are many tools available, to construct the filesystem to be used in the emulated network the author chose GBootRoot, see next subsection. Multiple instances of UML can use the same file system in parallel, the technique is called COW, Copy On Write and is more closely explained at the UML homepage [mLkpp02].

B.5 GBootRoot

In order to have a totally equal test and development environment GBootRoot was used to create a filesystem containing a Redhat 7.3 base installation. The redhat base installation was later extended with the necessary files. The root file system became approximately 350 Mb in size. A swap file system was also created, approximately 50 Mb in size.
B.6 Ethereal

The packet sniffer used was Ethereal [Com02], it was modified in order to support the AAAv6 protocol used to transfer data between the access router and the Mobile IPv6 nodes. A patch can be requested from the author.

B.7 Redhat 7.3

The system used to run the UML emulated network was a Redhat 7.3 system on a computer with the following hardware specifications.

PIII 800 Mhz
512 Mb ram
10 Gb hdd

As the network can contain up to 6 nodes, a faster processor could have been used, worth to notice is that the speed of the emulation is mostly dependent upon the amount of ram memory available, 256 meg of ram is the absolute minimum to run a network consisting of only two nodes.

B.8 Diameter libraries

Since the development of Diameter started in the AAA workgroup at IETF. The interest in Diameter from the industry has been large. The first Diameter library was released by Sun Microsystem [mic01], but only in binary form and restricted to non profit use. Not until late 2002 came the first opensource based implementation from Toshiba research institute [TAR02]. Both libraries can be used to implement a Diameter peer and server.

B.9 Router advertisement daemon

The daemon is known as RADVD [Fen02], the version used was modified to act as a access router for the access network. It handles both the netfilter towards the core network and the distribution of Diameter related information to the Mobile IPv6 nodes in the access network.
B.10 List of software

RedHat 7.3
Linux kernel 2.4.18
Ethereal 0.9.7 + patch
RADVD 0.7.2 + patch
MIPL 0.9.3-2.4.18 + patch
openDiameter
GBootRoot
Appendix C

Protocol specification

This chapter aims to give the reader a view of the different message options mentioned in this report. Many of the message options are similar to those used within AAAv6 [PE02] and the Diameter Mobile IPv6 Application [FL02]. These message options are modified to suit the purpose of this thesis. Also, the codes and subtypes defined in this chapter may collide with other applications since they are not assigned by IANA.

C.1 AAA challenge option

The AAA challenge option is always applied to the router advertisement. By the presence of the AAA challenge option the MN will now that the AR supports authentication using the diameter application for Mobile IPv6.

<table>
<thead>
<tr>
<th>0</th>
<th>7</th>
<th>8</th>
<th>15</th>
<th>16</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type= TBD</td>
<td>Length</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Challenge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Type 55 (AAA challenge option).

Length Should be set to 8.

Challenge This field contains a challenge.
C.2 AAA protocol messages

ICMPv6 is used to transfer data between the MN and the AR. To enable separation of data being transferred new message types had to be defined. The defined messages have the following general structure.

<table>
<thead>
<tr>
<th>Type=TBD</th>
<th>Code</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Message body ...

Type  The following types are defined:
- AAA Request: 230
- AAA Home Challenge Request: 231
- AAA Reply: 232
- AAA Teardown: 233

Code  The code field depends on the message type. Currently the following Code fields are defined:

For AAA Reply
- SUCCESS: 0
- NEW CHALLENGE: 1
- ADDRESS IN USE: 20
- INVALID CREDENTIAL: 50
- INVALID TIMESTAMP: 51
- AAAV6 FAILURE: 52

For AAA Teardown
- SUCCESS: 0

Message body  The message body may consist of one or more options.

No Code values are defined for the remaining AAA message types. The Code field MUST be set to zero.
C.2.1 AAA Protocol Message Options

The general structure of an AAA Protocol Message Option is defined as follows.

```
0 7 8 15 16 31
<table>
<thead>
<tr>
<th>Type=TBD</th>
<th>Subtype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option data .....</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Type** The Type field identifies the option. Currently, the following types are supported. The most significant bit of the Type indicates if the option is unskippable (0) or skippable (1).

**Subtype** Each option type may be further subdivided. The Subtype field identifies option at the next level of granularity.

**Length** The Length field indicates the size of the Option data in octets.

**Option data** The format of option data is depends on the type and subtype, and is defined below.

C.2.2 Client Identifier option

The Client Identifier option is used by AAA server in the visited network to identify the realm to route the AAA request, and by AAA home server in verifying the AAACredential submitted.

```
0 7 8 15 16 31
<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Identifier</td>
<td></td>
</tr>
</tbody>
</table>
```

**Type** 1 (unskippable)

**Subtype** Currently two subtypes are defined: NAI (0) and IPv6 address (1)

**Length** For subtype 1, the Length should be 16.

**identifier** For subtype 0, this field contains a NAI formatted according to RFC2486 [AB99]. For subtype 1, this field contains an IPv6 address.
C.2.3 Security Data option

The security data option is used to transfer credentials used to authenticate either the MN or the AAA home server. The type of authentication is identified by the subtype

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SPI</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Security data ...</td>
<td></td>
</tr>
</tbody>
</table>

**Type** 2 (unskippable).

**Subtype** Currently two subtypes are defined: AAA Credential (0) and AAA home server Authenticator (1).

**Length** Length of the option in octets, not including the first four octets.

**SPI** The security parameter index to be used in interpreting the Security Data.

**Security Data** The actual payload.

C.2.4 Challenge Option

The Challenge option is used to transport replay protectors between different pairs having a communication.

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Challenge</td>
<td></td>
</tr>
</tbody>
</table>

**Type** 3 (unskippable)

**Subtype** Currently two subtypes are defined:

- Local Challenge - Challenge issued by the attendant to the client (0)
- Client Challenge - Challenge issued by the client to AAA home server (2)

**Length** Length of the challenge in octets, usually 4

**Challenge** The actual challenge data.

### C.2.5 Generalized Key Reply Option

The Generalized Key Reply Option is used by the AAA home server to distribute keys for the MN to use when communicating with other nodes on the network, ie. AR.

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAH SPI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KEY SPI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Peer IPv6 Address

#### Encoded Key Data ...

**Type** 4 (unskippable)

**Subtype** Currently subtypes are defined for three entity pairs:

- Client-Attendant authentication key: Key to be used between the current attendant and the client for IPSec authentication (1)
- Client-Attendant encryption key: Key to be used between the current attendant and the client for IPsec encryption (2)
- MN-HA authentication key: Key to be used between the home agent and the client for IPsec authentication (4)

If the most significant bit of the Subtype value is 1, the “Peer IPv6 Address” field is present. Otherwise, it is absent.
Length Length of the option in octets except the first four octets.

AAAH SPI This field indicates the security association between the client and AAA home server, which should be used by the client to interpret the Encoded Key Data field.

Key SPI This field indicates the SPI value for the new security association into which the key should be inserted.

Peer IPv6 Address When present, this field indicates the IPv6 address of the peer. This is useful when the client does not already know the address to be used. This field is present in subtypes 128 and above.

Encoded Key Data This field contains the key, along with any other information required by the client to create the security association. The contents of the field MUST be encrypted by AAA home server as specified by AAAH SPI.

C.2.6 Timestamp Option

This option is used between the MN and the AAA home server as a replay protector. The reason for not using challenges between the MN and the AAA server is to lower the handover time, as challenges oftener require two roundtrips to finish the handover and authentication procedure.

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Length</th>
<th>Timestamp</th>
</tr>
</thead>
</table>

Type 5 (unskippable).

Subtype Currently no subtypes are defined. Should be zero.

Length Length of the Timestamp in octets.

Timestamp Timestamp value in some format mutually intelligible to the MN and AAA home server
C.2.7 IPv6 Address Option

This option is used by the client to tell the AAA home server about its current Care-of-address,

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IPv6 Address</td>
</tr>
</tbody>
</table>

**Type** 6 (unskippable).

**Subtype** Currently no subtypes are defined. Should be zero.

**Length** 16

**IPv6 Address** A valid IPv6 address.

C.2.8 Lifetime Option

This option is used to indicate the validity period of a successful AAA authentication.
In other words the MN has to reauthenticate before the lifetime expires.

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lifetime</td>
</tr>
</tbody>
</table>

**Type** 7 (unskippable).

**Subtype** Currently no subtypes are defined. Should be zero.

**Length** 4.

**Lifetime** Lifetime in seconds.
C.2.9 Embedded Data Option

This message option is used by the MN to transfer BU to the AAA home server for further transportation to the HA’s and CN’s. The AAA home server also uses this option when forwarding the BA’s from the CN’s and the HA’s.

<table>
<thead>
<tr>
<th>Type</th>
<th>who</th>
<th>Subtype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Embedded Data ...

**Type** 8 (unskippable).

**WHO** This field indicates who should process the embedded data. It is interpreted as follows (values in binary).

- 00 - Recipient of the AAA Protocol message (i.e., either the client or the attendant)
- 01 - AAAL
- 10 - AAAH
- 11 - reserved

**Subtype** Currently two subtypes are defined

- (0) – MN-HA binding update
- (1) – HA-MN binding acknowledgement
- (2) – EAP Request [2]
- (3) – EAP Response [2]

**Data** The actual embedded data itself. For subtype 0, this MUST be an IPv6 packet addressed to the HA, and containing a binding update. For subtype 1, this MUST be an IPv6 packet addressed to the CoA, and containing a binding acknowledgement from the HA.

Eg. to bundle the HA binding update with AAA processing, the client will first generate a binding update, and insert it into an embedded data option of the AAA Request message, with WH = 10 (binary) and Subtype = 0. Based on the value of WH, the attendant will extract the Embedded Data...
and forward it to AAA home server via the AAA server in the visited network. Based on the Subtype, AAA will forward the binding update to the home agent, and will receive a binding acknowledgement in reply. The attendant will forward the binding acknowledgement in an Embedded Data option to the AAA Reply message, with WH - 00 (binary) and Subtype = 1.

C.3 Diameter messages

The Mobile IPv6 application for diameter introduces four new command codes. The IANA defines the range 192 to 223 to be used in experimental purposes, these numbers are therefore used.

- AA-Registration-Request-Command (ARR) Code, TBD
- AA-Registration-Answer-Command (ARA) Code, TBD
- Home-agent-MIPv6-Request-Command (HOR) Code, TBD
- Home-agent-MIPv6-Answer-Command (HOA) Code, TBD

All message types are previously defined in the Diameter Mobile IPv6 Application [FL02]. But since the content of the message types are not defined, appendix C.4 defines AVPs to be used in the message types.

C.3.1 AA-Registration-Request-Command (ARR)

contains the following list of AVPs:

- Session-Id AVP
- User Name AVP
- Client challenge AVP
- Timestamp AVP
- User credential AVP
C.3.2 AA-Registration-Answer-Command (ARA)

- Session-Id AVP
- User Name AVP
- Client challenge AVP
- Timestamp AVP
- User credential AVP
- Authorization-Lifetime AVP

C.3.3 Home-agent-MIPv6-Request-Command (HOR)

- MIP-Binding-update AVP
- MIPv6-Home-Agent-Address AVP

C.3.4 Home-agent-MIPv6-Answer-Command (HOA)

- MIP-Binding-acknowledgement AVP
- MIPv6-Mobile-Node-Address AVP

C.4 Diameter AVP’s

The following attribute value pairs (AVPs) are used between the AR and the AAA server to convey information necessary to authenticate and authorize the Mobile IPv6 node.

C.4.1 Basic mode

The AVPs mentioned in this section are crucial in making the authentication work. The following AVPs are also used in Extended mode.
User Name AVP

The User-Name AVP (AVP Code 1) [RW00] is of type UTF8String, which contains the User-Name, in a format consistent with the NAI specification [AB99].

Client challenge AVP

The Client challenge AVP (AVP Code TBD) is of type numeric, which contains the challenge issued by the MN towards the AAA home server as a replay protector.

Timestamp AVP

The Timestamp (AVP Code TBD) is of type Time, and is used as a replay protector of messages exchanged between the MN and the AAA home server. Time is measured in seconds since January 1, 1970 00:00 UTC.

User credential AVP

The User credential AVP (AVP code TBD) is of type octetstring and is used to authenticate the user towards the AAA home server.

Session-Id AVP

The Session-Id AVP [CA02] (AVP Code 263) is of type UTF8String and is used to identify a specific session. All messages should only contain one Session-Id AVP and the same value should be used throughout the life of a session.

Authorization-Lifetime AVP

The Authorization-Lifetime AVP [CA02] (AVP Code 291) is of type Unsigned32 and contains the maximum number of seconds of service to be provided to the user before the user have to issue a new AA-Registration Request.
C.4.2 Extended mode

The AVPs mentioned in this section are only used when operating in Extended mode.

MIP-Binding-update AVP

The AVP [FL02] contains the binding update message sent by the MN to the HA, the message is of type octet string and has the AVP code 1. This AVP is only used in extended mode.

MIP-Binding-acknowledgement AVP

The AVP [FL02] contains the Mobile IPv6 binding update acknowledgement sent by the HA to the MN. The message is of type octetstring and has the AVP code 2. This AVP is only used in extended mode.

MIPv6-Mobile-Node-Address AVP

The AVP [FL02] contains the Mobile IPv6 nodes home address, it is of type IPv6Address and has the AVP code TBD.

MIPv6-Home-Agent-Address AVP

The AVP [FL02] contains the address of the Mobile IPv6 nodes HA. It is of type IPv6 address and has the AVP code TBD.

C.4.3 Key distribution

The CMS [CF02] DIAMETER application is supposed to handle the key exchange between the AR, MN and the AAA home server. The scheme is closer described in draft-ietf-aaa-diameter-cms-sec-04.txt.