Technical University Berlin

Telecommunication Networks Group

Perfomance Evaluation of AAA / Mobile IP Authentication

A. Hess, G. Schäfer
[hess,schaefer]@ee.tu-berlin.de

Berlin, 29/08/2001, Version 1.0

TKN Technical Report TKN-01-012

TKN Technical Reports Series

Editor: Prof. Dr.-Ing. Adam Wolisz

Abstract

This document[1] studies the performance of the current IETF approach to authenticating mobile nodes by means of an integrated *Authentication, Authorization and Accounting (AAA)* infrastructure. The report first describes the procedures and message formats of the Diameter Base Protocol and its application to Mobile IP as specified by the *Internet Engineering Taskforce (IETF)* in detail. After describing the simulation model for the protocol's operation, developed in the course of this study, we present and discuss results for a variety of parameter settings. The main findings of this study are: 1) the delay experienced by a mobile node in case of a full authentication dialogue involving entities of the mobile node's home network is largely determined by the end-to-end delay between the foreign and the home network, 2) the workload of AAA servers remains moderate in case of a load- and mobility model inspired by established values of GSM networks [14] as well as in case of a more progressive mobility model [16], and 3) the workload of AAA servers grows infinitly under both mobility models if cryptographic algorithms are used that require about 100 (30) times the processing capabilities of algorithms currently envisaged by the IETF (cryptographic hash functions and symmetric encryption). An important consequence of this finding is that the use of asymmetric cryptography would possibly lead to overload situations under the investigated conditions.

[1]This work has been supported by a grant from Siemens AG.

Contents

Chapter 1

Introduction

Upcoming next generation mobile communications networks aim not only to offer *Internet Protocol (IP)* connectivity to customers, but also to be designed themselves on the basis of the Internet Protocol suite. One important development in this direction is the deployment of the *Mobile IP* protocol [17, 18, 22] in the *Radio Access Network (RAN)* of next generation cellular networks. However, there still remain some open questions regarding the realization of the authentication of mobile devices during handover operations, which are currently subject to intensive research and development efforts inside and outside the *Internet Engineering Taskforce (IETF)*.

The principal approach of the IETF in this respect is to integrate authentication during Mobile IP registration with a general *Authentication, Authorization and Accounting (AAA)* infrastructure based on the so-called *Diameter* protocol [6].

1.1 Relation of this Report to other Work at TKN

This report has been written in the context of the project *Mobility in Multi-Domain, Multi-Technology, IP-based Networks* that is founded by Siemens AG, department Information and Communication Network (ICM N MC ST 3). The project is devided into three subprojects, one dealing with mobility mechanisms in All-IP mobile networks, the second one dealing with Quality of service in All-IP mobile networks, and the third subproject dealing with authentication in All-IP mobile networks.

The main focus of the authentication subproject is the problem of securely identifying entities for IP access in a mobile network with roaming capabilities. Strongly related to this problem is the task of managing the cryptographic keys that are needed for secure entity and data origin authentication. A special requirement to the authentication infrastructure arises out of the fact that mobile devices require frequent handovers from one access point to another. These handovers can either occur between access points of one single network provider ("intra-domain handover") or between access points of different network providers ("inter-domain handover"). An important issue in this context is the performance of the (re-)authentication during a handover operation.

While the first TKN report on authentication in All-IP based mobile networks [21] analyzed and compared authentication methods in wireless LANs (IEEE 802.11), GSM, UMTS Release '99, and the preliminary specifications of the IETF Mobile IP and AAA working groups with focus on the cryptographic protocols involved, the main objective of the present report is to get performance estimates for authentication-induced handover latency and the cryptographic processing requirements in intermediate nodes caused by authentication processing. This goal is accomplished by building and studying a discrete event simulation model of authentication processing during handover operations.

1.2 Overview of the Report

This report is organized as follows: the next chapter introduces the current concepts of the Internet Engineering Taskforce (IETF) regarding authentication for Mobile IP with the help of an AAA infrastructure, and explains the exchanged protocol data units in detail. Chapter 3 describes the simulation model and the two different mobility models that have been used in order to evaluate the model's performance under various assumptions on user mobility. Chapter 4 presents the results of the simulation study and chapter 5 summarizes the key findings and draws some conclusions.

Chapter 2

AAA Procedures and Message Formats for Mobile IP Authentication

The current standardization efforts of the IETF promote the use of the *Diameter* protocol [6, 9] for authenticating mobile nodes during Mobile IP registration. This protocol allows peers to exchange a variety of messages, whose basic format is specified in the *base protocol* that provides the following facilities:

- delivery of *attribute value pairs (AVP)*,
- capabilities negotiation,
- error notification, and
- extensibility, through addition of new commands and AVPs.

The Diameter base protocol provides the minimum requirements needed for an AAA transport protocol, as required in [2, 5, 13, 15]. The base protocol is not intended to be used by itself, and must be used with a Diameter application, such as Mobile IP [9, 17].

This chapter will give a brief description of the basic Diameter message formats and the attribute value pairs needed for AAA / Mobile IP authentication. The next section describes the base format as specified in [6]. Sections 2.2 and 2.3 discuss the messages including their attribute value pairs to be exchanged during a Mobile IP registration procedure according to [9]. The detailed formats of Mobile IP registration messages are explained in section 2.4, and section 2.5 describes the format of ESP-protected IP-packets exchanged between Diameter entities. In section 2.6, finally, the lengths of the Mobile IP and Diameter messages transmitted during a Mobile IP registration are summarized.

2.1 Diameter Base Message Formats

Figure 2.1 shows the common message format of all Diameter messages. Every message starts with a *version* field of length 8 bit with the current version being 1. This is followed by the *message*

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Ver      |                 Message Length                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R P r r r r r r|                  Command-Code                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Vendor-ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Hop-by-Hop Identifier                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      End-to-End Identifier                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AVPs ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 2.1: Common Format of Diameter Messages [6]

length field (its length is specified to be 2 octets even though the figure from [6] suggests 24 bit), that indicates the total length of the Diameter message including all header fields. The next field of length 8 bit is supposed to hold *command flags*. Up to now two flags have been defined: the *request flag (R)* which indicates a request messsage as opposed to an answer message, and the *proxiable flag (P)* which indicates that the message may be relayed ("proxied") to another diameter server for procesing as opposed to requiring that the message needs to be processed locally. The remaining six bits of the command flag are reserved for future use and must be set to zero. The next field, *command-code* specifies the type and purpose of the message. Its 24 bit address space is managed by the *Internet Assigned Numbers Authority (IANA)*. The *vendor-id* field of length 32 bit allows for vendor specific Diameter commands provided that the vendor sets this field to his IANA assigned *SMI Network Management Private Enterprise Code*. For all commands defined in IETF standards this field is set to zero. The *hop-by-hop-identifier* field is of length 32 bit and helps to match requests and replies. The sender of a request must ensure that this identifier is locally unique. The *end-to-end identifier*, also of length 32 bit, serves a similar purpose in an end-to-end relation, that is while the hop-by-hop identifier may be changed by relaying nodes (e.g. Diameter brokers) the end-to-end identifier uniquely identifies a request for the processing entities. The information relevant to a Diameter command is encapsulated in *attribute value pairs (AVP)*. For each command the required and optional AVPs are listed in the command definition.

Diameter commands are usually not specified by listing the exact message format as in figure 2.1, but by defining their content in *Augmented Backus-Naur Form (ABNF)* [11]. Figure 2.2 shows the generic ABNF format for specifying Diameter messages (for a complete description please refer to [6]).

A Diameter command definition always begins with the command name followed by "::=" and the diameter message specification. A diameter message is specified by defining its header, followed by a specification of zero or more "fixed" AVPs (which have to be present at some fixed position inside the message and are specified inside "< >"), zero or more required AVPs (specified inside "{ }"),

```
command-def      = command-name "::=" diameter-message

diameter-message = header  [ *fixed] [ *required] [ *optional]
                   [ *fixed]

header           = "<Diameter-Header:" command-id [r-bit] [p-bit] ">"

r-bit            = ", REQ"
                   ; If present, the ’R’ bit in the Command
                   ; Flags is set

p-bit            = ", PXY"
                   ; If present, the ’P’ bit in the Command
                   ; Flags is set

fixed            = [qual] "<" avp-spec ">"
                   ; Defines the fixed position of an AVP

required         = [qual] "{" avp-spec "}"
                   ; The AVP MUST be present

optional         = [qual] "[" avp-name "]"
                   ; The AVP may be present

qual             = [min] "*" [max]
                   ; See ABNF conventions, RFC 2234 section 6.6.
                   ; The absence of any qualifiers implies that
                   ; one and only one such AVP MUST be present.

min              = 1*DIGIT
                   ; The minimum number of times the element may
                   ; be present. The default value is zero.

max              = 1*DIGIT
                   ; The maximum number of times the element may
                   ; be present. The default value is infinity.

avp-spec         = diameter-name
                   ; The avp-spec has to be an AVP Name, defined
                   ; in the base or extended Diameter specifications.

avp-name         = avp-spec | "AVP"
                   ; The string "AVP" stands for *any* arbitrary
                   ; AVP Name, which does not conflict with the
                   ; required or fixed position AVPs defined in
                   ; the command code definition.
```

Figure 2.2: Specification of Diameter Commands in ABNF

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           AVP Code                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V M P r r r r r|                  AVP Length                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Vendor-ID (opt)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Data ...
+-+-+-+-+-+-+-+-+
```

Figure 2.3: Common Format of Attribute Value Pairs (AVP)

zero or more optional AVPs (specified inside "[ ]"), and again zero or more "fixed" AVPs. For each AVP the possible number of its occurence may be futher qualified (e.g. "0*3" preceding an AVP name specifies that zero to 3 AVPs of that type may be contained in the message). If this qualification is omitted exactly one AVP of that type must be present in the message with the exception that optional AVPs do not need to be present. The message header of a command is specified with the string "<Diameter Header:" followed by its *command-id* (24 bit command code, usually assigned by IANA), the specification of the Diameter command flags (Request and/or Proxiable) and a trailing ">".

Figure 2.3 illustrates the generic message format of Diameter AVPs. Every AVP starts with an identifying *AVP code* of length 32 bit, followed by a field of *AVP flags*. Up to now three AVP flags have been defined: the *vendor-specific* flag ("V") indicates that this is a vendor specific AVP and so the optional *vendor-id* field is present in the AVP header. The *mandatory* flag ("M") indicates that processing of this AVP is mandatory. If a Diameter node receives a message with an unknown AVP that has the mandatory flag set, it must reject the message. If a Diameter node receives a message with an unknown AVP that has this bit cleared, it may simply ignore the AVP. The *protection* flag ("P") specifies that an AVP may not be modified by intermediate Diameter nodes (e.g. relay nodes, brokers, etc.) as it is protected by another AVP that encapsulates a signature in *Cryptographic Message Syntax (CMS)*. The basic idea behind this is to provide end-to-end security for some AVPs as the basic Diameter security model just provides for hop-by-hop security (please refer to [7] for further details). All other bits of the AVP flags field are reserved for future use and have to be set to zero. After the AVP flags the header contains the *AVP length* field (24 bit) that indicates the length of this AVP including the AVP header. It is followed by the optional vendor-id field and the AVP specific *data* field.

The Diameter base protocol specifies the following base data types to be used in the data field of an AVP:

- *OctetString:* arbitrary data of variable length which must be aligned to a 32 bit boundary,
- *Integer32*, *Integer64*, *Unsigned32*, *Unsigned64:* a signed or unsigned value of the specified length in network byte order,

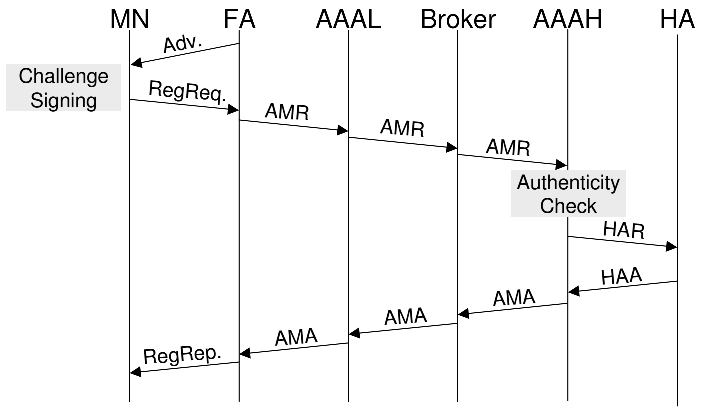

Figure 2.4: Message Flow in Integrated AAA / Mobile IP Authentication

- *Float32, Float64, Float128:* a floating point value of the specified length in network byte order,

- *Grouped:* the data field is specified as a series of AVPs which are contained in the "grouping" AVP with their headers, data and padding. Grouping of AVPs may be nested, i.e. a grouped AVP may contain grouped AVPs in its data field. If one of the contained AVPs is marked as a mandatory AVP, the mandatory flag of the grouping AVP must be set.

A Diameter application may further derive new data types from the above types, e.g. a commonly used derived data type is *IPAddress* which is derived from the OctetString data type and contains either an IPv4 or an IPv6 address with the most significant octet first. Other commonly used data types are *Time, UTF8String, DiameterIdentity,* and *Enumerated* (please refer to [6] for further details).

2.2 Diameter Messages for Mobile IP

Figure 2.4 recalls the overall message flow for an AAA / Mobile IP authentication in case the home agent is allocated in the home network as described in [21]. This section explains the format of the exchanged messages in more detail than [21] which was more focussed on a comparison of cryptographic protocols for mobile node authentication than on the exact message formats.

Foreign agents include random number challenges in their Mobile IP advertisement messages in order to allow for replay detection of Mobile IP registration messages send by mobile nodes. A mobile node wishing to register includes this random number challenge together with its *network access identifier (NAI)* and a so-called *MN-AAA authentication extension* in its registration message. The format of this authentication extension is defined in RFC 3012 and shown in figure 2.5 [8].

The *type* field of this extensions is set to "36" (to denote the generalized authentication extension) and the *subtype* field to "1" (to indicate the MN-AAA authentication subtype). The *length* field contains 4 plus the number of bytes in the authenticator. The *SPI* fields identifies the security association used to create the content of the *authenticator* field.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Subtype    |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              SPI                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Authenticator ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.5: The Generalized Mobile IP Authentication Extension [8]

2.2.1 The AMR Message

Upon reception of a Mobile IP registration message the foreign agent creates an *AA-Mobile-Node-Request (AMR)* message. The format of this message is defined in [9] which constitutes the base specification of the Diameter AAA / Mobile IP authentication procedure.

Figure 2.6 shows the ABNF definition of the AMR message. The message starts with a Diameter header whose command-code is set to “260” and that has the request- and the proxiable-bit set to one. Directly after the header follows the Diameter *Session-ID* AVP and the following mandatory AVPs:

- *Auth-Application-ID:* which contains “4” in its data field to indicate Mobile IP authentication,
- *User-Name:* is of type UTF8String and contains the user name in a format consistent with the NAI specification [1],
- *Destination-Realm:* is of type UTF8String and contains the *realm* of the destination Diameter server, i.e. the part of the user’s NAI following the “@”-sign,
- *Origin-Host:* identifies the endpoint which originated this Diameter message and is of type DiameterIdentity, a data type that uses UTF8 encoding and that has been derived from the base type OctetString,
- *Origin-Realm:* contains the realm of the originating Diameter entity and is encoded like the Destination-Realm AVP,
- *MIP-Reg-Request:* is of type OCtectString and contains the Mobile IP registration request message as send by the mobile node to the foreign agent,
- *MIP-MN-AAA-Auth:* is of type Grouped and contains data to simplify processing of the authentication data in the Mobile IP request by the target AAA server (usually the home AAA server AAAH). The contents of this AVP are the *security parameter index (SPI)* indicating the security association to be used to verify the authentication data, the length of the input to the authentication algorithm, the length of the authentication data, and an offset indicating the start of the authentication data inside the MIP-MN-AAA-Auth AVP.

```
<AA-Mobile-Node-Request> ::= < Diameter Header: 260, REQ, PXY >
                             < Session-ID >
                             { Auth-Application-Id }
                             { User-Name }
                             { Destination-Realm }
                             { Origin-Host }
                             { Origin-Realm }
                             { MIP-Reg-Request }
                             { MIP-MN-AAA-Auth }
                             [ MIP-Mobile-Node-Address ]
                             [ MIP-Home-Agent-Address ]
                             [ MIP-Feature-Vector ]
                             [ Authorization-Lifetime ]
                             [ MIP-FA-MN-Preferred-SPI ]
                             [ MIP-FA-HA-Preferred-SPI ]
                             [ MIP-Previous-FA-Host ]
                             [ MIP-Previous-FA-Addr ]
                             [ MIP-FA-Challenge ]
                             [ Destination-Host ]
                             [ Origin-State-Id ]
                           * [ AVP ]
                           * [ Proxy-Info ]
                           * [ Route-Record ]
```

Figure 2.6: ABNF of the AA Mobile Node Request Message (AMR)

The mandatory AVPs described so far may be followed by several optional AVPs:

- *MIP-Mobile-Node-Address:* is of type IPAddress and contains the mobile node's home address.
- *MIP-Home-Agent-Address:* is of type IPAddress and contains the mobile node's home agent address.
- *MIP-Feature-Vector:* is of type Unsigned32 and is added wih flag values set by the foreign agent or the foreign AAA server. It allows to specify if the assignement of a home address for the mobile node is requested, if this address must be allocated in the home domain, if the assignment of a home agent is requested, if a suitable home agent is available in the foreign domain, if and for which of the possible keys MN-HA-key, MN-FA-key, FA-HA-key a key should be provided by the home AAA server, if the home agent has to be allocated in the foreign network, and if the mobile node is acting with a co-located care-of-address.
- *Authorization-Lifetime:* is of type Unsigned32 and contains the number of seconds before the authorization and eventually distributed registration keys expire.
- *MIP-FA-MN-Preferred-SPI:* is of type Unsigned32 and contains the SPI the foreign agent would prefer to have assigned by the home agent in the MIP-FA-to-MN-Key AVP (see below).

- *MIP-FA-HA-Preferred-SPI:* like above but for the security association between foreign agent and home agent.
- *MIP-Previous-FA-Host:* is of type DiameterIdentity and contains the identity of the mobile node's old foreign agent. By including this AVP the Diameter client asks the Diameter server to return the registration keys that have previously been established between the old foreign agent and mobile node as well as between the old foreign agent and the home agent.
- *MIP-Previous-FA-Addr:* is of type IPAddress and contains the IP address of the mobile node's old foreign agent.
- *MIP-FA-Challenge:* is of type OctetString and contains the challenge advertised by the foreign agent to the mobile node. This AVP must be present in the AMR if RADIUS-style authentication is used.
- *Destination-Host:* the identity of the home AAA server in case the AMR generated is for continuation of a session that has previously been authorized by that AAA server.
- *Origin-State-Id:* is of type Unsigned32 and contains a monotonically increasing value that is advanced whenever a Diameter entity restarts with loss of previous state, for example upon reboot.

The AMR message is processed in the foreign AAA server which after possibly adding or modifying some optional AVPs sends this message towards the home AAA server.

2.2.2 The HAR Message

After receiving and procesing the AMR message the home AAA server generates a *Home Agent Request* message (HAR) and sends it to the home agent. The ABNF declaration of this message is shown in figure 2.7. The Diameter header of the message has the Command-Code field set to 262 and the Request- and Proxiable-bit set to one. In addition to some AVPs which are taken from the corresponding AMR message the HAR message may contain the following AVPs:

- *Accounting-Multi-Session-Id:* is a required AVP which is of type UTF8String. The Accounting-Multi-Session-Id AVP is used to link together multiple related accounting sessions, where each session would have a unique Accounting-Session-Id, but the same Acounting-Multi-Session-Id AVP.
- *MIP-Foreign-Agent-Host:* is also a required field and is of type DiameterIdentity. It contains the identity of the foreign agent which is copied from the value of the Origin-Host AVP in the AMR message.
- *MIP-MN-to-HA-Key:* is an optional AVP and of type OctetString. Its content is formatted arcording to [19] and it communicated the key to be used between the mobile node and the home agent to the mobile node.
- *MIP-MN-to-FA-Key:* analogous to MIP-MN-to-HA-Key but used to communicate the MN-to-FA key.

```
<Home-Agent-MIP-Request> ::= < Diameter Header: 262, REQ, PXY >
                             < Session-Id >
                             { Auth-Application-Id }
                             { Authorization-Lifetime }
                             { MIP-Reg-Request }
                             { Origin-Host }
                             { Origin-Realm }
                             { User-Name }
                             { Destination-Realm }
                             { Accounting-Multi-Session-Id }
                             { MIP-Foreign-Agent-Host }
                             [ Destination-Host ]
                             [ MIP-MN-to-HA-Key ]
                             [ MIP-MN-to-FA-Key ]
                             [ MIP-HA-to-MN-Key ]
                             [ MIP-HA-to-FA-Key ]
                             [ MIP-FA-to-MN-Key ]
                             [ MIP-FA-to-HA-Key ]
                             [ MIP-Mobile-Node-Address ]
                             [ MIP-Home-Agent-Address ]
                           * [ Filter-Rule ]
                             [ Session-Timeout ]
                             [ Origin-State-Id ]
                           * [ AVP ]
                           * [ Proxy-Info ]
                           * [ Route-Record ]
```

Figure 2.7: ABNF of the Home Agent Request Message (HAR)

- *MIP-HA-to-MN-Key:* is of type Grouped and contains four required AVPs:
  - *MIP-Peer-SPI:* is of type Unsigned32, and contains the security parameter index used to reference the key in the associated MIP-Session-Key AVP.
  - *MIP-Algorithm-Type:* is of type Enumerated, and contains the Algorithm identifier used to generate the associated Mobile IP authentication extension. Currently defined are the values *Prefix+Suffix MD5* (“0”) and *HMAC-MD5* (“1”).
  - *MIP-Replay-Mode:* is of type Enumerated and contains the replay mode the Home Agent should use when authenticating the Mobile Node. Currently defined are the values *None* (“0”), *Timestamps* (“1”), and *Nonces* (“2”).
  - *MIP-Session-Key:* is of type OctetString and contains the Session Key to be used between two Mobile IP entities.
- *MIP-HA-to-FA-Key:* is defined analogous to MIP-HA-to-MN-Key but used to communicate the HA-to-FA key.
- *MIP-FA-to-MN-Key, MIP-FA-to-HA-Key:* are also defined like MIP-FA-to-HA-Key with the exception that they do not require the MIP-Replay-Mode AVP.

```
<Home-Agent-MIP-Answer> ::= < Diameter Header: 262, PXY >
                            < Session-Id >
                            { Auth-Application-Id }
                            { Session-Timeout }
                            { Authorization-Lifetime }
                            { Result-Code }
                            { Origin-Host }
                            { Origin-Realm }
                            { Destination-Host }
                            { Accounting-Multi-Session-Id }
                            { MIP-Foreign-Agent-Host }
                            [ Error-Reporting-Host ]
                            [ MIP-Reg-Reply ]
                            [ MIP-Home-Agent-Address ]
                            [ MIP-Mobile-Node-Address ]
                            [ MIP-FA-to-MN-Key ]
                            [ MIP-FA-to-HA-Key ]
                            [ Filter-Rule ]
                            [ Origin-State-Id ]
                          * [ AVP ]
                          * [ Proxy-Info ]
                          * [ Route-Record ]
```

Figure 2.8: ABNF Definition of the Home Agent Answer Message (HAA)

- *Filter-Rule:* is of type UTF8String, and provides filter rules that need to be configured on the Foreign or Home Agent for the user (please refer to [9, section 4.10] for more details).

- *Session-Timeout:* is of type Unsigned32 and contains the maximum number of seconds of service to be provided to the user before termination of the session. A session terminated due to the Session-Timeout expiration must not generate a re-authentication and / or re-authorization.

2.2.3 The HAA Message

The home agent responds to the HAR message with a *Home Agent Answer* message (HAA). The format of this message is specified in the ABNF definition shown in figure 2.8. The message starts with the normal Diameter header which has the command-code field set to “262“ and the Proxiable-bit set to “1”. In addition to AVPs that have already been explained above, the message contains the following AVPs:

- *Result-Code:* is a required AVP that is of type Unsigned32 and indicates whether a particular request was completed successfully or whether an error occurred. For the Mobile IP application of Diameter five new values have been defined that indicate Mobile IP specific error conditions, e.g. authentication failure, failure of processing a registration request in the home agent, no home agent available, etc.

```
<AA-Mobile-Node-Answer> ::= < Diameter Header: 260, PXY >
                            < Session-Id >
                            { Auth-Application-Id }
                            { Authorization-Lifetime }
                            { Result-Code }
                            { Origin-Host }
                            { Origin-Realm }
                            { Destination-Host }
                            { Accounting-Multi-Session-Id }
                            [ Error-Reporting-Host ]
                            [ MIP-Reg-Reply ]
                            [ MIP-MN-to-FA-Key ]
                            [ MIP-MN-to-HA-Key ]
                            [ MIP-FA-to-MN-Key ]
                            [ MIP-FA-to-HA-Key ]
                            [ MIP-HA-to-MN-Key ]
                            [ MIP-Home-Agent-Address ]
                            [ MIP-Mobile-Node-Address ]
                          * [ Filter-Rule ]
                            [ Session-Timeout ]
                            [ Origin-State-Id ]
                          * [ AVP ]
                          * [ Proxy-Info ]
                          * [ Route-Record ]
```

Figure 2.9: ABNF Definition of the AA Mobile Node Answer Message (AMA)

- *Error-Reporting-Host:* in case the host setting the Result-Code is different from the one encoded in the Origin-Host AVP, this optional AVP contains the identity of the Diameter host that sent the Result-Code AVP to a value other than “2001” (Success). The data type of this AVP is DiameterIdentity.

- *MIP-Reg-Reply:* is of type OctetString and contains the Mobile IP registration reply message sent by the home agent to the foreign agent. In case the home AAA server generates keys to be used between the mobile node and the foreign or the home agent, these keys are included by the home agent into appropriate extensions to the Mobile IP registration message.

2.2.4 The AMA Message

After receiving the HAA message from the home agent (in case the home agent is allocated in the home domain) the home AAA server generates and sends a AA-Mobile-Node-Answer message (see also figure 2.9) to the foreign AAA server. The command-code field in the Diameter header is set to 260 and the Proxiable flag is set to “1”. For further explanations regarding specific AVPs please refer to the sections above, as all possible AVPs of the AMA message have already been explained.

Concerning the AVP *Destination-Host* which is specified to be mandatory in AMA-messages, it has to be mentioned, that the Diameter base protocol explicitely states that this AVP must not be present in answer messages [6, section 5.6].

2.3 Types of Authentication and AAA Message Contents

As already been stated in [21] a full AAA authentication is not needed for every handover of a mobile node from one foreign agent to another foreign agent. Three different situations can be distinguished:

1. The mobile node registers at a foreign agent that belongs to the same administrative domain like the foreign agent at which it is currently registered, and both foreign agents are served by the same AAA server. For this situation the current internet draft of the Diameter Mobile IP application [9] proposes an optimized procedure, in which the foreign AAA server provides the new foreign agent with the session keys that have been established for the old foreign agent, so that the new foreign agent can directly take over the role of the old foreign agent. In the remainder of this document we will refer to registration events of this type as a *type-1 handover*.
2. The mobile node registers at a foreign agent that belongs to the same administrative domain like the foreign agent at which it is currently registered, but both foreign agents are served by different AAA servers. This kind of handover events will be called *type-2 handovers*.
3. The mobile node wants to register at a foreign agent that belongs to an administrative domain where the mobile node has not registered before, or the session lifetime of the last AAA registration has expired, respectively. In this case a full AAA registration with involvement of the home AAA server is required. Handover events of this class will be called *type-3 handovers*.

Tables 2.1 to 2.4 summarize which AVPs are assumed to be necessary for each of the above types of handover events. Please note, that in the cases of type-1 and type-2 handovers, not all of the mandatory AVPs are really needed for the task of distributing the already established session keys for a mobile node to the new foreign agent. As the standardization of the Diameter protocol and its application for Mobile IP is not yet finished, the analysis in this report is aimed at evaluating the performance of handover authentication with an optimized approach that on one hand follows the present IETF standardization, but that on the other hand also tries to find an optimized mode of operation according to these standards.

For reasons of clarity tables 2.1 to 2.4 list the AVPs in the same order like the ABNF definition, but do not list optional AVPs that are not assumed to be necessary in any of the three handover event types and that follow the last AVP that occurs in at least one handover type.

2.4 Content of the Mobile IP Registration Messages

This section explains the formats of the Mobile IP registration messages and the Mobile IP registration extensions necessary for authentication based on an AAA infrastructure.

Figure 2.10 shows the format of the basic Mobile IP registration message according to [17]. It contains the following protocol fields:

Table 2.1: Assumptions on AVPs Exchanged in the AMR-Message

AVP	Type-1	Type-2	Type 3
< Diameter Header >	x	x	x
< Session-ID >	x	x	x
{ Auth-Application-Id }	x	x	x
{ User-Name }	x	x	x
{ Destination-Realm }	-	-	x
{ Origin-Host }	x	x	x
{ Origin-Realm }	x	x	x
{ MIP-Reg-Request }	-	-	x
{ MIP-MN-AAA-Auth }	-	-	x
[MIP-Mobile-Node-Address]	-	-	x
[MIP-Home-Agent-Address]	-	-	x
[MIP-Feature-Vector]	-	-	x
[MIP-Authorization-Lifetime]	-	-	-
[MIP-FA-MN-Preferred-SPI]	-	-	x
[MIP-FA-HA-Preferred-SPI]	-	-	x
[MIP-Previous-FA-Host]	-	-	-
[MIP-Previous-FA-Addr]	x	x	-

Table 2.2: Assumptions on AVPs Exchanged in the HAR-Message

AVP	Type-1	Type-2	Type 3
< Diameter Header >	-	-	x
< Session-ID >	-	-	x
{ Auth-Application-Id }	-	-	x
{ Authorization-Lifetime }	-	-	x
{ MIP-Reg-Request }	-	-	x
{ Origin-Host }	-	-	x
{ Origin-Realm }	-	-	x
{ User-Name }	-	-	x
{ Destination-Realm }	-	-	x
{ Accounting-Multi-Session-Id }	-	-	x
{ MIP-Foreign-Agent-Host }	-	-	x
[Destination-Host]	-	-	-
[MIP-MN-to-HA-Key]	-	-	x
[MIP-MN-to-FA-Key]	-	-	x
[MIP-HA-to-MN-Key]	-	-	x
[MIP-HA-to-FA-Key]	-	-	x
[MIP-FA-to-MN-Key]	-	-	-
[MIP-FA-to-HA-Key]	-	-	-

Table 2.3: Assumptions on AVPs Exchanged in the HAA-Message

AVP	Type-1	Type-2	Type 3
< Diameter Header >	-	-	x
< Session-ID >	-	-	x
{ Auth-Application-Id }	-	-	x
{ Session-Timeout }	-	-	x
{ Authorization-Lifetime }	-	-	x
{ Result-Code }	-	-	x
{ Origin-Host }	-	-	x
{ Origin-Realm }	-	-	x
{ Destination-Host }	-	-	x
{ Accounting-Multi-Session-Id }	-	-	x
{ MIP-Foreign-Agent-Host }	-	-	x
[Error-Reporting-Host]	-	-	-
[MIP-Reg-Reply]	-	-	x

Table 2.4: Assumptions on AVPs Exchanged in the AMA-Message

AVP	Type-1	Type-2	Type 3
< Diameter Header >	x	x	x
< Session-ID >	x	x	x
{ Auth-Application-Id }	x	x	x
{ Authorization-Lifetime }	x	x	x
{ Result-Code }	x	x	x
{ Origin-Host }	x	x	x
{ Origin-Realm }	x	x	x
{ Accounting-Multi-Session-Id }	x	x	x
[Error-Reporting-Host]	-	-	-
{ MIP-Reg-Reply }	-	-	x
[MIP-MN-to-HA-Key]	-	-	x
[MIP-MN-to-FA-Key]	-	-	x
[MIP-FA-to-MN-Key]	x	x	x
[MIP-FA-to-HA-Key]	x	x	x

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |S|B|D|M|G|V|rsv|          Lifetime             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Home Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Home Agent                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Care-of Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                         Identification                        +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Extensions ...
+-+-+-+-+-+-+-+-
```

Figure 2.10: Format of the fixed part of the Mobile IP Registration Request

- *Type:* this field contains the value “1” to identify a registration request message.
- The flags following the type-field allow to specify the following options:
  - *S:* Simultaneous bindings, if the ’S’ bit is set, the mobile node is requesting that the home agent retain its prior mobility bindings.
  - *B:* Broadcast datagrams, if the ’B’ bit is set, the mobile node requests that the home agent tunnel to it any broadcast datagrams that it receives on the home network.
  - *D:*Decapsulation by mobile node, if the ’D’ bit is set, the mobile node will itself decapsulate datagrams which are sent to the care-of address. That is, the mobile node is using a co-located care-of address.
  - *M:* Minimal encapsulation, if the ’M’ bit is set, the mobile node requests that its home agent use minimal encapsulation for datagrams tunneled to the mobile node.
  - *G:* GRE encapsulation, if the ’G’ bit is set, the mobile node requests that its home agent use GRE encapsulation for datagrams tunneled to the mobile node.
  - *V:* The mobile node requests that its mobility agent use Van Jacobson header compression over its link with the mobile node.
  - *rsv:* Reserved bits, sent as zero.
- *Lifetime:* The number of seconds remaining before the registration is considered expired. A value of zero indicates a request for deregistration. A value of 0xffff indicates infinity.
- *Home Address:* The IP address of the mobile node.
- *Home Agent:* The IP address of the mobile node’s home agent.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |     Code      |           Lifetime            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                          Home Address                         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           Home Agent                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 +                         Identification                        +
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Extensions ...
 +-+-+-+-+-+-+-+-
```

Figure 2.11: Format of the fixed part of the Mobile IP Registration Reply

- *Care-of Address:* The IP address for the end of the tunnel.
- *Identification:* A 64-bit number, constructed by the mobile node, used for matching Registration Requests with Registration Replies, and for protecting against replay attacks of registration messages.
- *Extensions:* The fixed portion of the Registration Request is followed by one or more of the Extensions.

Figure 2.11 shows the format of a Mobile IP registration reply message. It contains the following protocol fields [17]:

- *Type:* this field contains the value “3” to identify a registration reply message.
- *Code:* the value contained in this field indicats the result of the registration request.
- *Lifetime:* If the Code field indicates that the registration was accepted, the Lifetime field is set to the number of seconds remaining before the registration is considered expired. A value of zero indicates that the mobile node has been deregistered. A value of 0xffff indicates infinity.
- *Home Address:* The IP address of the mobile node.
- *Home Agent:* The IP address of the mobile node’s home agent.
- *Identification:* A 64-bit number used for matching registration requests with registration replies, and for protecting against replay attacks of registration messages. The value is based on the Identification field from the registration request message from the mobile node, and on the style of replay protection used in the security context between the mobile node and its home agent.
- *Extensions:* The fixed portion of the registration reply is followed by one or more extensions.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Length    |         SPI  ....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       ... SPI (cont.)          |       Authenticator ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.12: Format of a Mobile IP Authentication Extension

The common format of Mobile IP authentication extensions is depicted in figure 2.12. It contains the following protocol fields [17]:

- *Type:* the value contained in this field identifies this extension as one of the following MN-HA, MN-FA, or FA-HA authentication extension.
- *Length:* 4 plus the number of bytes in the Authenticator.
- *SPI:* the security parameter index identifying the security association that has to be used to check the contained authenticator.
- *Authenticator:* a cryptographic check value, that allows to check if any protocol field prior to this extension and including the type- and the length-field of this extension has been modified on its way from the peer that created this extension to the verifying peer entity (MN, FA, or HA). The default method for computing the authenticator is MD-5 in prefix+suffix mode. However, the actual method to be used is specified in the context of the security association that is identified by the SPI-field.

Figure 2.13 shows the common format of the *Generalized Key Request Extensions*. There are three extensions of this type, one for each of the MN-FA key, the MN-HA key and the FA-HA key. In the context of key distribution for Mobile IP with the help of an AAA infrastructure only the former two are used, as the FA-HA key is directly communicated to the FA and the HA by appropriate Diameter AVPs. The key request extensions contain the following protocol fields:

- *Type:* an identifier for this Mobile IP extension, multiple values have been assigned to identify MN-FA, MN-HA, and FA-HA key requests.
- *Subtype:* number assigned to identify the way in which the key request data is to be used when generating the registration key, e.g this field carries the number 7 for the MN-FA key request from AAA.
- *Length:* 4 plus the number of bytes in the field subtype data.
- *Mobilie Node SPI:* the security parameters index that the mobile node will assign for the security association created for use with the registration key.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |    Subtype    |            Length             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Mobile Node SPI                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  MN-(FA/HA) Key Request Subtype Data (not used with AAA) ...
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.13: Format of a Generalized Key Request Extension

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |    Subtype    |            Length             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             Lifetime                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             AAA SPI                           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           (FA/HA) SPI                         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |      Algorithm Identifier     |      Key Material ...
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.14: Format of an Unsolicited Key Material From AAA Extension

- *Subtype Data:* this field may contain data needed to carry out the creation of the registration key on behalf of the mobile node. It is not used in the context of AAA key distribution.

The common format of the *Unsolicited Key Material From AAA Extensions* is depicted in figure 2.14. Two extensions of this kind are included in the Mobile IP Registration Reply Messages in order to communicate the MN-FA and the MN-HA session keys. The data fields of these extensions are as follows:

- *Type, Subtype:* This extension uses subtype 7 (for MN-FA key), or subtype 1 (for MN-HA key), respectively, of the Generalized MN-FA Key Reply Extension [20] which is indicated by a value of 40 in the Type field.

- *Lifetime:* This field indicates the duration of time (in seconds) for which the contained key is valid.

- *AAA SPI:* A 32-bit opaque value, indicating the SPI that the mobile node must use to determine the algorithm to use for establishing the FA security information.

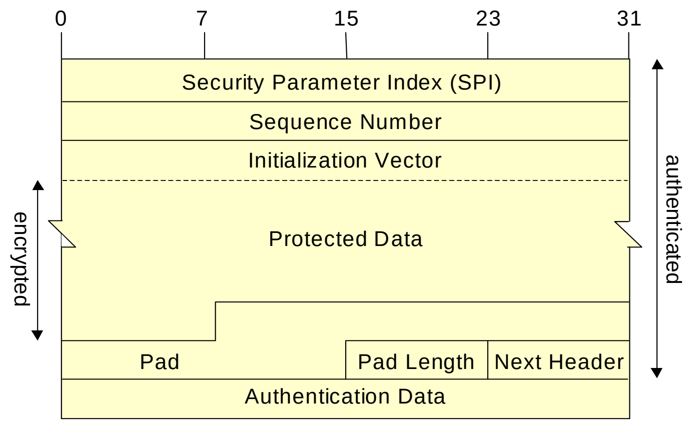

Figure 2.15: Format of the Encapsulating Security Payload

- *FA SPI:* A 32-bit opaque value, which the mobile node must use to index all the necessary information established for the FA security information after it is decoded.
- *Algorithm Identifier:* This field indicates the algorithm to be used for future computations of the MN-FA Authentication Extension
- *Key Material:* A random value of at least 64 bits.

2.5 Format of ESP-Protected IP Packets

The messages exchanged between two Diameter peer entities have to be protected, in order to attain a minimum level of security. The Diameter base specification advocates the mandatory use of a *hop-by-hop security model*, so that messages are protected between directly communucating Diameter entities. As a consequence, intermediate Diameter entities can modify the messages they relay and/or process. If end-to-end security is needed, an optional Diameter security application [7] can be used.

The Diameter base specification mandates the use of the IPSec security protocols to protect Diameter messages between FAs/HAs and AAA servers and the use of either *IPSec* [4] or the *Transport Layer Security (TLS)* protocol [12] to protect messages between Diameter servers (AAA servers, brokers, etc.). For reasons of simplicity, we assume the homogenous use of IPSec between all Diameter entities, more precisely the use of the *Encapsulating Security Payload (ESP)* protocol [3] of IPSec with message authentication and encryption.

Figure 2.15 shows the format of ESP-protected messages:

- The *SPI field* indicates the SA to be used for this packet.
- The *sequence number* provides replay protection.
- If the cryptographic algorithm in use requires an *initialization vector*, it is transmitted in the clear in every packet in the beginning of the payload.

Table 2.5: Lengths of of Mobile IP & Diameter Messages

Message	Type-1	Type-2	Type 3
Mobile IP Reg.Request	92 (– / 72)	134 (– / 114)	90 (– / 70)
Diameter AMR	236 (192 / 200)	236 (192 / 200)	436 (392 / 400)
Diameter HAR	–	–	532 (488 / 496)
Diameter HAA	–	–	492 (448 / 456)
Diameter AMA	340 (296 / 304)	340 (296 / 304)	476 (432 / 440)
Mobile IP Reg.Request	66 (– / 46)	66 (– / 46)	130 (– / 110)

- The *protected payload* contains the user payload of this packet, e.g. a TCP-fragment or an UDP-packet including the appropriate header.
- The *pad field* serves to ensure:
  - padding of the payload up to the required block length of the cipher in use, and
  - padding of the payload to right-justify the pad-length and next-header fields into the high-order 16 bit of a 32-bit word.
- The *pad length* indicates the amount of padding bytes added.
- The *next-header* field of the ESP header indicates the encapsulated payload.
- The optional *authentication-data* field contains a message authentication code, if present.

2.6 Summary

This chapter discussed the various message formats relevant to the AAA / Mobile IP registration and authentication procedures. Table 2.5 summarizes the lengths of the messages exchanged during a handover operation according to the classes distinguished in section 2.3. These values are used in the simulation model, described in the next chapter, that was build in order to evaluate the performance of AAA / Mobile IP authentication under various circumstances.

Chapter 3

A Simulation Model of AAA / Mobile IP Authentication

3.1 Overview over the simulation model

The simulation model was developed under Linux, using C++ and OMNeT++, which is an object-oriented modular discrete event simulator. The name itself stands for Objective Modular Network Testbed in C++. The description file of the network is written in the NED language of OMNet++. The NED language supports modular description of a network. A network consists of modules, connections and messages.

Modules are the active entites of a network (e.g. FA, HA, AAAL, ...). The functionality of each module is defined inside a single file, that is written in C++. Modules are linked together via connections of different types, which are described later on. Modules have the ability to communicate with each other by sending and receiving messages. There are two different types of messages that could be used inside OMNet++. *Normal messages*, which are transmitted over connections and *self-messages* which are sent from one module to the same module, so that transmitter and receiver are identical. Furthermore, self-messages are received at the same point of time as when they were sent. Self-messages could be scheduled at a certain point of time in future. We use this feature to model the different processing times of each module. For instance if an AAA local server needs 50 μs to de- and re- encrypt a message, we model this through scheduling a self-message at the corresponding point of time in future. When the AAA local server receives this self-message, it continues with the next step in the correct processing order.

Figure 3.1 shows the network we used for the simulations. The complete model consists of one Home Agent (HA), one AAA Home server (AAAH), two AAA Local server (AAAL), twenty Foreign Agents (FA) and twenty Mobile Environments (MOBEN). A MOBEN represents a cell of the network containing 4000 mobile nodes (see also section 3.2). The modules are connected with each other as follows:

- Mobile Environment $\Leftrightarrow$ Foreign Agent via Wireless link (2 MBit/s datarate).

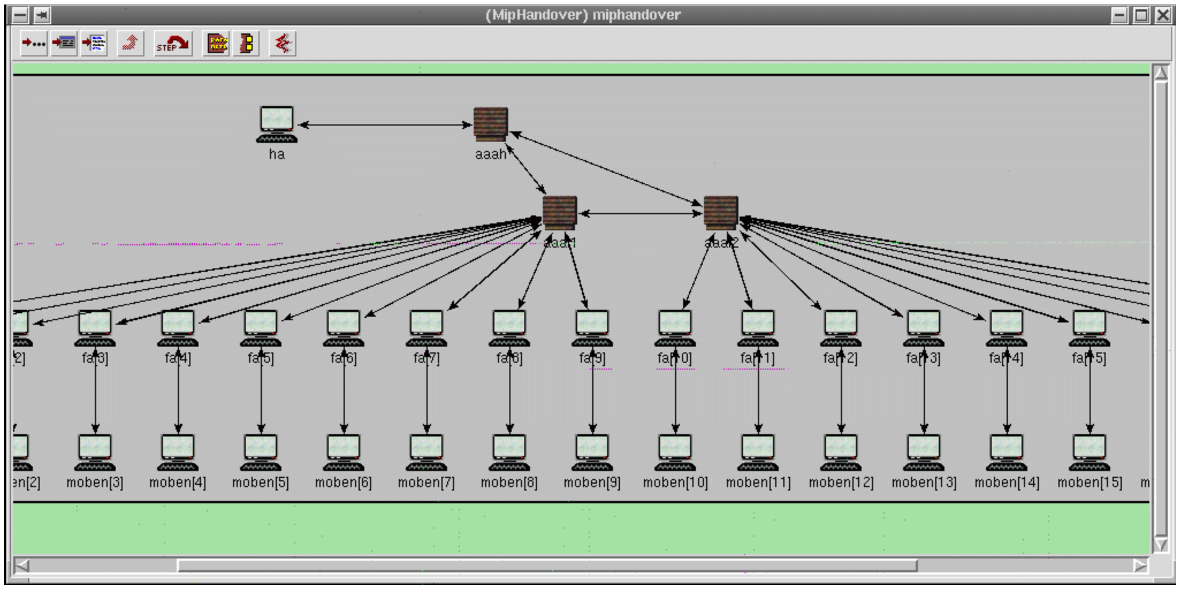

Figure 3.1: Overview of the model

- Foreign Agent ⇔ AAA Local Server via Local link (2 MBit/s datarate).
- AAA Local Server ⇔ AAA Local Server via Backbone link (100 MBit/s datarate).
- AAA Local Server ⇔ AAA Home Server: delay is dominating 20, 50, 100 ms.
- AAA Home Server ⇔ Home Agent via Local link (2 MBit/s datarate).

The data rate is specified in bits/second, and it is used for transmission delay calculation. The sending time of a message normally corresponds to the transmission of the first bit, and the arrival time of a message corresponds to the reception of the last bit. Furthermore, we take the end-to-end delay into account during the simulations.

In the following we differentiate between three different types of handover processes implemented inside the simulation model (see also section 3.7).

- **Handover-Type 1:** The standard handover, a mobile node changes its foreign agent, but keeps the same AAA local server.
- **Handover-Type 2:** An intra domain handover. A mobile node changes its foreign agent and its AAA local server.
- **Handover-Type 3:** A full authentication process involving the AAA home server and home agent.

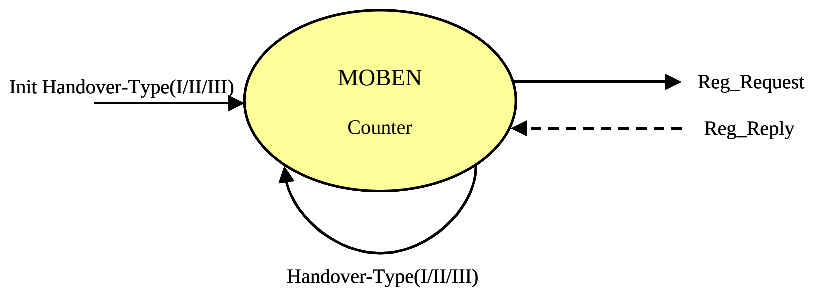

Figure 3.2: Mobile Environment

3.2 The Mobile Environment (MOBEN)

A mobile environment represents a cell with 4000 mobile nodes. Each mobile environment is connected via a wireless link with exactly one foreign agent. A counter inside a mobile environment stores the amount of already invoked handover processes. If the counter equals the former determined amount of handover processes to be simulated per mobile environment, no new handover events are generated. The simulation ends when no more messages reside inside the network.

Figure 3.2 depicts the model of a mobile environment. The straight line indicates request messages and the dashed one answer messages. At the beginning of the simulation the handover arrival process is initalized through the sending of Handover-Type(1/2/3) messages. Then later on, each mobile environment schedules its handover events on its own.

A mobile envrionment is able to send and to receive the following messages:

- Messages that could be received by a mobile environment:
  - Handover-Type(1/2/3)
  - Registration Reply.
- Messages that could be sent by a mobile environment:
  - Handover-Type(1/2/3)
  - Registration Request

If a mobile environment receives a handover-type(1/2/3) message, it will create a mobile ip registration request message. Then the message is sent via the wireless link to the foreign agent. Finally the mobile environment invokes a new handover event of the same type, by scheduling a self-message of type Handover-Type(1/2/3) at a defined point of time in future corresponding to the determined handover arrival process distribution.

A handover process is complete, when the mobile environment receives the corresponding mobile ip registration reply message.

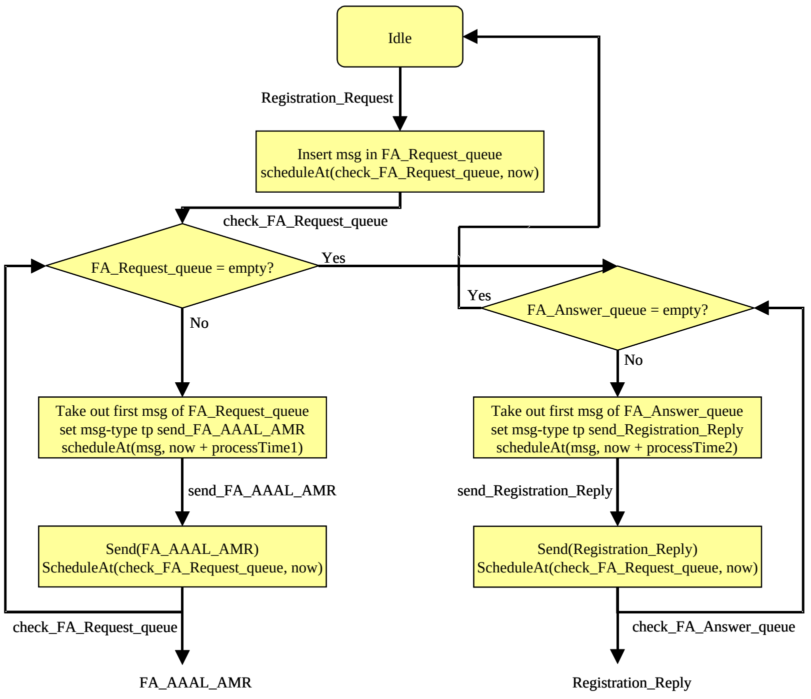
Idle
Registration_Request
Insert msg in FA_Request_queue
scheduleAt(check_FA_Request_queue, now)
check_FA_Request_queue
FA_Request_queue = empty?
Yes
No
Yes
FA_Answer_queue = empty?
No
Take out first msg of FA_Request_queue
set msg-type tp send_FA_AAAL_AMR
scheduleAt(msg, now + processTime1)
Take out first msg of FA_Answer_queue
set msg-type tp send_Registration_Reply
scheduleAt(msg, now + processTime2)
send_FA_AAAL_AMR
send_Registration_Reply
Send(FA_AAAL_AMR)
ScheduleAt(check_FA_Request_queue, now)
Send(Registration_Reply)
ScheduleAt(check_FA_Request_queue, now)
check_FA_Request_queue
check_FA_Answer_queue
FA_AAAL_AMR
Registration_Reply

Figure 3.3: Foreign Agent - Receiving a registration_request message

3.3 Foreign Agent (FA)

A foreign agent (see also figure 3.4) processes registration request messages in one direction and AAA server answer messages (AMA) in the other direction.

In the following section the single steps executed inside a foreign agent are explained and could also be seen in figure 3.3. Each foreign agent possesses two queues of type first-in-first-out (FIFO) and of unlimited length. If a foreign agent receives a registration_request_message from a mobile environment, the message is inserted into the fa_request_queue. In a next step, the foreign agent sends immediately a check_ fa_ request_ queue self-message to itself. Receiving this message, the foreign agent first checks the fa_request_queue. If there are messages waiting to be processed the foreign agent takes the first message out of the fa_request_queue. In the other case that there is no message inside the fa_request_queue the foreign agent controls the fa_answer_queue.

But following the first scenario (there was a message inside the fa_request_queue), the message then is transformed into a send_FA_AAAL_AMR self-message. The analysis and encryption of the message is represented through a defined period of processing time. After the expiration of this period of time, later on called *process time*, the self-message is sent.

Finally, if a send_FA_AAAL_AMR message is received by a foreign agent, the message type is set to FA_AAAL_AMR. After this, the message is immediately sent to the corresponding AAA local server and at the same time another check_fa_request_queue message is sent.

In the other direction, if a foreign agent receives an AA-Mobile-Node-Answer message, the proce-

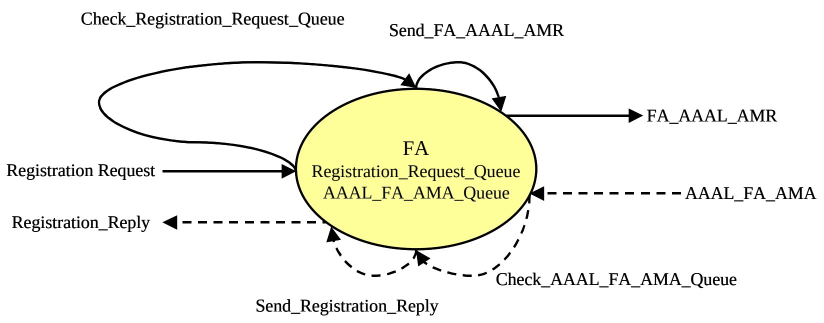

Figure 3.4: Foreign Agent

dure is analogous to the one described above. The AA-Mobile-Node-Answer is inserted into the FA_Answer_queue and a check_FA_Answer_queue self-message is scheduled. Receiving a message of this kind, first the foreign agent tries to get a message out of the FA_Answer_queue (otherwise the other queue is checked). Then the message type is set to send_registration_reply and the message is scheduled. After expiration of the corresponding process time the message is sent. Consequently, after receiving the send_registration_reply message, a registration_reply message is sent to the corresponding mobile environment and the next check_FA_Answer_queue self-message is scheduled.

As a summary we can state that there is a queue checking process which lasts until both queues of the foreign agent are empty.

3.4 AAA Local Server (AAAL)

The network possesses two AAA local servers, which are connected with each other. Further on, they are both connected the AAA home server and each AAA local server is connected to its own ten foreign agents. Each AAA local server possesses two queues of type FIFO and of unlimited length.

The sequence of the steps executed after the reception of a message are similar to the steps described above. Only the names and the process times differ.

The received (AAAL_AAAL_ or FA_AAAL_)AMR message is put into the AAAL_Request_queue. Then again by sending a self message a new queue checking process is invoked. Assumed that the AAAL_Request_queue is not empty the first message is taken out of the queue. The subsequent processing is dependent on the handover type and is further discussed in section 3.7.

3.5 AAA Home Server (AAAH)

The network contains one AAA home server. This AAA home server is connected with the home agent and with the two AAA local servers. The AAA home server receives an AAAL_AAAH_AMR message from one of the both AAA local servers and inserts it into the AAAH_Request_queue.

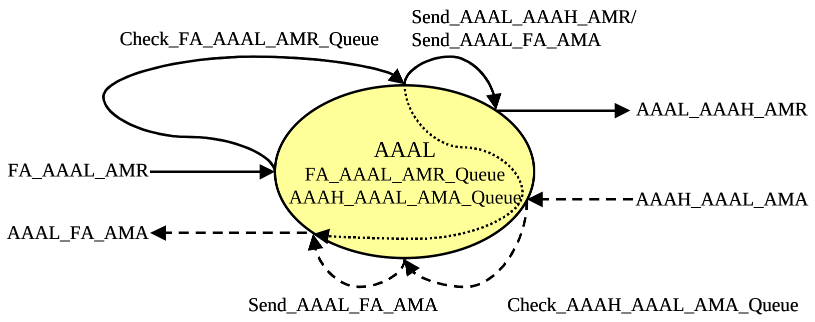

Figure 3.5: Local AAA-Server

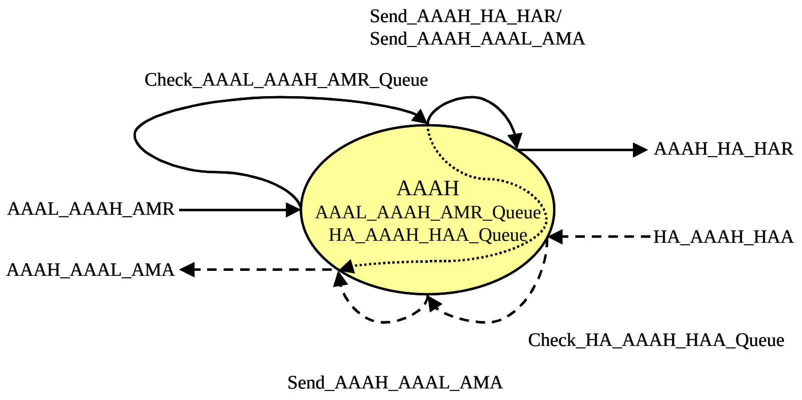

Figure 3.6: Home AAA-Server

Immediately after this, the AAA home server sends a check_AAAH_Request_queue message. The sequence of the scenario following does not differ from these described above. The first message of the AAAH_Request_queue is taken out. Then a send_AAAH_HA_HAR message is scheduled according to the corresponding process time. Receiving the send_AAAL_AAAH_AMR message, the message type is converted into AAAL_AAAH_HAR. After this, the message is sent to the connected home agent.

The case of receiving a HA_AAAH_HAA message is skipped, as the sequence of the steps is the same as already described.

3.6 Home Agent (HA)

The home agent possesses one FIFO queue of unlimited length. The received AAAH_HA_HAR messages are inserted into this queue. Then the home agent invokes a queue checking process. One after the other of the messages inside the queue are processed and sent as HA_AAAH_HAA messages to the AAA home server.

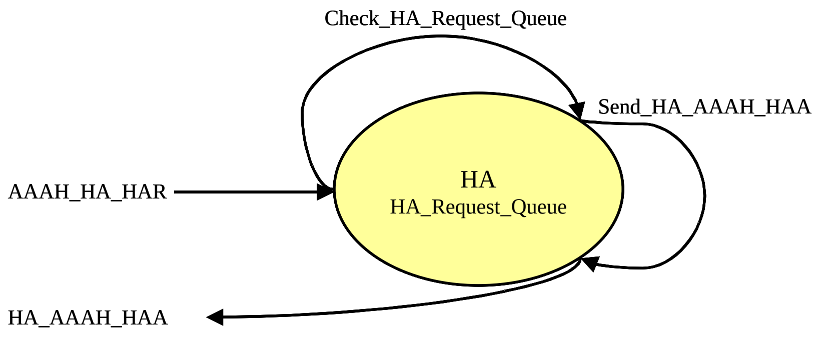

Figure 3.7: Home Agent (HA)

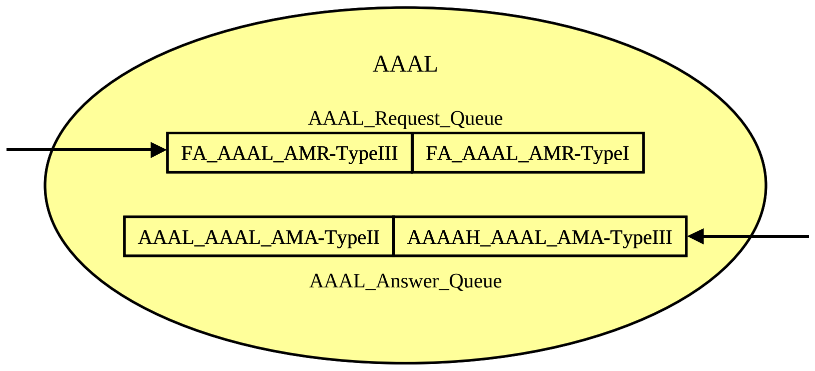

Figure 3.8: Possible situation inside an AAAL

3.7 The Handover Processes

In the following section the three types of handover events which could take place inside the simulation model are decribed in detail. In the following we deal the three cases of handover events seperately. During a simulation the three handover types may be processed in a mixed fashion. Thus the case could happen as pictured in figure 3.8

3.7.1 Type 1: The standard handover

A standard handover is, if a mobile node changes from one cell to another one, which belongs to the same domain as the first one. Hence the same AAA local server is responsible for the mobile node as before. The message flow of a handover process of type 1 is shown in figure 3.9.

Remark: In the following list there is a jump from step 8 to step 21. The missing steps will be described in the next two subsections. Using this numeration, identical sub-processes have the same step number through all three handover operations.

1. The mobile environment receives a Handover_Type1 message and creates a registration_request message.

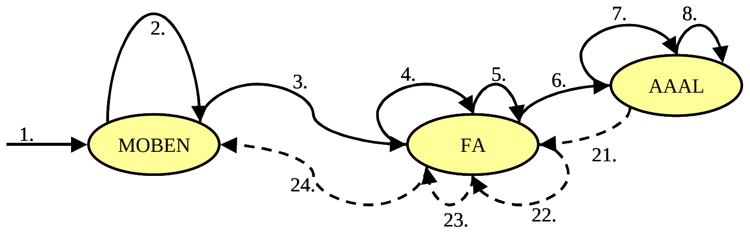

Figure 3.9: Message flow during a standard handover

2. The mobile environment creates a new Handover_Type1 message and schedules it corresponding to the Poisson distribution described in section 3.9.1.

3. The mobile environment sends the registration_request message to the responsible foreign agent.

4. The foreign agent receives a registration request message and inserts it into the FA_Request_queue. The foreign agent sends a self message to check the FA_Request_queue.

5. Receiving a check_FA_Request_queue message the first message (assumed that there is one) is taken out of the queue. This message is then transformed into a send_FA_AAAL_AMR self-message and scheduled.

6. Receiving the send_FA_AAAL_AMR self-message the foreign agent transforms the message into an FA_AAAL_AMR message and sends this message to the connected AAA local server.

7. Receiving the FA_AAAL_AMR message the AAA local server inserts the message into the AAAL_Request_queue and sends a check_AAAL_Request_queue message.

8. Receiving a check_AAAL_Request_queue message (assumed there is a message inside the queue) the first message is taken out of the queue. Now the message is transformed into a message of type send_AAAL_FA_AMA and scheduled.

21. Receiving a send_AAAL_FA_AMA message the AAA local server transforms the message into an AAAL_FA_AMA message and sends this message to the correct foreign agent.

22. The foreign agent receives an AAAL_FA_AMA message and inserts this message into the FA_Answer_queue. At the same time a check_FA_Answer_queue message is sent.

23. The foreign agent receives a check_FA_Answer_queue message, the first message (assumed that there is a message inside) is taken out of the queue. Now the foreign agent creates a send_Registration_Reply message and schedules this message.

24. The foreign agent receives a send_registration_reply message. The message type is set to registration_reply and sent to the connected mobile environment. The mobile environment receives a registration_reply message. Now the handover process with the identification number included in the registration_reply message is complete.

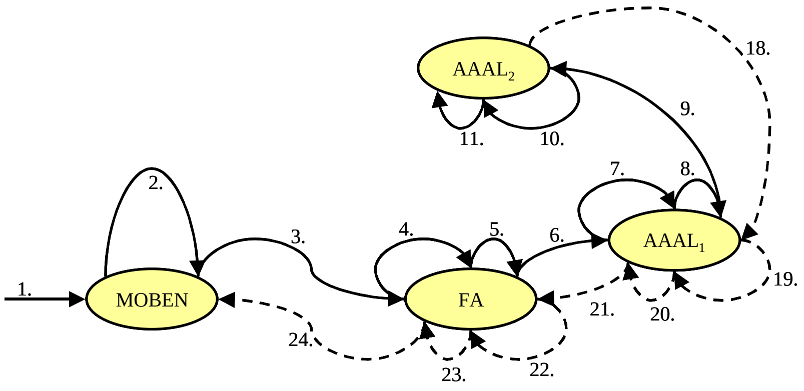

Figure 3.10: Message flow during an inter domain handover

3.7.2 Type 2: An optimized fast handover

The handover process of type 2 takes place if a mobile node changes the AAA local server area of responsibility. In this case a new AAA local server is supervising the authentication process of the mobile node. Therefore the new AAA local server has to communicate with the old AAA local server. The message flow of a handover process of type 2 is shown in figure 3.10. The steps 1 to 6 of a handover of type 2 do not differ of those from an standard handover. Hence we start here with step 7.

7. The $AAAL_1$ receives a check_FA_AAAL_queue message. The first message is taken out of the queue. Then the server transforms the message into a send_AAAL_AAAL_AMR message and schedules this self-message.

8. The $AAAL_1$ server receives a send_AAAL_AAAL_AMR message and transforms this into an AAAL_AAAL_AMR message. After this, the message is sent to the connected old AAA local server.

9. $AAAL_2$ receives an AAAL_AAAL_AMR message and inserts this message into the corresponding AAAL_Request_queue. At the same time a check_AAAL_Request_queue message is sent.

10. Receiving a check_AAAL_Request_queue message, the first message is taken out of the queue. The old AAA local server creates a send_AAAL_AAAL_AMA message and schedules this message.

18. Receiving a send_AAAH_AAAL_AMA message the AAA home server transforms the message into an AAAH_AAAL_AMA message and sends this message to the correct AAA local server.

19. $AAAL_1$ receives an AAAL_AAAL_AMA message. Then message is inserted into the corresponding AAAL_Answer_queue. At the same time the AAA local server sends a message of type check_AAAL_Answer_queue.

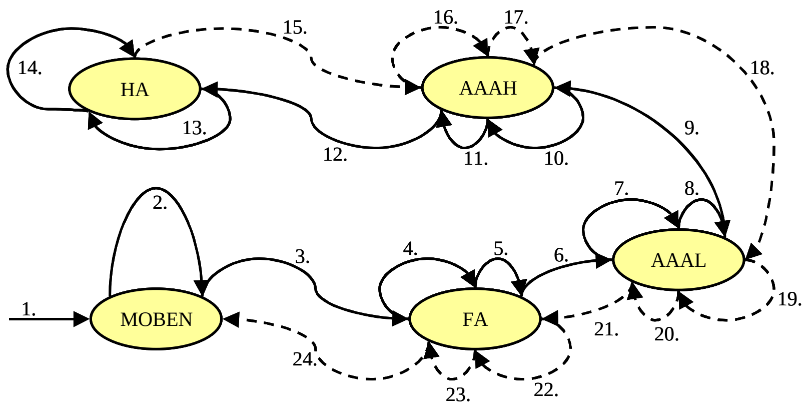

Figure 3.11: Message flow during a complete authentication process

20. Receiving a check_AAAL_Answer_queue message, the server takes out the first message of the queue. Subsequently the server transforms the message into a send_AAAL_FA_AMA message and schedules the message.

21. The AAA local server receives a send_AAAL_FA_AMA message. Now the proceeding is the same as already described above.

3.7.3 Type 3: The complete authentication process

A complete authentication process (see figure 3.11) takes place when a mobile node registers for the first time or when the session keys expire. In this case also the AAA home server and the home agent are involved. The description of the complete authentication process starts with step 11, the steps before are similar to those described above.

11. Receiving a check_AAAH_Request_queue message, the first message is taken out of the queue. Then the AAA home server creates a send_AAAH_HA_HAR message and schedules this message.

12. Receiving a send_AAAH_HA_HAR message the AAA home server transforms the message into an AAAH_HA_HAR message and sends this message to the connected home agent.

13. The home agent receives an AAAH_HA_HAR message, first the message is inserted into the HA_Request_queue. Then a check_HA_Request_queue message is sent.

14. The home agent takes the first message out of the queue. Then the home agent creates a send_HA_AAAH_HAA message and schedules the message.

15. The home agent receives a send_HA_AAAH_HAA message and transforms this message into an HA_AAAH_HAA message. The message is sent to the AAA home server.

16. The AAA home server, receives a HA_AAAH_HAA message and inserts this message into the AAAH_Answer_queue and sends a check_AAAH_Answer_queue message.

17. Receiving the check_AAAH_Answer_queue message the first item inside the queue is taken out. Then the AAA home server creates a send_AAAH_AAAL_AMA message and schedules this message. The further steps are identical to those already described above.

3.8 Process Parameters

In this chapter the dimensioning of the parameters is explained. The following assumptions were made:

- Hostname: 20 Bytes
- Network Access Identifier: 20 Bytes
- Authorization Lifetime: 4 Bytes
- Authenticator: 16/20 Bytes (MD5/SHA-1)
- Key: 16 Bytes
- TCP-Header (without options): 20 Bytes
- ESP-Header: 8 Bytes
- ESP-Authentication Extension: 16 Bytes
- Hashing 512 bit with MD-5: 1.5 μs
- Hashing 512 bit with SHA-1: 3.73 μs
- Encrypting n * 64 bit with DES: 3.06 μs + n * 1.51 μs
- Encrypting n * 64 bit with 3DES: 9.17 μs + n * 4.14 μs
- Transmission delay on wireless link:
  - Physical layer overhead 192 μs
  - MAC layer overhead 136 μs
  - IP layer overhead 80 μs
  - UDP layer overhead 32 μs

As written in [6] Diameter clients, such as Network Access Servers (NASes) and Foreign Agents MUST support IP Security. Diameter servers MUST support TLS, but the administrator MAY opt to configure IPSec instead of using TLS. Operating the Diameter protocol without any security mechanism is not recommended. We assumed the usage of IPSec, more precisely the IP Encapsulating Security Payload (ESP).

In table 3.1 all process times are depicted.

Table 3.1: Process Times

Action	Type-1	Type-2	Type 3
Action	HO-Type 1	HO-Type 2	HO-Type 3
MOBEN: MD5 + DES			
Encrypting + (Signing): Registration Request	3 μs	5 μs	3 μs
Decrypting: Registration Reply	3 μs	3 μs	5 μs
FA: MD5 + DES			
(De-) + Encrypting + Signing: AMR	48 μs	45 μs	88 μs
Decrypting + Signing: Registration Reply	69 μs	69 μs	100 μs
AAAL: MD5 + DES			
(De-) + Encrypting + Signing: AMR	–	91/112 μs	175 μs
(De-) + Encrypting + Signing: AMA	112 μs	133 μs	190 μs
AAAH: MD5 + DES			
(De-) + Encrypting + Signing: HAR	–	–	195 μs
(De-) + Encrypting + Signing: AMA	–	–	195 μs
HA: MD5 + DES			
(De-) + Encrypting + Signing: HAA	–	–	207 μs
MOBEN: SHA-1 + 3DES			
Encrypting + (Signing): Registration Request	8 μs	11 μs	8 μs
Decrypting: Registration Reply	8 μs	8 μs	11 μs
FA: SHA-1 + 3DES			
(De-) + Encrypting + Signing: AMR	131 μs	128 μs	238 μs
Decrypting + Signing: Registration Reply	188 μs	188 μs	270 μs
AAAL: SHA-1 + 3DES			
(De-) + Encrypting + Signing: AMR	–	247/304 μs	476 μs
(De-) + Encrypting + Signing: AMA	304 μs	362 μs	518 μs
AAAH: SHA-1 + 3DES			
(De-) + Encrypting + Signing: HAR	–	–	524 μs
(De-) + Encrypting + Signing: AMA	–	–	577 μs
HA: MD5 + DES			
(De-) + Encrypting + Signing: HAA	–	–	603 μs

3.9 Mobility Models

In the following, we describe two alternative mobility models that have been integraded into the simulation model.

3.9.1 Traffic Model 1

Each cell represents a portion of 4000 mobile nodes. Mobile nodes are just modeled at moments when an authentication operation has to be performed.

In every cell (mobile environment) three idependent processes generate handover operations of the three handover types:

- Mean inter-arrival time of Type-1 HO: 0.5 s
- Mean inter-arrival time of Type-2 HO: 4.5 s
- Mean inter-arrivel time of Type-3 HO: 3.6 s

This means that each mobile node performs:

- 1.8 Type-1 handover operations per hour
- 0.2 Type-2 handover operations per hour
- 1 Type-3 handover operation every 4, 8, 16, 32, 64 hours

In the first realised handover traffic model the inter arrival of handover operations is modeled with a Poisson process leading to exponentially distributed inter arrival times.

Within the area of a single cell, the average number of mobile nodes is invariant and therefore, the rate of incoming mobile nodes equals the rate of outgoing mobile nodes.

Mobile nodes enter/leave a certain cell area accroding to a Poisson random process.

In figure 3.12 the exponentially distributed interarrival times for handover type 1 are pictured. We see that with a high probability we receive small interarrival times and with a much smaller probability long interarrival times.

3.9.2 Traffic Model 2

The second traffic model is a simplified version of the model described in [16]. We adopted the idea of three different moving environments, but we did not include a map as a base for moving directions and decisions. In this traffic model a cell has the shape of a circle with a diameter of 1000m. From the mobility viewpoint, we consider the following mobile node categories:

- pedestrians with a speed following the Gaussian distribution with mean value 5 km/hr and variance 30%

- car passengers with a speed following the Gaussian distribution with mean value 20 km/hr and variance 30%
- users located in buildings.

The rate of each category stays invariant during simulations.

First we focus on a single cell. In each cell we have again 4000 mobile nodes which are divided into the three categories as follows:

- 2000 mobile nodes are located inside buildings
- 1000 pedestrians
- 1000 car passengers

The 2000 mobile nodes which are located inside buildings invoke only handover processes of type 3. We assume that these mobile nodes move inside the area of one cell and do not cross any cell borders, so they do not invoke handover processes of type 2 or type 3.

Further on, for the handover processes of type 1 we determine for each moving mobile node a distance (Gaussian distribution mean 500m and variance 500m) which the mobile node has to travel to reach a cell border and a speed corresponding to the adequate environment (pedestrian, car). Then the time needed to travel this distance is calculated and the handover process is correspondingly scheduled. This means if a mobile node moves in one direction (without turning), the distance could be maximal 1000m until the mobile node crosses a cell border, if we assume that the mobile does not change its moving direction. So we represent the distance a mobile node has to travel until reaching a cell border through a Gaussian distribution. Regarding the Gaussian distribution pictured in figure 3.13, we can see that it is unlikely that a mobile node has to travel only a bit more than 0m or 1000m to reach a cell border but that it is much more common that a mobile node has to travel (including changing the direction) a few hundred meters to change cells.

Handover processes of type 2 are handled in the same manner, beside that the mean value and the variance of the Gaussian distribution used in this case differ. We assume that a mobile node has got to travel a mean distance of 2500m to change the responsible AAA local server. So we use a Gaussian distribution with a mean value of 2500m and a variance of 100% to represent the distance each moving mobile node has to travel to change the AAA local server area of responsibility.

3.10 Summary

As a short summary of this chapter we state that our simulation model comprises two traffic models. The first one uses Poisson distributions to simulate the interarrival of handover processes. The second one differs between three types of mobile users (mobile users inside buildings, pedestrians and car passengers) and uses Gaussian distributions for describing the car and the pedestrian speed. Further on, we can choose between the two combinations DES + MD-5 or 3DES + SHA-1 for cryptographic operations In the following chapter we show the results for each traffic model in combination with the two combinations of cryptographic algorithms.

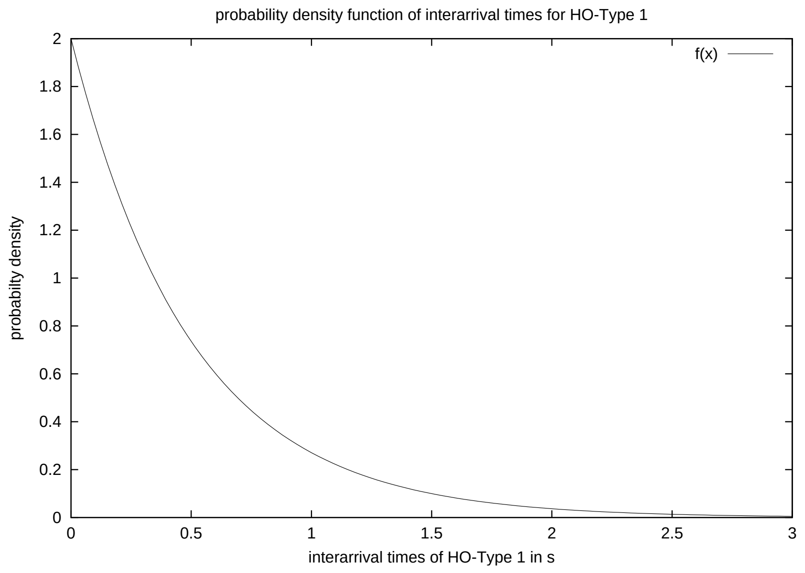

Figure 3.12: Exponentially distributed interarrival times

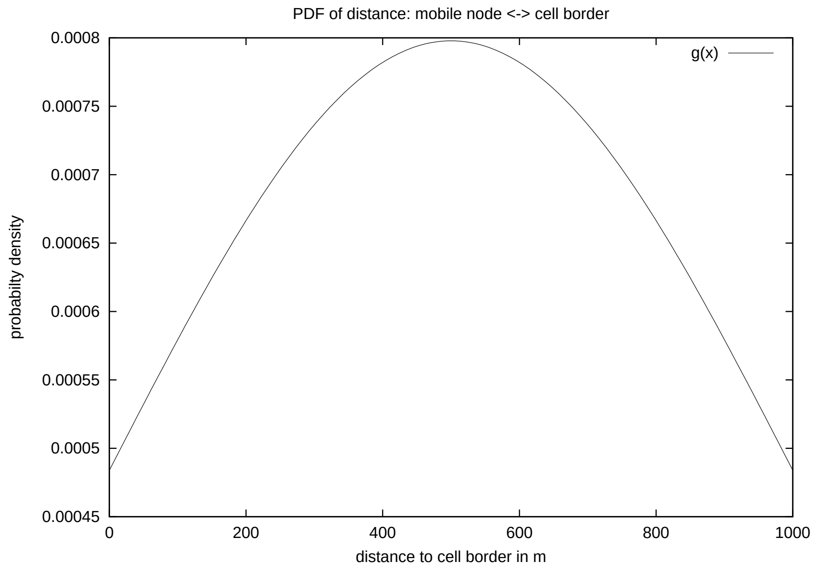

Figure 3.13: Gaussian distribution of distance: mobile node ↔ cell border

Chapter 4

Results of the Simulation Study

The main goal of the simulation model is to evaluate the performance of integrated AAA / Mobile IP authentication according to the following measures:

- Authentication delay, that is the time between the client starting to create a registration request until the completion of the registration after the last successful signature verification by the mobile node,
- Time spend for cryptographic computations of an authentication dialogue,
- Time spend for message transmission of an authentication dialogue and
- Throughput of an AAA server, that is the number of finished authentication tasks per Second.

Further more, we examine the queues that are involved in the authentication process, focusing on the lengths of the queues and on the time spend inside these queues.

4.1 Evaluation

In order to get results out of the simulation model we analyse and store different vaules at each entitiy of the model (see table 4.1). At each involved module we store the arriving time (the sending time) of each incoming (outgoing) message. With these stored values, we are able to calculate the complete authentication delay as well as the transmission time of each authentication process. Furthermore, we store the time each message spends in each visited queue. If a message is inserted into or taken out of a queue we store the length of the queue. And finally we have a look on the number of authentication tasks each AAA server executes per second.Analysing these values we get some information about the state of the model.

Table 4.1: Measured Parameters

	MOBEN	FA	AAAL	AAAH	HA
Arrival Time	x	x	x	x	x
Sending Time	x	x	x	x	x
Queue Time	-	x	x	x	x
Queue Length	-	x	x	x	x
Authentication Tasks per second	-	-	x	x	-

Table 4.2: Comparison: Traffic Model 1 $\Leftrightarrow$ Traffic Model 2

	HO-1	HO-2	HO-3
		Traffic Model 1	
Rel. percentage of HO events	80%	9%	11%
Mean interarrival times [ms]	500	4500	3600
		Traffic Model 2	
Rel. percentage of HO events	83%	14.5%	2.5%
Mean interarrival time [ms]	107.9	607.5	3600

4.2 Comparison of both traffic models

In this subsection we compare both traffic models with each other using different parameters, in order to show the differences between them. This is helpful regarding the later evaluation of the results. In table 4.2 we can see some basic differences between both models.

First we give a short summary of both traffic models:

- Traffic Model 1 [14]:
  - Each cell is assumed to handle 4000 mobile nodes and each mobile node performs:
    - 1.8 Type 1 handover operations
    - 0.2 Type 2 handover operations
    - 1 Type 3 handover operation every 4 hours
  - Mobile nodes are just modeled at the moments when an authentication operation has to be performed.
  - In every cell three independent processes generate handover operations of the three types:
    - Mean inter-arrival time of type 1 HO: 0.5 s
    - Mean inter-arrival time of type 2 HO: 4.5 s
    - Mean inter-arrival time of type 3 HO: 3.6 s

    * The inter-arrival process of handover operations is modeled with a Poisson process leading to exponentially distributed interarrival times.

- Traffic Model 2 [16]:
  - Mobile nodes are just modeled at the moments when an authentication operation has to be performed.
  - Each cell has the shape of a circle with diameter of 1 km.
  - In each cell there are:
    * 1000 pedestrians moving with a speed described through a Gaussian distribution with a mean value of 5 km/h and a variance of 30 %.
    * 1000 car passengers moving with a speed described through a Gaussian distribution with a mean value of 20 km/h and a variance of 30 %.
    * 2000 mobile nodes that reside inside buildings and do not move.
  - To invoke an authentication process of type 1, the moving mobile node has to travel a distance described through a Gaussian distribution with mean value 500 m and a variance of 100 %.
  - To invoke an authentication process of type 2, the moving mobile node has to travel a distance of 2500 m with a variance of 100 %.
  - Authentication operations of type 3 are generated in the same manner as in traffic model 1.

Before running the simulation model we determined a simulation end, that is reached when one cell has finished a certain amount of handover operations. In other words, we do not define a sharp end of the simulation, but an upper limit of handover processes. If one cell has reached the determined amount of successful handover operations the simulation is immediately stopped. Therefore, not every cell has executed the same amount of handover operations.

4.3 The influence of the traffic model on the authentication delay

Examining several runs of the simulation modul with both traffic models and using MD5/DES for the cryptographic operations, we receive the values (simulation end 320.000 handovers) of table 4.3. Running a simulation using traffic model 2, the execution needs only about a fourth of the simulation time compared to running the model with the first traffic model. Consequently we receive only about a fourth of handover operations of type 3, when using traffic model 2. Further more, we can see that the differences between the mean authentication delays for the corresponding handover types of each traffic model are infinitesimal. Thus we can conclude that the time spent inside waiting queues does only differ about the same quantity. This conclusion agrees with figure 4.1. There we can see the mean total waiting time for all handover operations of type 2 (the upper line corresponds to traffic model 2). In figures we receive the following results (mean value / standard deviation):

- Traffic Model 1: 0.96055 μs / 1.41022 ns

Table 4.3: Comparison: Traffic Model 1 ⇔ Traffic Model 2

	HO-1	HO-2	HO-3
		Traffic Model 1	
Mean authentication delay [ms]	4.93439	5.38846	5.25111
Standard deviation [ms]	3.26195e-06	1.63216e-06	2.0332e-05
		Traffic Model 2	
Mean authentication delay [ms]	4.94512	5.39541	5.25238
Standard deviation [ms]	1.34515e-05	5.41181e-06	2.92187e-06

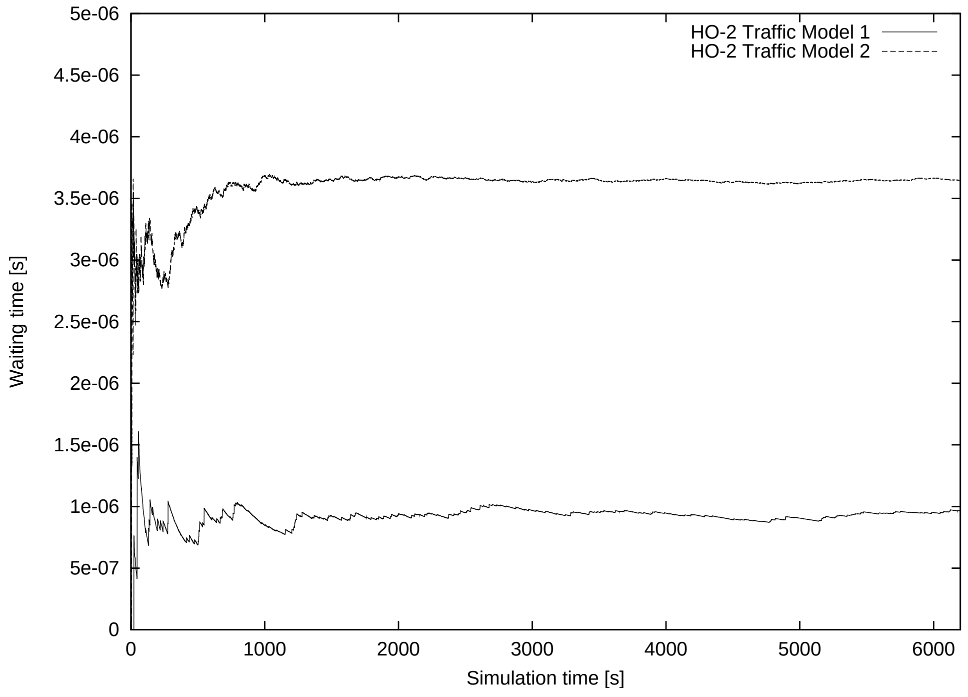

Figure 4.1: Mean waiting time of HO-2 operations

- Traffic Model 2: 3.68267 μs / 4.77504e ns.

The difference between both mean values is about 2.5 μs, which is not very much compared to the mean authentication delay over all handovers of type 2 (5.38846 ms).

Figure 4.2 depicts the number of authentication tasks per second, that $AAAL_1$ fullfills according to the traffic model. We can see that both graphs possess at the beginning a transient phase. The mean value of authentication tasks per second the AAA local server has to accomplish is 32.146 for traffic model 1 and 155.605 for traffic model 2.

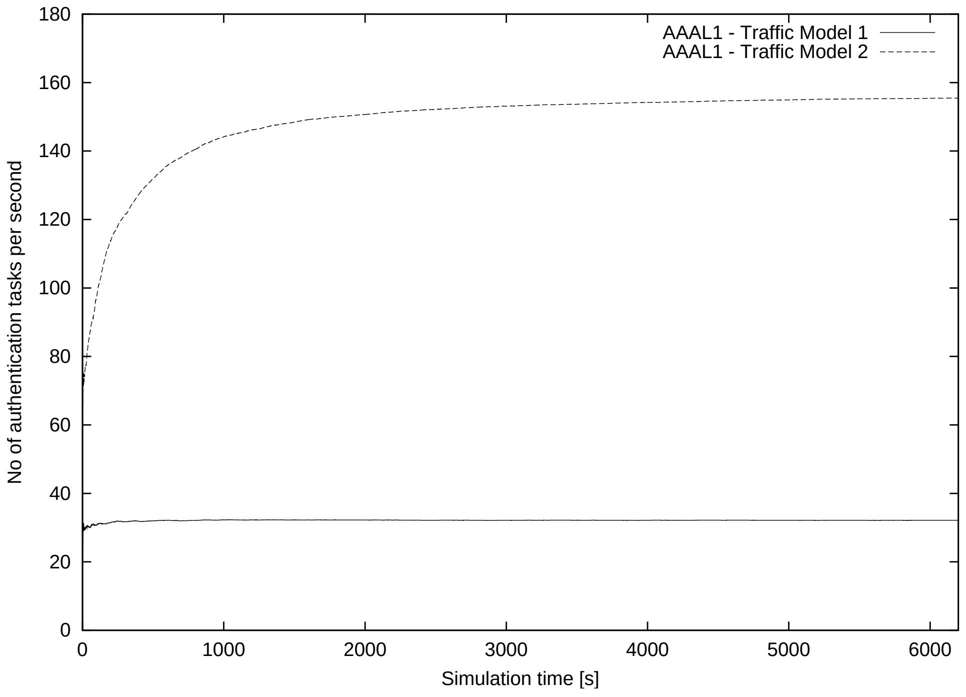

Figure 4.2: No of authentication tasks per second

To summarize this section, we state that both traffic models in combination with MD5/DES do not lead to a bottle-neck situation. The contrary is the case. During both scenarios the mean queue time over all handover operations of each type is so low that an influence on the authentication delay is not noticeable. Running the simulation model using traffic model 1, the values to be measured stay invariant after a short starting phase. On the other hand, using traffic model 2 the inital phase lasts much longer. Furthermore the usage of traffic model 2 results in a higher demand to the modules involved in handover operations of type 1 and type 2 compared to traffic model 1.

4.4 The influence of the time spent on cryptographic calculations

We have the choice between using MD5/DES and using SHA-1/3DES for cryptographic operations during a simulation run. The combination SHA-1/3DES needs more time to encrypt and sign a message than the combination MD5/DES. In this section we start with a comparison of both cryptographic proceedings. Figures 4.3, 4.4 and 4.5 depict the mean authentication delay for all three handover types and for both available cryptographic proceedings using traffic model 1. If we take a look at the figures of table 4.4, we can see that for example the difference between both cryptographic examples for handover type 1 counts 456.9e-6 s. Then in the last line we see that the additional time needed for using SHA-1/3DES instead of using MD5/DES already counts 404e-6 s. Further on, we should consider that the SHA-1 algorithm produces a message digest which is four bytes longer than the message digest produced by MD5. Thus the transmission time for an handover operation increases. Taking the wireless link between mobile node and foreign agent with a datarate of 2 Mbit/s, the additional

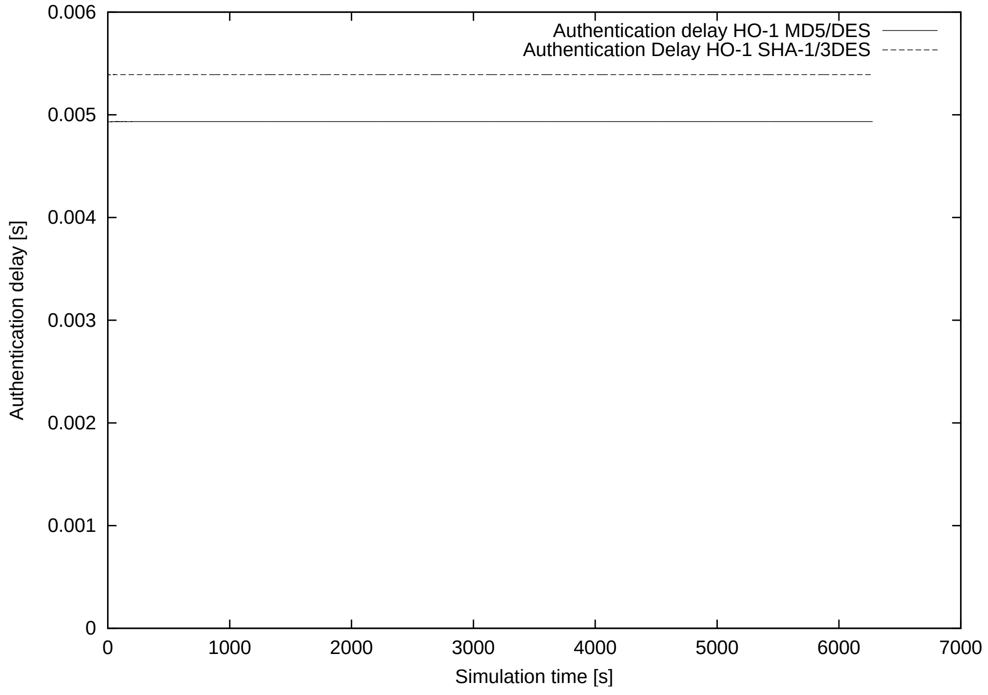

Figure 4.3: Authentication delay of HO-1 MD5/DES ⇔ SHA-1/3DES [Traffic Model1]

transmission time caused through four bytes counts 16e-6 s. If we take the additional transmission time the difference of the two mean values is explained.

Recapitulating we can say, that besides the higher processing times of SHA-1/3DES the usage of the more expensive cryptographic method results in slighly longer transmission times but far away from effectuating a bottle-neck situation. In both tested scenarios not one of the AAA servers is forced to work with full capacity.

Thus in a next step we started to multiply the processing times (based on the SHA-1/3DES processing times and message lengths) with different factors until a bottle-neck situation occured. The following results were received with the usage of traffic model 1.

Figures 4.6 and 4.7 depict the mean authentication delay of handover processes of type 1 for different multiple of the processing times for SHA-1/3DES. In table 4.5 the according values can be found. We see that the model stays stable until reaching a factor of 100. For factors bigger than 100 more and more messages are waiting to be processed inside the queues. The same behaviour is obeservable for handover operations of type 2 and type 3. Therefore, it is not possible to determine a mean value for the authentication delay for a factor of 100 or bigger.

The bottle-neck situation is caused through both AAA local servers. In figure 4.8 the average queue length of the AAAL_Request_queue and the AAAL_Answer_queue of $AAAL_2$ could be seen. The length of the displayed queues stay stable. But if we now add the graph of the AAAL_Request_queue for a factor of 100 (see figure 4.9) we see, that the system is not stable anymore. For factor 100 only the AAAL_Answer_queue is depicted in figure 4.8, and not the AAAL_Request_queue which is

Table 4.4: Comparison: MD5/DES ⇔ SHA-1/3DES

	HO-1	HO-2	HO-3
MD5/DES			
Mean authentication delay [s]	0.00493439	0.00538846	0.0525111
Standard deviation [s]	3.26195e-09	1.63216e-09	2.0332e-08
SHA-1/3DES			
Mean authentication Delay [s]	0.00539129	0.00624169	0.0546815
Standard deviation [s]	4.3599e-09	5.41181e-09	2.14126e-08
Difference [s]	456.9e-6	853.23e-6	2170.4e-6
Time needed for cryptographic operations:			
MD5/DES [s]	235e-6	458e-6	1158e-6
SHA-1/3DES [s]	639e-6	1248e-6	3225e-6
Difference [s]	404e-6	790e-6	2067e-6

Table 4.5: Different Processing Times

	Authentication Delay [s]		
factor	HO-1	HO-2	HO-3
1	0.00539149	0.00624153	0.0546817
10	0.0113637	0.0181181	0.0843644
20	0.0185353	0.0330223	0.119094
30	0.0264615	0.0505751	0.156691
40	0.0355517	0.0724441	0.1988
50	0.046709	0.102316	0.248057
60	0.0615469	0.14667	0.313088
70	0.0872176	0.233494	0.409185
80	0.152088	0.445609	0.595917
90	0.864617	2.40891	1.89257

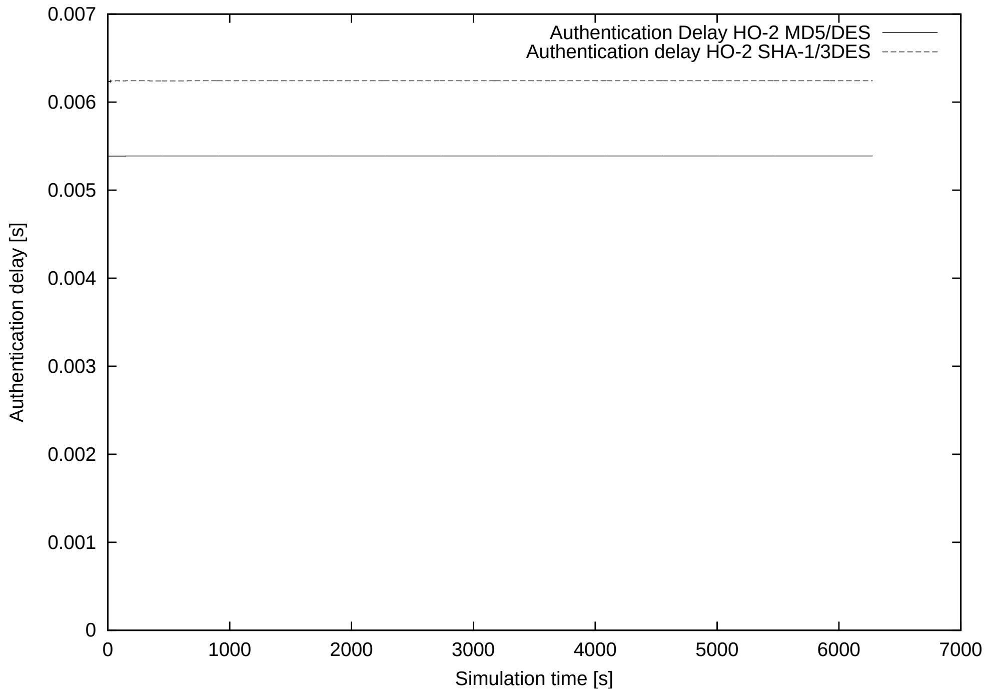

Figure 4.4: Authentication delay of HO-2 MD5/DES ⇔ SHA-1/3DES [Traffic Model 1]

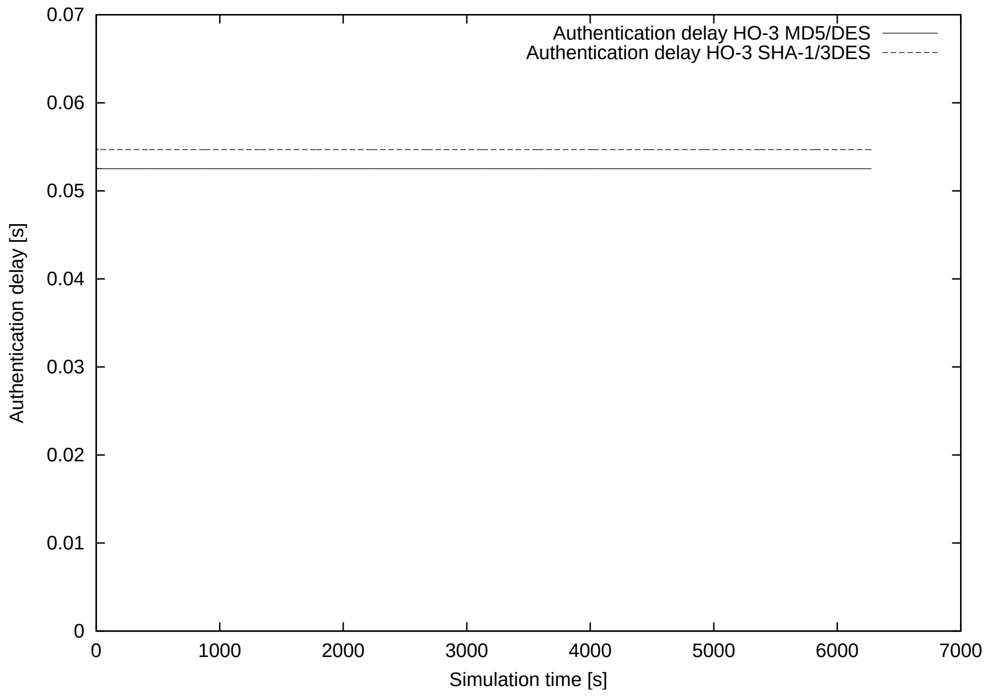

Figure 4.5: Authentication delay of HO-3 MD5/DES ⇔ SHA-1/3DES [Traffic Model 1]

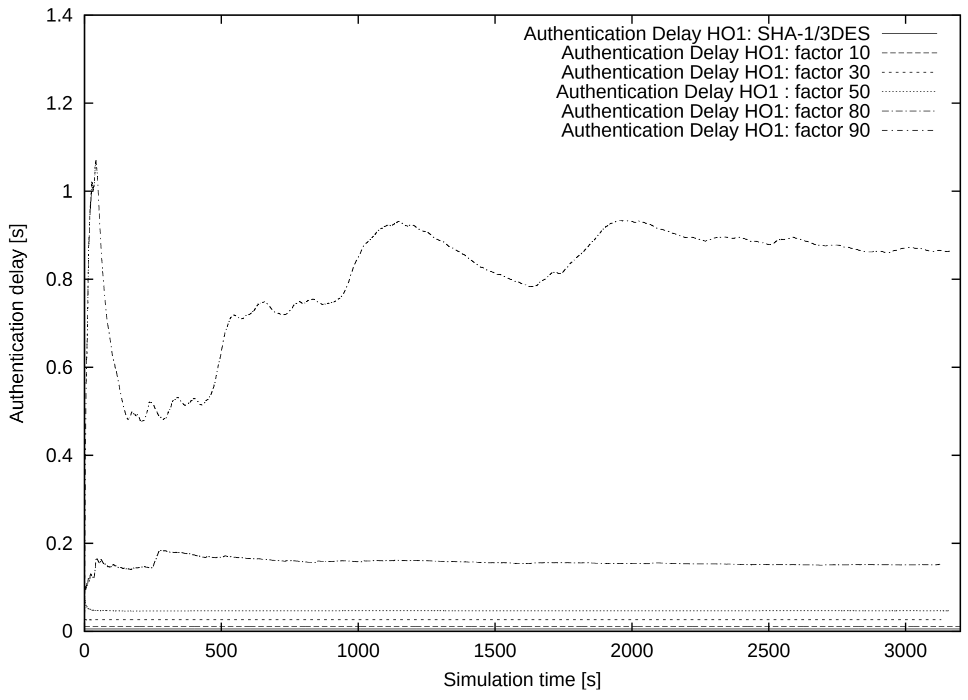

Figure 4.6: Authentication delay of HO1 for different processing times [Traffic Model 1]

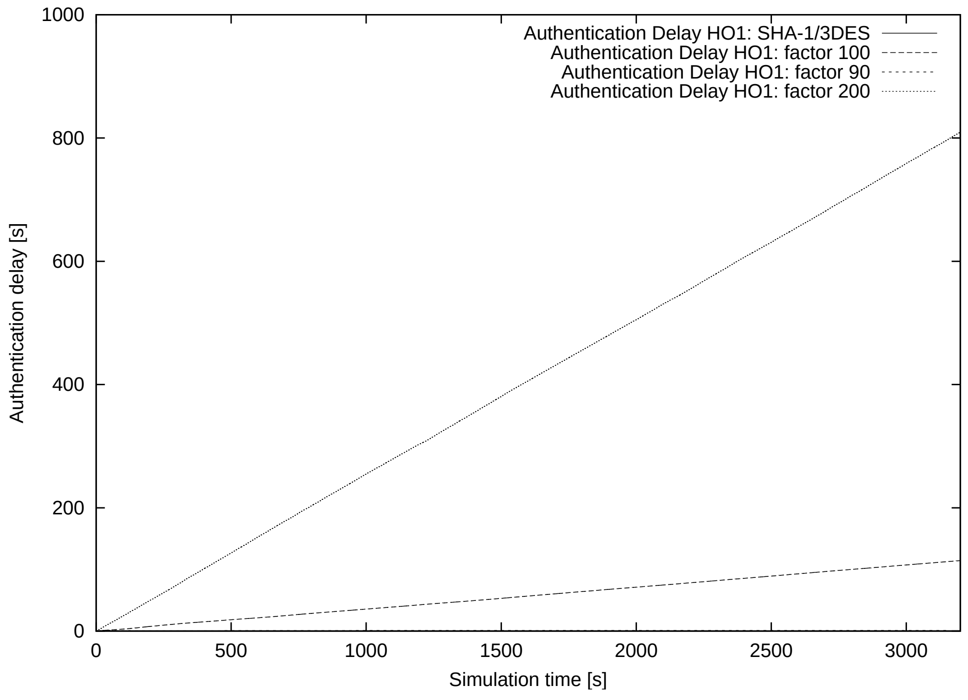

Figure 4.7: Authentication delay of HO1 for different processing times [Traffic Model 1]

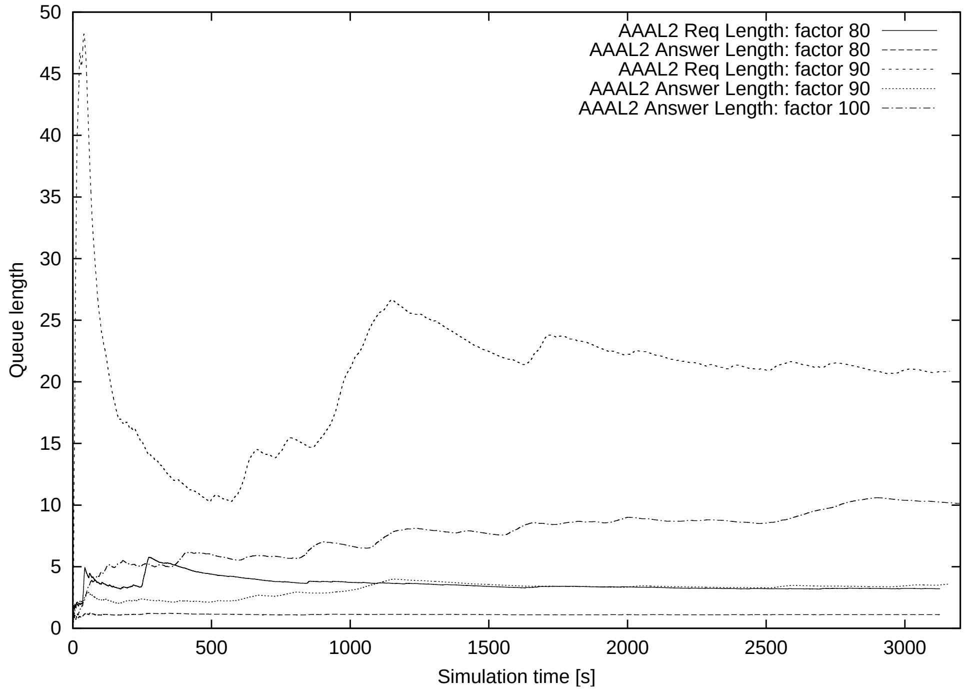

Figure 4.8: $AAAL_2$ queue lengths for different processing times [Traffic Model 1]

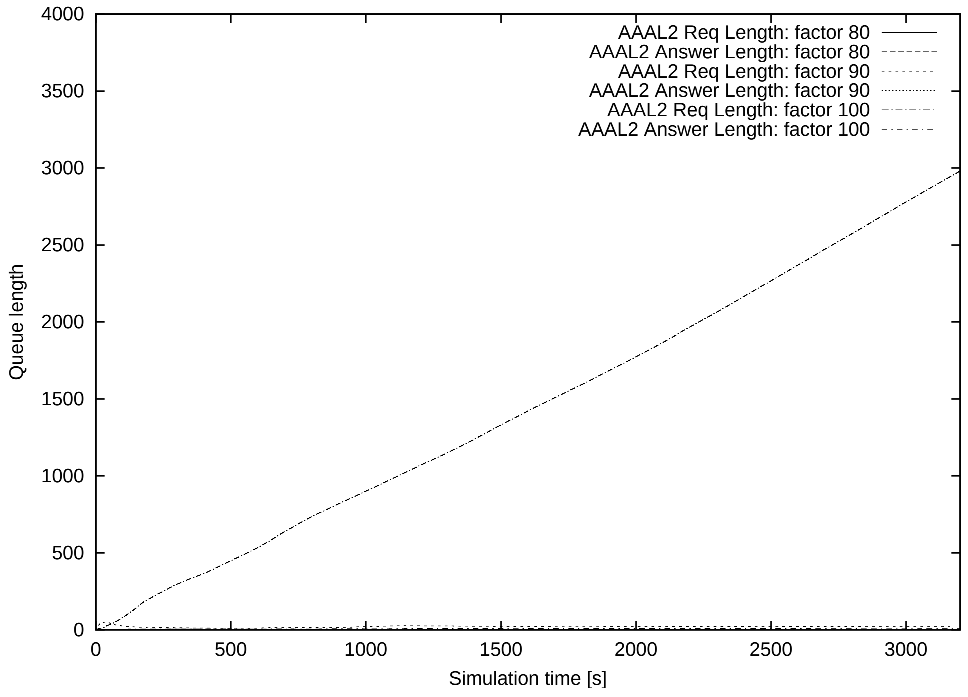

Figure 4.9: $AAAL_2$_Request_queue length for a factor of 100 [Traffic Model 1]

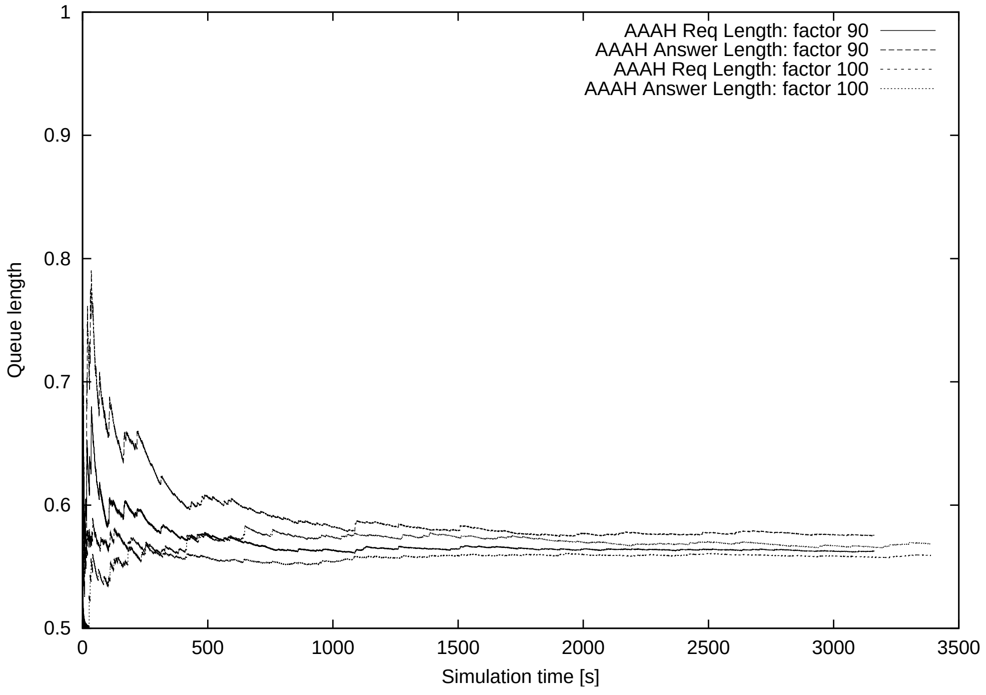

Figure 4.10: AAAH_Request_queue length for the factors of 90 and 100 [Traffic Model 1]

depicted in figure 4.9. There we see that the request queue of the local AAA server grows constantly. Furthermore, in figure 4.10 the lengths of the request queue and of the answer queue of the AAA home server are depicted. There we see that the effect of an infinitely growing request queue does not occur at the home server for a factor of 100.

Furthermore, we see that for a factor of 90 or bigger the mean authentication time for handovers of type 2 is bigger than the mean value for handovers of type 3. The reason for this is the accumulated queue time of the AAAL_Request_queue. An handover operation of type 3 resides twice in this queue compared to three times of an handover operation of type 2. We see that the AAAL_Request_queue is starting to dominate the behaviour of the system.

Remark: In a next step we ran the simulation model using traffic model 2 for traffic generation. The behaviour of the simulation model resembled the behaviour described above, except that the bottle-neck situation was reached already at a factor of 30. Running the simulation model with the processing times corresponding to the factor of 30, again the length of both AAAL_Request_queues grow constantly. Already at factor of 20 the mean authentication processing time for handover operations of type 2 is bigger than the mean value of handover operations of type 3. Using traffic model 2 for the traffic generation, the system earlier gets unstable. The enlargement of the processing times while using traffic model 2, has an obvious effect on the system. This is due to the higher arrival rate of handover requests compared to traffic model 1.

To summerize this chapter, we state that the simulation model starts to get instable at a factor of 100 for traffic model 1 and at a factor of 30 for traffic model 2.

Table 4.6: Different Processing Times - Traffic Model 2

	Authentication Delay [s]		
factor	HO-1	HO-2	HO-3
1	0.00540794	0.00626029	0.0547
10	0.01247	0.0221489	0.0870705
20	0.103446	0.340098	0.327097

Remark: In figures 4.6 and 4.8 we can see a strong overshoot at the beginning of the transient phase of the graph for the factor of 90. This is due to the manner in which initialize the simulation. In each cell we have one generator for each handover process. At the beginning of the simulation we schedule at once 20 handover processes of type 1, type 2 and type 3. So there are 60 scheduled events in total. Having now high cryptographic processing times, the queue length (authentication delay) rises quickly at the beginning of a simulation run. But if the cryptographic processing times are not too high, the system will level off.

4.5 Influence of the distance between home and local domain

Designing our model we assumed that between AAAL and AAAH not the datarate is the dominating factor for the transmission time but the end to end delay from the local domain to the home domain. Hence we examined the influence of different end to end delays on the behaviour of the model. Therefore, we ran the model using both traffic models and both variants of cryptographic processing. Figure 4.11 depicts the results for the usage of traffic model 1 in combination with MD5/DES for cryptographic operations. The figure shows the mean authentication delay of handover processes of type-3 for three different transmission delays between the local and the home area over the simulation time.

We can see that the starting phase behaviour does not differ from the further run of the curve. For this scenario we received the following authentication delay values:

- 20 ms: mean = 0.052511 s; standard deviation = 2.03320e-08 s;
- 50 ms: mean = 0.112511 s; standard deviation = 2.01804e-08 s;
- 100 ms: mean = 0.212511 s; standard deviation = 1.99212e-08 s.

If we subtract the mean value of case 20 ms from the mean value of case 50 ms, we receive a difference of 0.06 s which equals exactly two times the additional transmission delay of 30 ms. It is the same with the third case (100 ms). So if we look at the figures of table 4.7, we can see that in all three cases the mean waiting time is very small and the differences between them are most minimal. Further on, if we compare the mean number of authentication tasks each AAA server has accomplished per second, no distinctive features could be detected. For example we receive a mean value of 5.6 authentication tasks per second for the AAAH with a standard deviation of 5.6. Thus we can conclude that the

Table 4.7: Queue Times ⇔ Transmission Delay AAAL-AAAH

Queue	20 ms	50 ms	100 ms
Mean AAAH_Request_queue time [s]	1.80876e-07	1.31068e-07	1.59301e-07
Standard deviation [s]	2.26831e-11	1.53807e-11	2.19202e-11
Mean AAAL_Answer_queue time [s]	1.93764e-07	2.05934e-07	1.98173e-077
Standard deviation [s]	1.60362e-11	1.8572e-11	1.59238e-11

AAAH server is not running at full capacity, or in other words both queues of the AAAH server are empty from time to time.

We can state that for this scenario the influence of the transmission delay between local and home domain on the system is not noticeable.

In a next step we examined, what happens if we enlarge the cryptographic processing times. Figures 4.12 and 4.13 depict the mean authentication delay for handover operations of type 1 for an end to end delay of 50 ms between home and local domain and for different cryptographic processing times (based on SHA-1/3DES). Again we see that till a factor of 90 the system is stable and if we look at the graph for the factor 100, we see that the mean authentication delay starts to grow constantly.

So we can state, that an influence of the end to end delay between local and home domain on the system just increases the overall authentication delay experienced by a mobile node but has no further noticeable effect on the work load of the system.

4.6 Influence of the amount of mobile nodes per cell

In this section we examine the behaviour of the simulation model while increasing the number of mobile nodes per cell. In figure 4.14 we see the mean authentication delay for handover operations of type 2 for 4000, 8000, 16000, 32000 and 64000 mobile nodes per cell using traffic model 2. We see that the mean authentication delay increases slightly but that the system even can handle 64000 mobile nodes. 64000 mobile nodes in an area of 0.785 km^2 is already an enormous amount. The results for handovers of type 1 and 3 are similar. We focused our observations on handover processes of type 2, as the request queues of both AAA local servers are the most sensitve parts of the simulation model and handovers of type 2 reside three times inside these request queues. Thus we would early remark an effect on the system.

4.7 Conclusion

We have seen that neither the usage of MD5/DES nor the usage of SHA-1/3DES causes any problems. In both scenarios the queues of the AAA servers still get empty from time to time. Then in a next step we increased the cryptographic processing times taking the SHA-1/3DES processing times as base element. At a factor of 100 for traffic model 1 the simulation model gets unstable, due to a constantly

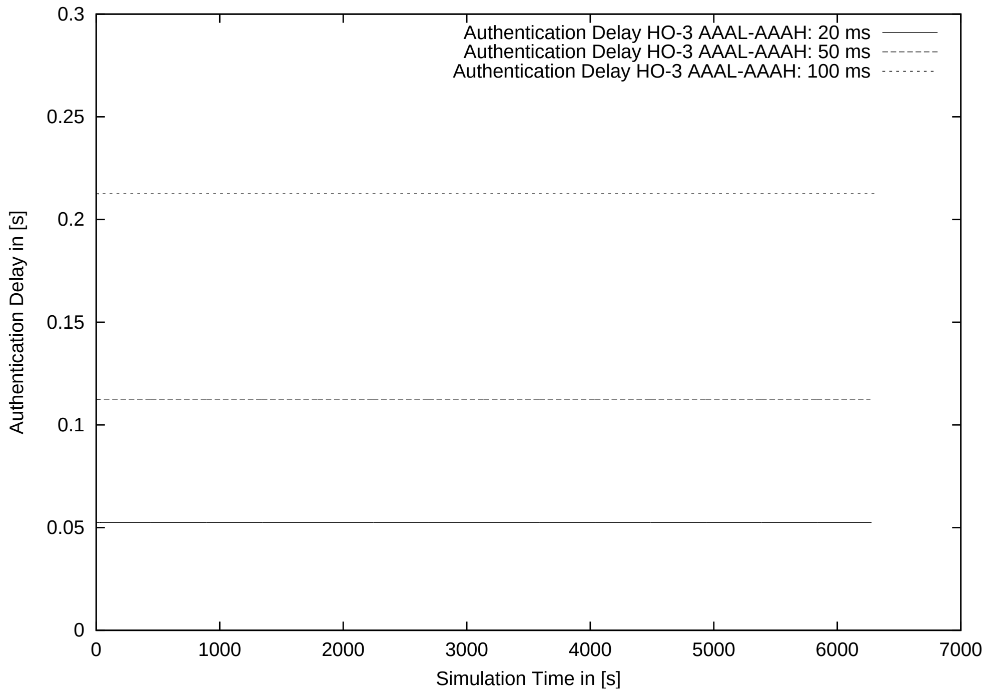

Figure 4.11: Authentication Delay of HO-3 ⇔ Delay AAAL-AAAH

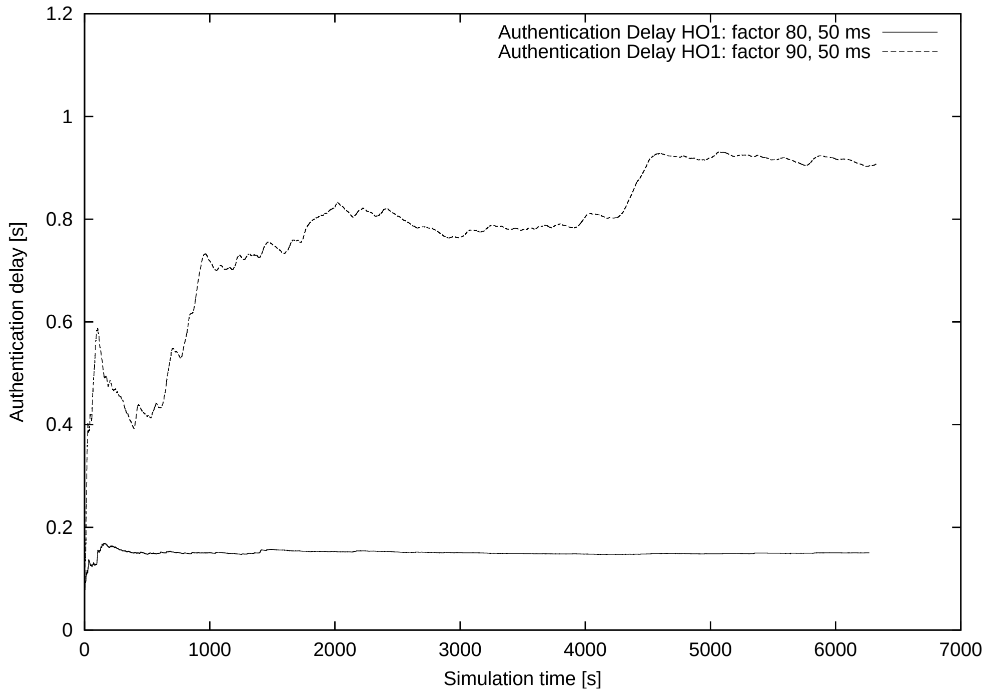

Figure 4.12: Authentication Delay of HO-1 ⇔ Delay AAAL-AAAH

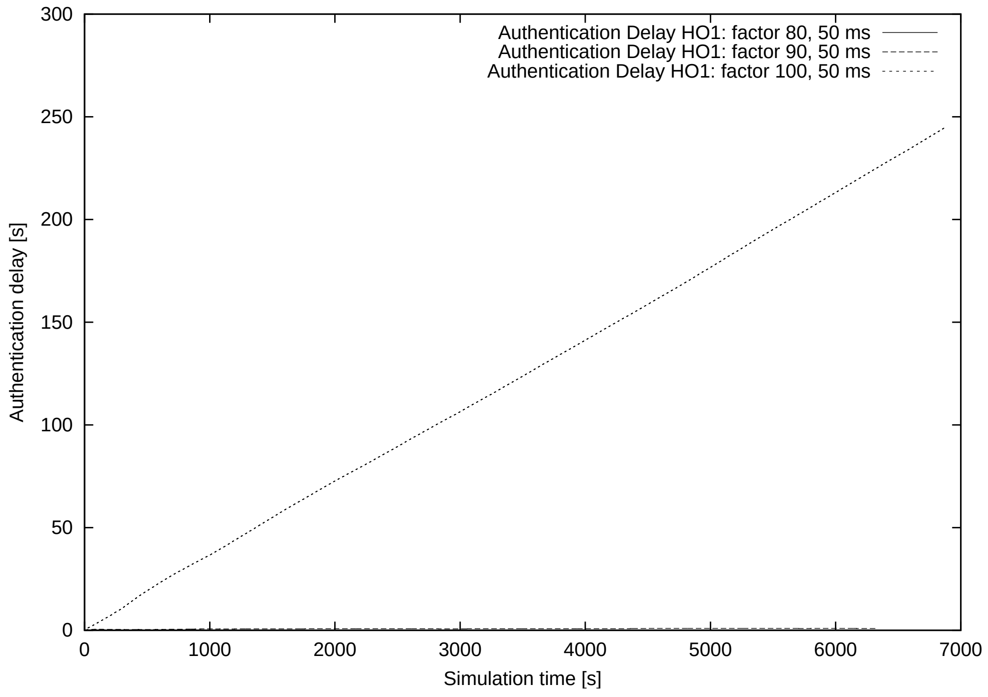

Figure 4.13: Authentication Delay of HO-1 ⇔ Delay AAAL-AAAH

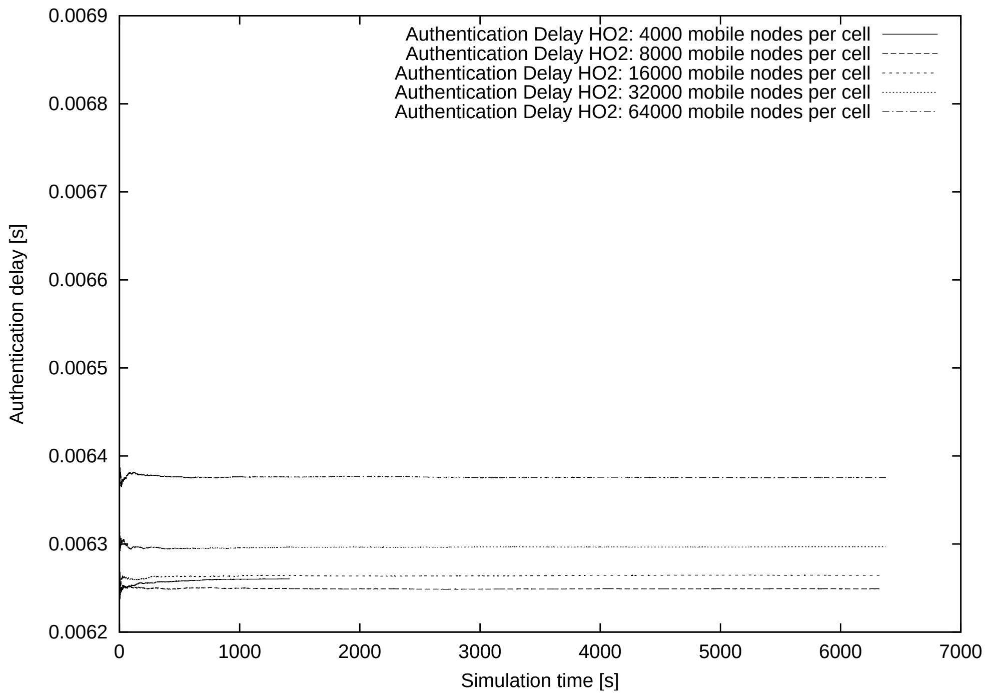

Figure 4.14: Authentication Delay of HO-2 ⇔ Mobile Nodes per Cell

growing AAAL_Request_queue. For a factor of 100 or bigger the incoming rate of messages arriving at the AAA local server is higher than the outgoing rate. Thus the request queue starts to grow infinitely. Using traffic model 2 for traffic generation, this situation already occurs at a factor of 30.

Following this we examined the influence of the transmission delay between home and local domain on the simulation model. We ran simulations using different end to end delays between AAAL and AAAH. But no influence on the system was noticeable, even when we increased the cryptographic processing, except the bigger authentication delay corresponding twice the end-to-end delay between AAAL and AAAH.

Finally, we examined the influence of the amount of mobile nodes per cell. We doubled the amount of mobile nodes per cell, starting at 4000 up to 64000. The mean authentication delay grew slightly but far away from causing a bottle-neck situation. The system reacts more or less insensitive to a higher amount of mobile nodes inside a cell or to a bigger distance between local and home domain.

To summarize this section, we can state that the AAA local servers are the modules which may cause a bottle-neck situation, if the processing times are high. For example, making use of asymmetric cryptography in the authentication dialogue may already result in processing times that are high enough to cause overload situations.

Chapter 5

Summary and Conclusions

This report has studied performance issues of the IETF's current approach to authenticate mobile nodes during Mobile IP registration on the basis of the *Authentication, Authorization and Accounting (AAA)* protocol *Diameter*.

The Diameter protocol, its message formats and the interworking with the Mobile IP registration procedure have been examined in detail in order to identify parameters for the performance study. As a result three types of handover operations and the lengths of the messages to be exchanged during these operations have been identified: a *type-1 handover* occurs when a mobile node changes into a radio cell which is handled by a different foreign agent which itself is connected to the same local AAA server. All draft versions of the Mobile IP application of the Diameter protocol up to version draft-ietf-aaa-diameter-mobileip-07.txt [10] include support for an optimized authentication dialogue which re-uses previously established session keys between the foreign agent and the mobile node, as well as between the foreign agent and the home agent. A handover of *type-2* occurs when a mobile node moves to a radio cell, that is managed by a foreign agent which is connected to a different local AAA server, but both the old and the new local AAA servers belong to the same administrative domain (e.g. network operator). While this case is not explicitely specified in the Diameter standardization drafts, it is a natural extension to the type-1 handover and it reflects current practice from other cellular networks, e.g. GSM. The third type of handover operations occurs when a mobile node registers for the first time in a foreign network, changes into a foreign network of another administrative domain or when a security context previously established with a visited foreign network expires. This handover type requires involvement of the AAA server of the mobile node's home network resulting in an increased delay to complete the handover due to one Internet roundtrip time. Thus, by minimizing the number of type-3 handover operations a better overall performance can be experienced by mobile users.

However, it has to be stated, that during the course of ongoing discussions in the IETF's AAA working group, the support for optimization of local handover operations has been eliminated from the current set of Diameter specifications. The reason for this is mainly, that there are a couple of open questions of local handover optimization for Mobile IP that need to be resolved before Diameter can support this feature, and that the Diameter specification has to be finished in order to get RFC status for it. As a consequence, the current set of Mobile IP and Diameter specifications requires a full authentication also involving the home AAA server for every handover performed by a mobile node.

Nevertheless, it is expected that support for local handover optimization will be integrated into the Diameter specifications as soon as the remaining issues are resolved in the Mobile IP standards. Therefore, we examined the performance of the three types of handover operations when deployed in a representative network configuration with two different mobility models.

The results of the simulation study show, that:

1. The delay experienced by a mobile node in case of a full authentication dialogue involving entities of the mobile node's home network is largely determined by the end-to-end delay between the foreign and the home network.
2. The workload of AAA servers remains moderate in case of a load- and mobility model inspired by established values of GSM networks [14] as well as in case of a more progressive mobility model [16].
3. The workload of AAA servers grows infinitly under both mobility models if cryptographic algorithms are used that require about 100 (30) times the processing capabilities of algorithms currently envisaged by the IETF (cryptographic hash functions and symmetric encryption).

An important consequence of the third finding is that the use of asymmetric cryptography would possibly lead to overload situations under the investigated conditions, as those operations often require processing capabilities in this range.

Bibliography

[1] B. Aboba and M. Beadles. The Network Access Identifier. RFC 2486, Jan 1999.

[2] B. Aboba and G. Zorn. Criteria for Evaluating Roaming Protocols. RFC 2477, Jan 1999.

[3] R. Atkinson and S. Kent. IP Encapsulating Security Payload (ESP). RFC 2406, 1998.

[4] R. Atkinson and S. Kent. Security Architecture for the Internet Protocol. RFC 2401, 1998.

[5] M. Beadles and D. Mitton. Criteria for Evaluating Network Access Server Protocols. Internet Draft, work in progress, June 2000. `http://www.ietf.org/internet-drafts/draft-ietf-nasreq-criteria-05.txt`.

[6] P. Calhoun, H. Akhtar, J. Arkko, E. Guttman, A. Rubens, and G. Zorn. Diameter Base Protocol. Internet Draft, work in progress, June 2001. `http://search.ietf.org/internet-drafts/draft-ietf-aaa-diameter-06.txt`.

[7] P. Calhoun, S. Farrell, and W. Bulley. Diameter CMS Security Application. Internet Draft, work in progress, June 2001. `http://search.ietf.org/internet-drafts/draft-ietf-aaa-diameter-cms-sec-%01.txt`.

[8] P. R. Calhoun and C. E. Perkins. Mobile IPv4 Challenge/Response Extensions. RFC 3012, Nov 2000.

[9] P. R. Calhoun and C. E. Perkins. Diameter Mobile IPv4 Application. Internet Draft, work in progress, June 2001. `http://search.ietf.org/internet-drafts/draft-ietf-aaa-diameter-mobileip%-06.txt`.

[10] P. R. Calhoun and C. E. Perkins. Diameter Mobile IPv4 Application. Internet Draft, work in progress, July 2001. `http://search.ietf.org/internet-drafts/draft-ietf-aaa-diameter-mobileip%-07.txt`.

[11] D. Crocker and P. Overell. Augmented BNF for Syntax Specifications: ABNF. RFC 2234, Nov 1997.

[12] T. Dierks and C. Allen. *The TLS Protocol Version 1.0*, November 1997. draft-ietf-tls-protocol-05.txt.

[13] S. Glass, T. Hiller, S. Jacobs, and C. Perkins. Mobile IP Authentication, Authorization, and Accounting Requirements. Internet RFC 2977, 2000.

[14] W. Hahn. IP-based Mobile Networks: Evaluation of Mobility Mechanisms for IP-based Networks. internal paper of the cooperation between Siemens ICM MC ST and Technical University of Berlin, 2001.

[15] T. Hiller and al. CDMA2000 Wireless Data Requirements for AAA. Internet Draft, work in progress, Sep 2000. `http://search.ietf.org/internet-drafts/draft-hiller-cdma2000-aaa-02.txt%`.

[16] J.G. Markoulidakis, G.L. Lyberopoulos, and M.E. Anagnostou. Traffic Model for Third Generation Cellular Mobile Telecommunication Systems. To appear in Intern. Journal of Wireless Information Networks. `http://www.telecom.ntua.gr/~ymark/`.

[17] C. Perkins. IP Mobility Support. Internet RFC 2002, 1996.

[18] C. Perkins. *Mobile IP Design Principles and Practices*. Number ISBN: 0-201-63469. Addison-Wesley Longman, Reading, MA, USA, 1998.

[19] C. Perkins and P. Calhoun. AAA Registration Keys for Mobile IP. Internet Draft, work in progress, May 2001. `http://www.ietf.org/internet-drafts/draft-ietf-mobileip-aaa-key-05.txt`.

[20] C. Perkins and P. Calhoun. Generalized Key Distribution Extensions for Mobile IP. Internet Draft, work in progress, July 2001. `http://www.ietf.org/internet-drafts/draft-ietf-mobileip-gen-key-00.txt`.

[21] G. Schäfer, A. Festag, and H. Karl. Current Approaches to Authentication in Wireless and Mobile Communication Networks. TKN Technical Report TKN-01-002, Mar 2001.

[22] J. Solomon. *Mobile IP: The Internet unplugged*. Number ISBN: 0-13-856246-6. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1998.