

MASTER

WLAN-GPRS roaming : based on mobile IP (v4)

van Sebille, T.C.

Award date:
2002

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain



WLAN – GPRS Roaming

Based on Mobile IP (v4)

Master's thesis, public version

T.C. van Sebille

author: T.C. van Sebille (415837)
Maastricht, December 2001 – October 2002

Vodafone in the Netherlands
Eindhoven University of Technology

TU/e technische universiteit eindhoven

author:

Tom van Seville

T.C.v.Seville@tue.nl

id.nr. 415837

supervisors:

Prof.ir. A.M.J. Koonen (TU/e)

Ir. J.J.B. Kwaaitaal (TU/e)

R. Crutzen (Vodafone NL)

Ing. A.B.P. Jongen (manager CDS, Vodafone NL)

Eindhoven University of Technology (TU/e)

Department of Electrical Engineering

group: Telecommunications Technology and Electromagnetism (TTE)

chairs: Electro-Optical Communications (ECO)/ Radio Communications (ECR)

The assignment was defined by and fulfilled at Vodafone in the Netherlands, Maastricht

department: Network/IT

Technology Development (TD)

Customer Data Solutions (CDS)

© T.C. van Seville October 2002

All rights are reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

Summary

This report describes my graduation work which I performed at Vodafone in the Netherlands, Maastricht. The project was under supervision of Ing. A.B.P. Jongen and R. Crutzen of the section Customer Data Solutions (CDS), which is part of Vodafone's Network/IT department. At the Eindhoven University of Technology (TU/e), Prof.ir. A.M.J. Koonen and Ir. J.J.B. Kwaaitaal of the section Telecommunication Technology and Electromagnetism at the faculty of Electrical Engineering supervised my graduation project. The assignment was to investigate how Vodafone's General Packet Radio Service (GPRS) network can be combined with new access technologies, like Wireless LAN (WLAN) and later the Universal Mobile Telecommunication System (UMTS) network. Points of special interest were the possibility of seamless handoffs and aspects regarding Authentication, Authorization and Accounting (AAA). The intention was to give a demonstration at the end.

A literature study pointed out that the right technology to use is Mobile IP. With this technology it is possible to make seamless handoffs at the IP layer (network layer) and maintain IP connectivity. Mobile IP is specified as an open standard by the Internet Engineering Task Force (IETF), which has developed many extensions regarding security issues and problems with some specific network implementations. I studied many of these additional specifications and drafts, in order to give a detailed recommendation how Vodafone NL could implement Mobile IP into their network. Especially the possibility to use the conventional AAA Internet infrastructure and the Subscriber Identity Module (SIM) with the GSM authentication infrastructure for creating security associations is very valuable. Furthermore, I created my own Mobile IP home network at the TU/e. In Maastricht I had several access technologies, such as WLAN, GPRS and Ethernet, which could be used as visited networks to demonstrate the seamless handoff handled by Mobile IP. I wrote a script that uses the Mobile IP software to make handoffs. This script contains an algorithm to choose the preferred interface to use.

The power of Mobile IP is that it is independent of the lower layer access technologies, such as GPRS, WLAN and UMTS. It enables higher layer protocols to maintain their session during a handoff, which results in a high quality mobility. Furthermore, the demonstration with the intelligent handoff algorithm proved the success of Mobile IP.

Tom van Sebille
© 22nd October 2002

Word of Gratitude

I would like to thank the following departments and persons for their support and cooperation in my graduation project. First of all, I would like to thank all my supervisors and the complete department Customer Data Solutions for the supervision, the opportunity to get to know Vodafone NL and the friendly cooperation. Furthermore, my demonstration was not possible without the support, input and cooperation of the Core IP department, High Level Design and Detailed Design. Of course I would like to thank SPACELABS (Society Pursuing Achievements in Communications Embracing Linux Architectural Basis, <http://www.spacelabs.nl>) for talking me into Linux and their support. Last but not least, I would like to thank my family and girlfriend Kim for their support at home.

Tom.

Table of contents

Introduction	5
Chapter 1 Mobile IP (v4).....	6
1.1 Introduction.....	6
1.2 Mobile IP Mechanism.....	8
Case 1: Foreign Agent Care-Of Address	9
Case 2: Co-located Care-Of Address	11
1.3 Mobile IP Details.....	13
1.3.1 Agent Advertisements and Solicitations.....	13
1.3.2 Registration Messages	14
1.3.3 Mobile IP Registration.....	16
Simultaneous Bindings.....	17
1.3.4 Tunneling Modes	18
1.3.5 ARP Issues	21
1.4 Specific Mobile IP Problems and Solutions	22
1.4.1 Mobile Node NAI Extension	22
1.4.2 NAT Traversal	23
1.4.3 Dynamic Home Agent Discovery.....	25
1.4.4 Route Optimization.....	26
1.4.5 Dynamic Key Distribution.....	26
1.4.6 Mobile Network with a Mobile Router.....	29
1.5 Commercial Implementations.....	31
1.6 Mobile IP in the Future.....	32
Chapter 2 Mobile IP Demonstration.....	33
2.1 Introduction.....	33
2.2 Software.....	33
2.2.1 HUT Dynamics Overview	34
2.3 Demo Set-Up.....	34
2.3.1 Initial Set-Up.....	35
Problem	37
2.3.2 Final Set-Up.....	38
2.3.3 HUT Dynamics Experiences.....	40
Dynamics MIP Routing.....	40
Dynamics MIP Control Commands	42
Complications	43
2.4 Handoff Script	44
2.4.1 Priority Interface	50
2.4.2 Issues.....	50
2.5 Future Work.....	51
Chapter 3 Conclusion and Recommendation.....	53
3.1 Conclusion.....	53
Appendix A List of References.....	55
Appendix B List of Acronyms.....	58

Appendix C	List of Figures.....	60
Appendix D	Configuration of Dynamics MIP Software.....	61
	GPRS Connection	61
	Initial Demo Set-Up	62
	Final Demo Set-Up	70
Appendix E	Handoff Script.....	78
Appendix F	Test Cases	88
	Proxy ARP with IP Forwarding	88
	Routing at MN Using Dynamics.....	89
Appendix G	Outdated Internet Drafts	91
	Route Optimization in Mobile IP	91
	AAA Registration Keys for Mobile IP.....	97
Appendix H	GPRS Overview	104
Appendix I	IP Packet Sizes	106
	Packet Length Distributions.....	106
Appendix J	Assignment.....	110
	Original Assignment	110
	Extended Assignment.....	111
Notes	112

Introduction

This report describes the research activities performed by T.C. van Seville for his master's degree at the faculty of Electrical Engineering of the Eindhoven University of Technology (TU/e). The assignment was formulated by Vodafone in the Netherlands (Vodafone NL). Vodafone is one of the leading mobile network operators in the Netherlands and has more than 3 million subscribers. It is part of the worldwide Vodafone Group, which has over 103 million subscribers in 28 countries. The graduation project was performed at the department of Network/IT, Technology Development, section Customer Data Solutions of Vodafone NL in Maastricht.

The focus of the research was aimed at seamless handoffs¹ and Authentication, Authorization, and Accounting (AAA) issues. Handoff procedures can be divided in the following categories.

- With respect to the initiating device:
 - Mobile Assisted HandOff (MAHO), like in GSM;
 - Mobile Controlled HandOff (MCHO);
 - Network Controlled HandOff (NCHO).
- With respect to the protocol layer:
 - Inter-system or inter-tech handover:
This is a layer 3 handover between different networks, which results in macro mobility;
 - Intra-system handover:
This is a layer 2 handover between different cells of the same network, which results in micro mobility.

More information can be found in [1, 2].

After studying some IEEE articles [3, 4], it became clear that Mobile IP was the major technology to enable layer 3 (IP) inter-system handoffs in hybrid networks. The power of Mobile IP is that it is independent of lower layer access technologies. It was decided to concentrate on this technology and build a demonstration set-up if possible. The demo should provide an increased feeling about the possibilities of Mobile IP. The original assignment and an extension are given in Appendix J.

In Chapter 1 the Mobile IP principles and specifications are discussed. It also handles some interesting additions, of which most are still in development. Chapter 2 discusses the demonstration set-up. Also an intelligent handoff script is discussed. Finally Chapter 3 gives the conclusions and recommendations for Vodafone NL. The appendices give much extra information, like all configuration files of the demo and some test scenarios.

¹ The terms handoff and handover are both used for the same concept.

Chapter 1 Mobile IP (v4)

1.1 Introduction

In the early 90s the Internet Engineering Task Force (IETF²) started a new Working Group “IP Routing for Wireless/Mobile Hosts (Mobile IP)”³ in the Internet Area, which was going to develop a mechanism which makes it possible for mobile users to always have the same IP address, wherever they are attached to the Internet: IP mobility. The purpose was that mobile users could roam between various networks while staying online and being able to maintain their IP connectivity. The various networks could be based on different access technologies, so the mechanism had to be independent of the underlying link layer technology.

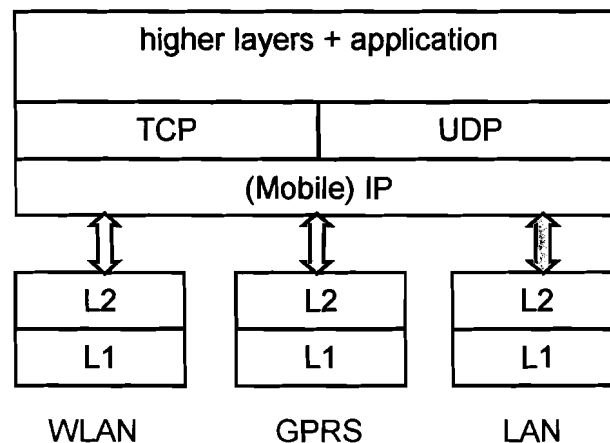


Figure 1-1 Protocol stack of a mobile user with three interfaces

In October 1996 the first RFC 2002 “IP Mobility Support” [5] was published that presented a new protocol, called Mobile IP for IPv4, which actually is an extension to the regular IP protocol, see Figure 1-1. In January 2002 RFC 3220 “IP Mobility Support for IPv4” [6, 7] obsoleted RFC 2002. The status of RFC 3220 is “Proposed Standard”, which means that it has entered the process to become an “Internet Standard”, but still is immature and may possibly evolve further. Parallel to this, the Working Group is working out some specific problems they came across. Recommended solutions are published in drafts which can be found on their website. Some of these problems will be discussed later in this chapter.

For IPv6 the working group has developed a similar mechanism, which was published as a draft “Mobility Support inIPv6” [8]. The main difference is that here it will be integrated in the IPv6 standard, while for IPv4 the mobility support is considered as optional. The mobile user needs specific client software to use Mobile IP.

The need for mobility support can be found in services that Internet users want to exploit when they are roaming on different networks. If you are downloading a large file or watching/listening to a media stream, you would not like the session to be disrupted when you’re moving between different (sub)networks. The same holds for a Voice-over-IP (VoIP) session: you do not want to set up your call again every time you change (sub)network.

² IETF: <http://www.ietf.org>

³ Mobile IP Working Group: <http://www.ietf.org/html.charters/mobileip-charter.html>

Another example is a mobile node that acts as a server which always has to be addressable at the same IP address for its clients. So, in general the use can be split up in two primitives:

1. availability (mobile servers must be addressable at a never-changing IP address)
2. continuity (maintain IP connectivity while changing point of attachment)

Especially for users with wireless interfaces, this results in an improved mobility.

Mobile IP is a macro mobility solution. It can handle IP handoffs between different access technologies or between different layer 3 (sub)networks. It is not suited for layer 2 cell handovers, which is called micro mobility.

This chapter describes the Mobile IP protocol according to RFC 3220. First a global description will be given in section 1.2 following two example cases. Then section 1.3 will explain some specific aspects in more detail. Finally, a few extra drafts, which handle specific problems, will be discussed in section 1.4. Security and network architectural issues are emphasized.

1.2 Mobile IP Mechanism

In IP networks, routing is based on stationary IP addresses. A node on a network is routable with normal IP routing by the IP address it is assigned on the network. The IP address is built up of two parts: the network number (or network prefix) and the host number (or local address). IP packets sent from node A to node B on a different network are routed towards the correct network based on the network number, which is the first part of the IP address. When a packet arrives at the network of node B, it is routed towards node B based on the host number, the last part of the IP address.

When node B is traveling and connected to the Internet via a different network than its home network, it may get an IP address, which is different from its Home Address. Mobile IP is the mechanism that enables other nodes to reach node B by its original home IP address of node B. Furthermore, with Mobile IP node B can use its original home IP address as source address. So Mobile IP provides transparent IP routing for mobile nodes, independent of their point of attachment to the Internet.

Conform RFC 3220, some definitions are given below, among which are the three principal components of Mobile IP: Home Agent, Foreign Agent and Mobile Node.

- **Mobile Node:**
The Mobile Node (MN) is the node (i.e. host or router) that roams between different networks and is able to use the same IP address. The MN may change its point of attachment to the Internet, without changing its constant IP address, called Home Address.
- **Home Network:**
The Home Network (HN) of an MN is the network to which IP packets with the Home Address as destination are routed using normal IP routing.
- **Home Address:**
The Home Address of an MN is a constant IP address assigned to the MN. It remains unchanged and belongs to the (sub)network of the HN. The MN will always use this address as source address, even when it is roaming.
- **Home Agent:**
The Home Agent (HA) of an MN is a router⁴ on the HN. It captures IP datagrams destined for the MN and tunnels them towards the MN when the MN is away from the HN. Furthermore, the HA performs some authentication and administration functions.
- **Correspondent Node:**
A Correspondent Node (CN) is an arbitrary node with which the MN is communicating.
- **Foreign Network:**
A Foreign Network (FN) is a network other than the HN of the MN.
- **Foreign Agent:**
A Foreign Agent (FA) is a router⁵ on an FN which provides routing services for registered MNs. The FA detunnels IP datagrams received from HAs and delivers them to the registered MNs. Furthermore, the FA performs authentication and administrative functions for MNs. It serves as default router for MNs. An FN does not necessarily need an FA to allow MIP operation.

⁴ In some implementations the HA can also be a host on the HN.

⁵ Here, the term router is related to the mobile nodes that are registered to this FA, i.e.: the FA will serve as default router at least for registered mobile nodes. In some implementations the FA can also be a host on the FN.

- **Mobility Agent:**
A Mobility Agent is a generic term for a Home Agent or a Foreign Agent.
- **Care-of Address:**
The Care-of Address (COA) is the IP address to which the HA tunnels datagrams destined for an MN. This tunnel termination point can either be an FA with which the involved MN is registered or the MN itself, when no FA is available. In the last case, the COA is called a co-located COA. It is the (temporary) IP address which was assigned to the MN on the visiting FN, for example by DHCP. Then, the MN will detunnel the encapsulated datagrams itself.
- **Mobility binding:**
The association of a Home Address, a COA and the remaining registration lifetime of an MN. Mobility bindings are stored in a registration table at the HA.

These definitions already give away a lot about the Mobile IP mechanism. In order to explain the Mobile IP mechanism roughly, two typical cases are worked out below. The IP addresses used in these cases are fictive and purely illustrative.

Case 1: Foreign Agent Care-Of Address

Case 1 describes an example of a standard configuration using an HA and an FA. The HN (1.2.3.0/24) is connected to the Internet via a router (the HA, 1.2.3.1), see Figure 1-2. The MN's fixed Home IP Address is 1.2.3.50.

The MN determines that it is at its HN by the reception of a specific message from its own HA: a Home Agent Advertisement. In general, HA Advertisements are sent by HAs and contain the IP address of the Home Agent, a lifetime, and of course MIP parameters about the services and capabilities of the HA. When an MN is at its HN, it can use normal IP routing and its own Home Address. In this case, the HA is implemented on the default router (gateway) of the HN.

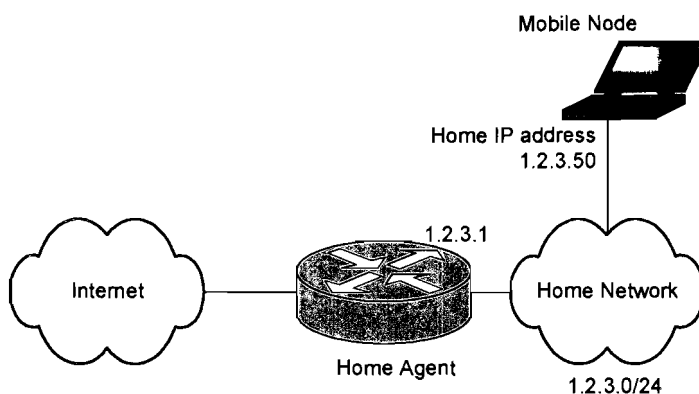


Figure 1-2 Case1 home situation

Suppose the MN is roaming and at some point attaches to FN 4.5.0.0/16, as shown in Figure 1-3. The MN detects its movement by the expiration of the advertisement lifetime⁶ of the

⁶ Here the lifetime of the ICMP message body is meant and not the registration lifetime of the Mobility Extension.

agent it was listening to without having received a new advertisement from the same agent. Another movement detection algorithm is based on differences in network prefixes.

The MN may try to get an IP address, for example by means of DHCP. But in general, success cannot be guaranteed. In this example case, suppose that the MN will not receive an IP address by DHCP, because its link layer MAC address is unknown at the DHCP server.

The MN discovers the availability of an FA by receiving a Foreign Agent Advertisement message, which presents one or more COAs to the MN. The COA does not have to be identical to the source address of the FA Advertisement message. For example, in case 1, the FA sends the FA Advertisement from the interface with IP address 4.5.0.1, but could present 11.22.0.9 as its COA.

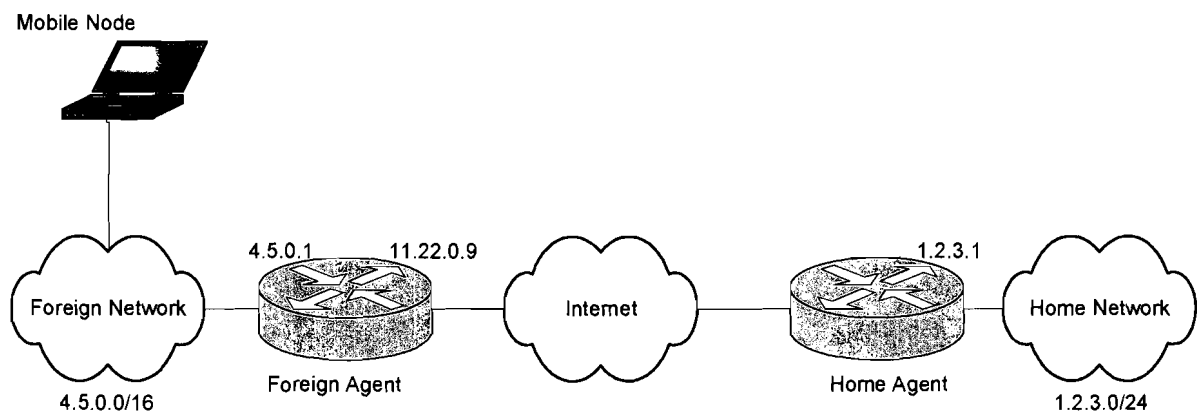


Figure 1-3 Case 1 foreign situation

Now the MN knows the FA and the COA it may use. The next step is to register itself with the FA and HA. This registration is necessary to create a mobility binding in the HA which is used for controlling the tunnel⁷ between HA and COA that is to be set up and maintained. The tunnel will be used for redirecting IP packets destined for the MN. In order to maintain the tunnel, this mobility binding has to be renewed or updated before the registration lifetime has expired. The MIP registration process uses the request-reply model. If an MN is registering via an FA, the FA will process the Registration Request message and then forward it to the HA. The HA replies to the FA, which will process this Registration Reply message again before relaying it back to the MN. All registration messages are authenticated (signed) to protect them against unwanted intervention and abuse.

Case 1 makes use of an FA, so the MN must register itself via the FA with its HA. The parameters for the mobility binding are exchanged during the registration. The MN send a Registration Request to the HA via the FA. After a successful registration, the HA will send a Registration Reply to the FA. This Registration Reply will arrive at the MN after being processed and relayed by the FA. Now the HA (1.2.3.1) will set up a tunnel to the COA, in this case the external interface of the FA (11.22.0.9), see Figure 1-4.

⁷ The concept of tunneling will be discussed in section 1.3.4.

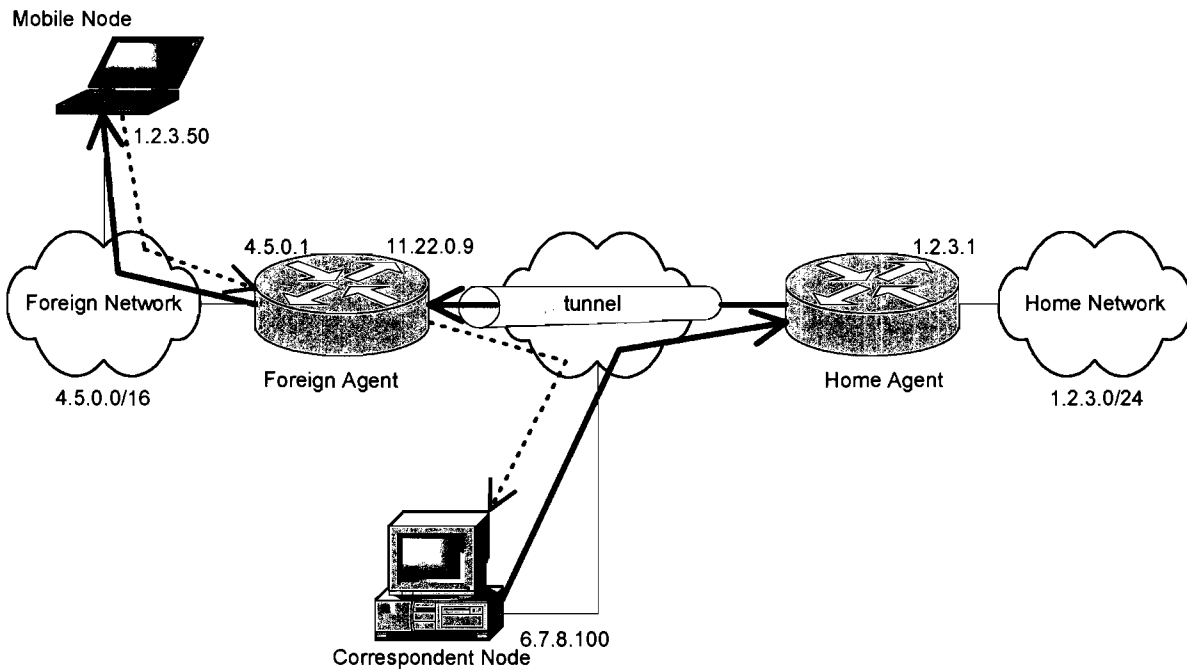


Figure 1-4 Case 1 routing scheme

The arrows in this figure show an example of how IP packets will be routed between an CN (6.7.8.100) and the MN. The grey arrows represent the packets sent from CN to MN, the black dotted arrows from MN to CN. The CN uses the MN's Home Address (1.2.3.50) to send IP packets to. These packets are routed towards the HN of the MN, based on the network number 1.2.3.0. At the HN the HA captures the packets and tunnels them towards the COA, thus towards the FA. The FA decapsulates the tunneled packets and puts them on the link of the MN. In this way the MN can receive the IP packets sent to its Home Address, while being attached on an FN.

Packets from MN to CN can use the normal IP routing. The MN uses the IP address of the CN (6.7.8.100) to send packets to. It places its Home Address 1.2.3.50 as source address in the IP packets, which in fact is topologically incorrect. The FA will serve as default router/gateway for the MN.

This way of routing is called triangular routing. The path from CN to MN via the tunnel is different than the path from MN to CN. Together it forms a triangle.

Case 2: Co-located Care-Of Address

Case 2 is a standard example not using an FA. Suppose that the MN attaches to FN 2.2.2.0/24, as shown in Figure 1-5. This FN does not have an FA installed, so here the MN will not receive any FA Advertisements. In this example, the MN receives an IP address (2.2.2.75) via DHCP.

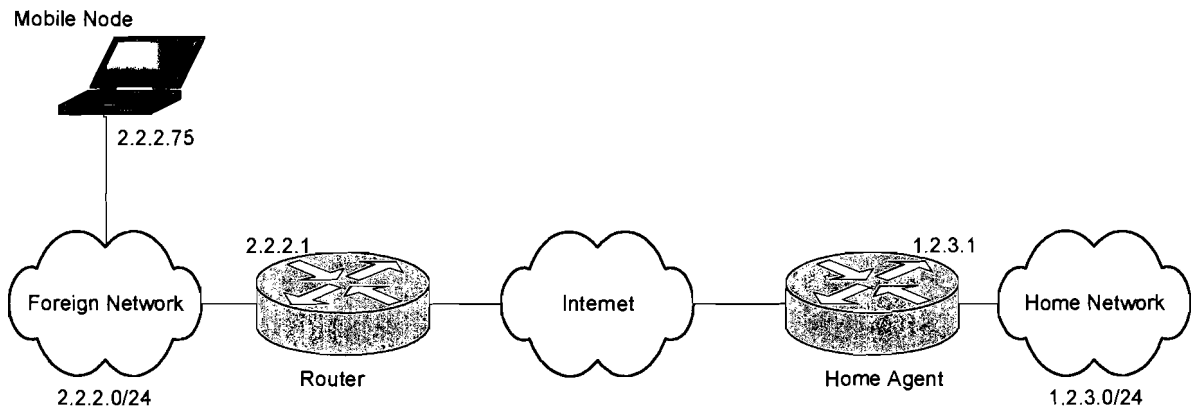


Figure 1-5 Case 2 foreign situation

The MN can use this address as COA. Because it is assigned to an interface of the MN itself, it is called a co-located Care-of Address. The MN registers itself directly with its HA (1.2.3.1) in order to create a mobility binding. The HA can now set up a tunnel to the COA, which actually is the MN in this case. This situation is depicted in Figure 1-6.

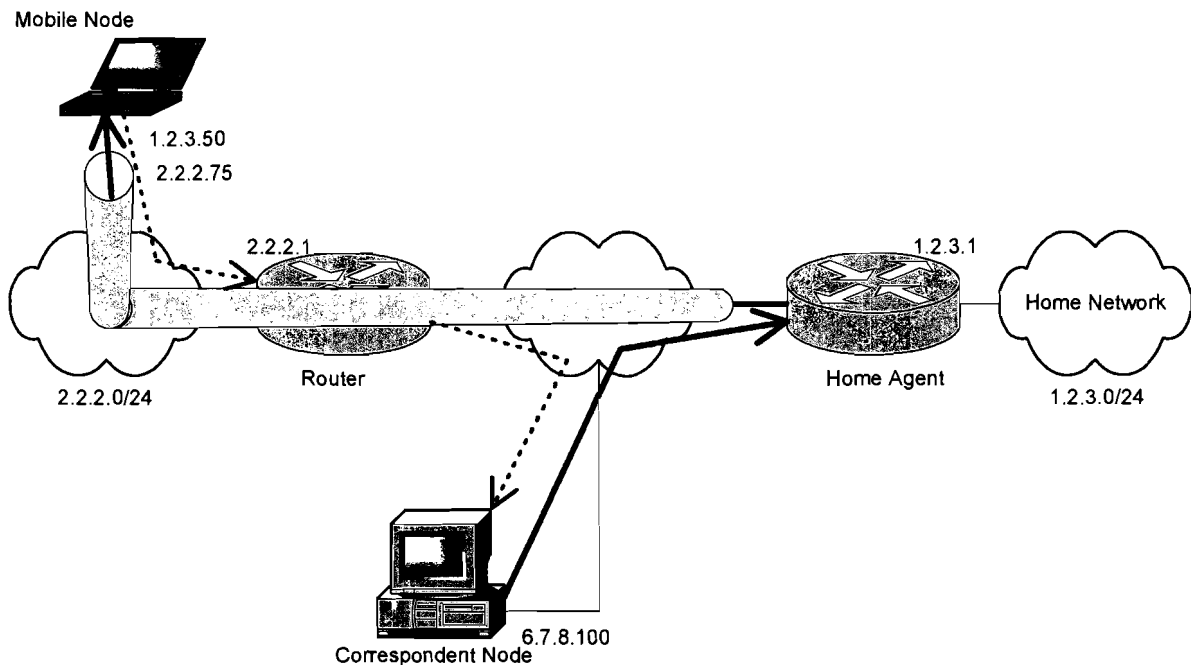


Figure 1-6 Case 2 routing scheme

Again, the arrows in the figure show a routing example. It is almost the same situation as in Case 1, but now the tunnel ends at the MN, which detunnels the packets itself. The MN thus can be reached by two IP addresses: 1.2.3.50 is used for data through the tunnel and 2.2.2.75 is used for MIP registration. The last one is the endpoint of the tunnel.

1.3 Mobile IP Details

This section describes some Mobile IP features that either are important to understand the basics or that are interesting for network architecture designers. The Mobile IP registration process is described, including the security aspects. Furthermore, agent advertisement, agent solicitation and tunneling will be discussed. The purpose of this section is to give a deeper functional overview of Mobile IP, with some essential and major details. Not all details nor the complete message formats are discussed. The interested reader can study the entire specification in [6, 7].

1.3.1 Agent Advertisements and Solicitations

For Mobile IP there are defined several specific messages. These messages are used for two main goals:

- Discovery of the Mobility Agents and their services;
- Registration of MNs to their respective HAs.

This section discusses some details about messages of the first bullet. The next section deals with the registration messages.

In the example cases it already was mentioned that HAs and FAs use specific messages to announce their presence, MIP requirements and capabilities to all nodes on a link. These messages are the Agent Advertisements. An HA sends Home Agent Advertisements, an FA Foreign Agent Advertisements.

The Mobility Agent uses broadcasting to send the messages. They send the Advertisements only on the links they serve. Normally the Advertisements are sent at some regular interval, but this is not required. Therefore, an Agent Advertisement may be requested by a so-called Agent Solicitation. When an MN broadcasts an Agent Solicitation message on a link, all available Mobility Agents on that link must reply with an Agent Advertisement. In this case the Advertisements should be sent directly to the requesting MN, thus not via broadcasting.

An Agent Advertisement message is composed of a Mobility Agent Advertisement extension added to a Router Advertisement message. The Router Advertisement message is defined in the ICMP Router Discovery Protocol (IRDP) [9, 10].

An Agent Solicitation message is identical to an ICMP Router Solicitation message with the TTL field set to 1. So it only is valid on the subnet it was sent.

A Mobility Agent Advertisement extension contains information about the services of the Mobility Agent. It announces whether the sending Mobility Agent serves as FA, HA or both. It informs the MNs about its services, the tunneling modes it supports, and in case of FA functionality, it advertises one or more COAs⁸. Agent Advertisements are not authenticated or encrypted.

A Mobility Agent Solicitation message does not contain any specific Mobile IP parameters.

⁸ Of course, an MN only uses one COA. It must try to use the first one advertised. Otherwise, it may try the other advertised COAs.

1.3.2 Registration Messages

When an MN is roaming on a Foreign Network, Mobile IP uses a tunnel to redirect the IP packets destined for the MN. This tunnel is set up and controlled by the MN and HA, in conjunction with the optionally FA. The control information is exchanged in Registration messages. This control information creates the mobility bindings at the HAs and entries in the registration tables of the FAs.

MIP uses the UDP protocol for the MIP registration messages. Well-known port number 434 has been assigned for this purpose at the Internet Assigned Numbers Authority (IANA⁹). The Registration Requests and Registration Replies are constructed by adding one or more extensions to a standard fixed portion, see Figure 1-7.

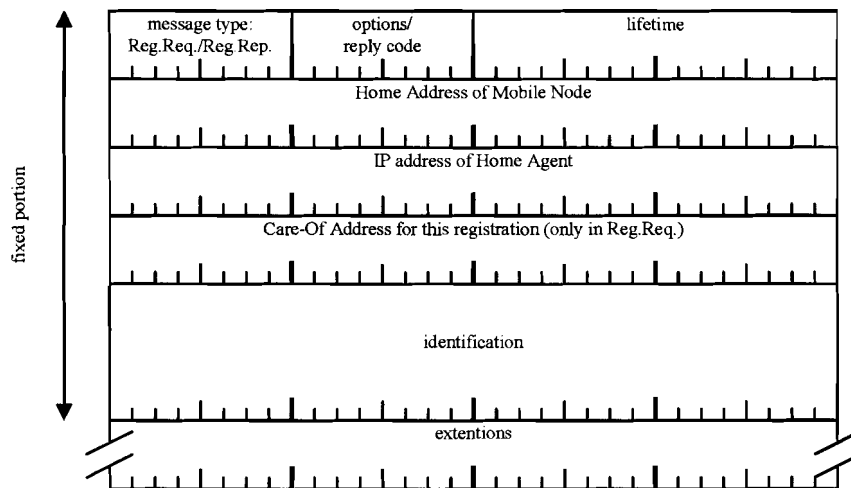


Figure 1-7 Fixed portion of Registration Request and Registration Reply message

The fixed portion of the Registration Request and Registration Reply only differs in one byte and one field. The fixed portion always includes

- the *Home Address* of the MN to identify the MN¹⁰;
- the *IP address of the HA* to identify the HA;
- a *COA* to inform the HA about the tunnel endpoint;
- and an *identification field* to identify the registration instance; it is used to match the Registration Request with the Registration Reply.

In the differing byte, the Registration Request includes some fields in which the MN can request specific MIP services. In the Registration Reply this byte is filled with a reply code of the HA, which informs the MN about the status of its registration. In this way the MN can verify whether the registration has been accepted and take an adequate action if an error code was returned.

If the MN is not configured with its Home Address, it may use the all-zeroes address (0.0.0.0). In section 1.4.1, it is discussed how an MN may use a Network Access Identifier (NAI) extension to identify itself to its HA in order to receive a Home Address.

In RFC 3220 the only extensions defined are authentication extensions, see Figure 1-8. These extensions secure the message exchange. Each MN has a security association with its HA.

⁹ IANA: <http://www.iana.org>; List of registered port numbers: <http://www.iana.org/assignments/port-numbers>

¹⁰ The MN has another possibility to identify itself, as will be discussed in section 1.4.1.

This association is identified with a 32-bits number, the Security Parameter Index (SPI). Optionally, an MN may have a security association with an FA and an FA may have a security association with an HA, all indexed by their respective SPI. The security association contains a secret, a 128-bits shared key, which is used to calculate a so-called authenticator from specific fields of the Registration messages. The authenticator is a mathematical message digest, according the HMAC-MD5¹¹ algorithm [11]. This is a kind of signature that can only be read by the owners of the shared key. It must protect the HA and MN against fake Registration messages from other nodes. An authentication extension contains the SPI to identify the node and the authenticator. So, it is not the Home Address or the identification field in the Registration Request, but the SPI that identifies the MN in the authentication extension.

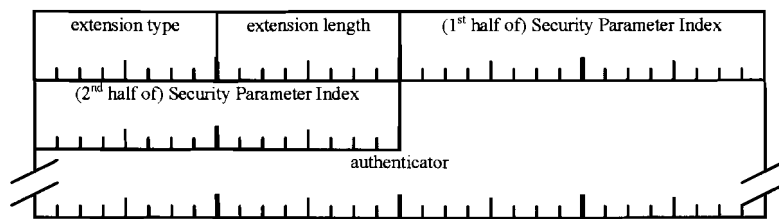


Figure 1-8 Authentication extension

There is one other form of attack that the Registration process is protected from: the replay attack. The identification field in the fixed portion of the Registration messages contains a timestamp or nonce¹² to assure that a Registration message is valid only once. This will prevent confusion when malicious nodes are repeating old messages.

Every Registration message consists of the fixed portion followed by one or more extensions. The authentication extensions have to be placed after all other extensions in order to protect them; the message digest is calculated over the UDP payload including all prior extensions. The order in which the authentication extensions have to be placed depends on the available security associations. The next example will clarify this. Figure 1-9 represents a Registration Request sent from MN to HA via an FA. In this case three security associations are present, one between MN-HA (MH), an optional one between MN-FA (MF), and an optional one between FA-HA (FH). Each of them has its own shared key to calculate or verify the related authentication extension.

¹¹ HMAC-MD5 = keyed Hashing for Message Authentication: a mechanism for message authentication using cryptographic hash functions. It calculates a Message Authentication Code. HMAC can be used with any iterative cryptographic hash function, e.g. MD5 (Message Digest algorithm 5), in combination with a secret shared key; RFC 2002 also supported the bare MD5 algorithm, which is not considered enough secure anymore.

¹² A nonce is a randomly chosen number, different from previous choices.

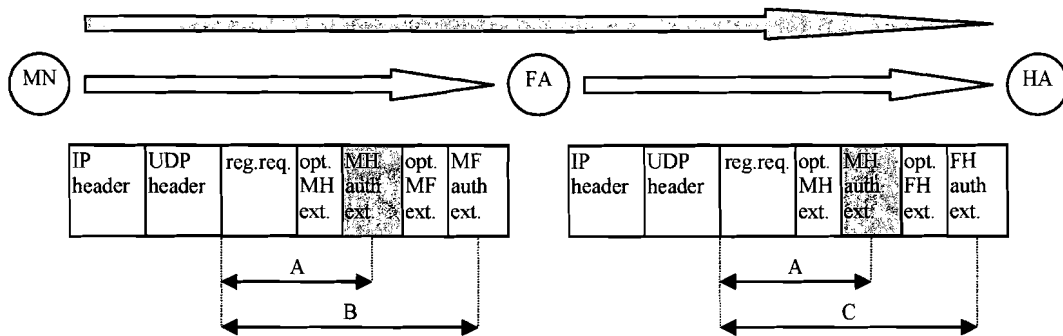


Figure 1-9 Authentication extension order

The Registration Request packet is shown twice; once in its composition sent from MN to FA, and once in its composition sent from FA to HA. When the packet is sent from MN to FA, the MN computes the MH authenticator over the data pointed out by arrow A, i.e. over the Registration Request, optional MH extensions and the first part of the MH Authentication Extension (not over the authenticator itself). The MF authenticator is computed over the data pointed out by arrow B. When the packet arrives at the FA, the FA removes and verifies the MF authenticator. If it succeeds, it calculates the FH authenticator over the data pointed out by arrow C. The FA adds it to the packet, which will be sent to the HA. The HA first removes and verifies the FH authenticator, after which it removes and verifies the MH authenticator. The Registration Reply message is analogically authenticated. In this way, the security associations are independent of each other.

Note that the figure also displays some optional undefined extensions (MH, MF and FH extensions) to indicate their places.

Not all authentication extensions are required, but there is one that must be present in each registration message: the MH authentication extension. An FA may request an MF authentication in case the MN and FA have a security association. The FA can indicate this request in the FA Advertisement. Furthermore, if an FA-HA security association exists, an FH authentication extension may be used. The last two optional security associations have been defined in order to prevent the use of malicious FAs.

1.3.3 Mobile IP Registration

In the previous section the Registration messages and the involved security aspects were discussed. This section will go into the registration process itself. The purpose of the registration process is supplying the HA and optionally FA the correct information to start the tunneling. For the HA this information is stored in a so-called mobility binding. An FA that serves several MNs can have registrations to multiple HAs. The FA keeps a visitor list of each registration, containing an entry for each MN that wants to register with the involved HA.

When an MN registers, it sends a registration request to the FA. After validation of the optional MF authentication extension, the FA creates a pending visitor entry, optionally adds an FH authentication extension, and forwards the message towards the HA. The pending visitor entry includes

- the MN's (layer 2, 3 and 4) *addressing information with respect to the FN*;
- the MN's *Home Address*;
- and the *HA address*.

It also contains a requested registration lifetime. When the HA receives the registration request, it first validates the optionally FH authentication extension. Then it validates the mandatory MH authentication extension. If these are all valid, the HA creates a mobility binding for the MN, which includes

- the MN's *Home Address*;
- its *COA*;
- and the *remaining registration lifetime*.

The HA replies with a Registration Reply sent towards the FA, indicating whether the registration was successful or why not. If the registration was successful, the FA activates the corresponding pending entry and forwards the reply to the MN. The reply also is secured with the authentication extensions. Now the HA can set up the tunnel to the COA.

In case of a co-located COA, this Registration process occurs directly between MN and HA.

Figure 1-10 shows the exchange of registration messages and the tunnel set up.

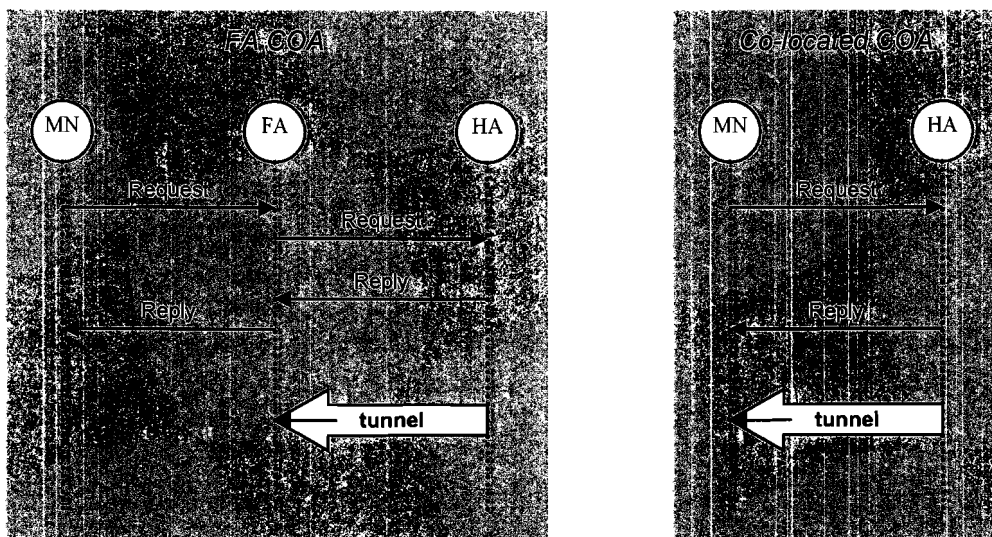


Figure 1-10 Registration Request and Reply

The HA and the FA keep the remaining registration lifetime for each MN. The MN is responsible for re-registration before the registration lifetime expires in order to maintain the tunnel. A mobility binding will be deleted after expiration of the lifetime. When an MN registers with a requested lifetime equal to zero, the mobility binding will be deleted. So in fact it is deregistration. Deregistration is very useful for an MN that has multiple simultaneous bindings and wants to delete one or more bindings but retain others. An MN should deregister all of its bindings when it returns to its Home Network.

Simultaneous Bindings

Mobile IP has a feature that allows the HA to make multiple mobility bindings to different COAs at the same time for one MN at its request. An HA that supports this “Simultaneous Mobility Bindings” option, can forward an IP datagram, destined for the MN, into each tunnel towards the MN’s multiple COAs. The MN will only process the first packet it receives and drop the others because those will be considered as duplicate. This may be very useful when an MN can use multiple points of attachment over different link layer technologies (recall Figure 1-1) and thus can be in the service area of more than one FA at the same time. The MN can request a mobility binding over a second interface to set up a tunnel, before the tunnel of

the first interface is disconnected. The MN may maintain both bindings or delete the first one. In this way no IP datagrams will be lost during an IP handoff between two different access technologies. This results in a seamless MN-initiated handoff at the network layer (layer 3). A seamless handoff by means of MIP is only possible between different interfaces, so not for example between two WLAN networks which are accessible via the same WLAN interface. There is a small restriction, which will be discussed at the end of section 1.3.5.

1.3.4 Tunneling Modes

The previous sections made clear that Mobile IP is an advanced routing mechanism, which in fact uses dynamic tunneling. Mobile IP supports three tunneling protocols. The basic tunneling protocol is the “IP within IP encapsulation” [12]. This must be supported by all HAs and FAs, and also by MNs that support co-located COAs. There are two other optional tunneling protocols specified: “minimal encapsulation within IP” [13] and “Generic Routing Encapsulation” (GRE) [14]. An MN may request one of these tunneling modes from its HA in the Registration Request. In case an FA is involved, the MN may only request a different tunneling mode when the FA advertised that it supports this.

IP within IP encapsulation is a very basic mode of tunneling. IP packets are encapsulated at a node, called the encapsulator. This is the tunnel entry point. Encapsulation means that an original IP packet is put into a new IP packet as payload, see Figure 1-11 (the tunneling part is displayed in the grey box). The IP header of the new packet, called the delivery header, contains the IP address of its decapsulator. This node decapsulates the new IP packet to retrieve the original IP packet that can be routed towards its original destination. In case of this tunneling protocol, decapsulation in fact is nothing more than removing the first IP header. The decapsulator is the endpoint of the tunnel. The purpose of this tunneling is to deliver the original IP packet to an intermediate node, the decapsulator, which would otherwise not be selected based on the IP destination address of the original IP header. In Mobile IP, the encapsulator is the HA, and the decapsulator is the FA or the MN.

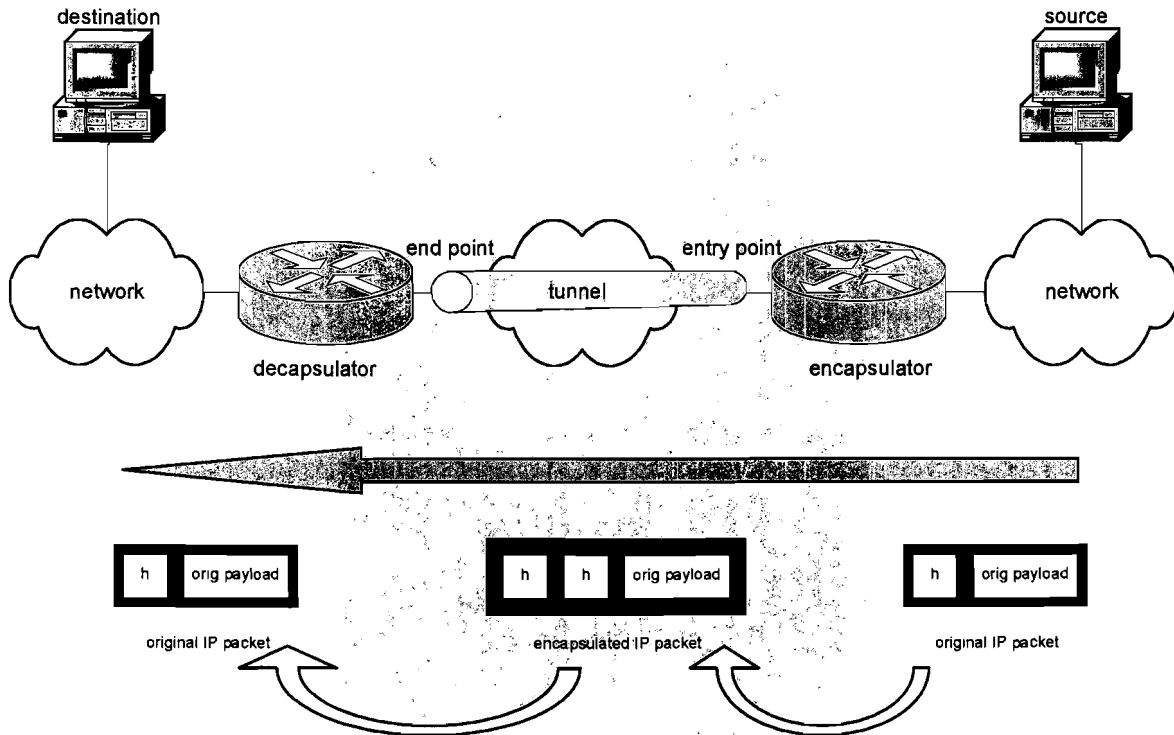


Figure 1-11 Tunneling scheme

Minimal encapsulation is a tunneling mode that is very much similar to IP within IP encapsulation. The difference is that minimal encapsulation does not encapsulate the full IP header of the original message, because several fields of this header and the new outer header are duplicated. In this way, header space can be saved, resulting in less overhead.

GRE is a more general tunneling protocol, which inserts an extra GRE header in between the original packet and the delivery header. GRE is not specifically for encapsulation of IP packets in an IP packet. It may encapsulate a packet of any protocol X into the payload of any protocol Y.

In section 1.2, it was already mentioned that the routing mechanism, introduced by Mobile IP, may result in topologically incorrect routing schemes. This holds for packets sent from MN to CN, when the MN is roaming on an FN. Mobile IP says the MN must use its Home Address in the source address field of the IP packets it sends. This Home Address does not belong to the network prefix of the FN. Nowadays, border gateways (routers) of an Autonomous System (AS¹³) are equipped with ingress and egress filters. These filters must protect the network from IP packets with spoofed IP addresses entering or leaving the AS. It just works as the opposite of routing. When an IP packet crosses the router, the router will use Reverse Path Forwarding (RPF) to find out whether the source IP address is “legal” or not. This means that if the link on which the packet arrives is registered as the default path for the source address of the IP packet, then the packet will be forwarded. Otherwise, the router discards the

¹³ An Autonomous System is a collection of routers and subnets under a single administrative authority, using a common Interior Gateway Protocol for routing packets. It is bounded by routers using the Border Gateway Protocol.

packet, because it is very likely that the source address was spoofed. This “spoofing filter” causes that the packets sent by the MN may not arrive at the CN.

In order to solve this problem, Mobile IP implements a feature called reverse tunneling, see RFC 3024 [15]. Reverse tunneling is not mandatory. An FA and HA notifies the MN about whether it supports this in their Agent Advertisement. An MN may request reverse tunneling in the Registration Request. When reverse tunneling is used, a bi-directional tunnel will be set up that also tunnels packets from COA to HA. In case of reverse tunneling, the tunnel will not be set up from the HA to the COA, but from the COA towards the HA. The registration process and tunnel set up is depicted in Figure 1-12.

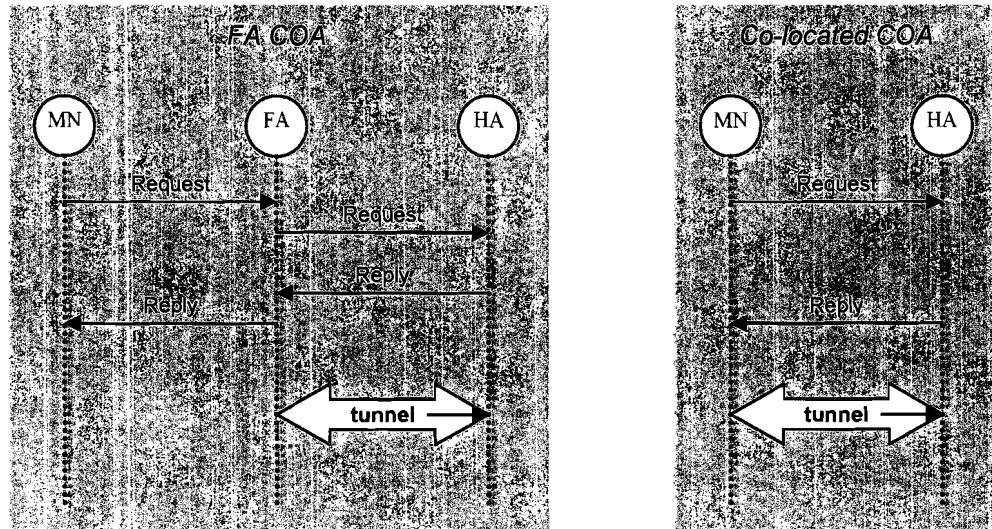


Figure 1-12 Registration process and tunnel set up using reverse tunneling

There are some more advantages of reverse tunneling, which can be found in RFC 3024 [15]. Figure 1-13 shows the routing scheme of Case 1 of section 1.2 when reverse tunneling is used. This is also called dog-legged routing.

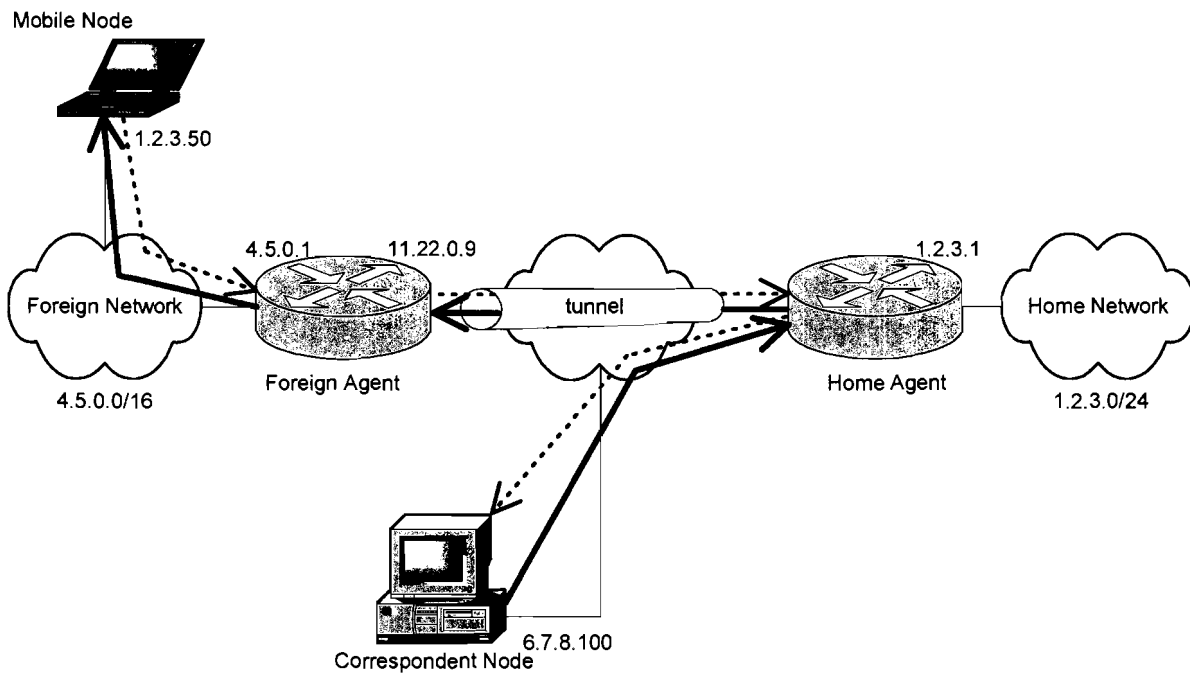


Figure 1-13 Reverse tunneling routing scheme

One could question what the difference between “normal tunneling” and Mobile IP is. Why should not an MN use just a normal tunneling method¹⁴ towards the HA without Mobile IP? There are some substantial differences

With Mobile IP the tunnel is not always bi-directional; reverse tunneling basically is only used when the uni-directional tunnel results in spoofing problems. Uni-directional tunneling saves overhead on the packets that do not need to be tunneled from MN to CN, compared to bi-directional tunneling. It also reduces the load on the links between COA and HA.

Furthermore, in Mobile IP the tunnel end point is not always the MN itself. It can also be an FA. In case the MN registers via an FA, the most advantageous difference is that the MN does not need to obtain an IP address (e.g. via DHCP) on the FN. The MN just uses its Home Address. If several MNs of the same HN register via the same FA, one tunnel will serve all MNs. This means that only one COA, so one IP address, is needed. In case of co-located COAs or when the MN is just using a normal tunnel (without Mobile IP), each MN will need an extra IP address, which forms a demanding load on the short running global set of public IP addresses.

1.3.5 ARP Issues

Mobile IP defines that the HA intercepts IP packets destined for the MN and tunnels them to the COA, when the MN is roaming on an FN. In order not to disturb the Address Resolution Protocol (ARP) [16], the HA may take special actions. ARP is used by nodes on the same link (e.g. Ethernet) to match the IP addresses with the layer 2 addresses¹⁵. If node A wants to send an IP packet to node B on the same link, it initially only knows the destination IP address. The IP packet is sent in a layer 2 packet labeled with the layer 2 address of node B for destination.

¹⁴ Think about VPN (Virtual Private Network) solutions.

¹⁵ For Ethernet these are the MAC (Medium Access Control) addresses; globally unique 48-bit identifiers.

So node A has to know the layer 2 address of node B. Node A has an ARP cache in which it stores a list that matches IP-addresses with layer 2 addresses for each node on the same link. The entries of this table have a short lifetime, about 30 seconds.

Node A may learn the layer 2 address of node B from formerly received IP packets or it may use ARP: if a node does not know the layer 2 address of a neighbor node (anymore), it may broadcast an ARP request on the link, which contains the requested IP address. Node B recognizes its IP address and responds to the requesting node A with an ARP reply to identify itself. If no answer was received on this ARP request, node A will assume that there is no node on the link with the requested IP address.

With Mobile IP, the HA has to intercept all packets destined for the roaming MNs. These IP packets will arrive at the link of the HN according to normal IP routing. But here the HA has to intercept these packets so it can forward them to the COAs. Therefore the HA will respond to ARP requests on behalf of the MNs. It will reply with its own layer 2 address. In this way all IP packets destined for roaming MNs will pass the HA. This principle is called proxy ARP.

Another necessary ARP packet is the Gratuitous ARP packet. A Gratuitous ARP packet is broadcasted on a link to update the entry of the sender in the ARP tables of all nodes on that link. When an MN leaves the HN and registers on an FN, the HA will send a Gratuitous ARP packet on the HN to update the ARP tables. When an MN returns to its HN, it uses Gratuitous ARP to update the ARP tables again.

Because of these ARP aspects, the MN is not allowed to use ARP at an interface connected to its HN and register from a FN at the same time. This would cause two interfaces (one of the MN and one of the HA) to claim the same IP address from the Ethernet point of view. The MIP specifications state that the MN should stop responding to ARP requests containing its Home IP Address before registering from a FN. This means that simultaneous bindings cannot be used when the HN is involved.

1.4 Specific Mobile IP Problems and Solutions

This section discusses some extra IETF documents from the Mobile IP Working Group, which may be interesting for a network operator that wants to implement Mobile IP for its customers.

Some documents are published as drafts, which means that they still are in the development phase and not officially published, in contrary to RFCs. The IETF reports: “Under no circumstances should an Internet-Draft be referenced by any paper, report, or Request-for-Proposal, nor should a vendor claim compliance with an Internet-Draft.” So, the items discussed in this section still are subject to change, but are very interesting to keep up with. Because some drafts already have been removed from the Working Group website, some drafts are given in Appendix G.

1.4.1 Mobile Node NAI Extension

In section 1.3.2, it was mentioned that there is an option for the MN to discover its Home Address when it is not configured with it. The MN then may use the MN NAI¹⁶ extension to identify itself. It sets the Home Address field of the Registration Request to 0.0.0.0. When the

¹⁶ Network Access Identifier, see [17].

HA has validated the Registration Request containing a Home Address field 0.0.0.0 and an MN NAI extension, it may assign a Home Address to the MN by starting to use this address in the Registration Reply. The MN must be able to assign its Home Address after extracting this information from the Registration Reply.

Besides the common fields, which are not discussed in this document, the MN NAI extension only contains a field with the MN NAI string. An example of such a NAI string:

Tom_van_Seille.CDS-WirelessWeb@Vodafone.NL

The MN NAI serves as the unique identifier for the MN, so the MN's Home Address does not always have to provide that function.

In order to authenticate the NAI extension, this NAI extension must appear in the Registration Request before the MH authentication extension and the MF authentication extension, if present. In this way, the FA as well as the HA can validate the NAI. Besides the acquisition of a Home Address, there are other possibilities that the MN NAI can be used for. For example, one could think of an FA that may use the NAI contained in the MN NAI extension to authenticate the MN at an AAA server in order to grant the MN access to the foreign domain.

1.4.2 NAT Traversal

Many private/internal LANs (e.g. office LANs) are connected to external networks (e.g. the Internet) via a Network Address Translator (NAT) [18]. A NAT allows hosts within a private network to transparently access hosts in the external network. It is configured with a block of external addresses for translating addresses of hosts in a private domain as they originate sessions to the external domain. For packets outbound from the private network, the source IP address and related fields such as IP, TCP, UDP and ICMP header checksums are translated. For inbound packets, the destination IP address and the checksums as listed above are translated. The NAT temporarily maps the private IP address to an external IP address.

In order to extend the flexibility and the number of simultaneous outbound sessions using the same external address range, a Network Address Port Translator (NAPT) may be used. Besides the address translation, an NAPT can also translate the layer 4 transport identifiers (TCP/UDP ports), so it can multiplex several outbound sessions (using multiple internal IP addresses) to one external IP address on different port numbers. Here the port numbers are also mapped. Figure 1-14 shows an example with an NAPT that has the public IP addresses 62.140.140.240 through 62.140.140.247 at the external side. A host 192.168.10.25 initiates an HTTP session (port 80) to 131.155.2.38 and chooses TCP source port 1025. The packets are drawn with the IP source (S) and destination (D) addresses and with the TCP/UDP source and destination port numbers. The NAPT temporarily maps IP address 62.140.140.245 and port number 3456 to it. Normally, the mapping will be deleted after the TCP session is ended. In case of UDP, there is no connection set up and thus no connection end. Therefore, the mapping will be deleted when no traffic of that UDP "session"¹⁷ has crossed the NAPT for a specified amount of time.

¹⁷ With UDP actually a session is not defined. Here the time that UDP packets can be exchanged over NAT is meant.

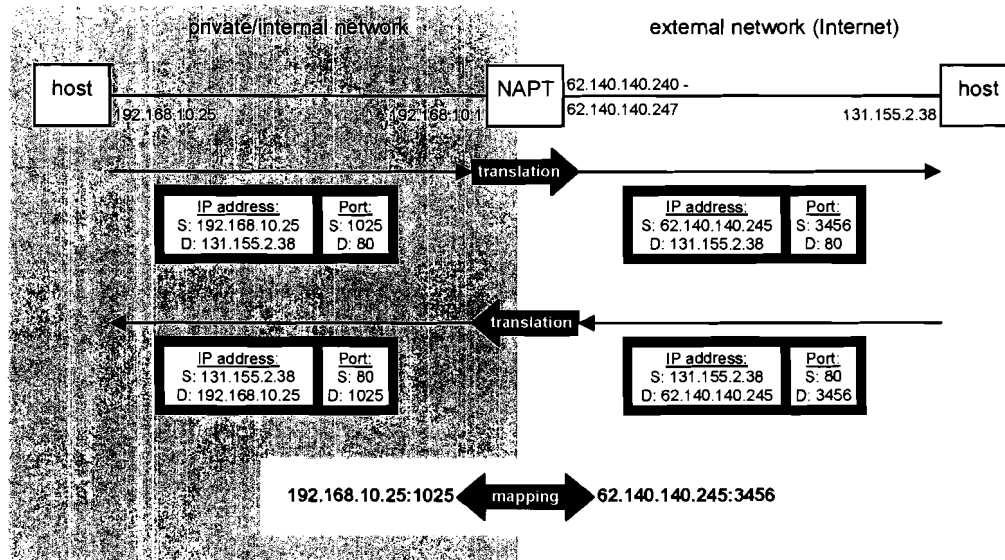


Figure 1-14 NAPT example

The term NAT often is used when both NAT as well as NAPT is meant and so will this document do. The advantage of NAT is that the internal hosts are protected against attacks from the outside, because inbound traffic is only allowed for a limited time after an outbound session was initiated. Furthermore, thanks to the multiplexing, the external IP range does not have to be as large as the number of internal hosts to allow all internal host access to the external network simultaneously. An NAPT just needs a much smaller block of external IP addresses. This saves some space on the set of globally unique IP addresses, which is running short.

A problem arises when an MN is roaming on an FN behind a NAT, while its HA is at the external side of the NAT. Mobile IP has one assumption that does not hold for this situation. Here, MNs and FAs are not uniquely identifiable by a routable IP address. A tunnel, set up from the HA to the COA, generally will not be able to pass the NAT. At the time of writing this report, the Working Group has made an Internet-draft [19] about a possible solution, which will be discussed here. It assumes that communication from behind a NAT to the HA on UDP port 434 is possible. After all, without this communication the Mobile IP Registration process would not be possible, not to mention the tunneling for redirecting the IP packets.

The solution is to use another tunneling mode: Mobile IP UDP tunneling. As TCP and UDP packets may traverse the NAT, the idea is to use a layer 4 tunneling mode. Since MIP uses UDP port 434 for the registration, it may assume that tunneling is only useful if this port can be used through the NAT. Therefore the proposed tunneling mode is UDP tunneling over port 434 (after all, you need to use a port if you want to use port translation!). The MN must not change the source port number. So, besides the control information, the data will now be tunneled through the same UDP "session". This results in IP in UDP, GRE in UDP, and minimal encapsulation in UDP tunneling. To make sure the mapping in the NAT will not be deleted, a keepalive packet must be sent when no packets have been sent for a specific period, called the keepalive interval. It is very likely that reverse tunneling also will be needed.

There are two new registration extensions defined for this purpose: the UDP Tunnel Request Extension and the UDP Tunnel Reply Extension.

In case of a co-located COA, the MN adds the UDP Tunnel Request Extension to its Registration message (before the MH authentication extension) to notify the HA that it is capable of handling MIP UDP tunneling. In case of an FA COA, the FA adds the UDP Tunnel Request Extension to its Registration message (before the FH authentication extension) to notify the HA that it is capable of handling MIP UDP tunneling; the MN does not know anything about the UDP tunneling then. In both cases the HA may detect that the MN/FA is behind a firewall because the COA differs from the IP source address. If the HA has detected this difference and supports MIP UDP Tunneling, it will reply with an MIP UDP Tunneling Reply Extension to notify the MN/FA that the it will use MIP UDP tunneling. The UDP tunnel extensions also exchange the tunneling mode to be used. Figure 1-15 shows a schematic view of this registration process in case an FA COA and reverse tunneling is used. After the registration the tunnel is already set up because it uses the same UDP “session”. In order to keep the UDP tunnel alive through the NAT (i.e. retain the address/port mapping), the FA sends a keepalive packet to the HA when no traffic has been sent for an UDP keepalive period.

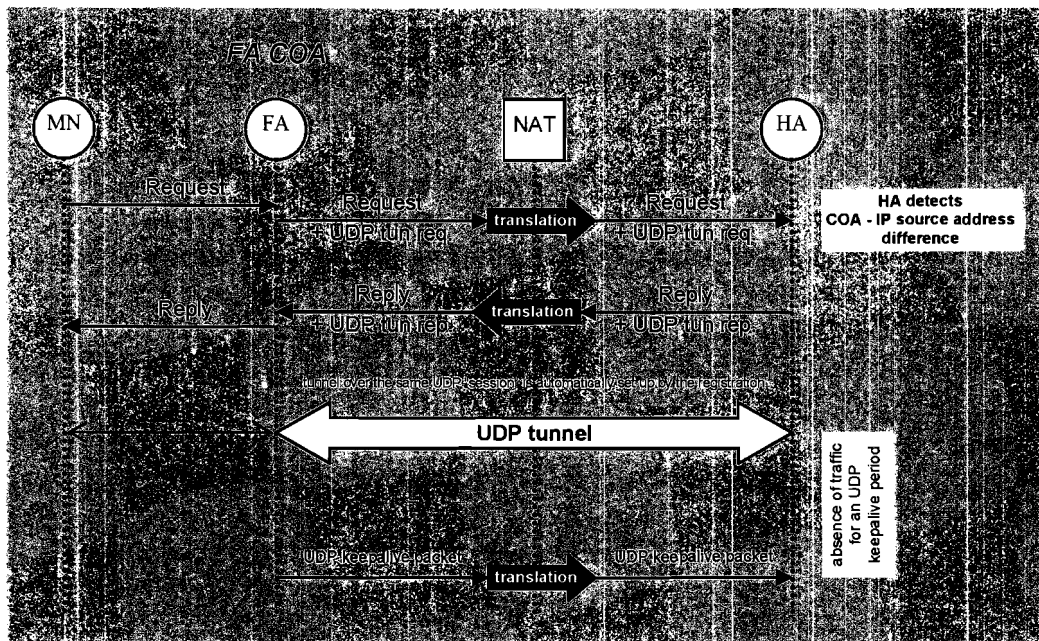


Figure 1-15 MIP UDP tunnel registration

1.4.3 Dynamic Home Agent Discovery

An MN does not have to be configured with its Home Agent IP address. It may also send a Registration Request to the Home subnet using subnet-directed broadcasting. The subnet-directed broadcast IP address is formed by using the subnet prefix appended with the all-ones host number. For Case 1 in section 1.2 this would be 1.2.3.255. HAs will respond to this Registration Request with a Registration Reply denying registration but containing the correct HA. After the reception of this Registration Request, the MN can retry to register with the correct HA IP address. This principle is called dynamic HA discovery. The only thing the MN has to know is the subnet prefix of its home network.

1.4.4 Route Optimization

The Working Group has been studying on a solution to optimize the routing. The idea was that the HA (or MN) notifies the CN about the MN's current COA, so that the CN itself can tunnel the IP packets directly to the COA. The return was that packets do not have to be routed to the HA first and then be re-directed into a tunnel towards the COA. Therefore the CN must maintain a list of Mobility Bindings itself. The Route Optimization draft [20] defined some new messages, of which the most significant is the Binding Update.

The details of the route optimization draft can be read in [20]. Shortly after draft version 11 updated version 10, it was removed from the website, because of two major issues:

- In order to be able to authenticate the Binding Update messages, a security association between HA/FA/MN and the CN is required. It is practically unfeasible to create these security associations with any CN, since an CN can be any node on the world.
- The CN must be MIPv4 aware to handle Binding Update messages, while MIP was designed to be transparent to CNs. This contradicts.

These issues caused the Route Optimization procedure to stop.

1.4.5 Dynamic Key Distribution

Since the Internet has become an all accepted medium and many people started using it every day for work and fun, security aspects are of growing importance. In order to be able to track hacker activities, many Internet Service Providers (ISPs) require authentication of users to grant the access and give user specific privileges. Protocols are developed that handle this Authentication, Authorization and Accounting, AAA for short.

Various Internet drafts and RFCs together define a way how Mobile IP and an AAA infrastructure should interwork. As some of these documents are still drafts, it is discussed only globally. Draft [21] specifies a mechanism to generate new MIP security associations during the MIP registration process via an FA. It uses the AAA infrastructure of which the interworking with MIP is defined in RFC 2977 [22]. The situation is shown schematically in Figure 1-16.

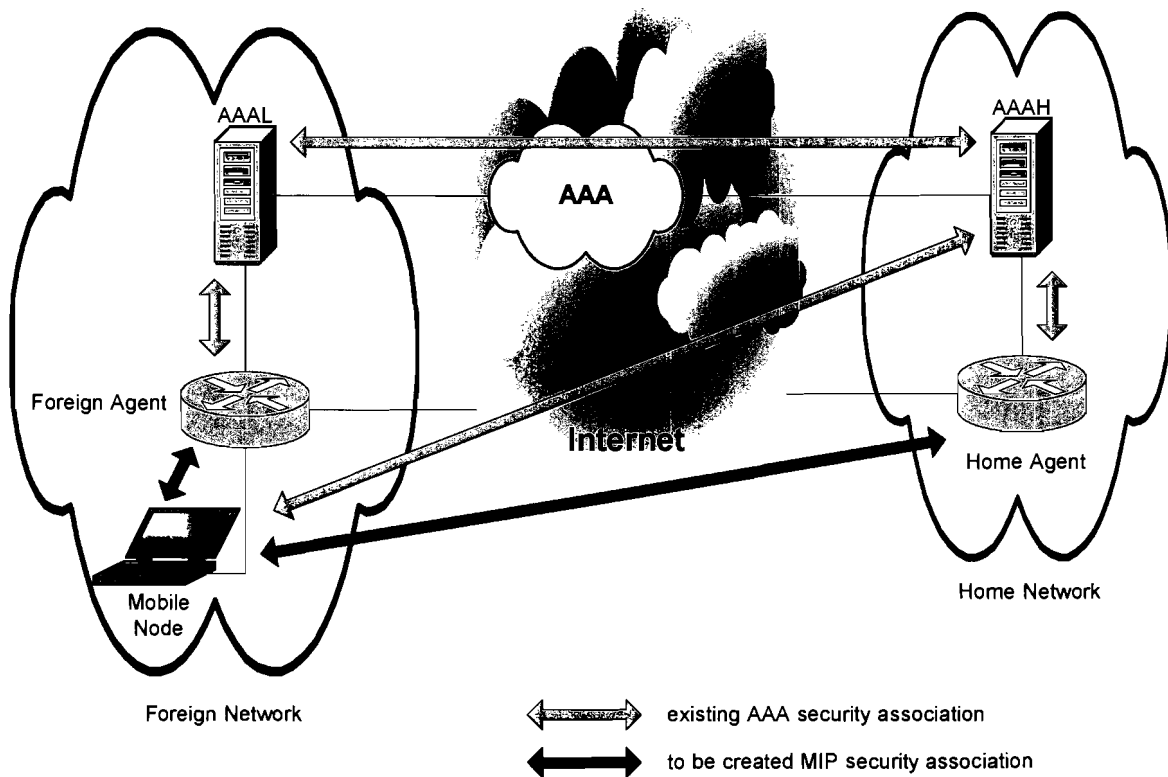


Figure 1-16 Interworking between AAA infrastructure and Mobile IP

The new MIP security associations to be created are derived from existing AAA security associations. An AAA security association is similar to an MIP security association; it contains an SPI and a shared key. The MIP security associations to be derived are those between MN and FA, and between MN and HA (if it did not already exist).

Draft [21] specifies new MIP extensions, which the MN can use during MIP registration to request a new MIP security association. These “AAA” extensions are the “MN-FA Key Request” extension to request a security association between MN and FA, the “MN-HA Key Request” extension to request a security association between MN and HA, and the “MN-AAA Authentication” extension to prove the MN identity to the AAA infrastructure. The MN identifies itself with its Home Address or an NAI.

Figure 1-16 shows the existing security associations that are used to create new ones. The MN is authenticated based on the AAA security association that it shares with its home authority AAA server (AAAH). The FA interworks with a local authority AAA server (AAAL): the FA runs the AAA client that exchanges the AAA information with the AAAL. AAA transport protocols, e.g. RADIUS¹⁸ or DIAMETER, implement a way to reliably forward AAA messages between AAA authorities on different domains (networks), for example from an AAAL (on an FN) to the AAAH (on the HN). For RADIUS this forwarding is called

¹⁸ Remote Authentication Dial In User Service [23].

RADIUS proxy. Based on the Home Address of the MN or on the home domain contained in its NAI, the AAAL forwards the AAA information to the correct AAAH¹⁹.

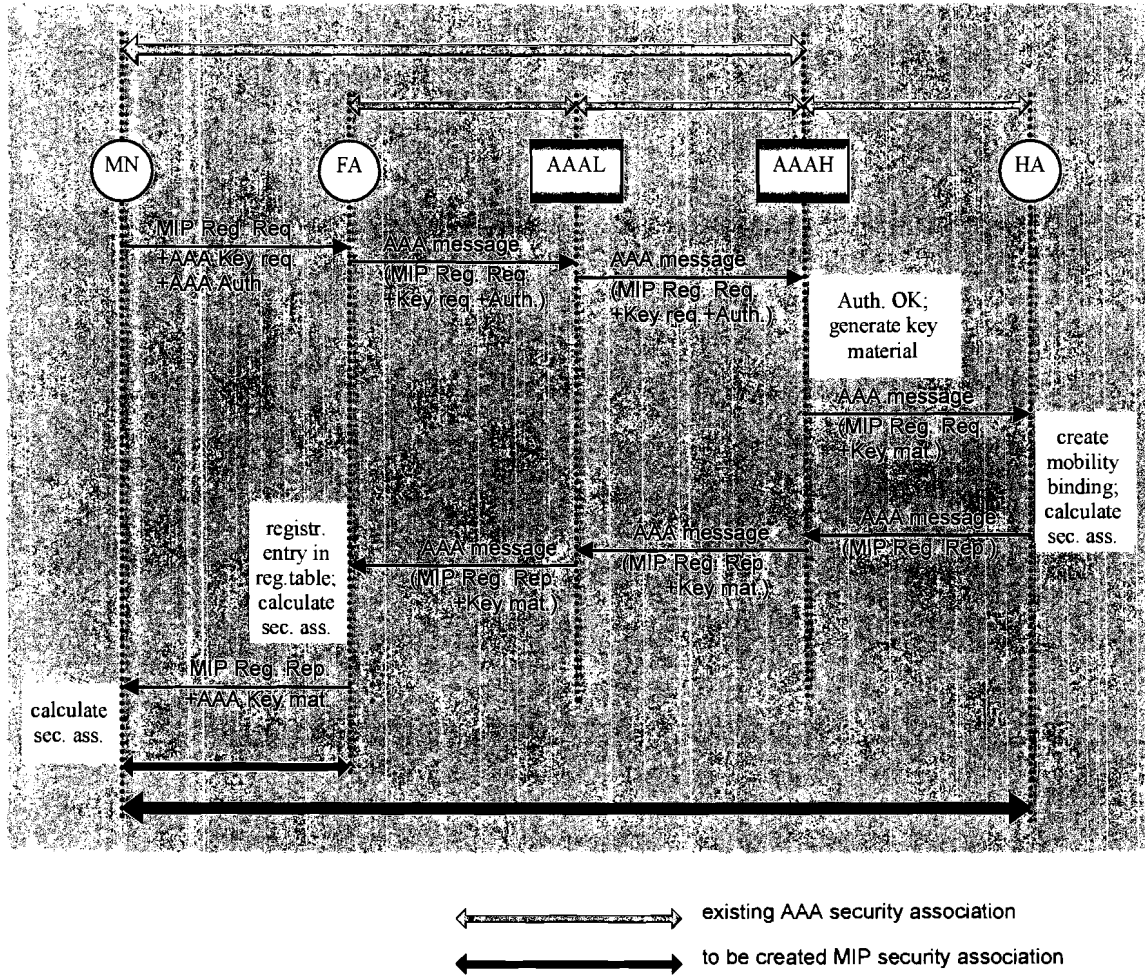


Figure 1-17 Message exchange

The message flow for a MN that needs both a new MN-FA and MN-HA security association is presented in Figure 1-17. The MN sends a MIP registration request containing an MN-FA Key Request, an MN-HA Key Request, and an AAA Authentication extension. The FA forwards this MIP registration request including the key requests and the AAA authenticator into the appropriate AAA format towards the AAAL, which in turn will forward the request to the AAAH of the MN. The existing AAA security associations assure the correct delivery. The AAAH verifies the information contained in the AAA Authentication extension in order to authenticate the MN. This is based on the AAA security association between the MN and its AAAH. Now, the AAAH generates key material which is used to calculate the new MIP security associations. Via the AAA protocol and the special MIP “AAA” extensions this key material is distributed to the nodes for which a new security association has to be generated. Key material for the new MN-FA MIP security association is distributed to the MN and the FA; key material for the new MN-HA MIP security association is distributed to the MN and the HA. The MIP Registration Request is forwarded from AAAH to HA along with the key material. The HA creates a mobility binding for the request and calculates the new MN-HA

¹⁹ Note that any AAA server can be an AAAL and/or AAAH, depending on the relation with the MN’s home domain.

MIP security association from the key material. The MIP Registration Reply is then sent back to MN via the AAA infrastructure, together with the key material for FA and MN. The FA calculates the new MN-FA MIP security association based on the key material and in the same way the MN calculates both MIP security associations. The MN may then authenticate the MIP message using the new MIP security association. Future MIP registrations can be sent via the MIP infrastructure using the new security associations.

The key material is a code of at least 64 bits. It is used as seed in the calculation process to generate the new shared key, of which Figure 1-18 shows a functional diagram. The new keys are computed based on the key material and the shared key of the existing AAA security associations. Besides the key material, also an SPI and a calculation algorithm, such as HMAC-MD5 [11], is distributed via the AAA infrastructure.

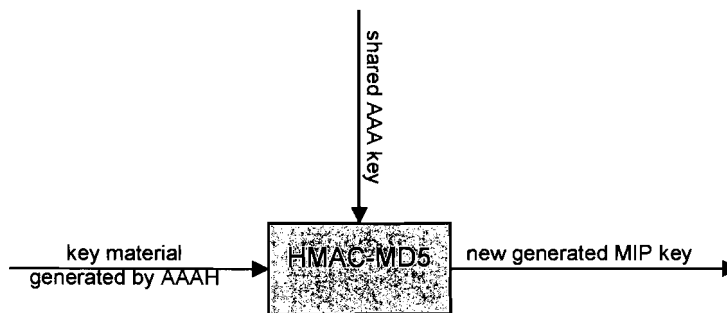


Figure 1-18 Key generation function

Basically this is how new MIP keys are generated. It is very useful when an MN attaches to an FN with an FA that does not already have a security association with the MN. An AAA security association between the AAAL and the AAAH server is the only required security association between the FN and the HN. The assumed existing security associations from which new are derived are shown in Figure 1-16 and Figure 1-17.

[25] specifies a similar method to generate new MIP security associations. The article describes how to use the existing security association between subscriber and mobile network operator, based on the subscriber's IMSI²⁰ and SIM²¹. A MN may identify itself with a NAI containing the IMSI. Furthermore, it uses a gateway between the internet AAA infrastructure and the AAA infrastructure of a GSM network. The advantage of this system is that this AAA infrastructure and many security associations between subscribers and mobile network operators already exist.

1.4.6 Mobile Network with a Mobile Router

The next step in the process towards all mobility is that routers will be mobile. The concept is very simple; however, the new possibilities are much greater. It allows mobile networking without awareness on the mobile network. In principle, hosts on the mobile network will not notice anything of the mobility. One could think of airplanes, ships, trains with an on-board LAN that is served by the Mobile Router (MR). Other examples are police cars or ambulances with LAN connected equipment.

²⁰ IMSI = International Mobile Subscriber Identifier, a globally unique number.

²¹ SIM = Subscriber Identity Module, the smartcard in a mobile terminal.

With mobile networking, one (or more) interface(s) of the MR will act as MN, see Figure 1-19. In the above examples, the interfaces will most likely be wireless: WLAN, GPRS, satellite communication.

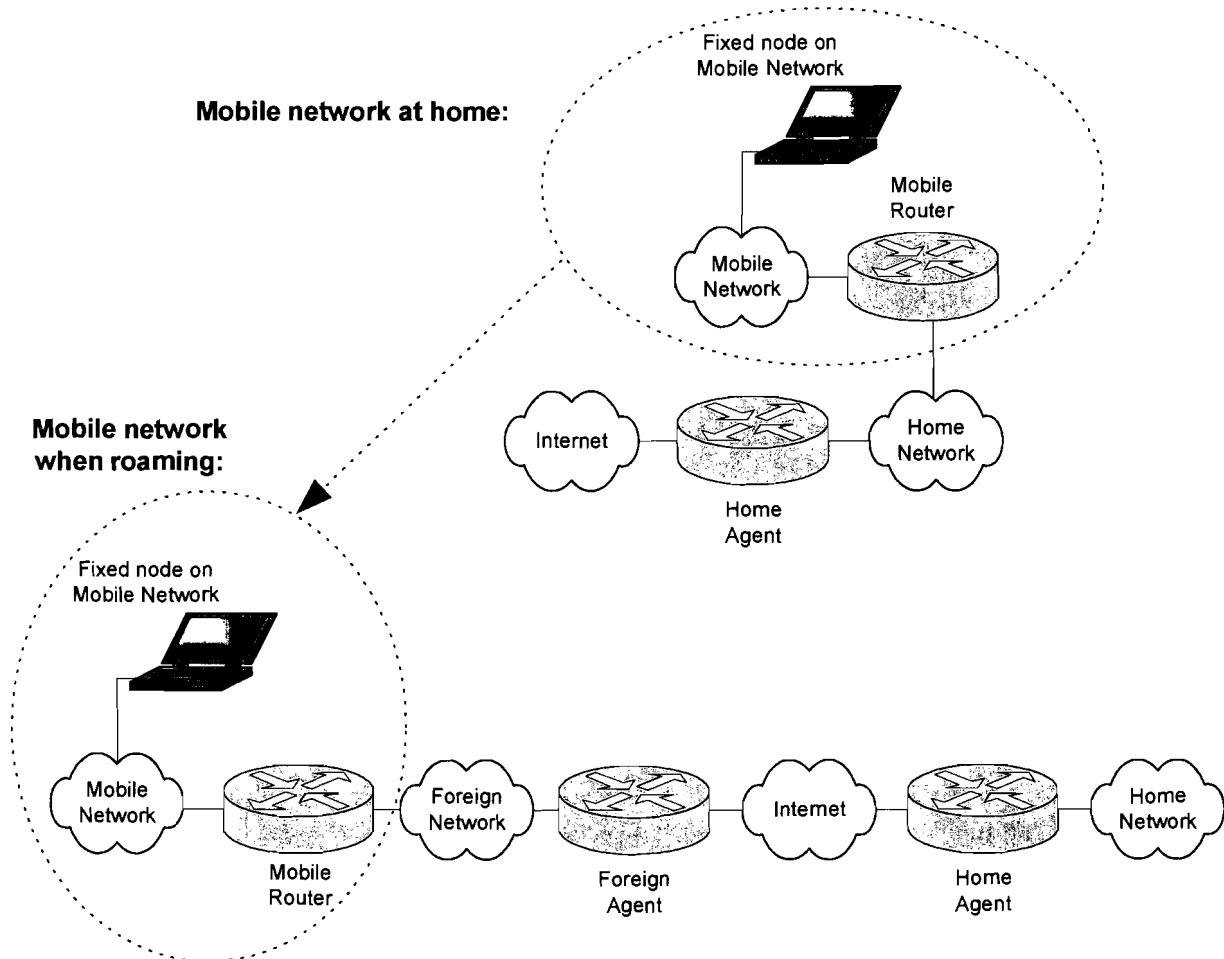


Figure 1-19 What is a mobile network?

In order to allow normal IP routing to the fixed hosts on the mobile network via the MR, an extra bi-directional tunnel from HA to MR is used. Without this MR-HA tunnel, packets coming from the HA-FA tunnel and destined for nodes on the Mobile Network will not be routed correctly towards the MR. The tunnel is set up after the tunnel to the COA address is set up and therefore goes through the HA-FA tunnel, as is drawn in Figure 1-20.

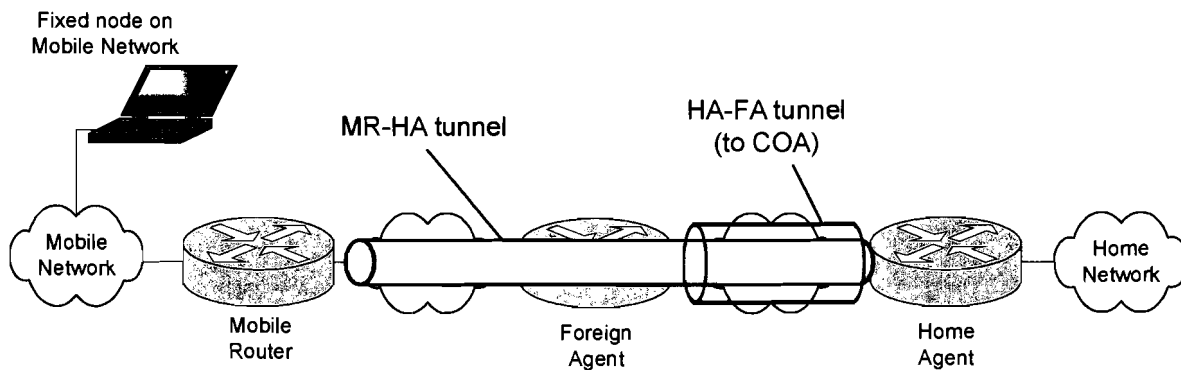


Figure 1-20 Extra MR-HA tunnel for routing

There is another solution to provide normal IP routing to the fixed hosts on the mobile network, but since it is more complex, it will not be discussed here.

The last step in mobility expansion is that the MR or a node on the mobile network acts as an FA for MNs roaming on the mobile network. This creates various new mobility scenarios.

1.5 Commercial Implementations

A growing part of Internet equipment vendors have implemented Mobile IP software in their products. Cisco Systems, supplier of most of Vodafone NL's IP equipment, has been supporting Mobile IP in Cisco IOS™ Software²² Releases 12.0(1)T and beyond. This means that any Cisco router can be configured as HA and/or FA. Other hardware vendors that implement MIP support are for example Lucent, Nortel, Motorola, 3Com.

As Mobile IPv4 is not implemented in the IPv4 standard, simply because the mobility support was added far after the standard was set, a mobile device has to use separate MIP software to be MIP enabled. For the PC, commercial MIP clients have been developed that work under MS Windows. Several MN client software packages are available at the time of writing this report. A small search over the Internet results in the products listed in Table 1-1. With this software, a mobile device, like a notebook computer, can be always online using LAN, WLAN, and GPRS for example.

Table 1-1 Available Mobile IP client software

vendor	Product	website
Lifix	Go!	http://www.lifix.fi
Birdstep	Intelligent Mobile IP	http://www.birdstep.com
Greenpacket	SONaccess	http://www.greenpacket.com
ipUnplugged	Roaming Client	http://www.ipunplugged.com
Netseal	Mobile Private Network Client	http://www.netseal.com

Note that some of the above MIP clients are integrated in more extensive software packs, which for example also implement VPN solutions.

²² Internetwork Operating System

1.6 Mobile IP in the Future

Mobile IP is still in development, that is, the basis has been “standardized” in RFCs, but there are still some problems to overcome. So has the problem to use an IPsec-based VPN in conjunction with MIP not been standardized yet. Another new feature in MIPv4 is MIP Proxy, which allows MIP-unaware nodes to be mobile, for example when roaming through a large WLAN infrastructure. This chapter did not describe all extra features defined by the Mobile IP Working Group, such as the “hierarchical MIP structure” or the “low latency handoffs”. It also did not mention the Mobile IPv6 specifications.

The main differences between MIPv4 and MIPv6, in favor of MIPv6, will be listed now, in order to gain a complete overview of the Mobile IP scope at this moment:

- What is known in MIPv4 as “route optimization” is built in as a fundamental part in MIPv6;
- MIPv6 route optimization can operate securely even without pre-arranged security associations;
- Support is also integrated into MIPv6 for allowing route optimization to coexist efficiently with routers that perform “ingress filtering”.

Chapter 2 Mobile IP Demonstration

2.1 Introduction

A part of the graduation project was to deliver a demonstration of a Mobile IP implementation. The demonstration should allow a Mobile Node (MN) to be always connected to the Internet using the Vodafone NL GPRS network and some WLAN access points at the Eindhoven University of Technology and Vodafone NL office. Furthermore, the MN should be able to use wired Ethernet connections. The deliverables were to show handoffs, if possible seamless. The handoffs should be managed automatically, based on criteria like availability, bandwidth and costs, or manually, so the user can overrule connectivity to its preferred interface.

2.2 Software

The Linux²³ Operating System was chosen as basis to build the MIP infrastructure for several reasons. Linux may be used without license fees and is available under the GNU²⁴ General Public License, which means that its source code is freely distributed and available to the general public. It is a system in which the user can fine-tune much more than for example in Microsoft Windows. It is a stable open source system, of which updates are very well spread over the Internet. One of the most easily accessible Linux distributions is Debian²⁵. Debian is (still being) developed by a large open community and still evolving. It is a package based system; each (set of) package(s) contains the code for a specific piece of software, which can be downloaded and installed separately. Together it forms the Debian distribution, which is based on the Linux kernel. The demo systems use Debian version 3.0 (named “Woody”) with Linux kernel 2.4.18.

A search on the Internet resulted in several available Mobile IP stacks that work with Linux. The names, together with some advantages and disadvantages are listed in Table 2-1. The choice was made to use HUT Dynamics v0.8.1 Mobile IP, because it is very well documented, there is an active mailing list, and it will certainly work under Debian with Linux kernel 2.4.18. An important disadvantage was inherent to this choice: the option of simultaneous bindings is not supported, so seamless handoffs could not be realized with this software.

²³ Linux is derived from Linus Torvalds (the man that built the first Linux kernel) and UNIX.

²⁴ GNU stands for “GNU’s Not Unix”, see <http://www.gnu.org>

²⁵ See <http://www.debian.org>

Table 2-1 Available Mobile IP stacks for Linux

stack:	pros:	cons:
Helsinki University of Technology: Dynamics Mobile IPv4 ²⁶ http://www.cs.hut.fi/Research/Dynamics/	<ul style="list-style-type: none"> • Works under Linux kernel 2.4.18 • Includes FA support • Supports MN decapsulation (co-located COA) • Supports reverse tunneling • No kernel modifications needed • Very well documented 	<ul style="list-style-type: none"> • No simultaneous bindings supported • "Old": not updated with recent features
Stanford University: Mosquitonet Mobile IP http://mosquitonet.stanford.edu/mip/	<ul style="list-style-type: none"> • Supports MN decapsulation (co-located COA) 	<ul style="list-style-type: none"> • Only 2.2.x Linux kernels, supported; needs kernel modifications • Does not include FA support
National University of Singapore: NUS MIPv4 for Linux http://opensource.nus.edu.sg/projects/mobileip/	<ul style="list-style-type: none"> • Includes FA support • Supports simultaneous bindings • Supports reverse tunneling 	<ul style="list-style-type: none"> • Works under Linux kernel 2.0.34

2.2.1 HUT Dynamics Overview

HUT Dynamics 0.8.1 contains three essential daemons²⁷, each for one of the MIP nodes: the MN daemon `dymnd`, the Foreign Agent (FA) daemon `dynfad` and the Home Agent (HA) daemon `dynhad`. The daemons are located on the Debian systems in `/usr/local/sbin/`. A daemon can be configured by a configuration (text) file containing the settings, like with most Debian parts. These files are `dymnd.conf`, `dynfad.conf`, respectively `dynhad.conf`, located in `/usr/local/etc/`. The Dynamics 0.8.1 software can be installed using the Debian packages, which are specially created for Debian users. Another option is downloading the source files and compiling it yourself. This last option was chosen for the demo. While daemons are running, some API commands can be given via the executables `dymn_tool`, `dynfa_tool`, respectively `dynha_tool`, also located in `/usr/local/sbin/`. On request, these executables also provide status information about the daemons and thus the MIP situation. To use pre-defined standard MIP configurations, one can run some simple set-up tools for MN, FA and HA. However, for the demo all settings were manually edited. Dynamics is supplied with good documentation; it has man pages for almost every daemon, executable and configuration file.

2.3 Demo Set-Up

The set-up of the demonstration took place in two steps. After the first set-up was implemented, an error sometimes occurred that was worked around by a change in the set-up.

²⁶ The engineers that created this research software are the people behind Lifix, which was mentioned in Table 1-1.

²⁷ A daemon is a process that runs in the background, doing automated processing.

The initial set-up will be discussed in section 2.3.1, the final set-up in section 2.3.2. The involved configuration files are given in Appendix D.

2.3.1 Initial Set-Up

As discussed in Chapter 1, for Mobile IP, a Home Network (HN) is needed. The Home Agent at the HN needs a public IP address, so that it can always be reached over the Internet. Because the campus network at the Eindhoven University of Technology provides a good IP infrastructure with few security related restrictions, this network was chosen to set up a HN. An IP range of eight public (i.e. globally unique) IP addresses were allocated for this project: 131.155.193.128-131.155.193.135. The architecture of the intended MIP demo set-up is shown in Figure 2-1. To keep the demonstration set-up simple, no FAs are used, so only co-located COAs will be used. The HA is a PC equipped with two Fast Ethernet network interfaces; one at the external and one at the internal side of the HN (131.155.193.128/29). This HN actually consists of only three nodes: (1) the WLAN Access Point (IEEE 802.11b, WiFi standard); (2) one interface of the HA; (3) the WLAN interface of the MN when it is at home.

The HA is a PC which functions as a router for the HN. To allow a fast and simple implementation in the ELE²⁸ network, the HA PC uses proxy ARP (explained in section 1.3.5) to route packets from the ELE network to the HN. From the ELE network it seems that the external interface of the HA PC, thus one layer-2 MAC address, has all internal IP addresses, besides its own IP address. In this way the ELE router does not need to change its IP routing tables with an entry for the HA PC as the default router for the HN. After setting up this proxy ARP feature for IP forwarding, a test was performed to check the functionality. This test is shown in the section “Proxy ARP with IP Forwarding” of Appendix F.

The nodes inside the HN are all configured with static IP addresses, which is done to keep the demonstration simple and tackle one problem at the same time. In principle this is not required for Dynamics MIP, because it supports dynamic HA discovery (discussed in section 1.4.3) and dynamic Home Address assignment using a MN NAI identifier (discussed in section 1.4.1). The HA has IP address 131.155.193.134, the WLAN AP 131.155.193.129, and the MN has a fixed Home Address 131.155.193.130 for its WLAN interface. The external interface of the HA PC has an IP address assigned by DHCP on the ELE network.

²⁸ Subnet of the faculty of Electrical Engineering

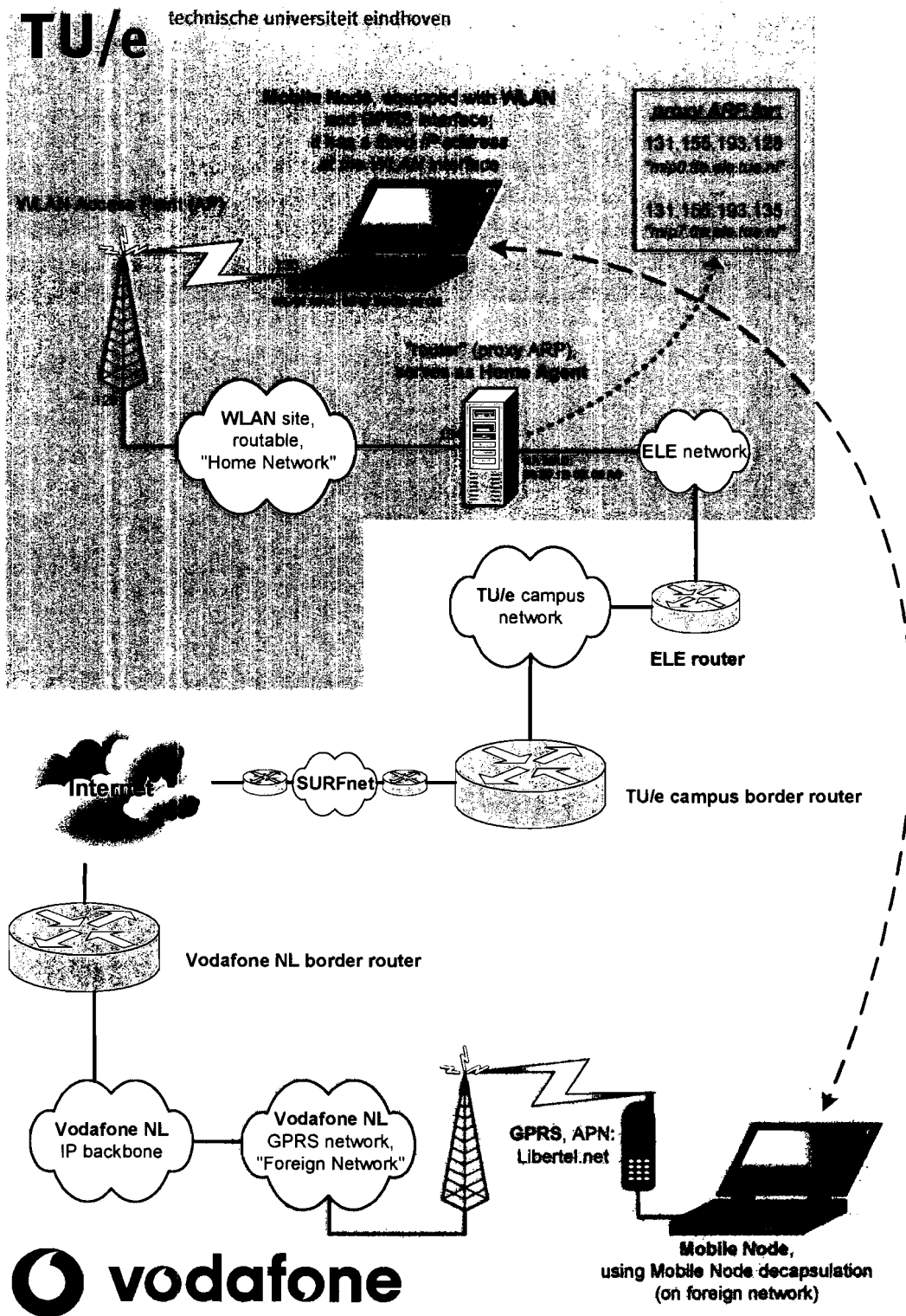


Figure 2-1 Architecture of initial MIP demo set-up

The TU/e campus network is protected with an ingress and egress filter at its border gateway, which is connected to SURFnet (the Dutch research and higher-educational backbone

network) which provides an Internet connection. Due to this filters, reverse tunneling should be used by a MN that is roaming outside the TU/e campus network.

The MN in this demo set-up can use various connections to FNs:

- At the ELE network:
 - A wired (Fast) Ethernet connection;
 - A wireless WiFi connection at different locations.
- At the Vodafone NL network:
 - A wired (Fast) Ethernet connection in the IP test lab;
 - A wireless WiFi connection at different points in the IP test lab;
 - A wireless GPRS connection²⁹, of which the APN terminates on the Internet.

The Vodafone NL IP test laboratory supplies public IP addresses. A MN needs a public IP COA address, i.e. not being behind a NAT, when no NAT traversal is supported in the MIP stack, like in Dynamics. The APN “libertel.net”³⁰ was specially designed by the Core IP team of Vodafone NL for this purpose.

Note that only the GPRS connection and the WLAN Access Point at the HN are shown in Figure 2-1 and Figure 2-2.

The network settings of the MN are configured such that the WLAN interfaces always connect to the same network. With Linux it is possible to configure a WLAN card to use different settings depending on the PCMCIA socket it is plugged into. It also is possible to make different settings for various WLAN cards, based on their MAC addresses. Using these (optional) configuration possibilities, the WLAN cards plugged into the MN are forced to use a specific WLAN Access Point. This is very useful, when various WLAN networks are accessible at the same location.



In the section “Initial Demo Set-Up” of Appendix D the configuration files are given for this set-up. The HA is configured to broadcast Agent Advertisements only on the internal interface (with IP address 131.155.193.134). This is how the MN may detect that it is at home. Furthermore, only MN-decapsulation (i.e. using co-located COAs) is configured, because no FAs are used. The MN is configured to use only reverse tunnels.

Problem

The set-up worked quite well according to the expectations. However, sometimes the HA system got in a very weird situation in which the IP forwarding did not work properly. Traces like the one described in the section “Proxy ARP with IP Forwarding” of Appendix F revealed that in that case, the HA did not forward IP packets from some arbitrary IP addresses to the HN, even when the session was initiated at the HN and the proxy ARP functionality worked correctly. One of those IP addresses was the default gateway of the ELE network. This meant that the MN could not communicate outside the ELE network, which is quite crucial.

Unfortunately, a reproducible case to generate the problem could not be found. It was suspected that this problem was caused by the proxy ARP and IP forwarding settings of the

²⁹ Appendix D contains the configuration files of the PPP connection that were created with `pppconfig` and edited for the GPRS connection in Debian.

³⁰  LIBERTEL was the former name of  **vodafone** in the Netherlands.

HA PC. After all, the HA functionality also uses proxy ARP and IP forwarding, which might conflict.

This problem was worked around by using a different set-up, in which the HA PC did not perform routing services apart from the MIP routing.

2.3.2 Final Set-Up

A sketch of the final demo set-up architecture is shown in Figure 2-2. Here the nodes of the former HN are connected directly to the ELE network. The IP addresses of those nodes were assigned statically; the eight “MIP allocated” public IP addresses could be used for this. All nodes retained the same IP address compared to the initial set-up. The big difference is that no extra router was placed between the nodes of the former HN and the ELE network. So, the HA PC does not have to perform other routing services than those inherent to MIP.

The configuration files of the HA and MN are given in the section “Final Demo Set-Up” of Appendix D. A little trick was used to make it seem that the HN is separate from the ELE network, at least for the MN.

When the MN connects with its home interface (i.e. the WLAN card with access to Access Point 131.155.193.129) to the ELE network, Linux is fine-tuned to configure this card with the fixed address 131.155.193.130 and the MIP software will recognize this as an address of the HN or as the MN Home Address, so the MN will think it is at its HN.

When the MN uses an interface that connects to the ELE network using a DHCP assigned IP address (thus not of the range 131.155.193.128-131.155.193.135), the MIP software will not recognize this address as one of the HN. Furthermore, in this final demo set-up the HA is configured not to broadcast Agent Advertisements, so the MN will not receive Home Agent Advertisements at the ELE network and think that it is at the HN. In this way, the ELE network will be considered as FN.

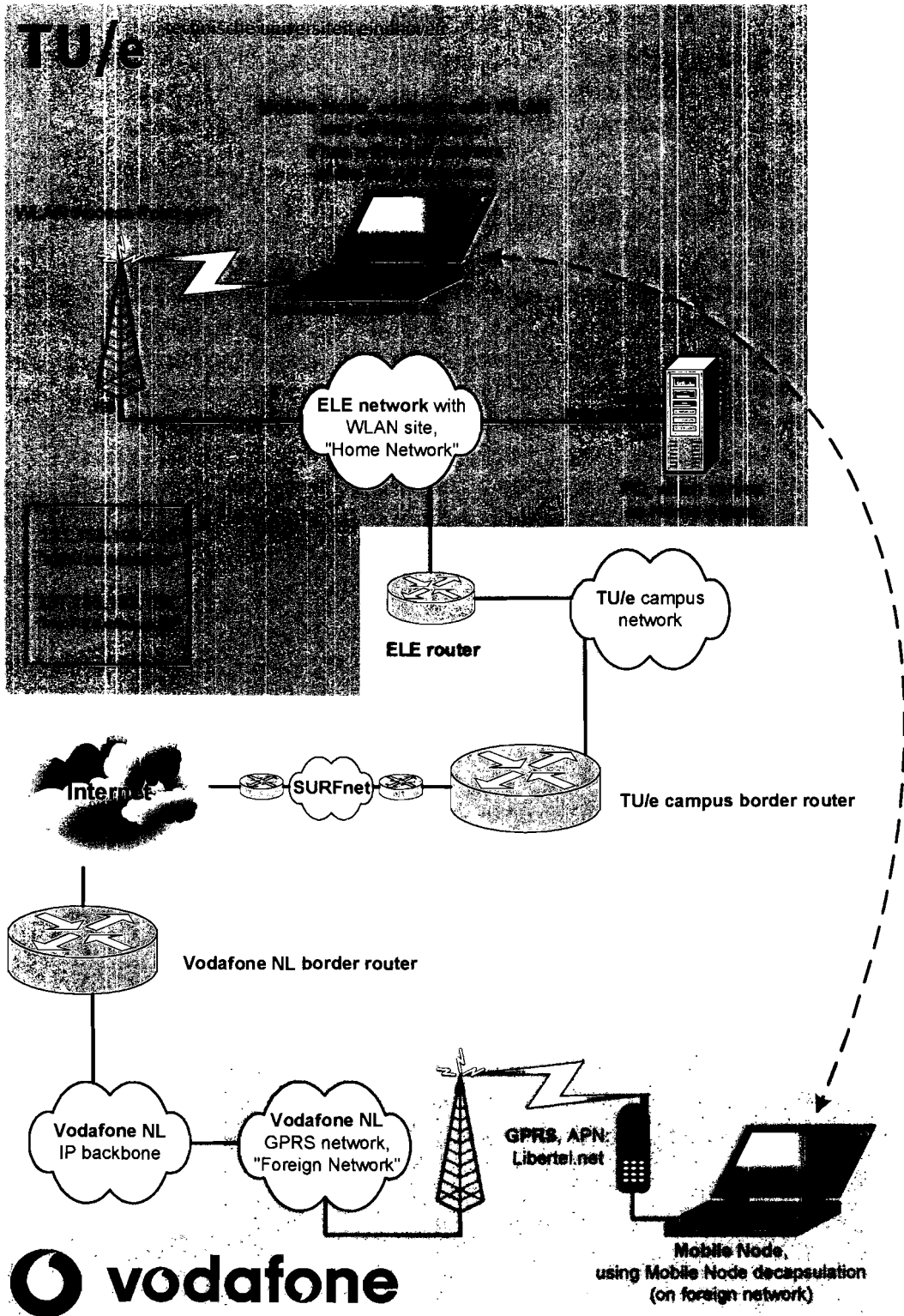


Figure 2-2 Architecture of final MIP demo set-up

2.3.3 HUT Dynamics Experiences

This section describes some experiences needed to know for understanding the handoff script of section 2.4. The Dynamics MIP routing is discussed in the next section. Furthermore, the commands to run and control Dynamics MIP are described in the subsequent section.

Dynamics MIP Routing

A PC running on Linux keeps an IP routing table. The routing table contains entries for destination nodes or networks and their corresponding gateway. A gateway 0.0.0.0 means that no gateway is necessary; the destination node or network is directly connected to the (sub)network of the related interface. The PC uses the routing table to look up the route when an IP packet is to be sent: it looks in the table from top to bottom for the destination node or network. The interface of the first match found will be used to send the packet over. When no match is found the default gateway is used. For the default gateway the destination is indicated with 0.0.0.0, which means “any node or network”. The default gateway in fact is the router that will forward the packet to the next network or router on its way to the destination node. An example of such a routing table for the Vodafone NL WLAN network is shown in Figure 2-3. The command to retrieve this table is `route -n`.

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
62.140.135.224 0.0.0.0 255.255.255.240 U 0 0 0 eth1
0.0.0.0 62.140.135.238 0.0.0.0 UG 0 0 0 eth1
```

Figure 2-3 Normal routing table example

There is only one entry for a destination network: 62.140.135.224 with subnet mask 255.255.255.240 (i.e. 62.140.135.224/28). This is the Ethernet to which the WLAN Access Point is connected. So an IP packet sent to a node on this network will be transmitted to the corresponding MAC address on `eth1`, the WLAN card. Other networks can be reached via gateway 62.140.135.238.

The routing discussion above is the introduction to how the routing with mobile IP is implemented, at least for Dynamics MIPv4. It will be explained according to the next example of Figure 2-4. This routing table is related to the situation above, but now MIP is active. Because a co-located COA is used, the MN has set-up a tunnel to its HA. This tunnel between MN and Mobility Agent is indicated with TUNLMNA.

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
131.155.193.134 62.140.135.238 255.255.255.255 UGH 0 0 0 eth1
62.140.135.224 0.0.0.0 255.255.255.240 U 0 0 0 eth1
0.0.0.0 0.0.0.0 0.0.0.0 U 0 0 0 TUNLMNA
```

Figure 2-4 Routing table example when using MIP

The routing table contains one unchanged entry; the one for the Ethernet network 62.140.135.224/28 on `eth1`. This means that packets sent to host on the same FN will be routed directly, without tunneling. This is compliant to the specification of RFC 3220 [6]. There are two new entries in the table. The first entry is the top entry that indicates that the HA 131.155.193.134 can be reached via gateway 62.140.135.238 on `eth1`. Registration messages send from MN to HA (remember that no FAs are used) will use this entry. The

second new entry fulfills the task of default gateway. It is related to the virtual interface TUNLMNA., with the MN Home Address (131.155.193.130) as source address. According to the routing table, the destination of this entry is any node or network. The gateway is 0.0.0.0 which means that every node is directly accessible at this interface. This looks strange, but is actually very simple. The TUNLMNA interface is a software interface that receives all packets sent to a destination address that does not match any other entry in the routing table. These packets are the packets to be tunneled to the HA. They use the MN Home Address (131.155.193.130) as source and the Correspondent Node (CN) address as destination. The TUNLMNA encapsulates the packets (HUT Dynamics MIP supports only IP within IP encapsulation): it places a new IP header in front of the original packet. This new header has the COA as source and the HA address as destination. Now, this new packet will be sent to the HA. According to the routing table the HA 131.155.193.134 can be reached via gateway 62.140.135.238 at eth1. The complete process is drawn in Figure 2-5.

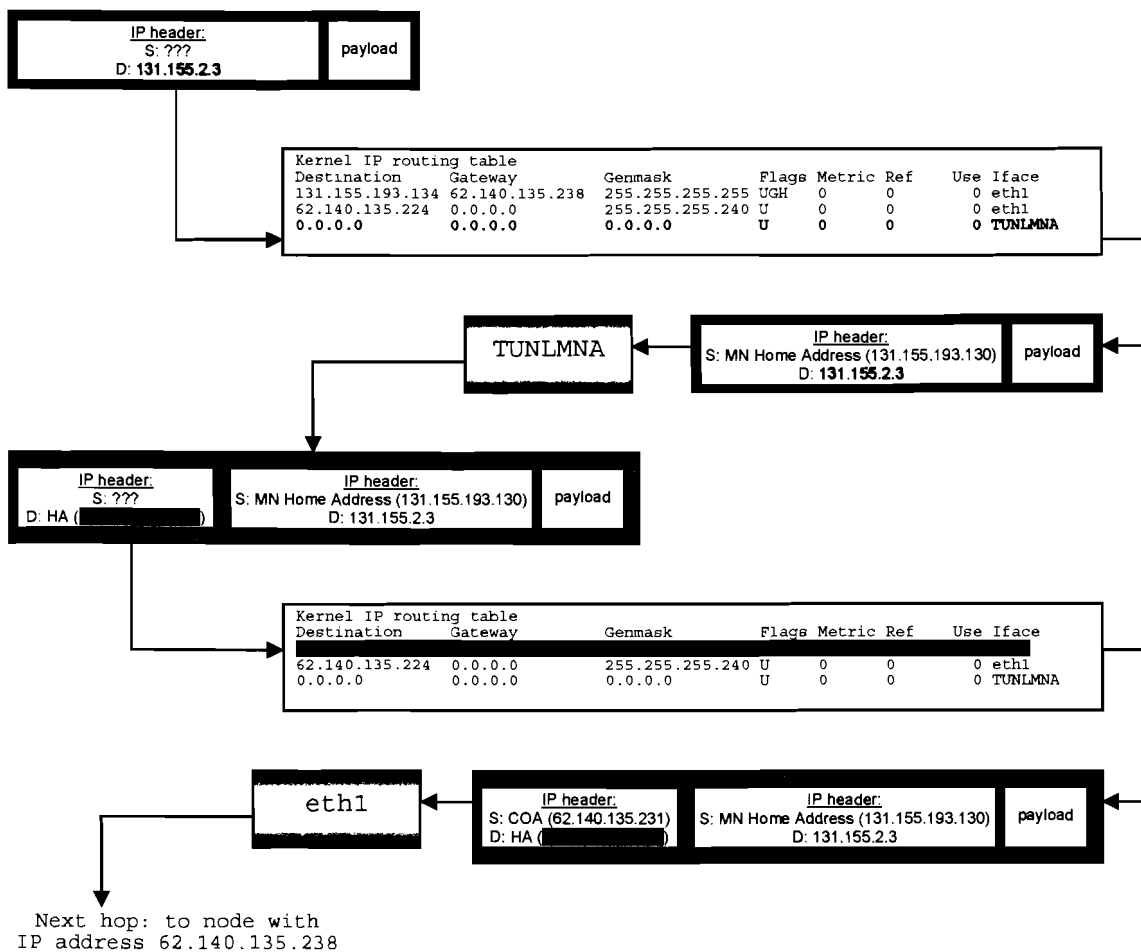


Figure 2-5 Routing at the MN using Dynamics

Proof of this routing functionality is given in section “Routing at MN Using Dynamics” of Appendix F, where ping packets are traced at interfaces TUNMNA and eth1, while using MIP.

Dynamics MIP Control Commands

The HUT Dynamics Mobile IP(v4) 0.8.1 implementation can be controlled by APIs. As mentioned before, for the MN this API is accessible via the program `/usr/local/sbin/dynmn_tool`. This section discusses some API commands used by the script that will be discussed in section 2.4. First, the HA should be started with the command `/usr/local/sbin/dynhad`, which automatically uses the configuration file `/usr/local/etc/dynhad.conf`. It is also possible to give a parameter that points to a different configuration file. When the MN is roaming on an FN, the following steps have to be taken to set up Dynamics MIP and register with the HA with a reverse tunnel (all API commands can be found in the manual pages included in the Dynamics software package):

1. Start the MN daemon with `/usr/local/sbin/dynmnd`, which automatically uses the configuration file `/usr/local/etc/dynmnd.conf`. It is also possible to give a parameter that points to a different configuration file. Status (to be retrieved with `/usr/local/sbin/dynmn_tool status`):

```
Mobile status:
state          Find Agent
local addr     131.155.193.130
co-addr        0.0.0.0
FA-addr        0.0.0.0
HA-addr        131.155.193.134
Home addr      131.155.193.130
tunnel is      down
tunneling mode full tunnel
info text      trying to connect
active devices 1
discarded msgs 0
```

2. The MN is now waiting for Agent Advertisements. However, since the MN is at an FN without FAs, it has to disconnect from this state in order to use a co-located COA. the command is `/usr/local/sbin/dynmn_tool disconnect`, resulting in the status:

```
Mobile status:
state          Disconnected
local addr     131.155.193.130
co-addr        0.0.0.0
FA-addr        0.0.0.0
HA-addr        131.155.193.134
Home addr      131.155.193.130
tunnel is      down
tunneling mode full tunnel
info text      trying to connect
active devices 1
discarded msgs 0
```

3. Now the MN-HA reverse tunnel can be set up with `/usr/local/sbin/dynmn_tool tunnel HA`; status:

```
Mobile status:
state          Connected
local addr     62.140.140.245
co-addr        62.140.140.245
FA-addr        131.155.193.134
HA-addr        131.155.193.134
Home addr      131.155.193.130
tunnel is      up
lifetime left   296s
tunneling mode full tunnel direct to HA
last request    4s ago; Fri Sep 27 17:16:36 2002
last reply      3s ago; Fri Sep 27 17:16:37 2002
reply code      0 - registration accepted
info text       connection established
last warning    connected - current_adv == NULL
active devices  1
discarded msgs  1
```

4. Now Mobile IP is fully running. The API allows giving interface updates, after which the MN tries to use the given interface, which is very useful for location updates. In this example only one interface is used: the GPRS ppp0 interface. The command is

```
/usr/local/sbin/dynmn_tool update ppp0;status:
```

```
Mobile status:
state           Connected
local addr     62.140.140.245
co-addr        62.140.140.245
FA-addr        131.155.193.134
HA-addr        131.155.193.134
Home addr      131.155.193.130
tunnel is      up
lifetime left  296s
tunneling mode full tunnel direct to HA
last request   4s ago; Fri Sep 27 17:17:24 2002
last reply     3s ago; Fri Sep 27 17:17:25 2002
reply code     0 - registration accepted
info text      connection established
last warning   connected - current_adv == NULL
active devices 1
discarded msgs 2
```

Steps 2 and 3 have to be made once to enter the MN-HA tunneling mode. These steps can only be taken on FNs. Then, only location updates (step 4) will serve for continuing Mobile IP usage. When the MN attaches to its HN and therefore gets its Home Address on the related interface, the daemon of course will not use the MN-HA tunneling.

Complications

Some small tests with HUT Dynamics Mobile IP, using different (W)LAN interfaces on the ELE and Vodafone NL networks plus an GPRS connection, resulted in the following complications:

1. According to the manual pages, there is a feature implemented in Dynamics to give a priority number to each Ethernet device. Dynamics will try to use the interface with the highest priority number. This feature was not tested in the demo set-up, because it only supports MAC address based interfaces, thus no ppp dial-up or GPRS connections. Besides, the intention was to make a more complex priority scheme;
2. When two interfaces connect to the same subnet, the Linux kernel only remembers the default gateway of the first connected interface. When the first interface is removed, Linux will not add the default gateway of the (remaining) second interface. Therefore the routing table will miss a default gateway. Dynamics MIP uses this standard Linux routing to install its complementary routing services (i.e. the route towards the HA and the TUNLMNA interface), so when the first interface is removed, Dynamics will not be able to install a route towards the HA;
3. When the MN daemon `dynmnd` is killed, no default gateway is left in the routing table, so the interfaces have to be re-initialized to correct IP routing;
4. Dynamics has an internal “location update” procedure for MNs. The update is triggered when the number of active devices (i.e. interfaces) changes. This number can be monitored via the status request (see the “Mobile status”-blocks in the previous section (“Dynamics MIP Control Commands”)). It changes for example when
 - A. a new interface is plugged in or got up;
 - B. an interface is pulled out or went down;

The location update re-registers the MN via a default interface, which probably is not the preferred interface.

Note that in case A the location update may be triggered before DHCP has assigned an IP

address. This is in conformance with the MIP specifications, because when using FAs, a local IP address does not have to be assigned. The interface may then use the MN Home Address as source.

The next section (2.4) discusses a script file that implements an automatic or manual handoff algorithm. The script also solves some of the above complications.

2.4 Handoff Script

The demo set-up is based on HUT Dynamics MIP. This software does not support the option of simultaneous bindings, so only one interface can provide IP connectivity at the same time. A script was created to use Dynamics on a MN. The intention of the script was to implement an automatic handoff procedure that decides which available interface to use, see Figure 2-6. The base for the algorithm can be found in [24]. It automatically selects the preferred interface, according to some criteria. Some advanced criteria are given in [1]. Furthermore, the script implements an option to overrule this automatic interface selection and manually select the preferred interface to use.

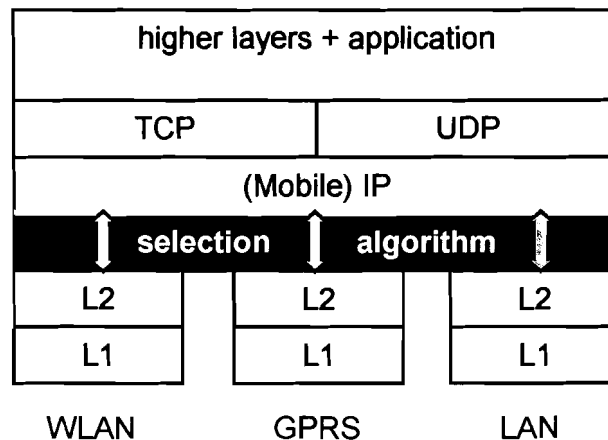


Figure 2-6 Selection algorithm chooses preferred interface

The script stores IP routing information of each Ethernet interface and ppp connection. With this information, the script can control the routing table. The script named MIPscriptMAN is given in Appendix E. It contains a lot of comment³¹ to explain the functionality. The script consists of the following parts:

1. Declaration of some variables and functions;
2. Initialization of the MIP process;
3. The “never-ending” loop for the automatic and manual interface selection.

The script reads the file MIPscriptPREF_FILE containing the manually given preferred interface. This file is written by the MIPscriptPREF script, also shown in Appendix E along with an example of MIPscriptPREF_FILE.

³¹ In Linux shells script, comment is preceded by a hash (#).

The IP routing information of each `eth` and `ppp` interface is stored in so-called variable arrays. The script allocates memory for at maximum 3 `eth` plus 3 `ppp` interfaces at the same time. Basically, the script works as follows:

- Every second it takes a snapshot of the IP routing configuration waiting for a change. The snapshot is stored in the variable `IP_ROUTE_SNAPSHOT`. A change means that an IP address, a subnet or default gateway has changed, for example due to an interface insertion or removal. The script reads the IP configuration with the command `ip route`, which only shows interfaces that already have been assigned an IP address. These interfaces can be used as co-located COAs.
- As soon as it detects a change in the configuration, the script will analyze it and will update to the (automatically chosen or manually given) preferred interface.
- Finally the administration is updated after which a new snapshot is taken to start over again.

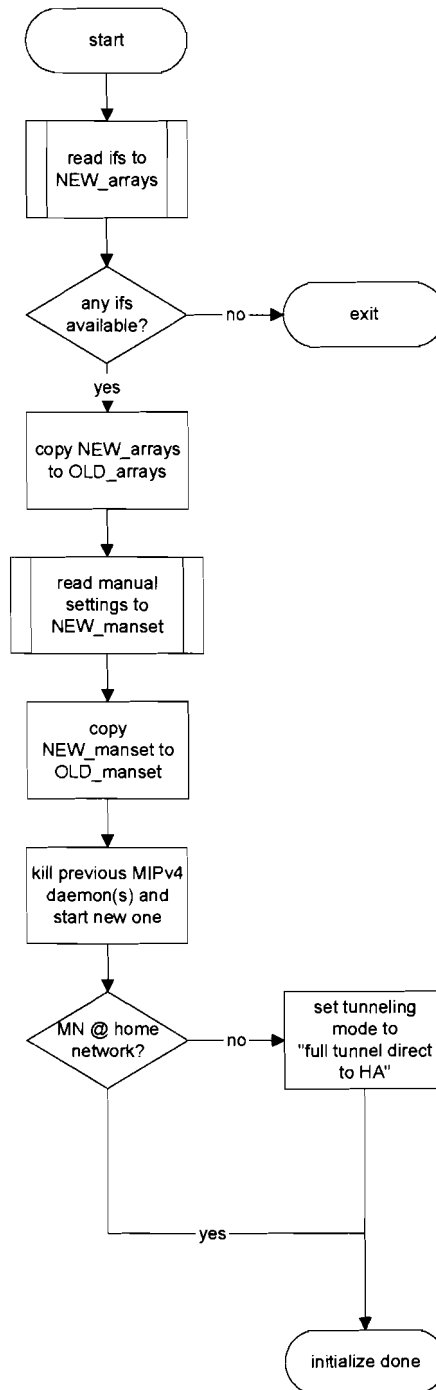
This process forms the “never ending” loop. The script can only be stopped using the escape command `CTRL+C`. A flowchart of the initialization part of the script is shown in Figure 2-7. The MN has to set the tunnel mode to “full tunnel to HA” to activate the co-located COA mode with reverse tunneling. The command is `dynmn_tool tunnel HA`, but can only be given when the MN is at an FN, otherwise `dynmn_tool` will hang. According to the manual, this is the result of a time-out which is set to infinity. The script therefore sets this tunneling mode as soon as the MN attaches to an FN for the first time. When a MN starts the script at its HN, the tunnel mode will not be set until the MN attaches to an FN.

The “never ending” loop is given in Figure 2-8. It implements the update to the preferred interface, which can be chosen automatically or manually.

Figure 2-9 (“read ifs to `NEW_arrays`”) presents the function `FILL_NEW_ARRAYS`, which reads the snapshot and writes the extracted IP routing information of all interfaces to the memory arrays.

Figure 2-10 shows how the manually given preference is implemented using the file `MIPscriptPREF_FILE`: the left flowchart is the principle of the separate script `MIPscriptPREF`; the right flowchart “read manual settings to `NEW_manset`” shows how `MIPscriptMAN` reads this information.

MIPscriptMAN "initialize"


 Figure 2-7 "Initialization phase" of handoff script³²

³² In the flowchart "ifs" stands for "interfaces".

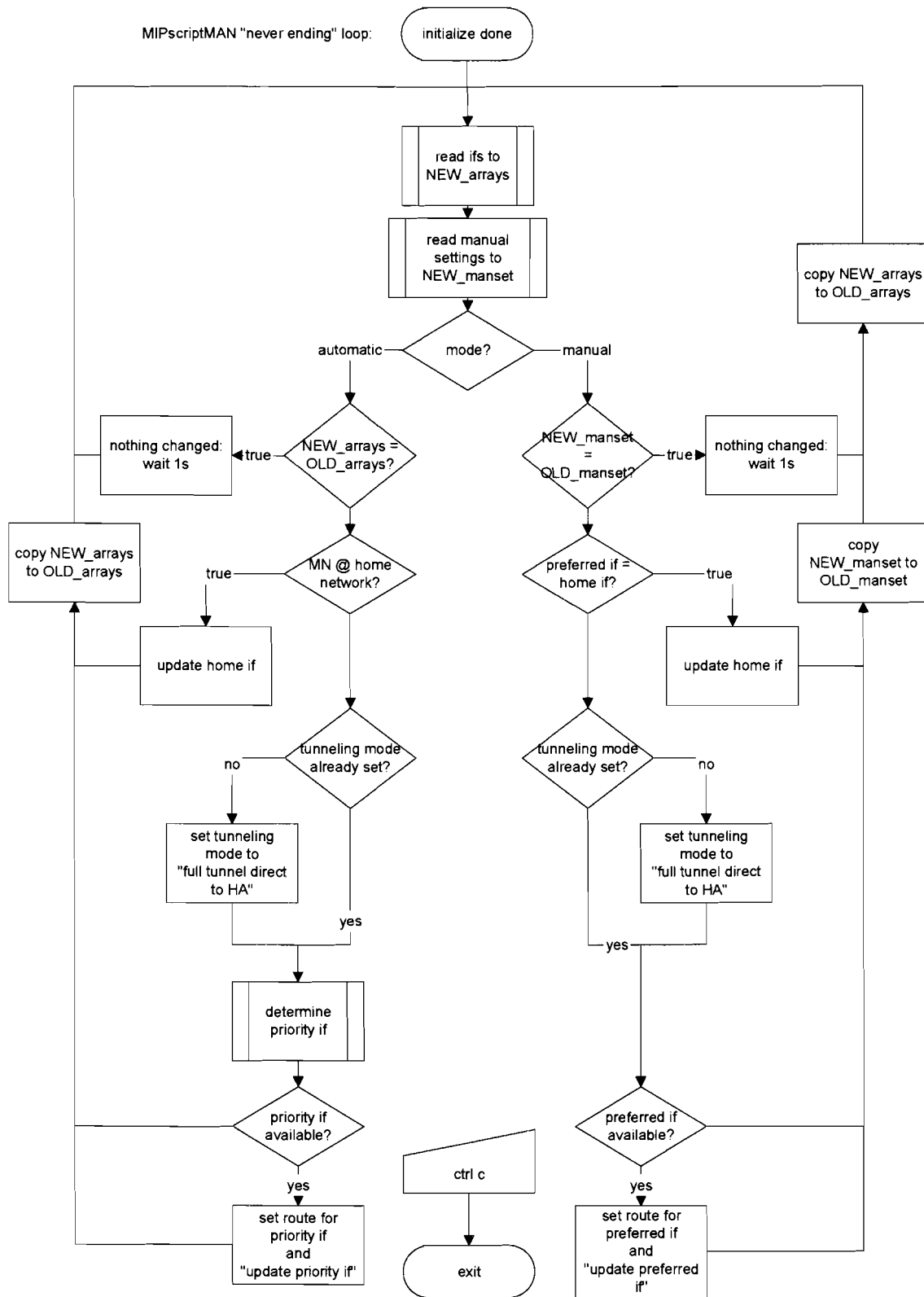


Figure 2-8 "Never ending loop" of handoff script

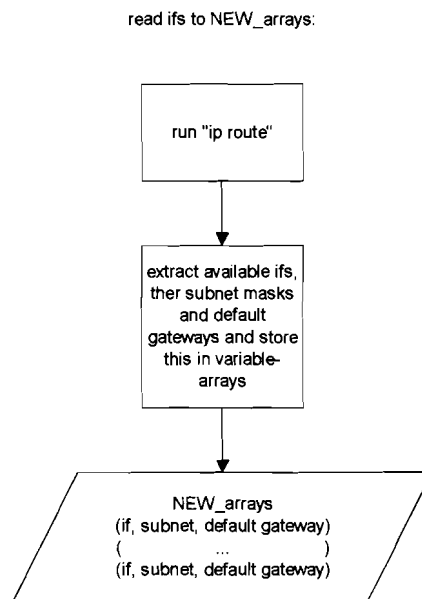


Figure 2-9 Filling the arrays

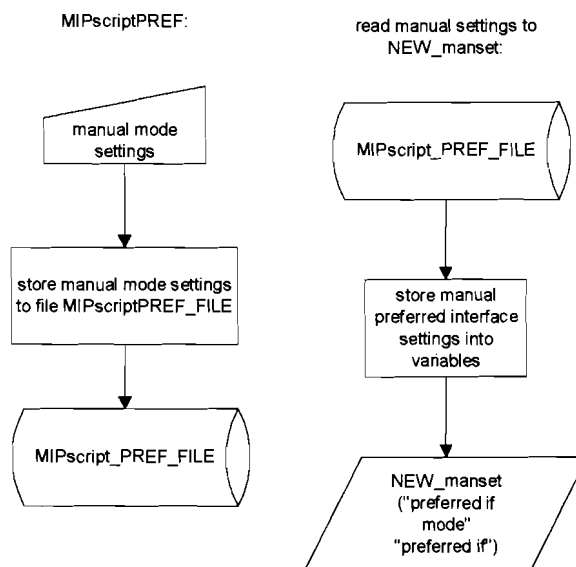


Figure 2-10 Manually given preference

While running the script a lot of status information is displayed. An example of such status information is shown in Figure 2-11. The status information results from the sequence: - start MIPscriptMAN in automatic mode, using the WLAN eth interface at the Vodafone NL testlab – activate the libertel.net GPRS ppp connection – pull out WLAN interface – put in the WLAN interface again. The different steps are separated in Figure 2-11 by means of shaded lines. With each shaded line the script starts a new cycle.

2.4.1 Priority Interface

In Figure 2-8 the function “determine priority if” is used. This block is responsible for the choice of the preferred interface in automatic selection mode. Through this interface the MN will set up the tunnel to its HA when it is roaming on FNs. Figure 2-12 shows the currently implemented algorithm. An eth interface always has a higher priority than a ppp interface. This is because the eth interface probably is cheaper (than for example a GPRS ppp interface) and has a higher throughput with a better QoS. Furthermore, if more than one interface of the same type exists, the algorithm will choose the newly added if possible or the first it encounters, i.e. the one with the lowest index number.

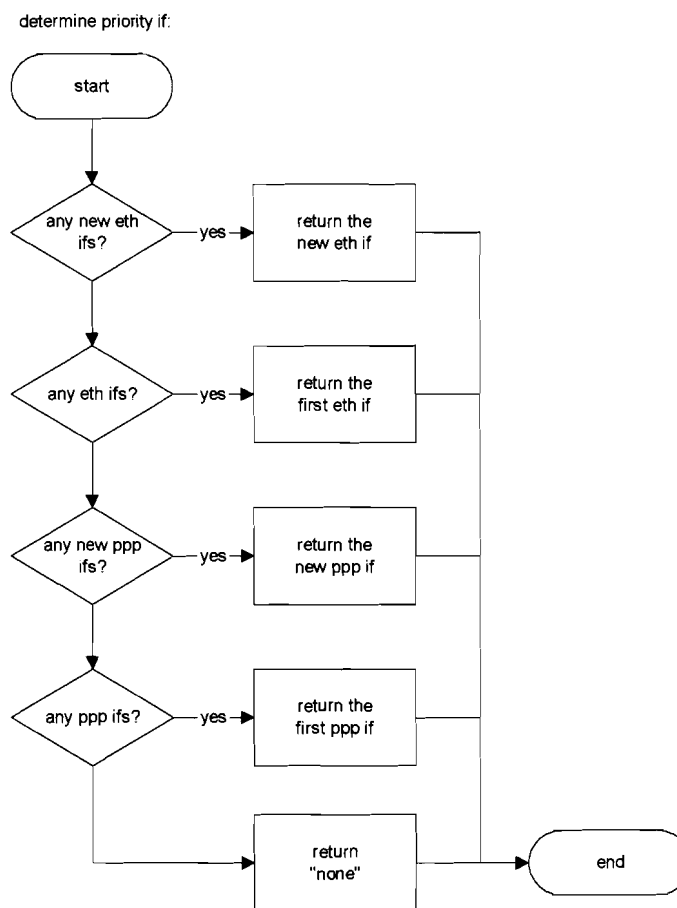


Figure 2-12 Algorithm to determine priority interface

2.4.2 Issues

The script discussed above still has some issues to be solved. One of those issues was already discussed under point 4 of the section “Complications” in section 2.3.3: the automatic location update procedure of the Dynamics implementation. This automatic update procedure may result in a double update, because the script will also try to update. However, the script will not try to update to the preferred interface until after an IP address was assigned. Sometimes the automatic update procedure wants to update to an interface that will not get an

IP address, because the DHCP server is slow or does not respond or because a WLAN interface is configured to use a WLAN that does not have radio coverage. The MN daemon will not go further until the DHCP client brings the interface down after a DHCP time-out.

The next issue makes it even more complicated. The DHCP client (the MN in the demo is configured with DHCP client `dhcp-client 2.0`) sometimes seems “confused”. When the MN does not receive a DHCP reply, for example because the DHCP server is slow or does not respond or because a WLAN interface is configured to use a WLAN that does not have radio coverage, the DHCP client sometimes wrongfully retrieves an outdated interface configuration, which it reminds from a formerly used network. Of course this outdated configuration will not work. It introduces an unwanted update by the automatic update procedure which cannot succeed. Installing a different DHCP client will probably solve this issue.

Due to this varying DHCP response times, the handoff times are not stable. After all, a handover can only be accomplished when an IP address has been assigned. Therefore it is almost impossible to give a significant value for this handoff time. In general, most handoff times are in between 0.5 and 5 seconds. Note that the simultaneous bindings option can solve this handoff time problem in most situations.

2.5 Future Work

To optimize the user value of the script, there are some things that should be solved or changed. The issues discussed above should be solved. The most important thing to improve is the function “choose priority if”, which runs an algorithm to select the preferred interface according to some criteria. In the current implementation, this function selects between interfaces only based on availability and type. Other useful criteria to select an interface or not are for example:

- a. Signal strength;
- b. Signal quality (SNR);
- c. Data throughput;
- d. Bit Error Rate (BER);
- e. Cost;
- f. Anticipation.

Points ‘a’ and ‘b’ both apply to radio networks and are very much related. In general, these layer 2 triggers are access technology dependent. The cost aspect ‘e’ points out that a Megabyte sent via WLAN probably will be cheaper than a Megabyte sent via GPRS or UMTS. An example of anticipation is that a MN should initiate a handoff when it senses a decreasing signal strength that will soon drop below threshold, see [1].

In future work the use of FAs should be tested. When the demo is built at a larger scale, a real HN can be implemented and the HA may use Agent Advertisements. Also some FNs should be implemented with FAs.

Furthermore, the Windows clients, which are available from Dynamics, can be used to test Dynamics MIP for Windows clients.

For Vodafone NL, the recommendation is to build an own MIP test network in the IP test laboratory. Most of their Cisco equipment can be set up as HA or FA in full compliance with the IETF specifications. So the option of simultaneous bindings will be available. With this

option, seamless handoffs are possible. Some of the software companies shown in Table 1-1 provide MIP client software which may be used on a trial basis for some restricted period. These clients are interesting if they support sufficient configuration options for a customized interface selection algorithm, just like the script for this Linux demo does.

Chapter 3 Conclusion and Recommendation

3.1 Conclusion

WLAN (e.g. WiFi) is a technology that will gain acceptance very fast in the Internet world, as it uses a license free radio spectrum, it is cheap, and it provides relatively fast network connections, compared to GPRS or an analogue modem connection over the PSTN. There are many different business models that can be applied, varying in who owns the WLAN site, who manages the site and who is allowed at the site in what way. At the so-called hotspots, WLAN is expected to provide a complementary service to GPRS and UMTS for mobile users.

Mobile IP (MIP) is very well suited for maintaining IP connectivity via various access technologies, like GPRS, Wireless LAN, Ethernet, UMTS in the near future, or whatever IP based access technology. Seamless handoffs can be made between different interfaces on different Foreign Networks (FNs) and using the simultaneous bindings option. Here the handoff time is zero. This enables world wide mobility. The mobile user only needs a one-time (client) software installation and configuration. The MN client software should implement an intelligent handoff algorithm, using the simultaneous bindings option to optimize handoffs.

There are two cases in which IP packets may get lost during MIP handoffs. The first case is when handoffs between different FNs using the same interface are made. Because an interface normally can only be connected to one network at the same time, no simultaneous bindings can be used here. The second case involves handoffs from the Home Network (HN) to a FN or vice versa. Normally, i.e. for Ethernet, it is impossible to be registered at an FN while using an interface at the Home Network, because of an ambiguity in the ARP mechanism. Both the MN and the HA would then be replying to ARP requests for the Home Address of the MN. Therefore, registrations at a FN cannot be made when the MN is at home and a seamless handoff from HN to FN or vice versa cannot be guaranteed.

However, this loss of IP packets should not be a problem, because IP is an unreliable (i.e. unacknowledged) connectionless network layer protocol. As long as the handoff time is shorter than the timeouts of higher layer connection-oriented protocols³³, there should be no problem.

Mobile IP itself is not sufficient to manage access at a WLAN hotspot, because it does not provide any data encryption, which probably is very desirable for business users. However, there are several products at the moment that already implement MIP within a VPN solution.

Mobile IP is an extension to the all accepted IP protocol that is still evolving; while writing this document, a new release of the specification was published in RFC 3344 [7]. The Mobile IP Working Group is still working on extra MIP options, as discussed in section 1.4. Other drafts deal with low latency handoffs and traversing IPsec-based VPN gateways.

The demonstration proved that the Dynamics MIP implementation works fine, also via GPRS, but that for real-time applications it is more useful to use an implementation with the

³³ E.g. the default for TCP is 30 seconds, but for many real-time applications this is unacceptable.

simultaneous bindings support. Furthermore, the handoff script with the intelligent interface selection procedure is very useful.

Appendix A List of References

- [1] Pahlavan K., Krishnamurthy P., Hatami A., Ylianttila M., Makela J., Pichna R. and Vallström J.
“Handoff in Hybrid Mobile Data Networks”
IEEE Personal Communications, Volume: 7 Issue: 2, Pages: 34-47, April 2000.
- [2] An N., Hu Y. and Sheriff R.
“A Handover Algorithm Support for Multimedia Service Provision in Heterogeneous Packet-Oriented Mobile Environments”
IEEE, in proc.: First International Conference on 3G Mobile Communication Technologies, 2000. (Conf. Publ. No. 471), Pages: 240 -244, 2000.
- [3] La Porta T., Salgarelli L. and Foster G.
“Mobile IP and Wide Area Wireless Data”
IEEE, in proc.: Wireless Communications and Networking Conference, 1999. WCNC, Volume: 3, Pages: 1528-1532, 1999.
- [4] Wietfeld C. and Gremmelmaier U.
“Seamless IP-based Service Integration across Fixed/Mobile and Corporate/Public Networks”
In proc.: Vehicular Technology Conference. 1999 IEEE 49th, Volume: 3, 1999, Pages: 1930-1934, 1999.
- [5] Perkins C.
“IP Mobility Support”
RFC 2002, October 1996.
Obsoleted by RFC 3220 [6].
- [6] Perkins C.
“IP Mobility Support for IPv4”
RFC 3220, January 2002.
This document contains some typos. While writing this report it was obsoleted by RFC 3344 [7].
- [7] Perkins C.
“IP Mobility Support for IPv4”
RFC 3344, August 2002.
- [8] Johnson D., Perkins C. and Arkko J.
“Mobility Support in IPv6”
work in progress, 1 June 2002.
Current version: <http://www.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-18.txt>
- [9] Postel J.
“INTERNET CONTROL MESSAGE PROTOCOL”
RFC 792, September 1981.

- [10] Deering S.
“ICMP Router Discovery Messages”
RFC 1256, September 1991.
- [11] Oehler M. and Glenn R.
“HMAC-MD5 IP Authentication with Replay Prevention”
RFC 2085, February 1997.
- [12] Perkins C.
“IP Encapsulation within IP”
RFC 2003, October 1996.
- [13] Perkins C.
“Minimal Encapsulation within IP”
RFC 2004, October 1996.
- [14] Li T., Hanks S., Meyer D. and Traina P.
“Generic Routing Encapsulation (GRE)”
RFC 2784, March 2000.
- [15] Montenegro G.
“Reverse Tunneling for Mobile IP, revised”
RFC 3024, January 2001.
- [16] Plummer D.
“An Ethernet Address Resolution Protocol”
RFC 826, November 1982.
- [17] Calhoun P. and Perkins C.
“Mobile IP Network Access Identifier Extension for IPv4”
RFC 2794, March 2000.
- [18] Srisuresh P. and Egevang K.
“Traditional IP Network Address Translator (Traditional NAT)”
RFC 3022, January 2001.
- [19] Levkowetz H. and Vaarala S.
“Mobile IP NAT/NAPT Traversal using UDP Tunnelling”
work in progress, 5 April 2002.
Used version: <http://www.ietf.org/internet-drafts/draft-ietf-mobileip-nat-traversal-02.txt>
- [20] Perkins C. and Johnson D.
“Route Optimization in Mobile IP”
work in progress, 6 September 2001.
*Used version: <http://www.ietf.org/internet-drafts/draft-ietf-mobileip-optim-11.txt>
This draft has been removed from the IETF website; a copy is enclosed in Appendix G.*

- [21] Perkins C. and Calhoun P.
“AAA Registration Keys for Mobile IP”
work in progress, 26 February 2002.
*Used version: <http://www.ietf.org/internet-drafts/draft-ietf-mobileip-aaa-key-09.txt>
This draft has been removed from the IETF website; a copy is enclosed in Appendix G.*
- [22] Glass S., Hiller T., Jacobs S. and Perkins C.
“Mobile IP Authentication, Authorization, and Accounting Requirements”
RFC 2977, October 2000.
- [23] Rigney C., Willens S., Rubens A. and Simpson W.
“Remote Authentication Dial In User Service (RADIUS)”
RFC 2865, June 2000.
- [24] Ylianttila M., Pichna R., Vallström J., Mäkelä J., Zahedi A., Krishnamurthy P. and Pahlavan K.
“Handoff Procedure for Heterogeneous Wireless Networks”
IEEE, in proc.: Global Telecommunications Conference, 1999. GLOBECOM '99,
Volume: 5, Pages: 2783-2787, 1999.
- [25] Haverinen H., Asokan N. and Määttänen T.
“Authentication and Key Generation for Mobile IP Using GSM Authentication and Roaming”
In proc.: IEEE International Conference on Communications, 2001. Volume: 8, Pages: 2453-2457, 2001.

NOTE *all RFCs can be found at the website of the IETF:
<http://www.ietf.org/rfc/rfcxxxx.txt> (where xxxxx is the 4-digit RFC number)*

Appendix B List of Acronyms

acronym	description
AAA	Authentication, Authorization and Accounting
AAAH	Home AAA server
AAAL	Local AAA server
API	Application Programming Interface
APN	Access Point Name
ARP	Address Resolution Protocol
AS	Autonomous System
BER	Bit Error Rate
BGW	Border GateWay
CN	Correspondent Node
COA	Care-Of Address
DHCP	Dynamic Host Configuration Protocol
EDGE	Enhanced Data for GSM Evolution
ETSI	European Telecommunications Standards Institute
FA	Foreign Agent
FH	FA-HA
FN	Foreign Network
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio System
GRE	Generic Routing Encapsulation
GRX	GPRS Roaming eXchange
GSM	Groupe Speciale Mobile, Global System for Mobile communications
HA	Home Agent
HN	Home Network
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMSI	International Mobile Subscriber Identification
IP	Internet Protocol
IRDP	ICMP Router Discovery Protocol
ISDN	Integrated Services Digital Network
LAN	Local Area Network
MAC	Medium Access Control
MAHO	Mobile Assisted HandOff
MCHO	Mobile Controlled HandOff
MF	MN-FA
MH	MN-HA
MIP	Mobile IP
MN	Mobile Node
MR	Mobile Router
MSISDN	Mobile Station ISDN Number
NAI	Network Access Identifier
NAT/NAPT	Network Address Translation/Network Address and Port Translation
NCHO	Network Controlled HandOff
PDP	Packet Data Protocol
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
RADIUS	Remote Authentication Dial In User Service
RFC	Request For Comments
RPF	Reverse Path Forwarding
SGSN	Serving GPRS Support Node
SIM	Subscriber Identity Module
SNR	Signal Noise Ratio

SPI	Security Parameter Index
TCP	Transmission Control Protocol
TTL	Time To Live
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System
UTRAN	UMTS Terrestrial Radio Access Network
WLAN	Wireless LAN
WLAN AP	WLAN Access Point
VoIP	Voice-over-IP
VPN	Virtual Private Network

Appendix C List of Figures

Figure 1-1 Protocol stack of a mobile user with three interfaces.....	6
Figure 1-2 Case 1 home situation.....	9
Figure 1-3 Case 1 foreign situation.....	10
Figure 1-4 Case 1 routing scheme.....	11
Figure 1-5 Case 2 foreign situation.....	12
Figure 1-6 Case 2 routing scheme.....	12
Figure 1-7 Fixed portion of Registration Request and Registration Reply message.....	14
Figure 1-8 Authentication extension.....	15
Figure 1-9 Authentication extension order.....	16
Figure 1-10 Registration Request and Reply.....	17
Figure 1-11 Tunneling scheme.....	19
Figure 1-12 Registration process and tunnel set up using reverse tunneling.....	20
Figure 1-13 Reverse tunneling routing scheme.....	21
Figure 1-14 NAPT example.....	24
Figure 1-15 MIP UDP tunnel registration.....	25
Figure 1-16 Interworking between AAA infrastructure and Mobile IP.....	27
Figure 1-17 Message exchange.....	28
Figure 1-18 Key generation function.....	29
Figure 1-19 What is a mobile network?.....	30
Figure 1-20 Extra MR-HA tunnel for routing.....	31
Figure 2-1 Architecture of initial MIP demo set-up.....	36
Figure 2-2 Architecture of final MIP demo set-up.....	39
Figure 2-3 Normal routing table example.....	40
Figure 2-4 Routing table example when using MIP.....	40
Figure 2-5 Routing at the MN using Dynamics.....	41
Figure 2-6 Selection algorithm chooses preferred interface.....	44
Figure 2-7 “Initialization phase” of handoff script.....	46
Figure 2-8 “Never ending loop” of handoff script.....	47
Figure 2-9 Filling the arrays.....	48
Figure 2-10 Manually given preference.....	48
Figure 2-11 Example of status information of MIPscriptMAN.....	49
Figure 2-12 Algorithm to determine priority interface.....	50

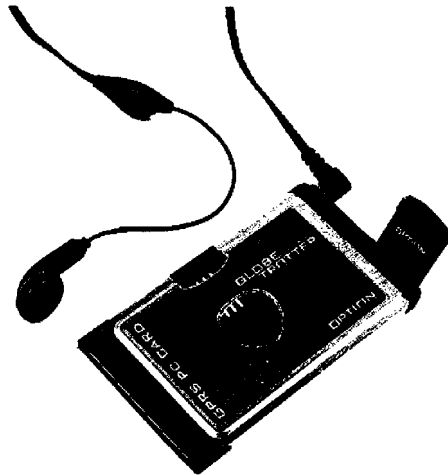
Appendix D Configuration of Dynamics MIP Software

This Appendix contains the following configuration files:

- The configuration files of the PPP connection that were created with `pppconfig` and edited for the GPRS connection in Debian (on MN);
- Dynamics Home Agent configuration file of initial demo set-up (on HA);
- Dynamics Mobile Node configuration file of initial demo set-up (on MN);
- Dynamics Home Agent configuration file of final demo set-up (on HA);
- Dynamics Mobile Node configuration file of final demo set-up (on MN).

GPRS Connection

The used MT was the Globetrotter GPRS PCMCIA from Option Wireless Technology³⁴, displayed below:



Option Globetrotter GPRS/GSM card

The configuration files of the PPP connection that were created with `pppconfig` and edited for the GPRS connection in Debian:

/etc/chatscripts/libnet

```
# This chatfile was generated by pppconfig 2.0.15.
# Please do not delete any of the comments. Pppconfig needs them.
#
# ispauth PAP
# abortstring
ABORT BUSY ABORT 'NO CARRIER' ABORT VOICE ABORT 'NO DIALTONE' ABORT 'NO DIAL TONE' ABORT 'NO
ANSWER' ABORT DELAYED
# modeminit
'' ATZ
```

```
# added by TomvS:
OK-AT-OK AT+CGDCONT=1,"IP","LIBERTEL.NET",,"1,1
```

```
# ispnumber
OK-AT-OK ATDT*99***1#
# ispconnect
```

³⁴ See <http://www.option.com>


```
CONNECT \d\c
# prelogin

# ispname
# isppassword
# postlogin

# end of pppconfig stuff
```

/etc/ppp/peers/libnet

```
# This optionfile was generated by pppconfig 2.0.15.
#
#
hide-password
noauth
connect "/usr/sbin/chat -v -f /etc/chatscripts/libnet"
debug
/dev/ttyS1
115200
defaultroute
noipdefault
user nettest
remotename libnet
ipparam libnet

usepeerdns

# added by TomvS:
lcp-echo-interval 0
```

Furthermore, to prevent errors during connection set-up the Van Jacobsen compression was disabled in `/etc/ppp/options`.

Initial Demo Set-Up

Dynamics Home Agent configuration file of initial demo set-up:
`/usr/local/etc/dynhad.conf`

```
# $Id: dynhad.conf,v 1.39 2001/10/20 13:36:07 jm Exp $
# Home Agent configuration file
#
# Dynamic hierarchical IP tunnel
# Copyright (C) 1998-2001, Dynamics group
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License version 2 as
# published by the Free Software Foundation. See README and COPYING for
# more details.
#
#####
#
# NOTE!
#   This is an example configuration file designed to give
#   perspective to the system configuration AND to provide
#   a basis for a working simple test environment.
#   The values of some of the parameters may not be the
#   same as the daemon's defaults, so don't get confused.
#
#####
#
# Interfaces to be used for Mobile IP services. Note that you have to configure
# each interface that may receive or send registration messages.
# interface: name of the interface, e.g. eth0
# ha_disc:
#   0 = do not allow dynamic HA discovery
#   1 = allow dynamic HA discovery with broadcast messages
```

```
# agentadv:
# 0 = do not send agent advertisements without agent solicitation
# 1 = send agent advertisements regularly
# -1 = do not send any (even solicited) agent advertisements
# interval: number of seconds to wait between two agentadv
#         (if allowed for this interface)
# force_IP_addr: local address to be forced for this interface
#                (can be used to select one of the multiple virtual
#                addresses); if not entered, the primary address of the
#                interface is used
INTERFACES_BEGIN
# interface  ha_disc  agentadv  interval  force_IP_addr
eth0        0        -1        10
eth1        0        1         10        131.155.193.134
INTERFACES_END

# Network Access Identifier (NAI) of this HA
# Unique identifier for this HA. A macro [interface] can be used to get
# the hardware address of an interface in dot-separated format.
# This is needed, if private address space is used in the home network.
# NetworkAccessIdentifier "[eth0]@example.com"

# Surrogate HA IP Address
# This is only needed, if private address space and a surrogate HA are used in
# the home network.
# SHAIPAddress 10.10.10.10

# Private HA Identifier at SHA
# Unique identifier (32-bit number) at SHA for this private HA.
# This is only needed, if private address space and a surrogate HA are used in
# the home network.
# PrivateHAIdentifier 1

# UDP port to listen for registration requests
# The default is 434
UDPPort 434

# Socket priority for signaling sockets (UDP) can be set with SO_PRIORITY to
# allow easier QoS configuration. If this argument is set, the given value is
# used as a priority for the signaling socket. E.g. CBQ class can be used to
# make sure that signaling is not disturbed by other traffic on a congested
# link.
# This feature is still undocumented and can be left commented.
#
# SocketPriority 1

# MaxBindings can be used to restrict the maximum number of Mobile Nodes
# that are concurrently attached to this Home Agent.
# The default is 20.
MaxBindings 20

# The default tunnel lifetime is suggested also by the HA.
# The default lifetime is 500.
HADefaultTunnelLifetime 500

# The Registration error reply interval should be restricted to
# avoid system overloading situations when receiving too much
# incorrect Registration Reply messages.
# The default value for RegErrorReplyInterval is 1 second.
RegErrorReplyInterval 1

# Triangle tunnel means that the packages to MNs are send via the HA, but
# packages from MN are routed directly (i.e. FA use normal IP routing).
# EnableTriangleTunneling < TRUE | FALSE >
EnableTriangleTunneling FALSE

# Reverse tunnel means bi-directional tunneling in which both the packages
# from and to MN are send via HA
# EnableReverseTunneling < TRUE | FALSE >
EnableReverseTunneling TRUE

#####
# The Home Agent needs to know what kind of security parameters each
# authorized Mobile Node uses. that is why there is a tbale that maps
# (in many-to-many relationship) SPI numbers, or SPI-number ranges to
# IP adresses - or IP-address ranges defined by network adresses and
```

```
# netmasks. The netmask may be defined in two ways: either in
# "bit offset notation" (the third row in the example) or in the
# "dotted decimal notation" (the fifth row in the example below).
# The list of Mobile Node information is separated between two
# keywords: AUTHORIZEDLIST_BEGIN and AUTHORIZEDLIST_END.
#
# < SPI | SPI-range          IP | network/netmask  >
# Example:

AUTHORIZEDLIST_BEGIN
# SPI          IP
#1000          192.168.240.2
#1001          192.168.240.3
#1002          0.0.0.0/0
#11000-11999   192.168.241.4
#12000         192.168.250.0/255.255.255.0
#13000-14000   192.168.251.0/28
<censored>     131.155.193.130
AUTHORIZEDLIST_END

# The Home Agents needs a security association for each authorized Mobile
# Node. The association includes following information.
#
# SPI (Security Parameter Index): a key for the other fields.
#
# Authentication Algorithm:
#   1: MD5/prefix+suffix (a.k.a. keyed-MD5) [RFC 2002]
#   4: HMAC-MD5 [RFC 2104]
#   5: SHA-1 [FIPS 180-1]
#   6: HMAC-SHA1 [RFC 2104]
# Note! MD5/prefix+suffix has known weaknesses and use of HMAC-MD5 is
# recommended. MD5/prefix+suffix algorithm is for backwards compatability with
# older versions that do not support more secure HMAC-MD5.
#
# Replay Protection Method:
#   0: none
#   1: timestamps
#   2: nonces
#
# Timestamp tolerance indicates how many seconds the MN's timestamp can differ
# from the HA's clock. 7 seconds is the recommended default value. This
# tolerance is checked only when timestamps are used for replay protection.
#
# The maximum lifetime for the binding is given in seconds.
# Special case: 65535 (or more) seconds means unlimited time (the binding will
# not expire)
#
# Shared Secret: a secret data known by MN and HA. It can be given as
# a HEX code string, i.e. two characters (0-F) correspond to one octet.
# The shared secret can also be given as a character string (e.g.
# "ABCDE" corresponds to 4142434445).
# Note: RFC 2002 specifies that the default key size is 128 bits (i.e.
# 16 bytes or 32 hex 'characters'). Dynamics supports also other key lengths.
#
# The SPI is the key identifier for the rest of the security parameters
# on the same line. SPI number ranges may be assigned the same security
# parameters.
#
# The list of Mobile Node information is separated between two
# keywords: SECURITY_BEGIN and SECURITY_END.
#
SECURITY_BEGIN
#   auth.   replay  timestamp      max      shared
# SPI   alg.   meth.  tolerance     lifetime  secret
<censored>4   1      20           600      <censored>
#1002   4      2           60       120      01020304050607
#10000  4      1           60       300      016A352B2F235E
#10001  4      1          120       180      0EF42BD234ECCAA2
SECURITY_END
#
#####
# Home Agent may have optional security associations with Foreign
# Agents. If the security association exists the session key can be
# encrypted with the help of shared secret and thus man-in-the-middle
# style attacks can be prevented. If no security association is set
# for a certain Foreign Agent - Home Agent pair, public key encryption
```

```

# (RSA) is used.
#
# When private address space is used, this list must have a security
# association with the surrogate HA instead of the FAs. Possible security
# associations with the FAs are then configured to the SHA.
#
# The following list contains the shared secrets indexed by SPI (and
# Foreign Agent IP address). The algorithm field specifies the method
# used for authentication and key distribution:
#   1: MD5/prefix+suffix (a.k.a. keyed-MD5) [RFC 2002]
#   4: HMAC-MD5 [RFC 2104]
#   5: SHA-1 [FIPS 180-1]
#   6: HMAC-SHA1 [RFC 2104]
# The format of the share secret field is identical to the one used with the
# MN-HA security association list above.
#
FA_SECURITY_BEGIN
# SPI      FA IP      Alg.      Shared Secret
#2001      192.168.0.1  4         0123456789ABCDEF
#2002      192.168.0.2  4         "eslkfj89jr3hduh3R!as"
FA_SECURITY_END
#
# The Highest FA public key can be protected from man-in-the-middle style
# attacks between the HFA and the HA with hash code. The use of this hash
# is optional, but recommended. The HA can have different ways of checking
# the hash code.
# Methods:
#   0: skip the hash code completely (not recommended)
#   1: if the hash code is received, check the public key with it
#   2: require the correct hash code for every registration message
#       with a public key (this may prevent the use of some organizations
#       which do not advertise the hash code)
PublicKeyHashMethod 1
#
#####

# The log messages are written through syslog service. The facility to be
# used defaults to LOG_LOCAL0, but it can be set with this parameter
# to any of the possible facilities (LOG_AUTHPRIV, LOG_DAEMON, and so on).
# The processing of log messages is defined in /etc/syslog.conf file.
SyslogFacility LOG_DAEMON

# Home Agents (and Foreign Agents) use unix domain sockets
# to communicate through their API interfaces.
# The group and owner must be names as strings, no groupIDs or userIDs are
# allowed. The file permissions are set in octal values like in chmod(1)
# The configuration parameters of the two API sockets are as follows:
HAAPIReadSocketPath "/var/run/dynamics_ha_read"
HAAPIReadSocketGroup "root"
HAAPIReadSocketOwner "root"
HAAPIReadSocketPermissions 0766
#
HAAPIAdminSocketPath "/var/run/dynamics_ha_admin"
HAAPIAdminSocketGroup "root"
HAAPIAdminSocketOwner "root"
HAAPIAdminSocketPermissions 0700
#
# Every configuration file must end to the keyword 'END'.
END

```

Dynamics Mobile Node configuration file of initial demo set-up:
/usr/local/etc/dynmnd.conf

```

# $Id: dynmnd.conf,v 1.56 2001/10/20 13:36:07 jm Exp $
# Mobile Node configuration file
#
# Dynamic hierarchial IP tunnel
# Copyright (C) 1998-2001, Dynamics group
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License version 2 as
# published by the Free Software Foundation. See README and COPYING for

```

```
# more details.
#
#####
#
# NOTE!
#   This is an example configuration file designed to give
#   perspective to the system configuration AND to provide
#   a basis for a working simple test environment.
#   The values of some of the parameters may not be the
#   same as the daemon's defaults, so don't get confused.
#
#   To get a minimal test working, you will need to check the
#   following items:
#     * MNHomeIPAddress
#     * HAIPAddress
#     * EnableFADecapsulation
#     * HomeNetPrefix (if using FA decapsulation or
#       dynamics HA address resolution)
#     * SPI and SharedSecret
#   The rest of the items should work with their preset values in
#   most cases and they can be used to fine tune the operations
#   after the basic operation have been tested successfully.
#
#####
#
# The Mobile Nodes's IP address in the Home Network.
# If using AAA (see UseAAA below), home address can be set to 0.0.0.0 in order
# to request a home address from the AAA infrastructure. This requires that
# also MN NAI is configured.
MNHomeIPAddress 131.155.193.130

# The Mobile Node's Network Access Identifier (NAI) [RFC2794]
# If configured, this NAI is used in registration requests to identify the
# mobile user for AAA services.
#
# MNNetworkAccessIdentifier "user@example.com"

# UseAAA < TRUE | FALSE >. TRUE enables AAA extensions (key requests using
# material from AAA, HA and home address discovery using AAA, etc.). This
# requires that MN NAI and AAA related items below are configured.
# FALSE disables these extensions.
UseAAA FALSE

# The IP address of Mobile Node's Home Agent. In case of a private HA address
# this is the address of the surrogate HA. If the HA address is unknown, set
# this to 0.0.0.0 and make sure that HomeNetPrefix is correct for dynamic
# HA address resolution or use AAA to discover HA address. If the HA has
# multiple interfaces, this should be the address of the "public" interface,
# i.e., the one toward default gateway (it has to be reachable from the foreign
# networks).
HAIPAddress 131.155.193.134

# If the HA has more than one interfaces, HAIPAddress should be configured to
# be the one reachable from the Internet (i.e., from the foreign networks the
# MN may visit). To allows MN to detect other HA's interfaces, their IP
# addresses may be configured here. MN will use this list in addition to
# HAIPAddress when determining whether an agent advertisement is from its own
# HA (i.e., when MN is at home). Multiple lines containing different addresses
# may be used to configure more than one alternative HA address.
# AlternativeHAIPAddress 10.1.2.3
AlternativeHAIPAddress 131.155.193.112

# AllowHomeAddrFromForeignNet < TRUE | FALSE >. TRUE allows AAA to assign
# a home agent and home address from the foreign network (assuming they are
# set to 0.0.0.0 above). FALSE means that both the home agent and the home
# address must be from the home domain.
AllowHomeAddrFromForeignNet FALSE

# The following configuration options PrivateHAIPAddress, PrivateHAIdentifier,
# and HANetworkAccessIdentifier are only used with home networks that use
# private IP addresses and a surrogate HA. In other cases they should be left
# commented.

# The private IP address of Mobile Node's Home Agent.
# Needed only, if surrogate HA is used.
# PrivateHAIPAddress 192.168.200.200
```

```
# The identifier for the private HA in SHA (unique 32-bit number)
# PrivateHAIdentifier 1

# Home Agent Network Access Identifier (NAI)
# If configured, this NAI is used to match the HA agent advertisements when
# a MN is determining whether it is at home or not. This is mainly used with
# private HA address that may not be globally unique.
#
# HANetworkAccessIdentifier "ha@example.com"

# EnableFADecapsulation < TRUE | FALSE >. TRUE enables a mode where
# the FA decapsulates the IP-within-IP encapsulated IP packets.
# FALSE disables this mode and sets the default mode where the
# MN decapsulates the IP-within-IP encapsulated IP packets.
# With FA decapsulation the MN uses its home address in the interface even in
# the foreign network and with MN decapsulation MN needs to acquire a
# co-located care-of address from the visited network (this needs an external
# program; see man pages for more information).
# The two modes cannot be used simultaneously.
EnableFADecapsulation FALSE

# Network address of home network (CIDR format: a.b.c.d/prefix_length)
# This is used with FA decapsulation and dynamics HA address resolution. If
# commented, the routing entry is not removed nor added. The home net entry
# may optionally be used with MN decapsulation - see MNDecapsRouteHandling
# option below.
#
# Example: 192.168.242.0/24
HomeNetPrefix 131.155.193.128/29

# Home net default gateway
# This entry can be used to force a gateway that the MN uses when it is
# at home. If this is left commented, the MN tries to use the default route
# that was in use when the program was started.
#
HomeNetGateway 131.155.193.134

#####
# a SPI (Security Parameter Index) must be defined for every MN.
# It is used for indexing the security association at the Home Agent.
SPI <censored>
#
# The SharedSecret is provided as a HEX number string. The shared secret can
# also be given as a character string
# (e.g. character string "ABCDE" corresponds to HEX number string 4142434445).
# Note: RFC 2002 specifies that the default key size is 128 bits (i.e.
# 16 bytes or 32 hex 'characters'). Dynamics supports also other key lengths.
# This shared secret is used with the HA. This must be commented out when using
# AAA infrastructure for key generation. In this case, the AAA related items
# below must be configured.
# SharedSecret < shared secret >
# SharedSecret 016A352B2F235E
SharedSecret <censored>
#
# Authentication algorithm
# 1: MD5/prefix+suffix (a.k.a. keyed-MD5) [RFC 2002]
# 4: HMAC-MD5 [RFC 2104]
# 5: SHA-1 [FIPS 180-1]
# 6: HMAC-SHA1 [RFC 2104]
# Note! MD5/prefix+suffix has known weaknesses and use of HMAC-MD5 is
# recommended. MD5/prefix+suffix algorithm is for backwards compatibility with
# older versions that do not support more secure HMAC-MD5.
AuthenticationAlgorithm 4
#
# Replay prevention method:
# 0: none
# 1: time stamps
# 2: nonces
ReplayMethod 1
#
# Mobile Node may have optional security associations with Foreign
# Agents. If the security association exists an additional Mobile Node -
# Foreign Agent Authentication Extension is added to the registration requests.
#
# The following list contains the shared secrets indexed by SPI (and
```

```
# Foreign Agent IP address). The algorithm field specifies the method
# used for key distribution (see the list above). The format of the share
# secret field is identical to the one used with the MN-HA security
# association list above.
#
FA_SECURITY_BEGIN
# SPI      FA IP      Alg.  Shared Secret
#2001     192.168.0.1  4     0123456789ABCDEF
#2002     192.168.0.2  4     "eslkfj89jr3hdhuh3R!as"
FA_SECURITY_END

# MN-AAA Authentication and Challenge/Response [RFC3012]

# If the MN does not have a security association with an FA, it may use AAA
# infrastructure for authentication. If this is used, also MN NAI
# ('MNNetworkAccessIdentifier' above) should be configured.

# SPI to be used in MN-AAA authentication.
# Reserved SPI values:
# 2 = CHAP_SPI, CHAP style authentication using MD5 [RFC 3012]
# 3 = MD5/prefix+suffix [draft-ietf-mobileip-aaa-key-03.txt]
# 4 = HMAC MD5 [draft-ietf-mobileip-aaa-key-03.txt]
# MN-AAA-SPI 12345

# Shared secret for MN-AAA authentication (see 'SharedSecret' above for format
# instructions)
# MN-AAA-SharedSecret "test"

# Algorithms to be used for MN-AAA authentication and key generation
# 1 = MD5/prefix+suffix (RFC 2002)
# 2 = RADIUS authentication (Sec. 8 of RFC 3012)
# 3 = MD5/prefix+suffix (RFC 2002) (alias for 1 above)
# 4 = HMAC-MD5 (Sec. 6 of RFC 3012; RFC 2104)
# 5 = SHA-1 (FIPS 180-1)
# 6 = HMAC-SHA1 (RFC 2104)
# Note: with algorithm 2, 'MN-AAA-SPI' should be set to reserved number
# CHAP_SPI (default: 2).
# MN-AAA-AuthenticationAlgorithm 4
# MN-AAA-KeyGenerationAlgorithm 4

#####
# TunnelingMode < 1 | 2 | 3 | 4 >
# The packets between the MN and a Correspondent Node (CN) can be routed using
# different routes. This option can be used to select, which mode will be
# selected.
# Possible values:
# 1 = automatic, prefer reverse tunnel (i.e. bi-directional tunnel)
# 2 = automatic, prefer triangle tunnel (i.e. tunnel only in CN->MN direction)
# 3 = accept only reverse tunnel
# 4 = accept only triangle tunnel
TunnelingMode 3

# When MN can get its own co-located care-of address and use reverse tunneling,
# the normal method is to set the default route to the tunnel. This means that
# all the packets destined to other networks than the current subnet in the
# visited network are send via the HA. If the co-located COA is public, it can
# be used for sessions that do not need constant IP address (e.g. most of the
# web browsing). The following configuration option specifies the routing
# operation that is used with the co-located COA.
# Possible values:
# 0 = set default route to the tunnel
# 1 = set only the home net route to the tunnel (the above HomeNetPrefix
#     options must be set)
# 2 = do not change the routing entries (i.e. some external means must be
#     used to direct traffic to the tunnel, e.g. manually adding host route
#     to a specific host)
MNDecapsRouteHandling 0

# DefaultTunnelLifetime is the lifetime suggested in registration
# The lifetime is defined in seconds, default value is 300.
# The request timer will be set according to this value. If the FA's agent
# advertisement has a smaller time, it is used instead.
# Special case: 65535 (or more) seconds means unlimited time (the binding will
# not expire)
```

```
# MNDefaultTunnelLifetime [ seconds ]
MNDefaultTunnelLifetime 300

# UDP port to be used for sending registration requests
# Port 434 is allocated for Mobile IP signaling and this should not be changed
# unless the network is known to use some other port (i.e. all the FAs and HAS
# must have the same port configured).
UDPPort 434

# Socket priority for signaling sockets (UDP) can be set with SO_PRIORITY to
# allow easier QoS configuration. If this argument is set, the given value is
# used as a priority for the signaling socket. E.g. CBQ class can be used to
# make sure that signaling is not disturbed by other traffic on a congested
# link.
# This feature is still undocumented and can be left commented.
#
# SocketPriority 1

# The log messages are written through syslog service. The facility to be
# used defaults to LOG_LOCAL0, but it can be set with this parameter
# to any of the possible facilities (LOG_AUTHPRIV, LOG_DAEMON, and so on).
# The processing of log messages is defined in /etc/syslog.conf file.
SyslogFacility LOG_DAEMON

# Ignore these interfaces. No agent advertisements are received nor
# agent solicitations sent for these interfaces.
IGNORE_INTERFACES_BEGIN
lo
dummy0
tun10
gre0
IGNORE_INTERFACES_END

# Other programs may set routing entries so that the data connection may
# fail. The MN can try to enforce the routes that it believes should be used.
# This operation should currently be used only with FA decapsulation. If the
# route enforcement is activated the MN daemon prevents certain route changes.
EnforceRoutes FALSE

# MN can be instructed to poll for current AP address when using a wireless
# LAN driver that supports wireless extensions. This can be used to speed up
# handoffs when using managed mode (BSS).
# Polling interval is configured in micro seconds
# (i.e., 1000000 equals to 1 second)
# -1 = AP polling disabled
APPollingInterval -1

# MN can be instructed to send periodic agent solicitations to find new FAs.
# Normally, MN uses agent solicitations when it does not have a valid agent
# advertisement. Periodic solicitation occurs even if the connection seems to
# be up. This will cause more broadcast messages and is thus disabled in the
# default configuration, but it can speed up handoffs in some environments.
# Solicitation interval is configured in micro seconds (usec)
# (i.e., 1000000 usec equals to 1 second). A random time between 0 and 0.5
# second will be added to solicitation intervals to prevent unwanted
# synchronization of broadcast messages. In addition, solicitations will not be
# send more often than once per second, so this interval should not be
# configured to be less than 1000000 usec.
# -1 = Periodic agent solicitation disabled
SolicitationInterval -1

#####
# Mobile Nodes use unix domain sockets to communicate through their API
# interfaces.
# The group and owner must be names as strings, no groupIDs or userIDs are
# allowed. The file permissions are set in octal values like in chmod(1).
# The configuration parameters of the two API sockets are as follows:
MNAPIReadSocketPath "/var/run/dynamics_mn_read"
MNAPIReadSocketGroup "root"
MNAPIReadSocketOwner "root"
MNAPIReadSocketPermissions 0666
#
MNAPIAdminSocketPath "/var/run/dynamics_mn_admin"
MNAPIAdminSocketGroup "root"
MNAPIAdminSocketOwner "root"
MNAPIAdminSocketPermissions 0700
```



```
#
# Every configuration file must end to the keyword 'END'.
END
```

Final Demo Set-Up

Dynamics Home Agent configuration file of final demo set-up: */usr/local/etc/dynhad.conf*

```
# $Id: dynhad.conf,v 1.39 2001/10/20 13:36:07 jm Exp $
# Home Agent configuration file
#
# Dynamic hierarchial IP tunnel
# Copyright (C) 1998-2001, Dynamics group
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License version 2 as
# published by the Free Software Foundation. See README and COPYING for
# more details.
#
#####
#
# NOTE!
#   This is an example configuration file designed to give
#   perspective to the system configuration AND to provide
#   a basis for a working simple test environment.
#   The values of some of the parameters may not be the
#   same as the daemon's defaults, so don't get confused.
#
#####
#
# Interfaces to be used for Mobile IP services. Note that you have to configure
# each interface that may receive or send registration messages.
# interface: name of the interface, e.g. eth0
# ha_disc:
#   0 = do not allow dynamic HA discovery
#   1 = allow dynamic HA discovery with broadcast messages
# agentadv:
#   0 = do not send agent advertisements without agent solicitation
#   1 = send agent advertisements regularly
#  -1 = do not send any (even solicited) agent advertisements
# interval: number of seconds to wait between two agentadvs
#           (if allowed for this interface)
# force_IP_addr: local address to be forced for this interface
#                (can be used to select one of the multiple virtual
#                addresses); if not entered, the primary address of the
#                interface is used
INTERFACES_BEGIN
# interface ha_disc agentadv interval force_IP_addr
eth0      0      -1      10      131.155.193.134
INTERFACES_END

# Network Access Identifier (NAI) of this HA
# Unique identifier for this HA. A macro [interface] can be used to get
# the hardware address of an interface in dot-separated format.
# This is needed, if private address space is used in the home network.
# NetworkAccessIdentifier "[eth0]@example.com"

# Surrogate HA IP Address
# This is only needed, if private address space and a surrogate HA are used in
# the home network.
# SHAIPAddress 10.10.10.10

# Private HA Identifier at SHA
# Unique identifier (32-bit number) at SHA for this private HA.
# This is only needed, if private address space and a surrogate HA are used in
# the home network.
# PrivateHAIdentifier 1

# UDP port to listen for registration requests
```

```
# The default is 434
UDPPort 434

# Socket priority for signaling sockets (UDP) can be set with SO_PRIORITY to
# allow easier QoS configuration. If this argument is set, the given value is
# used as a priority for the signaling socket. E.g. CBQ class can be used to
# make sure that signaling is not disturbed by other traffic on a congested
# link.
# This feature is still undocumented and can be left commented.
#
# SocketPriority 1

# MaxBindings can be used to restrict the maximum number of Mobile Nodes
# that are concurrently attached to this Home Agent.
# The default is 20.
MaxBindings 20

# The default tunnel lifetime is suggested also by the HA.
# The default lifetime is 500.
HADefaultTunnelLifetime 500

# The Registration error reply interval should be restricted to
# avoid system overloading situations when receiving too much
# incorrect Registration Reply messages.
# The default value for RegErrorReplyInterval is 1 second.
RegErrorReplyInterval 1

# Triangle tunnel means that the packages to MNs are send via the HA, but
# packages from MN are routed directly (i.e. FA use normal IP routing).
# EnableTriangleTunneling < TRUE | FALSE >
EnableTriangleTunneling FALSE

# Reverse tunnel means bi-directional tunneling in which both the packages
# from and to MN are send via HA
# EnableReverseTunneling < TRUE | FALSE >
EnableReverseTunneling TRUE

#####
# The Home Agent needs to know what kind of security parameters each
# authorized Mobile Node uses. that is why there is a table that maps
# (in many-to-many relationship) SPI numbers, or SPI-number ranges to
# IP addresses - or IP-address ranges defined by network addresses and
# netmasks. The netmask may be defined in two ways: either in
# "bit offset notation" (the third row in the example) or in the
# "dotted decimal notation" (the fifth row in the example below).
# The list of Mobile Node information is separated between two
# keywords: AUTHORIZEDLIST_BEGIN and AUTHORIZEDLIST_END.
#
# < SPI | SPI-range      IP | network/netmask >
# Example:

AUTHORIZEDLIST_BEGIN
# SPI      IP
#1000      192.168.240.2
#1001      192.168.240.3
#1002      0.0.0.0/0
#11000-11999 192.168.241.4
#12000      192.168.250.0/255.255.255.0
#13000-14000 192.168.251.0/28
<censored> 131.155.193.130
AUTHORIZEDLIST_END

# The Home Agents needs a security association for each authorized Mobile
# Node. The association includes following information.
#
# SPI (Security Parameter Index): a key for the other fields.
#
# Authentication Algorithm:
# 1: MD5/prefix+suffix (a.k.a. keyed-MD5) [RFC 2002]
# 4: HMAC-MD5 [RFC 2104]
# 5: SHA-1 [FIPS 180-1]
# 6: HMAC-SHA1 [RFC 2104]
# Note! MD5/prefix+suffix has known weaknesses and use of HMAC-MD5 is
# recommended. MD5/prefix+suffix algorithm is for backwards compatability with
# older versions that do not support more secure HMAC-MD5.
#
```

```
# Replay Protection Method:
# 0: none
# 1: timestamps
# 2: nonces
#
# Timestamp tolerance indicates how many seconds the MN's timestamp can differ
# from the HA's clock. 7 seconds is the recommended default value. This
# tolerance is checked only when timestamps are used for replay protection.
#
# The maximum lifetime for the binding is given in seconds.
# Special case: 65535 (or more) seconds means unlimited time (the binding will
# not expire)
#
# Shared Secret: a secret data known by MN and HA. It can be given as
# a HEX code string, i.e. two characters (0-F) correspond to one octet.
# The shared secret can also be given as a character string (e.g.
# "ABCDE" corresponds to 4142434445).
# Note: RFC 2002 specifies that the default key size is 128 bits (i.e.
# 16 bytes or 32 hex 'characters'). Dynamics supports also other key lengths.
#
# The SPI is the key identifier for the rest of the security parameters
# on the same line. SPI number ranges may be assigned the same security
# parameters.
#
# The list of Mobile Node information is separated between two
# keywords: SECURITY_BEGIN and SECURITY_END.
#
SECURITY_BEGIN
#      auth.      replay  timestamp      max      shared
# SPI   alg.      meth.   tolerance     lifetime  secret
<Sensored>4      1      20            600      <Sensored>
#1002   4         2         60         120      01020304050607
#10000  4         1         60         300      016A352B2F235E
#10001  4         1         120        180      0EF42BD234ECCAA2
SECURITY_END
#
#####
# Home Agent may have optional security associations with Foreign
# Agents. If the security association exists the session key can be
# encrypted with the help of shared secret and thus man-in-the-middle
# style attacks can be prevented. If no security association is set
# for a certain Foreign Agent - Home Agent pair, public key encryption
# (RSA) is used.
#
# When private address space is used, this list must have a security
# association with the surrogate HA instead of the FAs. Possible security
# associations with the FAs are then configured to the SHA.
#
# The following list contains the shared secrets indexed by SPI (and
# Foreign Agent IP address). The algorithm field specifies the method
# used for authentication and key distribution:
# 1: MD5/prefix+suffix (a.k.a. keyed-MD5) [RFC 2002]
# 4: HMAC-MD5 [RFC 2104]
# 5: SHA-1 [FIPS 180-1]
# 6: HMAC-SHA1 [RFC 2104]
# The format of the share secret field is identical to the one used with the
# MN-HA security association list above.
#
FA_SECURITY_BEGIN
# SPI      FA IP          Alg.      Shared Secret
#2001      192.168.0.1      4         0123456789ABCDEF
#2002      192.168.0.2      4         "eslkfj89jr3hdh3R!as"
FA_SECURITY_END
#
# The Highest FA public key can be protected from man-in-the-middle style
# attacks between the HFA and the HA with hash code. The use of this hash
# is optional, but recommended. The HA can have different ways of checking
# the hash code.
# Methods:
# 0: skip the hash code completely (not recommended)
# 1: if the hash code is received, check the public key with it
# 2: require the correct hash code for every registration message
#     with a public key (this may prevent the use of some organizations
#     which do not advertise the hash code)
PublicKeyHashMethod 1
#
```

```
#####  
# The log messages are written through syslog service. The facility to be  
# used defaults to LOG_LOCAL0, but it can be set with this parameter  
# to any of the possible facilities (LOG_AUTHPRIV, LOG_DAEMON, and so on).  
# The processing of log messages is defined in /etc/syslog.conf file.  
SyslogFacility LOG_DAEMON  
  
# Home Agents (and Foreign Agents) use unix domain sockets  
# to communicate through their API interfaces.  
# The group and owner must be names as strings, no groupIDs or userIDs are  
# allowed. The file permissions are set in octal values like in chmod(1).  
# The configuration parameters of the two API sockets are as follows:  
HAAPIReadSocketPath "/var/run/dynamics_ha_read"  
HAAPIReadSocketGroup "root"  
HAAPIReadSocketOwner "root"  
HAAPIReadSocketPermissions 0766  
#  
HAAPIAdminSocketPath "/var/run/dynamics_ha_admin"  
HAAPIAdminSocketGroup "root"  
HAAPIAdminSocketOwner "root"  
HAAPIAdminSocketPermissions 0700  
#  
# Every configuration file must end to the keyword 'END'.  
END
```

Dynamics Mobile Node configuration file of final demo set-up:
/usr/local/etc/dynmnd.conf

```
# $Id: dynmnd.conf,v 1.56 2001/10/20 13:36:07 jm Exp $  
# Mobile Node configuration file  
#  
# Dynamic hierarchial IP tunnel  
# Copyright (C) 1998-2001, Dynamics group  
#  
# This program is free software; you can redistribute it and/or modify  
# it under the terms of the GNU General Public License version 2 as  
# published by the Free Software Foundation. See README and COPYING for  
# more details.  
#  
#####  
# NOTE!  
# This is an example configuration file designed to give  
# perspective to the system configuration AND to provide  
# a basis for a working simple test environment.  
# The values of some of the parameters may not be the  
# same as the daemon's defaults, so don't get confused.  
#  
# To get a minimal test working, you will need to check the  
# following items:  
# * MNHomeIPAddress  
# * HAIPAddress  
# * EnableFADecapsulation  
# * HomeNetPrefix (if using FA decapsulation or  
# dynamics HA address resolution)  
# * SPI and SharedSecret  
# The rest of the items should work with their preset values in  
# most cases and they can be used to fine tune the operations  
# after the basic operation have been tested successfully.  
#  
#####  
# The Mobile Nodes's IP address in the Home Network.  
# If using AAA (see UseAAA below), home address can be set to 0.0.0.0 in order  
# to request a home address from the AAA infrastructure. This requires that  
# also MN NAI is configured.  
MNHomeIPAddress 131.155.193.130  
  
# The Mobile Node's Network Access Identifier (NAI) [RFC2794]  
# If configured, this NAI is used in registration requests to identify the  
# mobile user for AAA services.
```

```
#
# MNNetworkAccessIdentifier "user@example.com"

# UseAAA < TRUE | FALSE >. TRUE enables AAA extensions (key requests using
# material from AAA, HA and home address discovery using AAA, etc.). This
# requires that MN NAI and AAA related items below are configured.
# FALSE disables these extensions.
UseAAA FALSE

# The IP address of Mobile Node's Home Agent. In case of a private HA address
# this is the address of the surrogate HA. If the HA address is unknown, set
# this to 0.0.0.0 and make sure that HomeNetPrefix is correct for dynamic
# HA address resolution or use AAA to discover HA address. If the HA has
# multiple interfaces, this should be the address of the "public" interface,
# i.e., the one toward default gateway (it has to be reachable from the foreign
# networks).
HAIPAddress 131.155.193.134

# If the HA has more than one interfaces, HAIPAddress should be configured to
# be the one reachable from the Internet (i.e., from the foreign networks the
# MN may visit). To allow MN to detect other HA's interfaces, their IP
# addresses may be configured here. MN will use this list in addition to
# HAIPAddress when determining whether an agent advertisement is from its own
# HA (i.e., when MN is at home). Multiple lines containing different addresses
# may be used to configure more than one alternative HA address.
# AlternativeHAIPAddress 10.1.2.3
#AlternativeHAIPAddress 131.155.193.112

# AllowHomeAddrFromForeignNet < TRUE | FALSE >. TRUE allows AAA to assign
# a home agent and home address from the foreign network (assuming they are
# set to 0.0.0.0 above). FALSE means that both the home agent and the home
# address must be from the home domain.
AllowHomeAddrFromForeignNet FALSE

# The following configuration options PrivateHAIPAddress, PrivateHAIdentifier,
# and HANetworkAccessIdentifier are only used with home networks that use
# private IP addresses and a surrogate HA. In other cases they should be left
# commented.

# The private IP address of Mobile Node's Home Agent.
# Needed only, if surrogate HA is used.
# PrivateHAIPAddress 192.168.200.200

# The identifier for the private HA in SHA (unique 32-bit number)
# PrivateHAIdentifier 1

# Home Agent Network Access Identifier (NAI)
# If configured, this NAI is used to match the HA agent advertisements when
# a MN is determining whether it is at home or not. This is mainly used with
# private HA address that may not be globally unique.
#
# HANetworkAccessIdentifier "ha@example.com"

# EnableFADecapsulation < TRUE | FALSE >. TRUE enables a mode where
# the FA decapsulates the IP-within-IP encapsulated IP packets.
# FALSE disables this mode and sets the default mode where the
# MN decapsulates the IP-within-IP encapsulated IP packets.
# With FA decapsulation the MN uses its home address in the interface even in
# the foreign network and with MN decapsulation MN needs to acquire a
# co-located care-of address from the visited network (this needs an external
# program; see man pages for more information).
# The two modes cannot be used simultaneously.
EnableFADecapsulation FALSE

# Network address of home network (CIDR format: a.b.c.d/prefix_length)
# This is used with FA decapsulation and dynamics HA address resolution. If
# commented, the routing entry is not removed nor added. The home net entry
# may optionally be used with MN decapsulation - see MNDecapsRouteHandling
# option below.
#
# Example: 192.168.242.0/24
HomeNetPrefix 131.155.193.128/29
#HomeNetPrefix 131.155.192.0/22

# Home net default gateway
# This entry can be used to force a gateway that the MN uses when it is
```

```
# at home. If this is left commented, the MN tries to use the default route
# that was in use when the program was started.
#
HomeNetGateway 131.155.192.1

#####
# a SPI (Security Parameter Index) must be defined for every MN.
# It is used for indexing the security association at the Home Agent.
SPI <censored>
#
# The SharedSecret is provided as a HEX number string. The shared secret can
# also be given as a character string
# (e.g. character string "ABCDE" corresponds to HEX number string 4142434445).
# Note: RFC 2002 specifies that the default key size is 128 bits (i.e.
# 16 bytes or 32 hex 'characters'). Dynamics supports also other key lengths.
# This shared secret is used with the HA. This must be commented out when using
# AAA infrastructure for key generation. In this case, the AAA related items
# below must be configured.
# SharedSecret < shared secret >
# SharedSecret 016A352B2F235E
SharedSecret <censored>
#
# Authentication algorithm
# 1: MD5/prefix+suffix (a.k.a. keyed-MD5) [RFC 2002]
# 4: HMAC-MD5 [RFC 2104]
# 5: SHA-1 [FIPS 180-1]
# 6: HMAC-SHA1 [RFC 2104]
# Note! MD5/prefix+suffix has known weaknesses and use of HMAC-MD5 is
# recommended. MD5/prefix+suffix algorithm is for backwards compatability with
# older versions that do not support more secure HMAC-MD5.
AuthenticationAlgorithm 4
#
# Replay prevention method:
# 0: none
# 1: time stamps
# 2: nonces
ReplayMethod 1
#
# Mobile Node may have optional security associations with Foreign
# Agents. If the security association exists an additional Mobile Node -
# Foreign Agent Authentication Extension is added to the registration requests.
#
# The following list contains the shared secrets indexed by SPI (and
# Foreign Agent IP address). The algorithm field specifies the method
# used for key distribution (see the list above). The format of the share
# secret field is identical to the one used with the MN-HA security
# association list above.
#
FA_SECURITY_BEGIN
# SPI      FA IP      Alg.      Shared Secret
#2001      192.168.0.1  4         0123456789ABCDEF
#2002      192.168.0.2  4         "eslkfj89jr3hduh3R!as"
FA_SECURITY_END

# MN-AAA Authentication and Challenge/Response [RFC3012]

# If the MN does not have a security association with an FA, it may use AAA
# infrastructure for authentication. If this is used, also MN NAI
# ('MNNetworkAccessIdentifier' above) should be configured.

# SPI to be used in MN-AAA authentication.
# Reserved SPI values:
# 2 = CHAP_SPI, CHAP style authentication using MD5 [RFC 3012]
# 3 = MD5/prefix+suffix [draft-ietf-mobileip-aaa-key-03.txt]
# 4 = HMAC MD5 [draft-ietf-mobileip-aaa-key-03.txt]
# MN-AAA-SPI 12345

# Shared secret for MN-AAA authentication (see 'SharedSecret' above for format
# instructions)
# MN-AAA-SharedSecret "test"

# Algorithms to be used for MN-AAA authentication and key generation
# 1 = MD5/prefix+suffix (RFC 2002)
# 2 = RADIUS authentication (Sec. 8 of RFC 3012)
# 3 = MD5/prefix+suffix (RFC 2002) (alias for 1 above)
```

```
# 4 = HMAC-MD5 (Sec. 6 of RFC 3012; RFC 2104)
# 5 = SHA-1 (FIPS 180-1)
# 6 = HMAC-SHA1 (RFC 2104)
# Note: with algorithm 2, 'MN-AAA-SPI' should be set to reserved number
# CHAP_SPI (default: 2).
# MN-AAA-AuthenticationAlgorithm 4
# MN-AAA-KeyGenerationAlgorithm 4

#####
# TunnelingMode < 1 | 2 | 3 | 4 >
# The packets between the MN and a Correspondent Node (CN) can be routed using
# different routes. This option can be used to select, which mode will be
# selected.
# Possible values:
# 1 = automatic, prefer reverse tunnel (i.e. bi-directional tunnel)
# 2 = automatic, prefer triangle tunnel (i.e. tunnel only in CN->MN direction)
# 3 = accept only reverse tunnel
# 4 = accept only triangle tunnel
TunnelingMode 3

# When MN can get its own co-located care-of address and use reverse tunneling,
# the normal method is to set the default route to the tunnel. This means that
# all the packets destined to other networks than the current subnet in the
# visited network are send via the HA. If the co-located COA is public, it can
# be used for sessions that do not need constant IP address (e.g. most of the
# web browsing). The following configuration option specifies the routing
# operation that is used with the co-located COA.
# Possible values:
# 0 = set default route to the tunnel
# 1 = set only the home net route to the tunnel (the above HomeNetPrefix
# options must be set)
# 2 = do not change the routing entries (i.e. some external means must be
# used to direct traffic to the tunnel, e.g. manually adding host route
# to a specific host)
MNDecapsRouteHandling 0

# DefaultTunnelLifetime is the lifetime suggested in registration
# The lifetime is defined in seconds, default value is 300.
# The request timer will be set according to this value. If the FA's agent
# advertisement has a smaller time, it is used instead.
# Special case: 65535 (or more) seconds means unlimited time (the binding will
# not expire)
# MNDefaultTunnelLifetime [ seconds ]
MNDefaultTunnelLifetime 300

# UDP port to be used for sending registration requests
# Port 434 is allocated for Mobile IP signaling and this should not be changed
# unless the network is known to use some other port (i.e. all the FAs and HAS
# must have the same port configured).
UDPPort 434

# Socket priority for signaling sockets (UDP) can be set with SO_PRIORITY to
# allow easier QoS configuration. If this argument is set, the given value is
# used as a priority for the signaling socket. E.g. CBQ class can be used to
# make sure that signaling is not disturbed by other traffic on a congested
# link.
# This feature is still undocumented and can be left commented.
#
# SocketPriority 1

# The log messages are written through syslog service. The facility to be
# used defaults to LOG_LOCAL0, but it can be set with this parameter
# to any of the possible facilities (LOG_AUTHPRIV, LOG_DAEMON, and so on).
# The processing of log messages is defined in /etc/syslog.conf file.
SyslogFacility LOG_DAEMON

# Ignore these interfaces. No agent advertisements are received nor
# agent solicitations sent for these interfaces.
IGNORE_INTERFACES_BEGIN
lo
dummy0
tunl0
gre0
IGNORE_INTERFACES_END
```

```
# Other programs may set routing entries so that the data connection may
# fail. The MN can try to enforce the routes that it believes should be used.
# This operation should currently be used only with FA decapsulation. If the
# route enforcement is activated the MN daemon prevents certain route changes.
EnforceRoutes FALSE

# MN can be instructed to poll for current AP address when using a wireless
# LAN driver that supports wireless extensions. This can be used to speed up
# handoffs when using managed mode (BSS).
# Polling interval is configured in micro seconds
# (i.e., 1000000 equals to 1 second)
# -1 = AP polling disabled
APPollingInterval -1

# MN can be instructed to send periodic agent solicitations to find new FAs.
# Normally, MN uses agent solicitations when it does not have a valid agent
# advertisement. Periodic solicitation occurs even if the connection seems to
# be up. This will cause more broadcast messages and is thus disabled in the
# default configuration, but it can speed up handoffs in some environments.
# Solicitation interval is configured in micro seconds (usec)
# (i.e., 1000000 usec equals to 1 second). A random time between 0 and 0.5
# second will be added to solicitation intervals to prevent unwanted
# synchronization of broadcast messages. In addition, solicitations will not be
# send more often than once per second, so this interval should not be
# configured to be less than 1000000 usec.
# -1 = Periodic agent solicitation disabled
SolicitationInterval -1

#####
# Mobile Nodes use unix domain sockets to communicate through their API
# interfaces.
# The group and owner must be names as strings, no groupIDs or userIDs are
# allowed. The file permissions are set in octal values like in chmod(1).
# The configuration parameters of the two API sockets are as follows:
MNAPIReadSocketPath "/var/run/dynamics_mn_read"
MNAPIReadSocketGroup "root"
MNAPIReadSocketOwner "root"
MNAPIReadSocketPermissions 0666
#
MNAPIAdminSocketPath "/var/run/dynamics_mn_admin"
MNAPIAdminSocketGroup "root"
MNAPIAdminSocketOwner "root"
MNAPIAdminSocketPermissions 0700
#
# Every configuration file must end to the keyword 'END'.
END
```


Appendix E Handoff Script

This Appendix contains the following configuration files:

- Handoff script MIPscriptMAN (on MN);
- Manually handoff add-on MIPscriptPREF (on MN);
- Example of stored MIPscriptPREF_FILE (on MN).

Handoff script MIPscriptMAN³⁵:

/root/MIPscriptMAN

```
#!/bin/sh

# Scriptfile: MIPscript, 05-08-2002

# This file contains a script to handle the Mobile IPv4 implementation
# Dynamics 0.8.1 of the Helsinki University of Technology (HUT)
# in Finland
# http://www.cs.hut.fi/Research/Dynamics/
#
# The script is designed to use on a Mobile Node that only allows a
# co-located care-of address (COA), so it will never try to find and use
# a Foreign Agent.
#
# This script has been made by Tom van Sebille at the department
# of Electrical Engineering of the Eindhoven University of Technology
# in The Netherlands
#
# This file starts with the the declaration and definition of some
# variables and functions. These will be used in the script
# that follows. The script is a 'never ending' loop. When the script
# is terminated with ^c, the route table will very likely be corrupted.
# A re-plugin of the network interface(s) will update the table corectly.
#
# A second scriptfile, MIPscriptPREF, can be used to manually overrule the
# preferred interface to use. Two parameters are used:
# -1- a mode bit with possible values:
#     m: use the manual mode to choose preferred interface
#     a: use the automatic mode to choose preferred interface
# -2- the preferred interface that should be used in manual state
#     possible values: eth0, eth1, eth2, ppp0 or ppp1

# Get the fixed home IP-address of the Mobile Node
# and the IP-address of the Home Agent from
# the dynmnd.conf configuration file.
# Therefore the Mobile Node functionality of
# HUT Dynamics Mobile IPv4 0.8.1 has to be implemented on this machine.
MNHomeIPAddress=`cat /usr/local/etc/dynmnd.conf | egrep '^MNHomeIPAddress' | cut -d " " -f 2`
HAIPAddress=`cat /usr/local/etc/dynmnd.conf | egrep '^HAIPAddress' | cut -d " " -f 2`

# Definition of several arrays, one for each possible eth or ppp interface.
# Each array has 3 values: IP-address, subnet or ppp peer and default gateway.
# The latest information will be stored in the "NEW" arrays.
# Reference information is stored in "OLD" arrays.

declare -a eth0NEW
declare -a eth1NEW
declare -a eth2NEW
declare -a ppp0NEW
declare -a ppp1NEW
# All NEW arrays together form the 'NEW state'.

declare -a eth0OLD
declare -a eth1OLD
```

³⁵ Some lines are split due to the lay-out of this document. The script does not work properly when using split lines.

```

declare -a eth2OLD
declare -a ppp0OLD
declare -a ppp1OLD
# All OLD arrays together form the 'OLD state'.

# To keep track of a one-time change, a variable TUN_MODE_SET is used:
TUN_MODE_SET=0

# FILL_NEW_ARRAYS scans the current IP configuration and routing table and
# writes specific parameters to a temporary file which will be used to check if
# something has changed since previous check.
# The variable IP_ROUTE_SNAPSHOT is used as a buffer to make sure that all
# parameters are captured at the same time, so no changes will be encountered
# during the filling of the members of the NEW arrays:
# 0: IP-address
# 1: subnet/ppp peer
# 2: default gateway
# Extra future possibilities:
# 3: a bit indicating whether it is a WLAN interface or not
# (this cannot be concluded from the "ip route" command,
# "iwconfig" should be used)
#
## Sometimes the default gateway will be found a fraction of a second later
## than the IP-address and the subnet.
## To make sure that all changes have been captured, the filling of the NEW
## variables will be delayed by 0.2 seconds, after a subnet change has been
## detected.
## After those 0.2 seconds a new capture snapshot will be taken, which should
## contain the default gateway now.
## NOTE: It is very important that within these 0.2 seconds no interfaces are added
## or removed, because only one change at a time can be handled by this function!!!
## This is done in the first if-loop of FILL_NEW_ARRAYS
function FILL_NEW_ARRAYS {
  IP_ROUTE_SNAPSHOT=`ip route`
  if [ \
"$${eth0NEW[1]}" -o \
  "`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\d].*eth0.*kernel' | awk '{ print $1 }'" !=
"$${eth1NEW[1]}" -o \
  "`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\d].*eth1.*kernel' | awk '{ print $1 }'" !=
"$${eth2NEW[1]}" ] ; then
    #echo "sleep 0.2 s"
    sleep 0.2s
    IP_ROUTE_SNAPSHOT=`ip route`
  fi

  if [ "`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\d].*eth0.*kernel' | awk '{ print $1 }'" !=
"$${eth0NEW[1]}" ] ; then
    # Only change the default gateway if the subnet has changed,
    # otherwise EVAL_GWS will not function correctly.
    echo The subnet of eth0 has changed, so write new default gateway.
    eth0NEW[2]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^default.*eth0' | cut -d " " -f 3`
  fi
  eth0NEW[0]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\d].*eth0.*kernel' | awk '{ print $9 }'"
  eth0NEW[1]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\d].*eth0.*kernel' | awk '{ print $1 }'"

  if [ "`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\d].*eth1.*kernel' | awk '{ print $1 }'" !=
"$${eth1NEW[1]}" ] ; then
    # Only change the default gateway if the subnet has changed,
    # otherwise EVAL_GWS will not function correctly.
    echo The subnet of eth1 has changed, so write new default gateway.
    eth1NEW[2]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^default.*eth1' | cut -d " " -f 3`
  fi
  eth1NEW[0]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\d].*eth1.*kernel' | awk '{ print $9 }'"
  eth1NEW[1]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\d].*eth1.*kernel' | awk '{ print $1 }'"

  if [ "`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\d].*eth2.*kernel' | awk '{ print $1 }'" !=
"$${eth2NEW[1]}" ] ; then
    # Only change the default gateway if the subnet has changed,
    # otherwise EVAL_GWS will not function correctly.
    echo The subnet of eth2 has changed, so write new default gateway.
    eth2NEW[2]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^default.*eth2' | cut -d " " -f 3`
  fi
  eth2NEW[0]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\d].*eth2.*kernel' | awk '{ print $9 }'"
  eth2NEW[1]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\d].*eth2.*kernel' | awk '{ print $1 }'"

```

```

ppp0NEW[0]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\^d].*ppp0.*kernel' | awk '{ print $9 }'`
ppp0NEW[1]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\^d].*ppp0.*kernel' | awk '{ print $1 }'`
ppp0NEW[2]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^default.*ppp0' | cut -d " " -f 3`

ppp1NEW[0]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\^d].*ppp1.*kernel' | awk '{ print $9 }'`
ppp1NEW[1]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^[\^d].*ppp1.*kernel' | awk '{ print $1 }'`
ppp1NEW[2]=`echo "$IP_ROUTE_SNAPSHOT" | grep '^default.*ppp1' | cut -d " " -f 3`
}

# EVAL_GWS evaluates the gateways that have been found during FILL_NEW_ARRAYS
# If no default gateway was found, EVAL_GWS guesses the gateway based on 2 principles:
# 1: If an eth device misses a default gateway, it is assumed that this is because
# another eth device is connected to the same subnet and therefore already has
# configured the appropriate gateway. This gateway will be copied to the new
# eth interface.
# 2: If a ppp device misses a default gateway, it is assumed that this is because
# another interface has already set a default gateway. The gateway will be set by
# copying the peer IP-address, which is the default gateway for that interface.
function EVAL_GWS {
  # Check whether eth0 has a default gateway.
  # First check whether eth0 is up and whether a default gateway has already been found:
  if [ "${eth0NEW[0]}" != "" -a "${eth0NEW[2]}" = "" ]; then
    # eth0NEW[0] is not empty, so eth0 has an IP address and thus is up!
    # AND
    # no default gateway was found
    # check for other eth interfaces with same subnet:
    if [ "${eth0NEW[1]}" = "${eth1NEW[1]}" ]; then
      # eth0 is on the same subnet as eth1 and therefore gets the same default gateway:
      eth0NEW[2]="${eth1NEW[2]}"
    elif [ "${eth0NEW[1]}" = "${eth2NEW[1]}" ]; then
      # eth0 is on the same subnet as eth2 and therefore gets the same default gateway:
      eth0NEW[2]="${eth2NEW[2]}"
    fi
  fi

  # Do the same for eth1:
  if [ "${eth1NEW[0]}" != "" -a "${eth1NEW[2]}" = "" ]; then
    if [ "${eth1NEW[1]}" = "${eth0NEW[1]}" ]; then
      eth1NEW[2]="${eth0NEW[2]}"
    elif [ "${eth1NEW[1]}" = "${eth2NEW[1]}" ]; then
      eth1NEW[2]="${eth2NEW[2]}"
    fi
  fi

  # Do the same for eth2:
  if [ "${eth2NEW[0]}" != "" -a "${eth2NEW[2]}" = "" ]; then
    if [ "${eth2NEW[1]}" = "${eth0NEW[1]}" ]; then
      eth2NEW[2]="${eth0NEW[2]}"
    elif [ "${eth2NEW[1]}" = "${eth1NEW[1]}" ]; then
      eth2NEW[2]="${eth1NEW[2]}"
    fi
  fi

  # Now for the ppp interfaces:
  # First check whether ppp0 is up and whether a default gateway has already been found:
  if [ "${ppp0NEW[0]}" != "" -a "${ppp0NEW[2]}" = "" ]; then
    # ppp0 is up AND no gateway was found
    # so guess that the default gateway is the ppp peer
    ppp0NEW[2]="${ppp0NEW[1]}"
  fi

  # Do the same for ppp1:
  if [ "${ppp1NEW[0]}" != "" -a "${ppp1NEW[2]}" = "" ]; then
    ppp1NEW[2]="${ppp1NEW[1]}"
  fi
}

# COPY_ARRAYS copies the NEW arrays to the OLD arrays.
# The OLD arrays can be seen as reference values to keep track of changes.
function COPY_ARRAYS {
  eth0OLD[0]="${eth0NEW[0]}"
  eth0OLD[1]="${eth0NEW[1]}"
  eth0OLD[2]="${eth0NEW[2]}"
}

```

```

eth1OLD[0]={eth1NEW[0]}
eth1OLD[1]={eth1NEW[1]}
eth1OLD[2]={eth1NEW[2]}
eth2OLD[0]={eth2NEW[0]}
eth2OLD[1]={eth2NEW[1]}
eth2OLD[2]={eth2NEW[2]}
ppp0OLD[0]={ppp0NEW[0]}
ppp0OLD[1]={ppp0NEW[1]}
ppp0OLD[2]={ppp0NEW[2]}
ppp1OLD[0]={ppp1NEW[0]}
ppp1OLD[1]={ppp1NEW[1]}
ppp1OLD[2]={ppp1NEW[2]}
}

# COMPARE_ARRAYS compares the NEW arrays with the OLD reference arrays to see
# if some interface has been added, changed or removed.
# The function returns the following value:
#   nothing changed
#   OR
#   something changed
function COMPARE_ARRAYS {
  if [ \
    "${eth0NEW[0]}" = "${eth0OLD[0]}" -a "${eth0NEW[1]}" = "${eth0OLD[1]}" -a "${eth0NEW[2]}" =
    "${eth0OLD[2]}" -a \
    "${eth1NEW[0]}" = "${eth1OLD[0]}" -a "${eth1NEW[1]}" = "${eth1OLD[1]}" -a "${eth1NEW[2]}" =
    "${eth1OLD[2]}" -a \
    "${eth2NEW[0]}" = "${eth2OLD[0]}" -a "${eth2NEW[1]}" = "${eth2OLD[1]}" -a "${eth2NEW[2]}" =
    "${eth2OLD[2]}" -a \
    "${ppp0NEW[0]}" = "${ppp0OLD[0]}" -a "${ppp0NEW[1]}" = "${ppp0OLD[1]}" -a "${ppp0NEW[2]}" =
    "${ppp0OLD[2]}" -a \
    "${ppp1NEW[0]}" = "${ppp1OLD[0]}" -a "${ppp1NEW[1]}" = "${ppp1OLD[1]}" -a "${ppp1NEW[2]}" =
    "${ppp1OLD[2]}" \
  ]; then
    echo nothing changed
  else
    echo something changed
  fi
}

# HOME_IF checks whether one of the interfaces has the Mobile Nodes Home IP-address.
# The function returns the following value:
#   none: there is no interface which has the home IP-address
#   OR
#   <if>: this interface has the home IP-address
function HOME_IF {
  if [ "${eth0NEW[0]}" = "$MNHHomeIPAddress" ]; then
    echo eth0
  elif [ "${eth1NEW[0]}" = "$MNHHomeIPAddress" ]; then
    echo eth1
  elif [ "${eth2NEW[0]}" = "$MNHHomeIPAddress" ]; then
    echo eth2
  elif [ "${ppp0NEW[0]}" = "$MNHHomeIPAddress" ]; then
    echo ppp0
  elif [ "${ppp1NEW[0]}" = "$MNHHomeIPAddress" ]; then
    echo ppp1
  else
    echo none
  fi
}

# PRINT_ARRAYS prints the current status of all NEW and OLD arrays on the screen
function PRINT_ARRAYS {
  echo '-----'
  echo 'NEW state at `date`:
  echo '  eth0 IP           = '${eth0NEW[0]}
  echo '  eth0 subnet       = '${eth0NEW[1]}
  echo '  eth0 gateway      = '${eth0NEW[2]}
  echo '  eth1 IP           = '${eth1NEW[0]}
  echo '  eth1 subnet       = '${eth1NEW[1]}
  echo '  eth1 gateway      = '${eth1NEW[2]}
  echo '  eth2 IP           = '${eth2NEW[0]}
  echo '  eth2 subnet       = '${eth2NEW[1]}
  echo '  eth2 gateway      = '${eth2NEW[2]}
}

```

```

echo '      ppp0 IP      = '${ppp0NEW[0]}
echo '      ppp0 peer   = '${ppp0NEW[1]}
echo '      ppp0 gateway = '${ppp0NEW[2]}
echo '      ppp1 IP      = '${ppp1NEW[0]}
echo '      ppp1 peer   = '${ppp1NEW[1]}
echo '      ppp1 gateway = '${ppp1NEW[2]}
echo 'OLD state:'
echo '  eth0 IP          = '${eth0OLD[0]}
echo '  eth0 subnet      = '${eth0OLD[1]}
echo '  eth0 gateway     = '${eth0OLD[2]}
echo '  eth1 IP          = '${eth1OLD[0]}
echo '  eth1 subnet      = '${eth1OLD[1]}
echo '  eth1 gateway     = '${eth1OLD[2]}
echo '  eth2 IP          = '${eth2OLD[0]}
echo '  eth2 subnet      = '${eth2OLD[1]}
echo '  eth2 gateway     = '${eth2OLD[2]}
echo '  ppp0 IP          = '${ppp0OLD[0]}
echo '  ppp0 peer        = '${ppp0OLD[1]}
echo '  ppp0 gateway     = '${ppp0OLD[2]}
echo '  ppp1 IP          = '${ppp1OLD[0]}
echo '  ppp1 peer        = '${ppp1OLD[1]}
echo '  ppp1 gateway     = '${ppp1OLD[2]}
echo '-----'
}

# PRIORITY_IF returns the interface which has the highest priority.
# This means that through this interface the tunnel to the
# Home Agent will be set up, if the Mobile Node is not at the Home
# Network.
# This function may contain very complex algorithms to determine the
# interface with the highest priority.
#
# The current function only checks whether there are eth interfaces
# and if so, it returns the new added or the first it finds.
# Otherwise it will return the new added ppp interface or the first
# ppp interface it finds.
#
# In more advanced future algorithms the signal strength and/or signal
# quality (for WLAN interfaces), data throughput, delay time,
# cost, etc. can be implemented.
function PRIORITY_IF {
  # First look for new added eth interfaces:
  if [ "${eth0OLD[0]}" != "${eth0NEW[0]}" -a "${eth0NEW[0]}" != "" ]; then
    echo eth0
  elif [ "${eth1OLD[0]}" != "${eth1NEW[0]}" -a "${eth1NEW[0]}" != "" ]; then
    echo eth1
  elif [ "${eth2OLD[0]}" != "${eth2NEW[0]}" -a "${eth2NEW[0]}" != "" ]; then
    echo eth2
  # Then look for available eth interfaces:
  elif [ "${eth0NEW[0]}" != "" ]; then
    echo eth0
  elif [ "${eth1NEW[0]}" != "" ]; then
    echo eth1
  elif [ "${eth2NEW[0]}" != "" ]; then
    echo eth2
  # Then look for new added ppp interfaces:
  elif [ "${ppp0OLD[0]}" != "${ppp0NEW[0]}" -a "${ppp0NEW[0]}" != "" ]; then
    echo ppp0
  elif [ "${ppp1OLD[0]}" != "${ppp1NEW[0]}" -a "${ppp1NEW[0]}" != "" ]; then
    echo ppp1
  # Finally look for available ppp interfaces:
  elif [ "${ppp0NEW[0]}" != "" ]; then
    echo ppp0
  elif [ "${ppp1NEW[0]}" != "" ]; then
    echo ppp1
  # There is no interface available:
  else
    echo none
  fi
}

# READ_MAN_SETTINGS reads the manual preferred interface settings
# (i.e. preferred interface mode and the preferred interface)
# from the file MIPscriptPREF_FILE that is written by the script MIPscriptPREF.

```

```
# It writes these values to variables.
function READ_MAN_SETTINGS {
    PREFIFMODE_NEW=`cat ./MIPscriptPREF_FILE|egrep "mode"|cut -d " " -f 4`
    #echo PREFIFMODE_NEW = $PREFIFMODE_NEW
    PREFIF_NEW=`cat ./MIPscriptPREF_FILE|egrep "Manual"|cut -d " " -f 4`
    #echo PREFIF_NEW = $PREFIF_NEW
}

# COMPARE_MAN_SETTINGS compares the current manual preferred interface settings with
# the reference (OLD) manual overrule settings.
function COMPARE_MAN_SETTINGS {
    if [ "$PREFIFMODE_NEW" = "$PREFIFMODE_OLD" -a "$PREFIF_NEW" = "$PREFIF_OLD" ]; then
        echo "nothing changed"
    else
        echo "something changed"
    fi
}

# COPY_MAN_SETTINGS will copy the NEW manual preferred interface settings to
# the OLD (reference) ones:
function COPY_MAN_SETTINGS {
    PREFIFMODE_OLD=$PREFIFMODE_NEW
    PREFIF_OLD=$PREFIF_NEW
}

#####
# The real script starts here: #
#####

##### First we have an initialization phase
# It is defined as a function so it can be easily "turned off" by
# commenting it out.
function INITIALIZE {
    # The NEW arrays will be filled for the first time
    FILL_NEW_ARRAYS
    EVAL_GWS

    # Check for an available interface, if none, quit
    if [ "`PRIORITY_IF`" = "none" ]; then
        echo Connect to the Internet before running MIPscript,
        echo MIPscript will be terminated.
        exit
    fi

    # The reference arrays (OLD) have to be the same for a start:
    COPY_ARRAYS
    # Check the status:
    PRINT_ARRAYS

    # Set the NEW and OLD manual preferred interface settings to default values.
    # This means that the script will start with the automatic mode:
    ./MIPscriptPREF a eth0
    READ_MAN_SETTINGS
    COPY_MAN_SETTINGS

    # Kill all current Mobile Node daemons, if any:
    killall dynmnd 2> /dev/null
    # Start new HUT Dynamics 0.8.1 Mobile Node daemon:
    dynmnd
    # Only the first time the Mobile Node starts using a foreign network,
    # the tunneling mode must be set to "tunnel HA", so the daemon
    # will never try to use Foreign Agents:
    if [ "`HOME_IF`" = "none" ]; then
        dynmn_tool disconnect
        echo @@@@ Set tunneling mode to tunnel-direct-to-Home-Agent:
        dynmn_tool tunnel HA
        TUN_MODE_SET=1
    fi
}

# Run the initialization:
```

```

INITIALIZE

##### After the initialization the 'never ending' loop will be entered:
while [ 1 ]; do
  FILL_NEW_ARRAYS
  EVAL_GWS
  READ_MAN_SETTINGS

  if [ "$PREFIXMODE_NEW" = "a" ]; then
    # automatic mode
    if [ "`COMPARE_ARRAYS`" = "nothing changed" ]; then
      sleep 1s
    else
      echo "Something changed in automatic mode:"
      PRINT_ARRAYS
      if [ "`HOME_IF`" != "none" ]; then
        dynmn_tool update `HOME_IF`

        # The following part is probably unnecessary: START WASTE
        #echo Renew default gateway of home interface:
        #route
        #if [ "`HOME_IF`" = "eth0" ]; then
        #  #route add default gw ${eth0NEW[2]} dev eth0
        #elif [ "`HOME_IF`" = "eth1" ]; then
        #  #route add default gw ${eth1NEW[2]} dev eth1
        #elif [ "`HOME_IF`" = "eth2" ]; then
        #  #route add default gw ${eth2NEW[2]} dev eth2
        #elif [ "`HOME_IF`" = "ppp0" ]; then
        #  #route add default gw ${ppp0NEW[2]} dev ppp0
        #elif [ "`HOME_IF`" = "ppp1" ]; then
        #  #route add default gw ${ppp1NEW[2]} dev ppp1
        #fi
        # END WASTE
        echo new route:
        route -n

      else
        # Only the first time the Mobile Node starts using a foreign network,
        # the tunneling mode must be set to "tunnel HA", so the daemon
        # will never try to use Foreign Agents:
        #if [ "$TUN_MODE_SET" != "1" ]; then
        if [ "$TUN_MODE_SET" != "1" -a "`PRIORITY_IF`" != "none" ]; then
          dynmn_tool disconnect
          echo @@@@ Set tunneling mode to tunnel-direct-to-Home-Agent:
          dynmn_tool tunnel HA
          TUN_MODE_SET=1
        fi
        echo Priority interface = `PRIORITY_IF`
        if [ "`PRIORITY_IF`" = "none" ]; then
          echo There is no interface available!
        elif [ "`PRIORITY_IF`" = "eth0" ]; then
          echo flush route...
          route
          echo add route...
          route add $HAIPAddress gw ${eth0NEW[2]} dev eth0
          echo new route:
          route -n
          echo dyn update:
          dynmn_tool update eth0
        elif [ "`PRIORITY_IF`" = "eth1" ]; then
          echo flush route...
          route
          echo add route...
          route add $HAIPAddress gw ${eth1NEW[2]} dev eth1
          echo new route:
          route -n
          echo dyn update:
          dynmn_tool update eth1
        elif [ "`PRIORITY_IF`" = "eth2" ]; then
          echo flush route...
          route
          echo add route...
          route add $HAIPAddress gw ${eth2NEW[2]} dev eth2
          echo new route:
          route -n
          echo dyn update:

```

```

    dynmn_tool update eth2
elif [ "`PRIORITY_IF`" = "ppp0" ]; then
    echo flush route...
    routef
    echo add route...
    route add $HAIPAddress gw ${ppp0NEW[2]} dev ppp0
    echo new route:
    route -n
    echo dyn update:
    dynmn_tool update ppp0
elif [ "`PRIORITY_IF`" = "ppp1" ]; then
    echo flush route...
    routef
    echo add route...
    route add $HAIPAddress gw ${ppp1NEW[2]} dev ppp1
    echo new route:
    route -n
    echo dyn update:
    dynmn_tool update ppp1
fi
fi
COPY_ARRAYS
fi
else # PREFIFMODE_NEW = m
# manual mode
if [ "`COMPARE_MAN_SETTINGS`" = "nothing changed" ]; then
    sleep 1s
else
    echo "Something changed in manual mode:"
    PRINT_ARRAYS
    if [ "$PREFIF_NEW" = "`HOME_IF`" ]; then
        dynmn_tool update `HOME_IF`

# The following part is probably unnecessary: START WASTE
#echo Renew default gateway of home interface:
#routef
#if [ "`HOME_IF`" = "eth0" ]; then
#    #route add default gw ${eth0NEW[2]} dev eth0
#elif [ "`HOME_IF`" = "eth1" ]; then
#    #route add default gw ${eth1NEW[2]} dev eth1
#elif [ "`HOME_IF`" = "eth2" ]; then
#    #route add default gw ${eth2NEW[2]} dev eth2
#elif [ "`HOME_IF`" = "ppp0" ]; then
#    #route add default gw ${ppp0NEW[2]} dev ppp0
#elif [ "`HOME_IF`" = "ppp1" ]; then
#    #route add default gw ${ppp1NEW[2]} dev ppp1
#fi
# END WASTE
    echo new route:
    route -n

else
# Only the first time the Mobile Node starts using a foreign network,
# the tunneling mode must be set to "tunnel HA", so the daemon
# will never try to use Foreign Agents:
#if [ "$TUN_MODE_SET" != "1" ]; then
if [ "$TUN_MODE_SET" != "1" -a "`PRIORITY_IF`" != "none" ]; then
    dynmn_tool disconnect
    echo @@@@ Set tunneling mode to tunnel-direct-to-Home-Agent:
    dynmn_tool tunnel HA
    TUN_MODE_SET=1
fi
echo Preferred interface = $PREFIF_NEW
if [ "$PREFIF_NEW" = "eth0" -a "${eth0NEW[0]}" != "" ]; then
    echo flush route...
    routef
    echo add route...
    route add $HAIPAddress gw ${eth0NEW[2]} dev eth0
    echo new route:
    route -n
    echo dyn update:
    dynmn_tool update eth0
elif [ "$PREFIF_NEW" = "eth1" -a "${eth1NEW[0]}" != "" ]; then
    echo flush route...
    routef
    echo add route...

```



```

route add $HAIPAddress gw ${eth1NEW[2]} dev eth1
echo new route:
route -n
echo dyn update:
dynmn_tool update eth1
elif [ "$PREFIF_NEW" = "eth2" -a "${eth2NEW[0]}" != "" ]; then
echo flush route...
route f
echo add route...
route add $HAIPAddress gw ${eth2NEW[2]} dev eth2
echo new route:
route -n
echo dyn update:
dynmn_tool update eth2
elif [ "$PREFIF_NEW" = "ppp0" -a "${ppp0NEW[0]}" != "" ]; then
echo flush route...
route f
echo add route...
route add $HAIPAddress gw ${ppp0NEW[2]} dev ppp0
echo new route:
route -n
echo dyn update:
dynmn_tool update ppp0
elif [ "$PREFIF_NEW" = "ppp1" -a "${ppp1NEW[0]}" != "" ]; then
echo flush route...
route f
echo add route...
route add $HAIPAddress gw ${ppp1NEW[2]} dev ppp1
echo new route:
route -n
echo dyn update:
dynmn_tool update ppp1
else
echo The requested interface is not available, no changes made.
fi
fi
COPY_MAN_SETTINGS
fi
# In case an interface changed during manual mode, update reference arrays:
COPY_ARRAYS
fi
done
# end

```

Manually handoff add-on MIPscriptPREF: ***/root/MIPscriptPREF***

```

#!/bin/sh

# Scriptfile: MIPscriptPREF, 05-08-2002

# This file contains a script to handle the Mobile IPv4 implementation
# Dynamics 0.8.1 of the Helsinki University of Technology (HUT)
# in Finland
# http://www.cs.hut.fi/Research/Dynamics/
#
# The script is designed to use on a Mobile Node that only allows a
# co-located care-of address (COA), so it will never try to find and use
# a Foreign Agent.
#
# This script has been made by Tom van Sebille at the department
# of Electrical Engineering of the Eindhoven University of Technology
# in The Netherlands
#
# This file is an add-on to the scriptfile MIPscript, which implements an
# automated update procedure for HUT Dynamics 0.8.1. This scriptfile can be
# used to manually overrule the preferred interface to use.
# Two parameters are used:
# -1- a mode bit with possible values:
# m: use the manual mode to choose preferred interface
# a: use the automatic mode to choose preferred interface

```

```
# -2- the preferred interface that should be used in manual state
# possible values: eth0, eth1, eth2, ppp0 or ppp1

# These parameters will be written to a file MIPscriptPREF_FILE that is
# used in MIPscript to set the preferred interface.

# NOTE: If the Homa Agent sends HA advertisements, then this script will
# not work if the Mobile Node is at the home network. This is because
# HUT Dynamics 0.8.1 cannot be overruled in the home situation.

# First, check the parameters validity:
if [ "$1" != "m" -a "$1" != "a" ]; then
    echo "First parameter must be m or a for manual or automatic mode."
    exit
fi

# Check second parameter
if [ "$1" == "m" -a "$2" != "eth0" -a "$2" != "eth1" -a "$2" != "eth2" -a \
    "$2" != "ppp0" -a "$2" != "ppp1" ]; then
    echo "Incorrect interface given."
    exit
fi

# Write the output to file MIPscripPREF_FILE
echo -e "Preferred interface mode: $1\nManual preferred interface: $2" > MIPscriptPREF_FILE

# end
```

Example of stored MIPscriptPREF_FILE:
/root/MIPscriptPREF_FILE

```
Preferred interface mode: a
Manual preferred interface: eth0
```

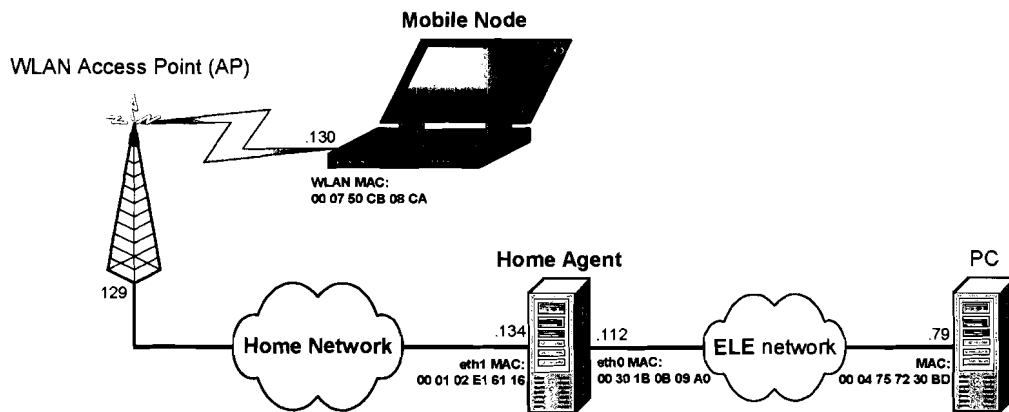
Appendix F Test Cases

This Appendix contains the following test cases:

- Proxy ARP with IP Forwarding;
- Tracing ping packets with MIP;

Proxy ARP with IP Forwarding

The test consisted of sniffing some packets that are forwarded by the HA PC, see Figure 2-2. The HA PC was configured to use proxy ARP at the external interface for the nodes of the (internal) HN. The packets sniffed during the test were “ping” packets from the MN (131.155.193.130) at the HN to a PC (131.155.193.79) at the ELE network. The set-up is shown in the figure below (only the host numbers are shown).



Proxy ARP test case set-up

The program used for sniffing the packets on both interfaces of the HA PC is `tcpdump`. The table below presents the packets in time. The `tcpdump` command line is also shown. The proof of the proxy ARP is given in the shaded line.

Ping packets from 131.155.193.130 to 131.155.193.79:

sniffed at 131.155.193.134	sniffed at 131.155.193.112
<pre> root:~# tcpdump -n -i eth1 ip host 131.155.193.130 or host 131.155.193.112 or host 131.155.193.134 or ip broadcast tcpdump: listening on eth1 11:57:36.698188 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:37.087898 131.155.193.79 > 131.155.193.130: icmp: echo reply 11:57:37.689049 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:37.689222 131.155.193.79 > 131.155.193.130: icmp: echo reply 11:57:38.689066 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:38.689227 131.155.193.79 > 131.155.193.130: icmp: echo reply 11:57:39.689343 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:39.689520 131.155.193.79 > 131.155.193.130: icmp: echo reply 11:57:40.689143 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:40.689392 131.155.193.79 > 131.155.193.130: icmp: echo reply 11:57:41.689041 arp who-has 131.155.193.134 tell 131.155.193.130 11:57:41.689101 arp reply 131.155.193.134 ie-at 0:1:2:e1:61:16 11:57:41.689830 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:41.690227 131.155.193.79 > 131.155.193.130: icmp: echo reply 11:57:42.689224 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:42.689415 131.155.193.79 > 131.155.193.130: icmp: echo reply </pre>	<pre> root:~# tcpdump -n -i eth0 ip host 131.155.193.130 or host 131.155.193.112 or host 131.155.193.79 or ip broadcast tcpdump: listening on eth0 11:57:36.698249 arp who-has 131.155.193.79 tell 131.155.193.112 11:57:36.705336 arp reply 131.155.193.79 ie-at 0:4:75:72:30:bd 11:57:36.705391 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:36.711174 arp who-has 131.155.193.130 tell 131.155.193.79 11:57:37.087898 arp reply 131.155.193.130 ie-at 0:30:1b:b:9:a0 11:57:37.689049 131.155.193.79 > 131.155.193.130: icmp: echo request (DF) 11:57:37.689205 131.155.193.79 > 131.155.193.130: icmp: echo reply 11:57:38.689102 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:38.689211 131.155.193.79 > 131.155.193.130: icmp: echo reply 11:57:39.689386 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:39.689502 131.155.193.79 > 131.155.193.130: icmp: echo reply 11:57:40.689183 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:40.689372 131.155.193.79 > 131.155.193.130: icmp: echo reply 11:57:41.689860 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:41.690208 131.155.193.79 > 131.155.193.130: icmp: echo reply 11:57:42.689274 131.155.193.130 > 131.155.193.79: icmp: echo request (DF) 11:57:42.689397 131.155.193.79 > 131.155.193.130: icmp: echo reply </pre>

Routing at MN Using Dynamics

The next test is to check the routing process at the MN. The test is meant to verify the routing process discussed in the section “Dynamics MIP Routing” of section 2.3.3. During the test the MN sends four ping packets to node 131.155.2.3. The MN Home Address is 131.155.193.130, the HA address is 131.155.193.134 and the used co-located COA is 62.140.135.231. The table below presents the ping packets in time, sniffed at interfaces TUNLMNA and eth1. The shaded line represents the first IP-within-IP tunneled packet. The

source and destination IP addresses of both IP headers are shown. From the table it can be calculated that the tunneling process takes about 0.02ms on the MN (Intel PIII, 600MHz).

Ping packets from MN to 131.155.2.3:

sniffed at TUNLMNA:	sniffed at eth1:
<pre> root:~# tcpdump -n -i TUNLMNA tcpdump: listening on TUNLMNA 16:22:29.563894 131.155.193.130 > 131.155.2.3: icmp: echo request (DF) 16:22:29.579570 131.155.2.3 > 131.155.193.130: icmp: echo reply (DF) 16:22:30.559493 131.155.193.130 > 131.155.2.3: icmp: echo request (DF) 16:22:30.569837 131.155.2.3 > 131.155.193.130: icmp: echo reply (DF) 16:22:31.559476 131.155.193.130 > 131.155.2.3: icmp: echo request (DF) 16:22:31.570088 131.155.2.3 > 131.155.193.130: icmp: echo reply (DF) 16:22:32.560314 131.155.193.130 > 131.155.2.3: icmp: echo request (DF) 16:22:32.591103 131.155.2.3 > 131.155.193.130: icmp: echo reply (DF) </pre>	<pre> root:~# tcpdump -n -i eth1 ip host 62.140.135.231 or host 131.155.193.130 or ip broadcast or broadcast tcpdump: listening on eth1 16:22:29.563894 62.140.135.231 > 131.155.193.130: icmp: echo request (DF) (ipip) 16:22:29.579570 131.155.193.134 > 62.140.135.231: 131.155.2.3 > 131.155.193.130: icmp: echo reply (DF) (ipip) 16:22:30.559515 62.140.135.231 > 131.155.193.134: 131.155.193.130 > 131.155.2.3: icmp: echo request (DF) (ipip) 16:22:30.569837 131.155.193.134 > 62.140.135.231: 131.155.2.3 > 131.155.193.130: icmp: echo reply (DF) (ipip) 16:22:31.559496 62.140.135.231 > 131.155.193.134: 131.155.193.130 > 131.155.2.3: icmp: echo request (DF) (ipip) 16:22:31.570088 131.155.193.134 > 62.140.135.231: 131.155.2.3 > 131.155.193.130: icmp: echo reply (DF) (ipip) 16:22:32.560336 62.140.135.231 > 131.155.193.134: 131.155.193.130 > 131.155.2.3: icmp: echo request (DF) (ipip) 16:22:32.591103 131.155.193.134 > 62.140.135.231: 131.155.2.3 > 131.155.193.130: icmp: echo reply (DF) (ipip) </pre>



Appendix G Outdated Internet Drafts

Route Optimization in Mobile IP

Mobile IP Working Group Charles Perkins
INTERNET DRAFT Nokia Research Center
6 September 2001 David B. Johnson
Carnegie Mellon University

Perkins and Johnson Expires 6 March 2002 [Page ii]

Route Optimization in Mobile IP
draft-ietf-mobileip-optim-11.txt

Internet Draft Route Optimization in Mobile IP 6 September 2001

Status of This Memo

1. Introduction

This document is a submission by the mobile-ip Working Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the MOBILE-IP@STANDARDS.NORTELNETWORKS.COM mailing list.

The base Mobile IP protocol [12], allows any mobile node to move about, changing its point of attachment to the Internet, while continuing to be identified by its home IP address. Correspondent nodes send IP datagrams to a mobile node at its home address in the same way as with any other destination. This scheme allows transparent interoperation between mobile nodes and their correspondent nodes, but forces all datagrams for a mobile node to be routed through its home agent. Thus, datagrams to the mobile node are often routed along paths that are significantly longer than optimal. For example, if a mobile node is visiting some subnet, even datagrams from a correspondent node on the same subnet must be routed through the Internet to the mobile node's home agent (on its home network), only then to be tunneled back to the original subnet for final delivery. This indirect routing delays the delivery of the datagrams to mobile nodes, and places an unnecessary burden on the networks and routers along their paths through the Internet.

Distribution of this memo is unlimited.

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

In this document, we will define extensions to the operation of the base Mobile IP protocol to allow for better routing, so that datagrams can be routed from a correspondent node to a mobile node without going to the home agent first. We refer collectively to these extensions as Route Optimization.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at:
<http://www.ietf.org/ietf/lid-abstracts.txt>
The list of Internet-Draft Shadow Directories can be accessed at:
<http://www.ietf.org/shadow.html>.

Route Optimization extensions provide a means for nodes to cache the binding of a mobile node and to then tunnel their own datagrams directly to the care-of address indicated in that binding, bypassing the mobile node's home agent. Extensions are also provided to allow datagrams in flight when a mobile node moves, and datagrams sent based on an out-of-date cached binding, to be forwarded directly to the mobile node's new care-of address.

Abstract

Using the base Mobile IP protocol, all datagrams destined to a mobile node are routed through that mobile node's home agent, which then tunnels each datagram to the mobile node's current location. This document defines Route Optimization messages and extensions to the base protocol to optimize datagram routing to a mobile node. Using these protocol extensions, correspondent nodes may cache the binding of a mobile node, and then tunnel their datagrams for the mobile node directly to the care-of address, bypassing the mobile node's home agent. Extensions are also provided to allow datagrams in flight when a mobile node moves, and datagrams sent based on an out-of-date cached binding, to be forwarded directly to the mobile node's new binding.

All operation of Route Optimization that changes the routing of IP datagrams to the mobile node is authenticated using the same type of mechanisms defined in the base Mobile IP protocol. This authentication generally relies on a mobility security association established in advance between the sender and receiver of such messages. The association can be created using ISAKMP [7], or any of the registration key establishment methods specified in [11].

After Section 2 gives some extra terminology, Section 3 provides an overview of the basic protocol operations associated with Route Optimization. Section 4 defines the message types used to update binding caches. Subsequent sections show the formats for the messages, explaining the function of fields within each message. Home agent considerations are given in Section 7, and foreign agent considerations in Section 8.

Perkins and Johnson Expires 6 March 2002 [Page i]

Perkins and Johnson Expires 6 March 2002 [Page 1]

Internet Draft Route Optimization in Mobile IP 6 September 2001

Internet Draft Route Optimization in Mobile IP 6 September 2001

Contents

Status of This Memo 1
Abstract 1
1. Introduction 1
2. Terminology 2
3. Route Optimization Overview 2
3.1. Binding Caches 3
3.2. Foreign Agent Smooth Handoff 4
4. Route Optimization Message Formats 5
4.1. Binding Warning Message 6
4.2. Binding Request Message 7
4.3. Binding Update Message 8
4.4. Binding Acknowledge Message 12
5. Route Optimization Authentication Extension 13
5.1. Modified Registration Request Message 13
6. Format of Smooth Handoff Extensions 14
6.1. Previous Foreign Agent Notification Extension 14
6.2. Modified Mobility Agent Advertisement Extension 16
6.3. Binding Warning Extension 17
7. Miscellaneous Home Agent Operations 18
7.1. Home Agent Rate Limiting 18
7.2. Managing Binding Updates for Correspondent Nodes 18
8. Miscellaneous Foreign Agent Operations 18
8.1. Previous Foreign Agent Notification 19
8.2. Maintaining Binding Caches 20
8.3. Rate Limiting 20
9. Security Considerations 20
10. Acknowledgement 21
A. Mobility Security Association Management 23
B. Using a Master Key at the Home Agent 24
Addresses 25

2. Terminology
This document introduces the following terminology, in addition to that used to describe the base Mobile IP protocol:
Binding cache
A cache of mobility bindings of mobile nodes, maintained by a node for use in tunneling datagrams to those mobile nodes.
Binding update
A message indicating a mobile node's current mobility binding, and in particular its care-of address.
Registration Lifetime
The registration lifetime is the time duration for which a binding is valid. The term remaining registration lifetime means the amount of time remaining for which a registration lifetime is still valid, at some time after the registration was approved by the home agent.
Security Parameters Index (SPI)
An index identifying a security context between a pair of nodes among the contexts available in the Mobility Security Association. SPI values 0 through 255 are reserved [2].
Triangle Routing
A situation in which a Correspondent Host's packets to a Mobile Host follow a path which is longer than the optimal path because the packets must be forwarded to the Mobile Host via a Home Agent.
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

3. Route Optimization Overview

This section provides an overview of the protocols and operations of Route Optimization. These can be divided into two main parts:

1. Updating binding caches
2. Managing smooth handoffs between foreign agents

Perkins and Johnson Expires 6 March 2002 [Page 2]

Internet Draft Route Optimization in Mobile IP 6 September 2001

The first part of the document goes into detail about binding cache maintenance, and then smooth handoff is considered.

3.1. Binding Caches

Route Optimization provides a means for any node to maintain a binding cache containing the care-of address of one or more mobile nodes. When sending an IP datagram to a mobile node, if the sender has a binding cache entry for the destination mobile node, it MAY tunnel the datagram directly to the care-of address indicated in the cached mobility binding.

In the absence of any binding cache entry, datagrams destined for a mobile node will be routed to the mobile node's home network in the same way as any other IP datagram, and then tunneled to the mobile node's current care-of address by the mobile node's home agent. This is the only routing mechanism supported by the base Mobile IP protocol. With Route Optimization, as a side effect of this indirect routing of a datagram to a mobile node, the original sender of the datagram may be informed of the mobile node's current mobility binding, giving the sender an opportunity to cache the binding.

Any node may maintain a binding cache to optimize its own communication with mobile nodes. A node may create or update a binding cache entry for a mobile node only when it has received and authenticated the mobile node's mobility binding. As before, each binding in the binding cache also has an associated lifetime, specified in the Binding Update message in which the node obtained the binding. After the expiration of this time period, the binding is deleted from the cache. In addition, a node cache MAY use any reasonable strategy for managing the space within the binding cache. When a new entry needs to be added to the binding cache, the node MAY choose to drop any entry already in the cache, if needed, to make space for the new entry. For example, a least-recently used (LRU) strategy for cache entry replacement is likely to work well.

When a mobile node's home agent intercepts a datagram from the home network and tunnels it to the mobile node, the home agent may deduce that the original source of the datagram has no binding cache entry for the destination mobile node. The home agent SHOULD then send a Binding Update message to the original source node, informing it of the mobile node's current mobility binding. No acknowledgment for such a Binding Update message is needed, since additional future datagrams from this source node intercepted by the home agent for the mobile node will cause transmission of another Binding Update. For a Binding Update to be authenticated by the original source node, the source node and the home agent must have established a mobility security association.

Perkins and Johnson Expires 6 March 2002 [Page 3]

Internet Draft Route Optimization in Mobile IP 6 September 2001

Similarly, when any node (e.g., a foreign agent) receives a tunneled datagram, if it has a binding cache entry for the destination mobile node (and thus has no visitor list entry for this mobile node), the node receiving this tunneled datagram may deduce that the tunneling node has an out-of-date binding cache entry for this mobile node. In this case, the receiving node SHOULD send a Binding Warning message to the mobile node's home agent, advising it to send a Binding Update message to the node that tunneled this datagram. A correspondent node can determine the mobile node's home agent from the binding cache entry, because the home agent address is learned from the Binding Update that established this cache entry. The address of the node that tunneled this datagram can be determined from the datagram's header, since the address of the node tunneling this datagram is the outer source address of the encapsulated datagram. As in the case of a Binding Update sent by the mobile node's home agent, no acknowledgment of this Binding Warning is needed, since additional future datagrams for the mobile node tunneled by the same node will cause the transmission of another Binding Warning. However, unlike the Binding Update message, no authentication of the Binding Warning message is necessary, since it does not directly affect the routing of IP datagrams to the mobile node.

When sending an IP datagram, if the sending node has a binding cache entry for the destination node, it SHOULD tunnel the datagram to the mobile node's care-of address using the encapsulation techniques used by home agents, and described in [9, 10, 3, 4].

3.2. Foreign Agent Smooth Handoff

When a mobile node moves and registers with a new foreign agent, the base Mobile IP protocol does not notify the mobile node's previous foreign agent. IP datagrams intercepted by the home agent after the new registration are tunneled to the mobile node's new care-of address, but datagrams in flight that had already been intercepted by the home agent and tunneled to the old care-of address when the mobile node moved are likely to be lost and are assumed to be retransmitted by higher-level protocols if needed. The old foreign agent eventually deletes its visitor list entry for the mobile node after the expiration of the registration lifetime.

Route Optimization provides a means for the mobile node's previous foreign agent to be reliably notified of the mobile node's new mobility binding, allowing datagrams in flight to the mobile node's previous foreign agent to be forwarded to its new care-of address. This notification also allows any datagrams tunneled to

the mobile node's previous foreign agent, from correspondent nodes with out-of-date binding cache entries for the mobile node, to be forwarded to its new care-of address. Finally, this notification

Perkins and Johnson Expires 6 March 2002 [Page 4]

Internet Draft Route Optimization in Mobile IP 6 September 2001

allows any resources consumed by the mobile node at the previous foreign agent (such as radio channel reservations) to be released immediately, rather than waiting for its registration lifetime to expire.

As part of the registration procedure, the mobile node MAY request that its new foreign agent attempt to notify its previous foreign agent on its behalf, by including a Previous Foreign Agent Notification extension in its Registration Request message sent to the new foreign agent. The new foreign agent then builds a Binding Update message and transmits it to the mobile node's previous foreign agent as part of registration, requesting an acknowledgment from the previous foreign agent. The extension includes only those values needed to construct the Binding Update message that are not already contained in the Registration Request message. The authenticator for the Binding Update message is computed by the mobile node using the security association shared with its previous foreign agent. This notification will typically include the mobile node's new care-of address, allowing the previous foreign agent to create a binding cache entry for the mobile node to serve as a forwarding pointer [5] to its new location. Any tunneled datagrams for the mobile node that arrive at its previous foreign agent after the forwarding pointer has been created can then be re-tunneled to the mobile node's new care-of address.

For this smooth handoff to be secure during registration with a new foreign agent, the mobile node and the previous foreign agent must have a security association. The security association is used to authenticate the notification sent to the previous foreign agent.

The Mobility Agent Advertisement extension of the agent advertisement message is revised under route optimization to include a bit indicating that the foreign agent supports smooth handoffs.

The mobile node is responsible for occasionally retransmitting a Binding Update message to its previous foreign agent until the matching Binding Acknowledge message is received, or until the mobile node can be sure that foreign agent has expired its binding. The mobile node is likely to select a small timeout value for the lifetime available to such bindings sent to previous foreign agents.

4. Route Optimization Message Formats

Route Optimization defines four message types used for management of binding cache entries. These message types fit in the numbering space defined in the base Mobile IP specification for messages sent to UDP port 434. Each of these messages begins with a one-octet field indicating the type of the message. The binding cache

Perkins and Johnson Expires 6 March 2002 [Page 5]

Internet Draft Route Optimization in Mobile IP 6 September 2001

management messages in this section are carried by way of UDP, sent to port 434.

The following type codes are defined in this document:

- | | |
|----|-----------------------------|
| 16 | Binding Warning message |
| 17 | Binding Request message |
| 18 | Binding Update message |
| 19 | Binding Acknowledge message |

Route Optimization also requires one minor change to existing Mobile IP messages: a new flag bit must be added to the Registration Request message, replacing a previously unused, reserved bit in the message.

This section describes each of the new Route Optimization messages and the change to Registration Request message.

4.1. Binding Warning Message

A Binding Warning message is used to transmit advice that a Binding Update is needed by one or more correspondent nodes or foreign agents. This happens when the suggested recipients are likely to have either no binding cache entry or an out-of-date binding cache entry for some mobile node. When any node detunnels a datagram destined for the mobile node, if it is not the current foreign agent for the destination mobile node, that foreign agent SHOULD send a Binding Warning message to the mobile node's home agent. If the foreign agent does not have any information about the mobile node's home agent, the foreign agent SHOULD send a Binding Warning message to the sender of the datagram (i.e., the correspondent node).

```

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      |                               Reserved      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Mobile Node Home Address |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Target Node Addresses   |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Binding Warning message is illustrated above, and contains the following fields:

- | | |
|----------|----------------------------------|
| Type | 16 |
| Reserved | Sent as 0; ignored on reception. |

Mobile Node Home Address
The home address of the mobile node to which the Binding Warning message refers.

Target Node Addresses
Zero or more addresses of nodes. Each address identifies a node that should be the target of a Binding Update message sent by the home agent. If no addresses are present, the recipient of the message is the intended target for the message.

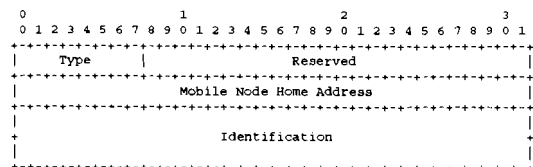
A home agent will receive a Binding Warning message if a node maintaining a binding cache entry for one of the home agent's mobile nodes uses an out-of-date entry. When a home agent receives a Binding Warning message, it SHOULD send a Binding Update message to each target node address identified in the Binding Warning, giving it the current binding for the mobile node identified in the mobile node home address field of the Binding Warning.

When a mobile node receives a new Care-of Address, it MAY send a Binding Warning message to its Home Agent, requesting that the home agent send Binding Update messages to one or more correspondent nodes. This feature MAY be used by the mobile node when it returns to its home network, so that the Home Agent will send out Binding Updates with zero lifetimes to all the mobile node's correspondent nodes. It is important for the correspondent nodes to delete their binding cache entries for the mobile node when the mobile node no longer has a Care-of Address.

If a foreign agent receives a packet for a mobile node for which there isn't any visitor list or binding cache information available, the foreign agent SHOULD send the Binding Warning to the correspondent node that transmitted the undeliverable message.

4.2. Binding Request Message

A Binding Request message is used by a node to request a mobile node's current mobility binding from a mobile node or the mobile node's home agent.



The format of the Binding Request message is illustrated above, and contains the following fields:

- Type 17
- Reserved Sent as 0; ignored on reception.
- Mobile Node Home Address The home address of the mobile node to which the Binding Request refers.
- Identification A 64-bit sequence number, assigned by the node sending the Binding Request message, used to assist in matching requests with replies, and in protecting against replay attacks.

When the home agent receives a Binding Request message, it consults its home list and determines the correct binding information to be sent to the requesting node. Before satisfying the request, the home agent is required to check whether or not the mobile node has allowed the information to be disseminated. If the mobile node specified the private (P) bit in its Registration Request message, then the home agent must make no further attempt to satisfy Binding Requests on behalf of that mobile node. In this case, the home agent SHOULD return a Binding Update in which both the care-of address is set equal to the mobile node's home address and the lifetime is set to zero. Such a Binding Update message indicates that the binding cache entry for the specified mobile node should be deleted.

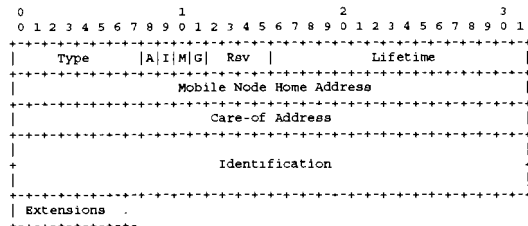
4.3. Binding Update Message

The Binding Update message is used for notification of a mobile node's current mobility binding. Subject to rate-limiting

provisions, it SHOULD be sent by the mobile node's home agent in the following situations:

- in response to a Binding Request message,
- in response to a Binding Warning message,
- in response to the reception of a Binding Warning extension to a Registration Request,
- in response to the reception of a packet destined for a mobile node.

A Binding Update SHOULD also be sent by a mobile node, or by the foreign agent with which the mobile node is registering, when notifying the mobile node's previous foreign agent that the mobile node has moved.



The format of the Binding Update message is illustrated above, and contains the following fields:

- Type 18
- A The 'A' (acknowledge) bit is set by the node sending the Binding Update message to request a Binding Acknowledge message be returned.
- I The 'I' (identification present) bit is set by the node sending the Binding Update message if the identification field is present in the message.

- M If the 'M' (minimal encapsulation) bit is set, datagrams MAY be tunneled to the mobile node using the minimal encapsulation protocol [10]
- G If the 'G' (Generic Record Encapsulation, or GRE) bit is set, datagrams MAY be tunneled to the mobile node using GRE [3]
- Rsv Reserved. Sent as 0; ignored on reception.
- Lifetime The number of seconds remaining before the binding cache entry must be considered expired. A value of all ones indicates infinity. A value of zero indicates that no binding cache entry for the mobile node should be created and that any existing binding cache entry (and visitor list entry, in the case of a mobile node's previous foreign agent) for the mobile node should be deleted. The lifetime is typically equal to the remaining lifetime of the mobile node's registration.

- Mobile Node Home Address The home address of the mobile node to which the Binding Update message refers.
- Care-of Address The current care-of address of the mobile node. When set equal to the home address of the mobile node, the Binding Update message instead indicates that no binding cache entry for the mobile node should be created, and any existing binding cache entry (and visitor list entry, in the case of a mobile node's previous foreign agent) for the mobile node should be deleted.
- Identification If present, a 64-bit number, assigned by the node sending the Binding Request message, used to assist in matching requests with replies, and in protecting against replay attacks.

Each Binding Update message indicates the binding's maximum lifetime. When sending the Binding Update message, the home agent SHOULD set this lifetime to the remaining registration lifetime. A node wanting to provide continued service with a particular binding cache entry MAY attempt to reconfirm that mobility binding before the expiration of the registration lifetime. Such reconfirmation of a binding cache entry may be appropriate when the node has indications (such as an open transport-level connection to the mobile node) that the binding

cache entry is still needed. This reconfirmation is performed by the node sending a Binding Request message to the mobile node's home agent, requesting it to reply with the mobile node's current

mobility binding in a new Binding Update message. Note that the node maintaining the binding SHOULD also keep track of the home agent's address, to be able to fill in the destination IP address of future Binding Requests.

As stated in Section 4.2, if the home agent chooses to respond to a Binding Request for a mobile node that set the 'P' bit in its registration, or which has returned home, it MUST send a Binding Update with the care-of address set to the mobile node's home address and with the lifetime set to zero. The home agent may also send such zero-lifetime Binding Updates to correspondent nodes named in a Binding Warning extension to a registration request that has the 'P' bit set or which is effecting de-registration of the mobile node. Finally, the home agent MAY also send such zero-lifetime Binding Updates to foreign agents from which the mobile node was previously registered but which are no longer serving the mobile node. This would allow such foreign agents to immediately reclaim any state information that pertained to the mobile node without waiting for the requisite lifetime to expire.

When a node receives a Binding Update message, it is required to verify the authentication in the message, using the mobility security association it shares with the sender's home agent. In such cases, the authentication data is found in the Route Optimization or Smooth Handoff authentication extension (Section 5), which is required. If the authentication succeeds, then a binding cache entry SHOULD be updated for use in future transmissions of data to the mobile node. Otherwise, an authentication exception SHOULD be raised.

Under all circumstances, the sending of Binding Update messages is subject to the rate limiting restriction described in Section 7.1.

When using nonces for replay protection, the identification field in the Binding Update message is used differently, to still allow replay protection even though the Binding Update is not being sent in reply to a request directly from the target node. In this case, the home agent is required to set the high-order 32 bits of the identification field to the value of the nonce that will be used by the home agent in the next Binding Update message sent to this node. The low-order 32 bits of the identification field are required to be set to the value of the nonce being used for this message.

Thus, on each Binding Update message, the home agent communicates to the target node, the value of the nonce that will be used next time. If no Binding Updates are lost in the network, the home agent and the target node can remain synchronized with respect to the nonces being

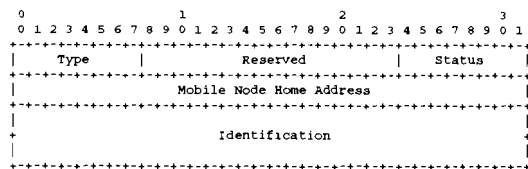
Perkins and Johnson Expires 6 March 2002 [Page 11]

Internet Draft Route Optimization in Mobile IP 6 September 2001

used. If, however, the target node receives a Binding Update with what it believes to be an incorrect nonce, it MAY resynchronize with the home agent by using a Binding Request message.

4.4. Binding Acknowledge Message

A Binding Acknowledge message is used to acknowledge receipt of a Binding Update message. It SHOULD be sent by a node receiving a Binding Update message in which the acknowledge (A) bit is set; if in addition that message also contains a valid authentication extension and Identification, the Binding Acknowledge MUST be sent.



The format of the Binding Acknowledge message is illustrated above, and contains the following fields:

Type	19
Status	If the Status is nonzero, this acknowledgment is negative. For instance, if the Binding Update was not accepted, but the incoming datagram has the Acknowledge flag set, then the status code should be set appropriately in the Binding Acknowledge message.
Reserved	Sent as 0; ignored on reception.
Mobile Node Home Address	Copied from the Binding Update message being acknowledged.
Identification	Copied from the Binding Update message being acknowledged, if present there.

Perkins and Johnson Expires 6 March 2002 [Page 12]

Internet Draft Route Optimization in Mobile IP 6 September 2001

Allowable values for the Status include:

- 128 reason unspecified
- 129 administratively prohibited
- 130 insufficient resources
- 131 sending node failed authentication
- 133 identification mismatch
- 134 poorly formed Binding Update

Up-to-date values of the Code field are specified in the most recent "Assigned Numbers" [13]

5. Route Optimization Authentication Extension

The Route Optimization Authentication extension is used to authenticate Route Optimization management messages sent with an SPI corresponding to the source IP address of the message. This extension is subtype TBD of the Generalized Authentication Extension [2]. The authenticator value is computed, as before, from the stream of bytes including the shared secret, the UDP payload (that is, the Route Optimization management message), all prior extensions in their entirety, and the type, subtype, length, and SPI of this extension, but not including the authenticator field itself nor the UDP header. This extension is required to be used in any Binding Update message sent by the Home Agent or the Mobile Node.

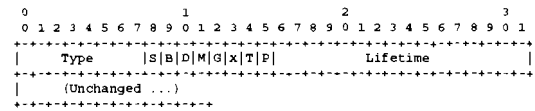
5.1. Modified Registration Request Message

One bit is added to the flag bits in the Registration Request message to indicate that the mobile node would like its home agent to keep its mobility binding private. Normally, the home agent sends Binding Update messages to correspondent nodes as needed to allow them to cache the mobile node's binding. If the mobile node sets the private ('P') bit in the Registration Request message, the home agent MUST NOT send the mobile node's binding in any Binding Update message. Instead, each Binding Update message SHOULD give the mobile node's care-of address equal to its home address, and SHOULD give a lifetime value of 0.

Thus, the Registration Request message under Route Optimization begins as shown below:

Perkins and Johnson Expires 6 March 2002 [Page 13]

Internet Draft Route Optimization in Mobile IP 6 September 2001



P The private ('P') bit is set by the node sending the Binding Update message to indicate that the home agent MUST keep its mobility binding private. In any Binding Update message sent by the mobile node's home agent, the care-of address SHOULD be set equal to the mobile node's home address, and the lifetime SHOULD be set equal to 0.

The other flag bits are as defined in the base Mobile IP specification [12] and for Reverse Tunneling [8]

6. Format of Smooth Handoff Extensions

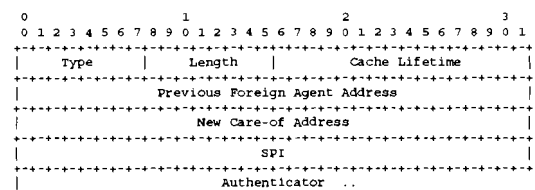
This section specifies the format for messages which are used to enable smooth handoff from a mobile node's previous foreign agent to its new foreign agent when a mobile node initiates a new registration.

6.1. Previous Foreign Agent Notification Extension

The Previous Foreign Agent Notification extension MAY be included in a Registration Request message sent to a mobility agent (either a foreign agent or the mobile node's home agent). It instructs the mobility agent to send a Binding Update message to the mobile node's previous foreign agent on behalf of the mobile node, to notify it that the mobile node has moved. The previous foreign agent SHOULD then delete the mobile node's visitor list entry and, if a new care-of address is included in the Binding Update message, create a binding cache entry for the mobile node with its new care-of address. The Previous Foreign Agent Notification extension contains only those values not otherwise already contained in the Registration Request message that are needed for the new foreign agent to construct the Binding Update message.

Perkins and Johnson Expires 6 March 2002 [Page 14]

Internet Draft Route Optimization in Mobile IP 6 September 2001



```

-----
Type      96
Length    14 plus the length of the authenticator
Cache Lifetime
The number of seconds remaining before the binding
cache entry created by the previous foreign agent must
be considered expired. A value of all ones indicates
infinity. A value of zero indicates that the previous
foreign agent MUST NOT create a binding cache entry for
the mobile node once it has deleted the mobile node's
visitor list entry. The cache lifetime value is copied
into the lifetime field of the Binding Update message.

Previous Foreign Agent Address
The IP address of the mobile node's previous foreign
agent to which the new foreign agent should send a
Binding Update message on behalf of the mobile node.

New Care-of Address
The address for the new mobility agent to send in the
Binding Update message to the previous foreign agent.

SPI       Security Parameters Index (4 bytes). An opaque
identifier. The SPI is copied over into the Smooth
Handoff authentication extension by the new foreign
agent.

Authenticator
The authenticator value to be used in the Route
Optimization Authentication extension in the Binding
Update message sent by the new foreign agent to the
mobile node's previous foreign agent. This authenticator
is calculated only over the Binding Update message body.

```

Perkins and Johnson Expires 6 March 2002 [Page 15]
 Internet Draft Route Optimization in Mobile IP 6 September 2001

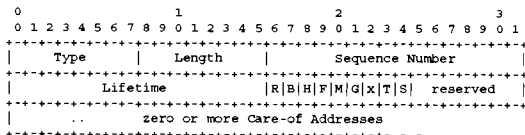
If a binding cache entry is created at the mobile node's previous foreign agent, it is treated in the same way as any other binding cache entry. The New Care-of Address in the extension SHOULD be either the care-of address being registered in the new registration (to cause IP datagrams from the previous foreign agent to be tunneled to the new foreign agent) or the mobile node's home address (to cause the previous foreign agent to delete its visitor list entry only for the mobile node, but not forward datagrams for it). This latter feature is especially valuable when a mobile node returns to its home network.

Mobile nodes SHOULD assign a small value to the Cache Lifetime, so that the binding created at the previous foreign agent will not take up space in the foreign agent's binding cache for very long.

The Binding Update sent by the mobility agent to the previous foreign agent MUST have the IP address of the foreign agent as the source address in the IP header. Conceptually, the mobility agent is "forwarding" a Binding Update to the previous foreign agent, albeit in a way that is specialized to the needs of the mobile node to reestablish connectivity with the fewest number of packet transmissions over its own link. The mobility agent MUST set the "A" bit in the Binding Update message, so that the previous foreign agent will know to send a Binding Acknowledge message back to the mobile node.

6.2. Modified Mobility Agent Advertisement Extension

Performing smooth handoffs requires one minor change to the existing Mobile IP Mobility Agent Advertisement extension [12]. A new flag bit, the 'S' bit, replaces a previously unused reserved bit in the extension, to indicate that the foreign agent supports smooth handoffs. By default, every foreign agent that supports smooth handoffs SHOULD support at least the establishment of a registration key by using elliptic curve key exchange [11].



Thus, the proposed modification to the Mobility Agent Advertisement extension, illustrated above, keeps the advertisement almost the

Perkins and Johnson Expires 6 March 2002 [Page 16]
 Internet Draft Route Optimization in Mobile IP 6 September 2001

same as in the base Mobile IP specification, except for adding the following bit:

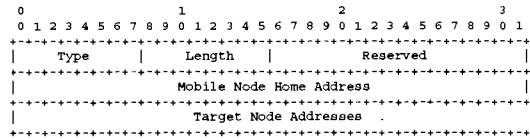
S The 'S' smooth handoff bit is set by the foreign agent sending the agent advertisement message to indicate that it supports the smooth handoffs, and thus the Registration Key Request extension [11].

More detailed information about the handling of this extension by foreign agents is deferred until Section 8.1.

6.3. Binding Warning Extension

A mobile node MAY append a Binding Warning Extension to a Registration Request. The Binding Warning extension is used to advise a mobile node's home agent that one or more correspondent nodes are likely to have either no binding cache entry or an

out-of-date binding cache entry for the mobile node sending the Registration Request.



The format of the Binding Warning extension is illustrated above, and contains the following fields:

Type 16
 Reserved Sent as 0; ignored on reception.
 Mobile Node Home Address The home address of the mobile node to which the Binding Warning message refers.
 Target Node Addresses One or more addresses of the correspondent nodes that need to receive Binding Update messages. Each node should be the target of a Binding Update message sent by the home agent.

Perkins and Johnson Expires 6 March 2002 [Page 17]
 Internet Draft Route Optimization in Mobile IP 6 September 2001

When a home agent receives a Binding Warning extension as part of a valid Registration Request, it SHOULD send a Binding Update message to each target node address identified in the Binding Warning, giving it the current binding for the mobile node identified in the mobile node home address field of the Binding Warning.

When a mobile node returns to its home network, it SHOULD append a Binding Warning extension to the Registration Request message sent to its Home Agent, instructing its home agent to send Binding Update messages (naturally, with zero lifetimes) to one or more correspondent nodes. It is important for the correspondent nodes to delete their binding cache entries for the mobile node when the mobile node no longer has a Care-of Address.

7. Miscellaneous Home Agent Operations

7.1. Home Agent Rate Limiting

A home agent is required to provide some mechanism to limit the rate at which it sends Binding Update messages to the same node about any given mobility binding. This rate limiting is especially important because it is expected that, within the short term, most Internet nodes will not support maintenance of a binding cache. In this case, continual transmissions of Binding Update messages will only waste processing resources at the home agent and correspondent node, and along the Internet path between these nodes.

7.2. Managing Binding Updates for Correspondent Nodes

The home agent MAY keep a list of correspondent nodes from which it has received Binding Acknowledgements for Binding Updates for active registrations (i.e., registrations which have not yet timed out). In this case, when the home agent receives a valid Registration Request, it MAY transmit new Binding Updates to each correspondent node that is on its list for the particular mobile node. In order to know which correspondent nodes correctly received the Binding Updates, the home agent SHOULD set the "A" bit in the Binding Update, requesting an acknowledgement.

Rate-limiting MUST be employed by a Home Agent offering this service, as specified in section 7.1.

8. Miscellaneous Foreign Agent Operations

This section details various operational considerations important for foreign agents wishing to support smooth handoff. This includes

Perkins and Johnson Expires 6 March 2002 [Page 18]
 Internet Draft Route Optimization in Mobile IP 6 September 2001

processing Previous Foreign Agent Notification extensions, and the maintenance of up-to-date binding cache entries.

8.1. Previous Foreign Agent Notification

When a foreign agent receives a Previous Foreign Agent Notification extension, it creates a Binding Update for the previous foreign agent, using the specified SPI and precomputed authenticator sent to it by the mobile node.

When the previous foreign agent receives the Binding Update, it will authenticate the message using the mobility security association and SPI specified in the Binding Update. If the message authentication is correct, the visitor list entry for this mobile node at the previous foreign agent will be deleted and a Binding Acknowledge message returned to the sender. In addition, if a new care-of address was included in the Binding Update message, the previous foreign agent will create a binding cache entry for the mobile node; the previous foreign agent can then tunnel datagrams to the mobile node's new care-of address using that binding cache, just as any node maintaining a binding cache. The previous foreign agent is also expected to return a Binding Acknowledge message to the mobile node.

Note that this Binding Acknowledge is addressed to the mobile node, and SHOULD be tunneled using the new binding cache entry. The tunneled acknowledgment then SHOULD be delivered directly to the new foreign agent, without having to go to the home network. This creates an interesting problem for the new foreign agent when it receives the acknowledgment before the Registration Reply from the home agent. It is suggested that the new foreign agent deliver the acknowledgment to the mobile node anyway, even though the mobile node is technically unregistered. If there is concern that this provides a loophole for unauthorized traffic to the mobile node, the new foreign agent could limit the number of datagrams delivered to the unregistered mobile node to this single instance. Alternatively, a new extension to the Registration Reply message can be defined to carry along the acknowledgment from the previous foreign agent. This latter approach would have the benefit that fewer datagrams would be transmitted over bandwidth-constrained wireless media during registration.

When the Binding Acknowledge message from the previous foreign agent is received by the new foreign agent, it detunnels it and sends it to the mobile node. In this way, the mobile node can discover that its previous foreign agent has received the Binding Update message. The mobile node must be certain that its previous foreign agent has been notified about its new care-of address, because otherwise the previous foreign agent could become a "black hole"

Perkins and Johnson Expires 6 March 2002 [Page 19]

Internet Draft Route Optimization in Mobile IP 6 September 2001

for datagrams destined for the mobile node based on out-of-date binding cache entries at other nodes. The new foreign agent has no further responsibility for helping to update the binding cache at the previous foreign agent, and does not retransmit the message even if no acknowledgment is received.

If the acknowledgment has not been received after sufficient time, the mobile node is responsible for retransmitting another Binding Update message to its previous foreign agent. Although the previous foreign agent may have already received and processed the Binding Update message (the Binding Acknowledge message may have been lost in transit to the new foreign agent), the mobile node SHOULD continue to retransmit its Binding Update message until the previous foreign agent responds with a Binding Acknowledge.

8.2. Maintaining Binding Caches

The binding cache entry built by the previous foreign agent from the information in the Previous Foreign Agent Notification extension MAY be deleted from its Binding Cache at any time, and these cache entries are expected to be created with short lifetimes (see section 6.1). In this case, the previous foreign agent will be unable to find a current care-of address for subsequently arriving tunneled datagrams for the mobile node.

8.3. Rate Limiting

A foreign agent MUST provide some mechanism to limit the rate at which it sends Binding Warning messages to the same node about any given mobility binding. This rate limiting is especially important because it is expected that, within the short term, many Internet nodes will not support maintenance of a binding cache. In this case, continual transmissions of Binding Warning messages will only waste processing resources at the foreign agent and correspondent node, and along the Internet path between these nodes.

9. Security Considerations

The calculation of the authentication data supplied with the Route Optimization and Smooth Handoff authentication extensions in section 5 is specified to be the same as in the base Mobile IP document for ease of implementation. There is a better method available (HMAC), specified in RFC 2104 [6]. If the base Mobile IP specification is updated to use HMAC, then this route optimization specification SHOULD also be updated similarly.

Perkins and Johnson Expires 6 March 2002 [Page 20]

Internet Draft Route Optimization in Mobile IP 6 September 2001

10. Acknowledgement

Expanding the Binding Warning to allow a mobile node to send a list of correspondent nodes to the Home Agent was suggested by Mohamad Khalil, Emad Qaddoura, Haseeb Akhtar, and Liem Le of Nortel Networks. Pete McCann of Lucent also contributed text specifying additional considerations under which the home agent could send zero-lifetime Binding Updates in section 4.3.

Perkins and Johnson Expires 6 March 2002 [Page 21]

Internet Draft Route Optimization in Mobile IP 6 September 2001

References

- [1] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. Request for Comments (Best Current Practice) 2119, Internet Engineering Task Force, March 1997.
- [2] P. Calhoun and C. E. Perkins. Mobile IP Foreign Agent Challenge/Response Extension, December 2000.
- [3] S. Hanks, T. Li, D. Farinacci, and P. Traina. Generic Routing Encapsulation (GRE). Request for Comments (Informational) 1701, Internet Engineering Task Force, October 1994.
- [4] S. Hanks, T. Li, D. Farinacci, and P. Traina. Generic Routing Encapsulation over IPv4 networks. Request for Comments (Informational) 1702, Internet Engineering Task Force, October 1994.
- [5] David B. Johnson. Scalable and Robust Internetwork Routing for Mobile Hosts. In Proceedings of the 14th International Conference on Distributed Computing Systems, pages 2--11, June 1994.
- [6] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. Request for Comments (Informational) 2104, Internet Engineering Task Force, February 1997.
- [7] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet Security Association and Key Management Protocol (ISAKMP) Request for Comments (Proposed Standard) 2408, Internet Engineering Task Force, November 1998.
- [8] G. Montenegro. Reverse Tunneling for Mobile IP. Request for Comments (Proposed Standard) 2344, Internet Engineering Task Force, May 1998.
- [9] C. Perkins. IP Encapsulation within IP. Request for Comments (Proposed Standard) 2003, Internet Engineering Task Force, October 1996.
- [10] C. Perkins. Minimal Encapsulation within IP. Request for Comments (Proposed Standard) 2004, Internet Engineering Task Force, October 1996.
- [11] C. Perkins and D. Johnson. Registration Keys for Route Optimization (work in progress). Internet Draft, Internet Engineering Task Force, December 1997.

Perkins and Johnson Expires 6 March 2002 [Page 22]

Internet Draft Route Optimization in Mobile IP 6 September 2001

- [12] C. Perkins, Editor. IP Mobility Support version 2 (work in progress) draft-ietf-mobileip-rfc2002-bis-03.txt, September 2000.
- [13] J. Reynolds and J. Postel. Assigned Numbers. Request for Comments (Standard) 1700, Internet Engineering Task Force, October 1994.

A. Mobility Security Association Management

One of the most difficult aspects of Route Optimization for Mobile IP in the Internet today is that of providing authentication for all messages that affect the routing of datagrams to a mobile node. In the base Mobile IP protocol, only the home agent is aware of the mobile node's mobility binding and only the home agent tunnels datagrams to the mobile node. Thus, all routing of datagrams to the mobile node while away from its home network is controlled by the home agent. Authentication is currently achieved based on a manually established mobility security association between the home agent and the mobile node. Since the home agent and the mobile node are both owned by the same organization (both are assigned IP addresses within the same IP subnet), this manual configuration is manageable, and (for example) can be performed while the mobile node is at home.

However, with Route Optimization, authentication is more difficult to manage, since a Binding Update may in general need to be sent to almost any node in the Internet. Since no authentication or Key distribution protocol is generally available in the Internet today, the Route Optimization procedures defined in this document MAY make use of the same type of manual Key distribution discussed in the base Mobile IP protocol. For use with Route Optimization, a mobility security association held by a correspondent node or a foreign agent

must include the same parameters as required by base Mobile IP [12]

For a correspondent node to be able to create a binding cache entry for a mobile node, the correspondent node needs a mobility security association with either the mobile node or its home agent. This mobility security association, though, could be used in creating and updating binding cache entries at this correspondent node for all mobile nodes served by this home agent. Doing so places the correspondent node in a fairly natural relationship with respect to the mobile nodes served by this home agent. For example, the mobile nodes may represent different people affiliated with the same organization owning the home agent, with which the user of the correspondent node often collaborates. The effort of establishing such a mobility security association with the relevant home agent may be more manageable (appendix B) than the effort of doing so with each mobile node. It is similarly possible for a home agent to have a

Perkins and Johnson Expires 6 March 2002 [Page 23]

Internet Draft Route Optimization in Mobile IP 6 September 2001

manually established mobility security association with the foreign agents often used by its mobile nodes, or for a particular mobile node to have a manually established mobility security association with the foreign agents serving the foreign networks that it often visits.

In general, if the movement and communication patterns of a mobile node or the group of mobile nodes served by the same home agent are sufficient to justify establishing a mobility security association with the mobile node's home agent, users or network administrators are likely to do so. Without establishing a mobility security association, nodes will not currently be able to authenticate the values transmitted in Route Optimization extensions.

B. Using a Master Key at the Home Agent

Rather than storing each mobility security association that it has established with many different correspondent nodes and foreign agents, a home agent MAY manage its mobility security associations so that each of them can be generated from a single master key. With the master key, the home agent could build a key for any given other node, for example by computing the node-specific key as

MD5 (node-address | master-key | node-address)

where node-address is the IP address of the particular node for which the home agent is building a key, and master-key is the single master key held by the home agent for all mobility security associations it has established with correspondent nodes. The node-specific key is built by computing an MD5 hash over a string consisting of the master key with the node-address concatenated as a prefix and as a suffix.

Using this scheme, when establishing each mobility security association, the network administrator managing the home agent computes the node-specific key and communicates this key to the network administrator of the other node through some secure channel, such as over the telephone. The mobility security association is configured at this other node in the same way as any mobility security association. At the home agent, though, no record need be kept that this key has been given out. The home agent need only be configured to know that this scheme is in use for all of its mobility

Perkins and Johnson Expires 6 March 2002 [Page 25]

security associations (perhaps only for specific set of its mobile nodes).

When the home agent needs a mobility security association as part of Route Optimization, it builds the node-specific key based on the master key and the IP address of the other node with which it is

Perkins and Johnson Expires 6 March 2002 (Page 24)

Internet Draft Route Optimization in Mobile IP 6 September 2001

attempting to authenticate. If the other node knows the correct node-specific key, the authentication will succeed; otherwise, it will fail as it should.

Addresses

The working group can be contacted via the current chairs:

Basavaraj Patil Nokia Corporation 6000 Connection Drive M/S M8-540 Irving, TX 75039 USA Phone: +1 972-894-6709 Fax: +1 972-894-5349 Email: Raj.Patil@nokia.com	Phil Roberts Megisto Corp. Suite 120 20251 Century Blvd Germantown MD 20874 USA Phone: +1 847-202-9314 Email: PROBERTS@MEGISTO.COM
--	---

Questions about this memo can also be directed to the authors:

Charles E. Perkins Communications Systems Lab Nokia Research Center 313 Fairchild Drive Mountain View, California 94043 USA Phone: +1-650 625-2986 Fax: -1 650 625-2502 Email: charliep@iprg.nokia.com	David B. Johnson Dept. Computer Science - MS 132 6100 Main Street Houston, Texas 77005-1892 USA Phone: +1-713-348-3063 Fax: +1-713-348-5930 E-mail: dbj@cs.rice.edu
--	--

AAA Registration Keys for Mobile IP

Mobile IP Working Group
INTERNET DRAFT
26 February 2002

Charles E. Perkins
Nokia Research Center
Pat R. Calhoun
Black Storm Systems

AAA Registration Keys for Mobile IP
draft-ietf-mobileip-aaa-key-09.txt

Status of This Memo

This document is a submission by the mobile-ip Working Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the mobile-ip@sunroof.eng.sun.com mailing list.

Distribution of this memo is unlimited.

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at:
<http://www.ietf.org/ietf/1id-abstracts.txt>
The list of Internet-Draft Shadow Directories can be accessed at:
<http://www.ietf.org/shadow.html>.

Abstract

AAA servers, such as RADIUS and DIAMETER, are in use within the Internet today to provide authentication and authorization services for dial-up computers. Mobile IP requires strong authentication between the mobile node and its home agent. When the mobile node shares a security association with its home AAA server, however, it is possible to use that security association to create derivative

security associations between the mobile node and its home agent, and again between the mobile node and the foreign agent currently offering connectivity to the mobile node. This document specifies extensions to the Mobile IP Registration Reply packet that can be used to create such security information at the mobile node.

Perkins, Calhoun Expires 26 August 2002 [Page 1]

Internet Draft AAA Keys for Mobile IP 26 February 2002

1. Introduction

AAA servers, such as RADIUS [13] and DIAMETER [4], are in use within the Internet today to provide authentication and authorization services for dial-up computers. Such services are likely to be equally valuable for mobile nodes using Mobile IP [12] when the nodes are attempting to connect to foreign domains with AAA servers. Requirements for interactions between AAA and Mobile IP are outlined in RFC 2977 [6], that document describes an infrastructure which enables AAA servers to authenticate and authorize network access requests from mobile nodes. See also appendix B. The Mobile IP Registration Request is considered to be a request for network access. It is then possible to augment the functionality of the Mobile IP mobility agents so that they can translate between Mobile IP registration messages and the messages used within the AAA infrastructure architected in RFC 2977. Mobility agents and AAA servers that conform to the requirements of RFC 2977 can be considered as appropriate network entities to support the message types specified in this document. Please consult RFC 2977 for further details.

Mobile IP requires strong authentication between the mobile node and its home agent. When the mobile node shares a security association

with its home AAA server, however, it is possible to use that security association to create derivative security associations between the mobile node and its home agent, and again between the mobile node and the foreign agent currently offering connectivity to the mobile node. This document specifies extensions to the Mobile IP Registration messages that can be used to create those security associations at the mobile node.

AAA servers typically use the Network Access Identifier (NAI) [1] to uniquely identify the mobile node; the mobile node's home address is not always necessary to provide that function. Thus, it is possible for a mobile node to authenticate itself, and be authorized for connection to the foreign domain, without having any home address. However, for Mobile IP to work, the mobile node is required to have a security association with its home agent. When the Mobile IP Registration Reply packet is authenticated by the MN-AAA Authentication Extension [3], the mobile node can verify that the keys contained in the extensions were produced by the AAA server, and thus may be reliably used to create security associations with the home agent, or alternatively with the foreign agent.

It is also assumed that the AAA entities involved (i.e., the AAAH, AAAL, and the AAA interface features of the foreign agents and home agents) all have means outside of the scope of this document for exchanging keys. The extensions within this document are intended to work with any AAA protocol suite that allows for such key exchange.

Perkins, Calhoun Expires 26 August 2002 [Page 2]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

2. Terminology

security association

The information shared by two network nodes that enables them to carry out the operations needed for some security protocol that the nodes intend to operate. For the purposes of this document, all security associations will contain the following information:

key a number, kept secret. Only nodes in possession of the key have any hope of using the security algorithm to obtain correct results.

SPI Security Parameters Index. This number enables selection of one security association in case that several exist between the two nodes operating a security procedure.

Also for the purposes of this document, a mobile node is allowed to have a security association with another node even though it does not necessarily know the IP address of that node. It is only required that the mobile node use the security association for purpose in accordance with the expectations of the other node.

security algorithm

A set of rules for using input data and a secret key for producing data for use in security protocols. For example, HMAC-MD5 [8] is the security algorithm that all nodes using Mobile IP must implement for the purposes of producing and verifying authentication data.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [2]. Other terminology is used as defined in the base Mobile IP specification [12].

Furthermore, in order to simplify the discussion, we have used the word "Extension" instead of "Subtype of the Generalized Extension" in many cases. So, for instance, instead of using the phrase "The Unsolicited MN-FA Key Material From AAA Subtype of the Generalized MN-FA Key Reply Extension", we would instead use the phrase "The Unsolicited MN-FA Key Material From AAA Extension".

Perkins, Calhoun Expires 26 August 2002 [Page 3]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

3. Overview of Operations with Key Extensions

When a mobile node depends on an AAA infrastructure to obtain authorization for network connectivity and Mobile IP registration, it may not have any pre-existing security relationships with either its home agent, or the foreign agent controlling the access to the foreign network. The extensions defined in this document allow a AAA agent to supply key material to mobile nodes to be used as the basis of its security association with mobile agents (foreign agents and home agents). The AAA agent that will act on these extensions is part of the AAA infrastructure, and is typically identified within the foreign domain by methods outside the scope of this specification (see appendix B).

The key material is requested by the mobile node in new extensions to Mobile IP Registration Request messages, and supplied to the mobile node in extensions to the Mobile IP Registration Reply messages. The method by which key material is supplied to the mobility agents themselves is out of scope for this document, and would depend on the particular details of the security architecture for the AAA servers in the foreign and home domains (see RFC 2977 and appendix B). For the purposes of this document, we assume that there is a suitable AAA infrastructure available to the foreign agents, and that the mobile node does have a security association with at least one AAA server in its home domain.

The protocol and messages in this document are intended to facilitate the following operations which may occur between the mobile node, AAA

server, home agent, and foreign agent. However, the only message flows specified in this document are the Registration Request between the mobile node and the foreign agent, and Registration Reply between the foreign agent and the mobile node.

1. When a mobile node travels away from home, it may not have a security association with its home agent, perhaps because it does not yet have a home address.
2. If the mobile node does not have a Mobility Security Association with the foreign agent, it SHOULD include an MN-FA Key Request extension (see Section 10) as part of its Registration Request that it sends to the Foreign Agent.
3. Similarly, if the mobile node does not have a Mobility Security Association with the home agent, it MUST add an MN-HA Key Request extension (see Section 11) as part of its Registration Request that it sends to the Foreign Agent.

Perkins, Calhoun Expires 26 August 2002 [Page 4]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

4. If one or more Key Request extensions were added, the mobile node adds the MN-AAA Authentication extension is added to its Registration Request.
5. By action of the foreign agent, which is presumed to be also a participant in some AAA protocol, the mobile node's key requests and authentication data are transferred, typically after reformatting to fit into the appropriate AAA messages, which are out of scope for this document.
6. At the time the information within the MN-AAA Authentication extension is verified by the AAA server, the AAA server also generates Key Material, if it has been requested by the mobile node.
7. The respective AAA keys are distributed to the Home and Foreign Agent via the AAA protocol.
8. The mobile node first generates the key using the Key Material provided, according to its security association with the AAA. Using that key, the mobile node authenticates the Reply message. If the Reply passes authentication and contains the Unsolicited MN-HA Key Material From AAA extension (see section 9), the generated key is then used to establish the mobile node's security association with its home agent, and is used to authenticate the MN-HA authentication extension.
9. Similarly, if the Reply passes authentication and contains the Unsolicited MN-FA Key Material From AAA extension (see section 8), the mobile node generates the key using the Key Material provided, according to its security association with the AAA. The resulting key is used to establish the mobile node's security association with its new foreign agent, and is used to compute the authentication data used in the Mobile-Foreign authentication extension.

Any registration reply containing the Unsolicited MN-HA Key Material From AAA extension MUST also contain a subsequent Mobile Home Authentication Extension, created using the generated MN-HA key. Similarly, a reply containing the Unsolicited MN-FA Key Material From AAA extension MUST also contain a subsequent Mobile Foreign Authentication Extension, created using the the MN-FA key.

4. Mobility Security Associations

Mobility Security Associations between Mobile IP entities (mobile nodes, home agents, foreign agents) contain both the necessary cryptographic key information, and a way to identify

Perkins, Calhoun Expires 26 August 2002 [Page 5]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

the cryptographic algorithm which uses the key to produce the authentication information typically included in the Mobile Home Authentication extension or the Mobile Foreign Authentication extension. In order for the mobile node to make use of key material created by the AAA server, the mobile node also has to be able to identify and select the appropriate cryptographic algorithm that uses the key to produce the authentication.

The algorithm identifiers are tabulated in the list of Authentication Algorithms allowable as values for the "Attribute Type" (5) (i.e., "Authentication Algorithm"), one of the classifications in the tabulated Attribute Types for "IPSEC Security Association Attributes". See <http://www.iana.org/assignments/isakmp-registry> for the full listing of all Attribute Types and other Attributes for IPSEC Security Associations.

Mobility Security Associations shared between mobile nodes and home agents also require a replay protection method. The following table contains the supported replay methods.

Replay Method	Name	Reference
1	None	RFC 3220 [12]
2	Timestamps	RFC 3220 [12]
3	Nonces	RFC 3220 [12]

5. Key Material Creation and Derivation

This section contains the procedures followed in the creation of the Key Material by AAA servers, and the key derivation procedures used

by mobile nodes. Note that the AAA servers will also make use of the derivation procedures to deliver the keys via the AAA protocol. AAA servers that follow these procedures will produce results that can be understood by mobile nodes. Mobility agents (home agent, foreign agent) will faithfully transcribe the results into the appropriate Mobile IP extensions.

The example that follows makes use of HMAC-MD5 [7]. All mobile nodes and mobility agents implementing Mobile IP, and implementing the extensions specified in this document, MUST implement HMAC-MD5 [12]. Other cryptographic functions MAY also be used.

The following steps are performed on the AAA server:

1. The AAA server identifies the mobile node. If the Home Address field of the Registration Request is either zero (0), or all ones

Perkins, Calhoun Expires 26 August 2002 [Page 6]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

(1), then the Mobile Node's NAI is used instead of the mobile node's home address.

2. The AAA server generates a random [5] value of at least 64 bits to be used as the Key Material.
3. The AAA server provides the random value for later insertion into the Key extension, in the "Key Material" field.

The following steps are performed on the mobile node:

1. The mobile node calculates
 $key = HMAC-MD5 (Key\ Material \parallel home\ address)$
2. The mobile node creates the security association, using the key and the other relevant information in the Key Extension.

The secret key used within the HMAC-MD5 computation is the AAA-key pointed to by the AAA SPI, which has been previously configured as the basis for a security association between the mobile node and the AAA server creating the key.

6. Generalized Key Request/Reply Extensions

The extensions in this section are Generalized Extensions, and have subtypes as specified in section 7.

6.1. Generalized MN-FA Key Request Extension

Figure 1 illustrates the Generalized MN-FA Key Request Extension.

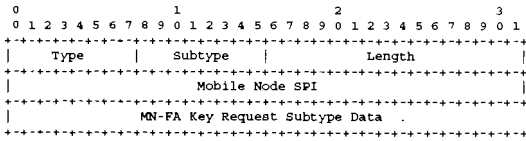


Figure 1: The Generalized Mobile IP MN-FA Key Request Extension

Perkins, Calhoun Expires 26 August 2002 [Page 7]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

Type	TBD (not skippable) (see [12] and section 13)
Subtype	a number assigned to identify the way in which the Key Request Data is to be used when generating the registration key
Length	The 16-bit Length field indicates the length of the extension. It is equal to the number of bytes in the MN-FA Key Request Subtype Data plus 4 (for the Mobile Node SPI field), and SHOULD be at least 20.
Mobile Node SPI	The Security Parameters Index that the mobile node will assign for the security association created for use with the registration key.
MN-FA Key Request Subtype Data	Data needed to carry out the creation of the registration key on behalf of the mobile node.

The Generalized MN-FA Key Request Extension defines a set of extensions, identified by subtype, which may be used by a mobile node in a Mobile IP Registration Request message to request that some other entity create a key for use by the mobile node with the mobile node's new foreign agent.

6.2. Generalized MN-FA Key Reply Extension

The Generalized MN-FA Key Reply extension supplies a registration key requested by using one of the subtypes of the Generalized MN-FA Key Request extension. Figure 2 illustrates the format Generalized MN-FA Key Reply Extension.

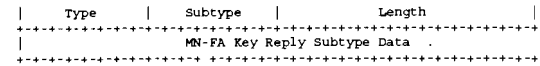
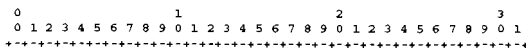


Figure 2: The Generalized Mobile IP MN-FA Key Reply Extension

Type TBD (not skippable) (see [12] and section 13)

Perkins, Calhoun Expires 26 August 2002 [Page 8]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

Subtype	a number assigned to identify the way in which the MN-FA Key Reply Subtype Data is to be decrypted to obtain the registration key
Length	The 16-bit Length field is equal to the number of bytes in the MN-FA Key Reply Subtype Data.
MN-FA Key Reply Subtype Data	An encoded copy of the key to be used between the mobile node and the foreign agent, along with any other information needed by the recipient to create the designated Mobility Security Association.

For each subtype, the format of the MN-FA Key Reply Subtype Data has to be separately defined according to the particular method required to set up the security association.

In some cases, the MN-FA Key supplied in the data for a subtype of this extension comes by a request which was sent using a subtype of the Generalized MN-FA Key Request Extension. In that case, the SPI to be used when employing the security association defined by the registration key is the same as given in the original request.

6.3. Generalized MN-HA Key Request Extension

Figure 3 illustrates the Generalized MN-HA Key Request Extension.

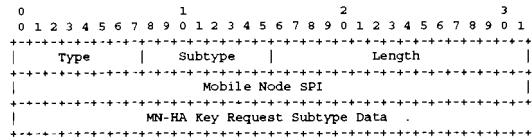


Figure 3: The Generalized Mobile IP MN-HA Key Request Extension

Type	TBD (not skippable) (see [12] and section 13)
Subtype	a number assigned to identify the way in which the Key Request Data is to be used when generating the registration key

Perkins, Calhoun Expires 26 August 2002 [Page 9]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

Length	The 16-bit Length field indicates the length of the extension. It is equal to the number of bytes in the MN-HA Key Request Subtype Data plus 4 (for the Mobile Node SPI field), and SHOULD be at least 20.
Mobile Node SPI	The Security Parameters Index that the mobile node will assign for the security association created for use with the registration key.
MN-HA Key Request Subtype Data	Data needed to carry out the creation of the registration key on behalf of the mobile node.

The Generalized MN-HA Key Request Extension defines a set of extensions, identified by subtype, which may be used by a mobile node in a Mobile IP Registration Request message to request that some other entity create a key for use by the mobile node with the mobile node's new home agent.

6.4. Generalized MN-HA Key Reply Extension

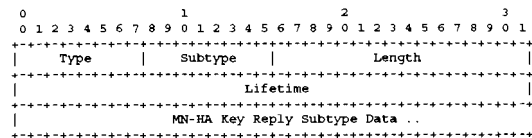


Figure 4: The Generalized Mobile IP MN-HA Key Reply Extension

Type	TBD (not skippable) (see [12] and section 13)
Subtype	a number assigned to identify the way in which the MN-HA Key Reply Subtype Data is to be decrypted to obtain the registration key

Length The 16-bit Length field indicates the length of the extension. It is equal to the number of bytes in the MN-HA Key Reply Subtype Data plus 4 (for the Lifetime field)

Perkins, Calhoun Expires 26 August 2002 [Page 10]

Internet Draft AAA Keys for Mobile IP 26 February 2002

Lifetime This field indicates the duration of time (in seconds) for which the MN-HA key is valid.

MN-HA Key Reply Subtype Data
An encrypted copy of the key to be used between the mobile node and its home agent, along with any other information needed by the mobile node to create the designated Mobility Security Association with the home agent.

For each subtype, the format of the MN-HA Key Reply Subtype Data has to be separately defined according to the particular method required to set up the security association.

7. Key Request/Reply Subtypes

The extension subtypes in this section are subtypes of the Generalized Extensions specified in section 6.

8. Unsolicited MN-FA Key Material From AAA Subtype

The Unsolicited MN-FA Key Material From AAA extension, shown in figure 5, uses subtype 7 of the Generalized MN-FA Key Reply Extension (see section 6.2).

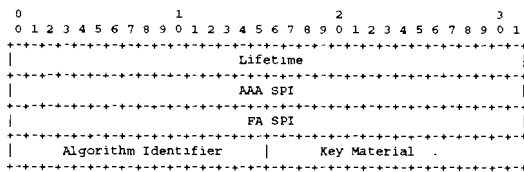


Figure 5: The Unsolicited MN-FA Key Material From AAA Subtype-Specific Data

Perkins, Calhoun Expires 26 August 2002 [Page 11]

Internet Draft AAA Keys for Mobile IP 26 February 2002

lifetime This field indicates the duration of time (in seconds) for which the MN-FA 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.

FA SPI The SPI for the Security Association to the FA that the MN creates as a result of processing this extension

Algorithm Identifier This field indicates the algorithm to be used (selected from among the values in the "Authentication Algorithm" table cited in section 4) for future computations of the Mobile-Foreign Authentication Extension.

Key Material A random [5] value of at least 64 bits.

The Key Material is added by the AAA server for use by the mobile node in creating the MN-FA key, which is used to secure future Mobile IP registrations with the same foreign agent. The Unsolicited MN-FA Key Material From AAA extension MUST appear in the Registration Reply before the Mobile-Foreign Authentication extension.

Once the mobile node creates the FA Security Information, by using the algorithm indexed by the AAA SPI, it stores the FA Security Information indexed by the FA SPI in its list of Mobile Security Associations.

If the foreign agent receives a Registration Reply that has no Unsolicited MN-FA Key Material From AAA extension, and thus cannot establish a Mobility Security Association with the mobile node, the foreign agent MAY change the Code value of the Registration Reply to MISSING_MN_FA (see section 12), effectively causing the registration to fail.

9. Unsolicited MN-HA Key Material From AAA Subtype

The Unsolicited MN-HA Key Material From AAA is subtype 1 of the Generalized MN-HA Key Reply Extension (see section 6.4)

Perkins, Calhoun Expires 26 August 2002 [Page 12]

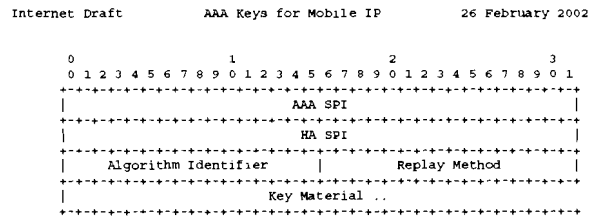


Figure 6: The Unsolicited MN-HA Key Material From AAA Subtype-Specific Data

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 HA security information.

HA SPI The SPI for the Security Association to the HA that the MN creates as a result of processing this extension

Algorithm Identifier This field indicates the algorithm to be used for future computations of the MN-HA Authentication Extension (see section 4)

Replay Method This field contains the replay method to be used for future Registration messages (see section 4)

Key Material A random [5] value of at least 64 bits.

The Unsolicited MN-HA Material Key From AAA subtype-specific data is shown in figure 6. The Mobile Node creates the MN-HA key using the Key Material that has previously been configured for securing all such communication requirements with the AAA server which will be contacted within the AAA infrastructure (see appendix B). The key is intended for use by the mobile node to secure future Mobile IP registrations with its home agent. The MN-HA Key Reply MUST appear in the Registration Reply before the MN-HA Authentication extension.

Once the mobile node creates the MN-HA Key, by using the algorithm specified in the AAA SPI, it stores the HA Security Information indexed by the HA SPI in its list of Mobile Security Associations.

Perkins, Calhoun Expires 26 August 2002 [Page 13]

Internet Draft AAA Keys for Mobile IP 26 February 2002

The mobile node uses the Identification field data from the Registration Request as its initial synchronization data with the home agent.

10. MN-FA Key Request From AAA Subtype

The MN-FA Key Request From AAA subtype data uses subtype 7 of the Generalized MN-FA Key Request Extension (see section 6.1). The MN-FA Key Request From AAA extension MUST appear in the Registration Request before the MN-AAA Authentication extension. The subtype data field is zero in length.

11. MN-HA Key Request From AAA Subtype

The MN-HA Key Request From AAA subtype data uses subtype 7 of the Generalized MN-HA Key Request Extension (see section 6.3). The MN-HA Key Request From AAA extension MUST appear in the Registration Request before the MN-AAA Authentication extension. The subtype data field is zero in length.

12. Error Values

Each entry in the following table contains the name of Code [12] to be returned in a Registration Reply, the value for the Code, and the section in which the error is first mentioned in this specification.

Error Name	Value	Section
MISSING_MN_FA	107	8

13. IANA Considerations

The numbers for the Generalized Extensions in section 6 are taken from the numbering space defined for Mobile IP registration extensions defined in RFC 3220 [12] as extended in RFC 2356 [10]. The numbers suggested in this section are already in use by implementations which have been tested for interoperability.

The number 7, assigned to the Unsolicited MN-HA Key Material From AAA Subtype extension, was taken from the numbering space defined for the Generalized MN-HA Key Reply Extension (see section 6.4)

Perkins, Calhoun Expires 26 August 2002 [Page 14]

Internet Draft AAA Keys for Mobile IP 26 February 2002

The number 7, assigned to the MN-FA Key Request From AAA Subtype extension, was taken from the numbering space defined for the Generalized MN-FA Key Request Extension (see section 6.1)

The number 1, assigned to the Unsolicited MN-FA Key Material From AAA Subtype extension, was taken from the numbering space defined for the Generalized MN-FA Key Reply Extension (see section 6.2)

The number 7, assigned to the MN-HA Key Request From AAA Subtype extension, was taken from the numbering space defined for the Generalized MN-HA Key Request Extension (see section 6.3)

The Code values specified for errors, listed in section 12, MUST NOT conflict with any other code values listed in RFC 3220, RFC 3024 [9], or RFC 2956 [10]. They are to be taken from the space of error values conventionally associated with rejection by the foreign agent (i.e., 64-127)

Section 4 introduces the Algorithm Identifier namespace that requires IANA management. This specification makes use of 1-3; all other values other than zero (0) are available for assignment, pending review and approval by a Designated Expert [11]

Section 4 introduces the Replay Method Identifier namespace that requires IANA management. This specification makes use of 1-3; all other values other than zero (0) are available for assignment, pending review and approval by a Designated Expert [11].

14. Security Considerations

The extensions in this document are intended to provide the appropriate level of security for Mobile IP entities (mobile node, foreign agent, and home agent) to operate Mobile IP registration protocol. The security associations resulting from use of these extensions do not offer any higher level of security than what is already implicit in use of the security association between the mobile node and the AAA.

Since the extensions defined in this specification only carries Key Material, which is used to derive keys, it does not expose any data that could be used in an attack aimed at recovering the key shared between the mobile node and the AAA. The authors do not believe this specification introduces new security risks.

Perkins, Calhoun Expires 26 August 2002 [Page 15]

Internet Draft AAA Keys for Mobile IP 26 February 2002

15. Acknowledgements

Thanks to Fredrik Johansson and the members of the IESG for their useful comments on this document.

Perkins, Calhoun Expires 26 August 2002 [Page 16]

Internet Draft AAA Keys for Mobile IP 26 February 2002

References

- [1] B. Aboba and M. Beadles. The Network Access Identifier. Request for Comments (Proposed Standard) 2486, Internet Engineering Task Force, January 1999.
- [2] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. Request for Comments (Best Current Practice) 2119, Internet Engineering Task Force, March 1997.
- [3] P. Calhoun and C. E. Perkins. Mobile IP Foreign Agent Challenge/Response Extension. Request for Comments (Proposed Standard) 3012, Internet Engineering Task Force, December 2000.
- [4] P. Calhoun, A. Rubens, H. Akhtar, and E. Guttman. DIAMETER Base Protocol (work in progress) Internet Draft, Internet Engineering Task Force. draft-ietf-aaa-diameter-07.txt, July 2001.
- [5] D. Eastlake, 3rd, S. Crocker, and J. Schiller. Randomness Recommendations for Security. Request for Comments (Informational) 1750, Internet Engineering Task Force, December 1994.
- [6] S. Glass, T. Hiller, S. Jacobs, and C. Perkins. Mobile IP Authentication, Authorization, and Accounting Requirements. Request for Comments (Proposed Standard) 2977, Internet Engineering Task Force, October 2000.
- [7] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. Request for Comments (Informational) 2104, Internet Engineering Task Force, February 1997.
- [8] D. Kristol and L. Montulli. HTTP State Management Mechanism. Request for Comments (Proposed Standard) 2109, Internet Engineering Task Force, February 1997.
- [9] Editor G. Montenegro. Reverse Tunneling for Mobile IP, revised. Request for Comments (Proposed Standard) 3024, Internet Engineering Task Force, January 2001.
- [10] G. Montenegro and V. Gupta. Sun's SKIP Firewall Traversal for Mobile IP. Request for Comments (Informational) 2356, Internet Engineering Task Force, June 1998.
- [11] T. Narten and H. Alvestrand. Guidelines for Writing an IANA Considerations Section in RFCs. Request for Comments (Best

Perkins, Calhoun Expires 26 August 2002 [Page 17]

Internet Draft AAA Keys for Mobile IP 26 February 2002

Current Practice) 2434, Internet Engineering Task Force, October 1998.

- [12] C. Perkins. IP Mobility Support. Request for Comments (Proposed Standard) 3220, Internet Engineering Task Force, December 2001.
- [13] C. Rigney, A. Rubens, W. Simpson, and S. Willens. Remote Authentication Dial In User Service (RADIUS) Request for Comments (Proposed Standard) 2865, Internet Engineering Task Force, June 2000.

A. Changes Since Previous Revision

In this revision of the document, there have been several major changes as a result of suggestions received during Last Call.

- Generalized Key Extensions previously specified in another document have been instead specified in this document in order that this document can be self-contained and not dependent on the standardization status of the other document.
- Additional explanation has been included for the purposes of clarifying the problem space and solution approach.
- An appendix has been added to describe the expected AAA infrastructure that will produce the keys that are to be distributed within the extensions specified in this document.
- Ladder diagrams have been included to illustrate the expected message flows containing the extensions defined in this document.
- HMAC-MD5 has been mandated for implementation by the mobile node, for compatibility with RFC 3220 [12]. The example text has been modified accordingly (see section 5)
- A table of Algorithm Identifiers has been identified as the numbering space for algorithm selection when establishing the security association using the keys distributed with the extensions in this document. See section 4.
- A terminology section has been added.
- This appendix has been added.

Perkins, Calhoun Expires 26 August 2002 [Page 18]

Internet Draft AAA Keys for Mobile IP 26 February 2002

B. AAA Infrastructure

In this appendix, we attempt to capture the main features of a basic model for operation of AAA servers that is assumed for understanding of the use of the Mobile IP registration extensions described in this

document. This information has been adapted from the discussion in RFC 2977 [6]

Within the Internet, a mobile node belonging to one administrative domain (called the home domain) often needs to use resources provided by another administrative domain (called the foreign domain). An foreign agent that handles the mobile node's Registration Request is likely to require that the mobile node provide some credentials that can be authenticated before access to the resources is permitted. These credentials may be provided as part of the Mobile-AAA Authentication extension [3], relying on the existence of an AAA infrastructure such as is described in this section, and also described in RFC 2977 and RFC 3012 [3]. Such credentials are typically managed by entities within the mobile node's home domain. They may be also used for setting up secure communications with the mobile node.

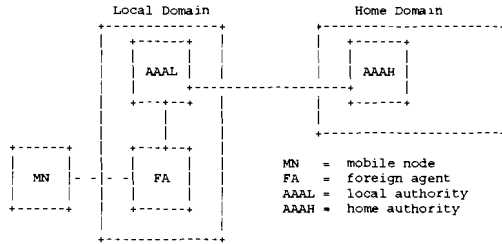


Figure 7: AAA Servers in Home and Local Domains

The foreign agent often does not have direct access to the data needed to verify the credentials. Instead, the foreign agent is expected to consult an authority (typically in the same foreign domain) in order to request proof that the mobile node has acceptable credentials. Since the foreign agent and the local authority (AAAL)

Perkins, Calhoun Expires 26 August 2002 [Page 19]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

are part of the same administrative domain, they are expected to have established, or be able to establish for the necessary lifetime, a secure channel for the purposes of exchanging sensitive (access) information, and keeping it private from (at least) the visiting mobile node.

The local authority (AAAL) itself may not have enough information stored locally to carry out the verification for the credentials of the mobile node. In contrast to the foreign agent, however, the AAAL is expected to be configured with enough information to negotiate the verification of mobile node credentials with its home domain. The home and foreign domains should be configured with sufficient security relationships and access controls so that they can negotiate the authorization, and also enable the mobile node to acquire security associations with the foreign domain. requested resources. For the purposes of the key exchanges specified within this document, the authorization is expected to depend only upon secure authentication of the mobile node's credentials.

Once the authorization has been obtained by the local authority, and the authority has notified the foreign agent about the successful negotiation, the foreign agent can deliver the Registration Reply to the mobile node along with the desired key material.

In figure 7, there might be many mobile nodes from many different Home Domains. Each Home Domain provides a AAAH that can check credentials originating from mobile nodes administered by that Home Domain. There is a security model implicit in figure 7, and it is crucial to identify the specific security associations assumed in the security model. These security associations are illustrated in figure 8, and are considered to be relatively long-lived security associations.

First, it is natural to assume that the mobile node has a security association with the AAAH, since that is roughly what it means for the mobile node to belong to the home domain.

Second, from the model illustrated in figure 7 it is clear that AAAL and AAAH have to share a security association, because otherwise they could not rely on the authentication results, authorizations, nor even the accounting data which might be transacted between them. Requiring such bilateral security relationships is, however, in the end not scalable; the AAA framework MUST provide for more scalable mechanisms, but the methods by which such a broker model is to be created are out of scope for this document. See RFC 2977 for more details.

Finally, from figure 7, it is clear that the foreign agent can naturally share a security association with the AAAL. This is

Perkins, Calhoun Expires 26 August 2002 [Page 20]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

necessary in order for the model to work because the foreign agent has to have a way to find out that it is permissible to allocate the local resources to the mobile node, and further to transmit any successful Registration Reply to the mobile node.

Figure 8 illustrates the natural security associations we understand from our proposed model. Note that there may be, by mutual agreement between AAAL and AAAH, a third party inserted between AAAL and AAAH to help them arbitrate secure transactions in a more scalable fashion. The broker model which has been designed to enable such

third-party processing should not have any effect on the Mobile IP extensions specified in this document, and so no description is provided here; see RFC 2977 [6] for more details.

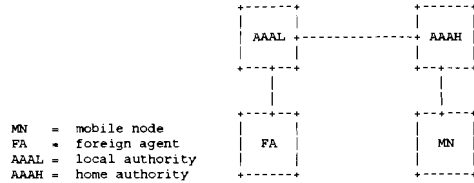


Figure 8: Security Associations

Nodes in two separate administrative domains (for instance, AAAH and AAAL) often must take additional steps to verify the identity of their communication partners, or alternatively to guarantee the privacy of the data making up the communication. While these considerations lead to important security requirements, as mentioned above in the context of security between servers, we consider the exact choice of security associations between the AAA servers to be beyond the scope of this document. The choices are unlikely to depend upon Mobile IP, or any specific features of the general model illustrated in figure 7. On the other hand, the security associations needed between Mobile IP entities are of central importance in the design of the key exchange extensions in this document.

One further detail deserves mention. The key associations to be established between the mobile node and the foreign agent have

Perkins, Calhoun Expires 26 August 2002 [Page 21]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

to be communicated to the foreign agent as well as to the mobile node. The way that the key is distributed to the foreign agent is not relevant to any material in this document, and is expected to be handled as part of the AAA protocol processing between the AAAH and AAAL, and the further AAA protocol processing between the AAAL and the foreign agent. Any method by which the key can be securely transmitted to the AAAL and then relayed (possibly with re-encryption) to the foreign agent, is expected to be outside the jurisdiction of any Mobile IP specification, and thus compatible (by reason of non-interference) with the protocol extensions specified in this document.

C. Message Flow for Requesting and Receiving Registration Keys

In this section, we show message flows for requesting and receiving a registration key from the AAA infrastructure, described in section B. Challenge values, as specified in [3], might be added to the Advertisement and Registration messages for additional replay protection, but are not illustrated here.

Diagram 9 illustrates the message flow for the case when the mobile node explicitly requests a registration key.

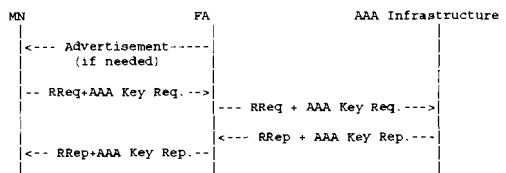


Figure 9: Message Flows for Requesting and Receiving Registration Keys

In diagram 9, the following message flow is illustrated:

1. The foreign agent disseminates an Agent Advertisement. This advertisement MAY have been produced after receiving an Agent Solicitation from the mobile node (not shown in the diagram).

Perkins, Calhoun Expires 26 August 2002 [Page 22]
 Internet Draft AAA Keys for Mobile IP 26 February 2002

2. The mobile node creates a Registration Request including the MN-HA Key Request and/or MN-FA Key Request, as needed, along with an authorization-enabling authentication extension as required by Mobile IP [12]
3. The foreign agent relays the Registration Request either to its locally configured AAA Infrastructure (see appendix B), according to local policy.
4. The foreign agent receives a Registration Reply with the appropriate indications for authorizing connectivity for the mobile node, which also includes the necessary AAA Key Reply extensions. Along with this Registration Reply, the foreign agent may also receive key material by some other secure method appropriate for communications between it and its local AAA

infrastructure.

- The foreign agent relays the Registration Reply to the mobile node, along with the new Key Reply extensions to be used by the mobile node to establish security associations with the relevant mobility agents (foreign agent and/or home agent)

Diagram 10 illustrates the message flow for the case when the mobile node receives an unsolicited registration key from the AAA Infrastructure.

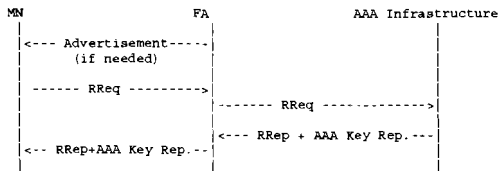


Figure 10: Message Flow for Receiving Unsolicited Registration Keys

In diagram 10, the following message flow is illustrated:

- The foreign agent disseminates an Agent Advertisement. This advertisement MAY have been produced after receiving an Agent Solicitation from the mobile node (not shown in the diagram)

Perkins, Calhoun Expires 26 August 2002 [Page 23]

Internet Draft AAA Keys for Mobile IP 26 February 2002

- The mobile node creates a Registration Request including an authorization-enabling authentication extension as required by Mobile IP [12].
- The foreign agent relays the Registration Request either to its locally configured AAA Infrastructure (see appendix B), according to local policy.
- The foreign agent receives a Registration Reply with the appropriate indications for authorizing connectivity for the mobile node, which also includes the necessary AAA Key Reply extensions. Along with this Registration Reply, the foreign agent may also receive key material by some other secure method appropriate for communications between it and its local AAA infrastructure.
- The foreign agent relays the Registration Reply to the mobile node, along with the new Key Reply extensions to be used by the mobile node to establish security associations with the relevant mobility agents (foreign agent and/or home agent)

Addresses

The working group can be contacted via the current chairs:

Baavaraj Patil Nokia 6000 Connection Dr. Irving, TX. 75039	Phil Roberts Megisto Corp. Suite 120 20251 Century Blvd Germantown MD 20874
---	---

USA	USA
Phone: +1 972-894-6709	Phone: +1 847-202-9314
Email: Baavaraj.Patil@nokia.com	Email: PROberts@MBGISTO.com

Questions about this memo can also be directed to the authors:

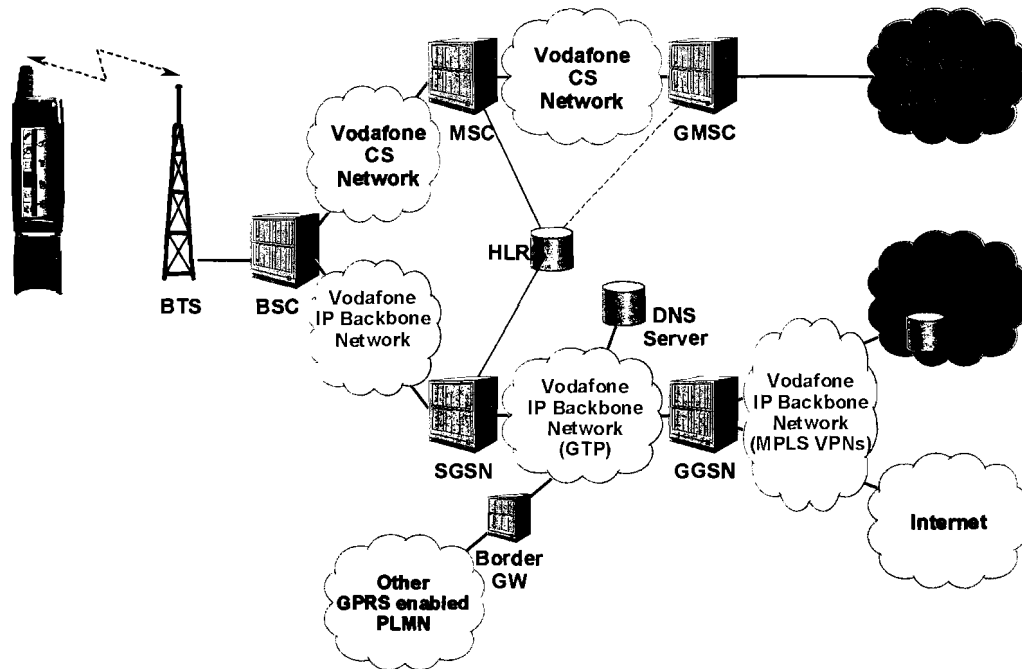
Perkins, Calhoun Expires 26 August 2002 [Page 24]

Internet Draft AAA Keys for Mobile IP 26 February 2002

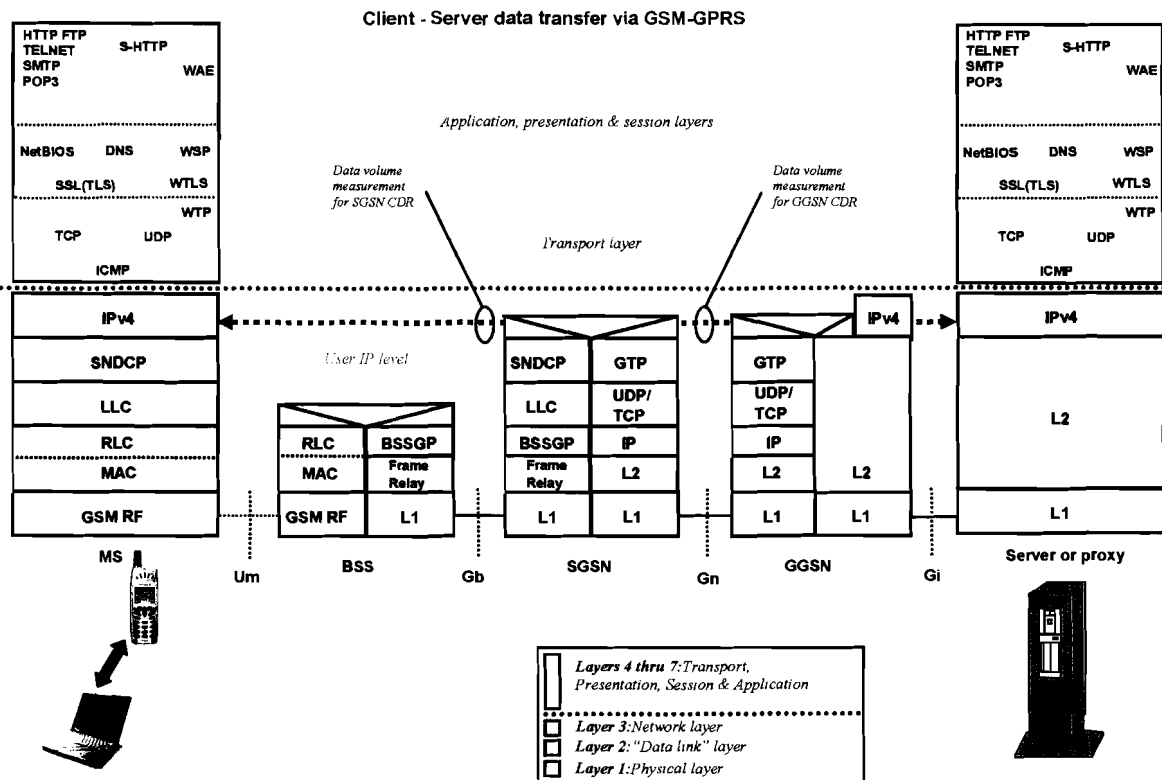
Charles E. Perkins Communications Systems Lab Nokia Research Center 313 Fairchild Drive 200 Mountain View, California 94043 USA Phone: +1-650 625-2986 EMail: charliep@prg.nokia.com Fax: +1 650 625-2502	Pat R. Calhoun Black Storm Networks 250 Cambridge Avenue, Suite Palo Alto, California, 94306 USA Phone: +1 650-617-2932 Email: pcalhoun@iameter.org Fax: +1 650-786-6445
--	---

Perkins, Calhoun Expires 26 August 2002 [Page 25]

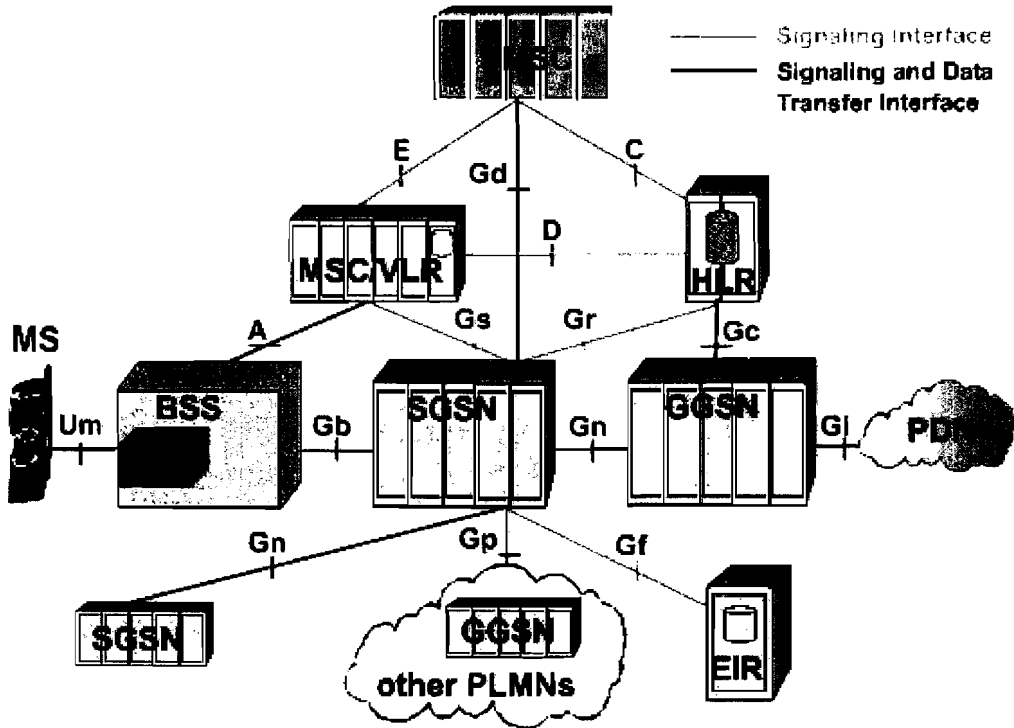
Appendix H GPRS Overview



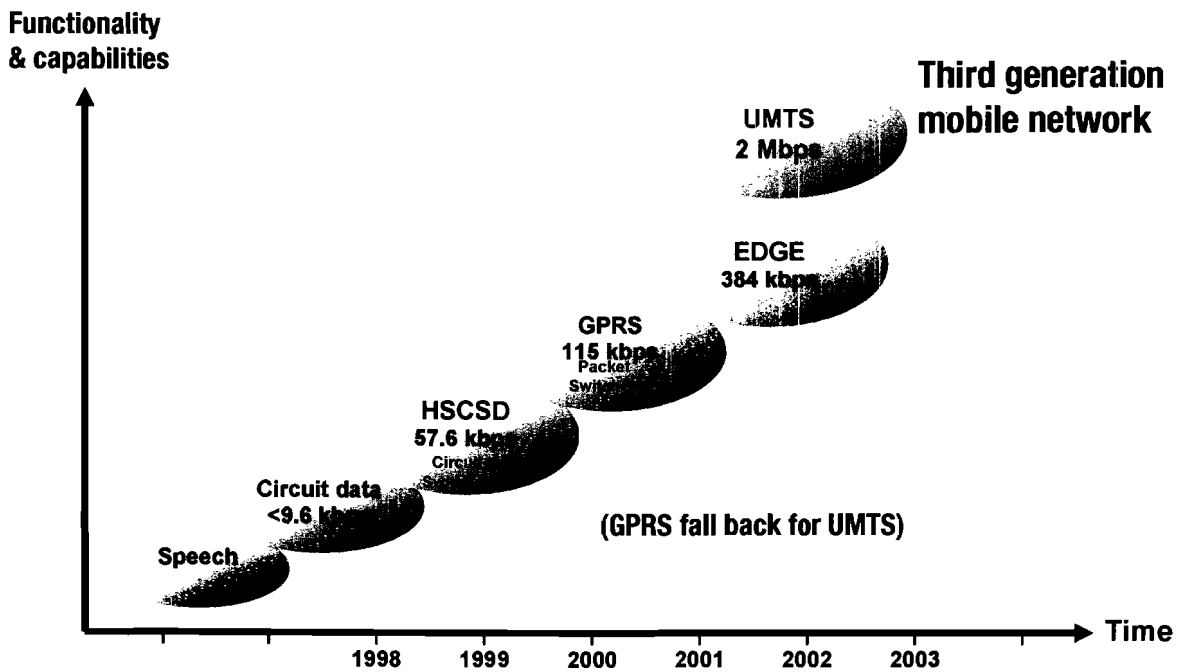
GSM/GPRS network entities overview



GPRS protocol stacks overview



GSM/GPRS network interfaces overview



Global view of technologies and data rates

Appendix I IP Packet Sizes

It is very difficult to determine THE average IP packet size, as this is dependent of the payload. Sniffing many packets at a representative point on the Internet and analyzing the results will give an impression. The following text and images were copied from the site http://www.caida.org/analysis/AIX/plen_hist/

Packet Length Distributions

The packet length distribution seen at NASA Ames Internet Exchange (AIX) is fairly constant, with no substantial trend observed between May 1999 and February 2000.

The following graphs show the distribution of IP packet sizes seen at the NASA Ames Internet Exchange (AIX). These packet length distributions are generated from multiple short traces collected at various times of day over two approximately one-week periods. Because these distributions contain contributions from the different workloads carried by the network at different times of day, they should represent more of an 'average' picture of the packet size distribution than any individual trace. However, no attempt has been made to normalize the contributions of individual traces. The distributions presented here are simply those of the concatenated traces.

Additionally, these distributions have been plotted for packet sizes less than 1600 bytes only. This allows the structure of the distributions to be presented in greater detail, but ignores the very small fraction of large packets that appear in these traces (typically less than 0.005% of packets).

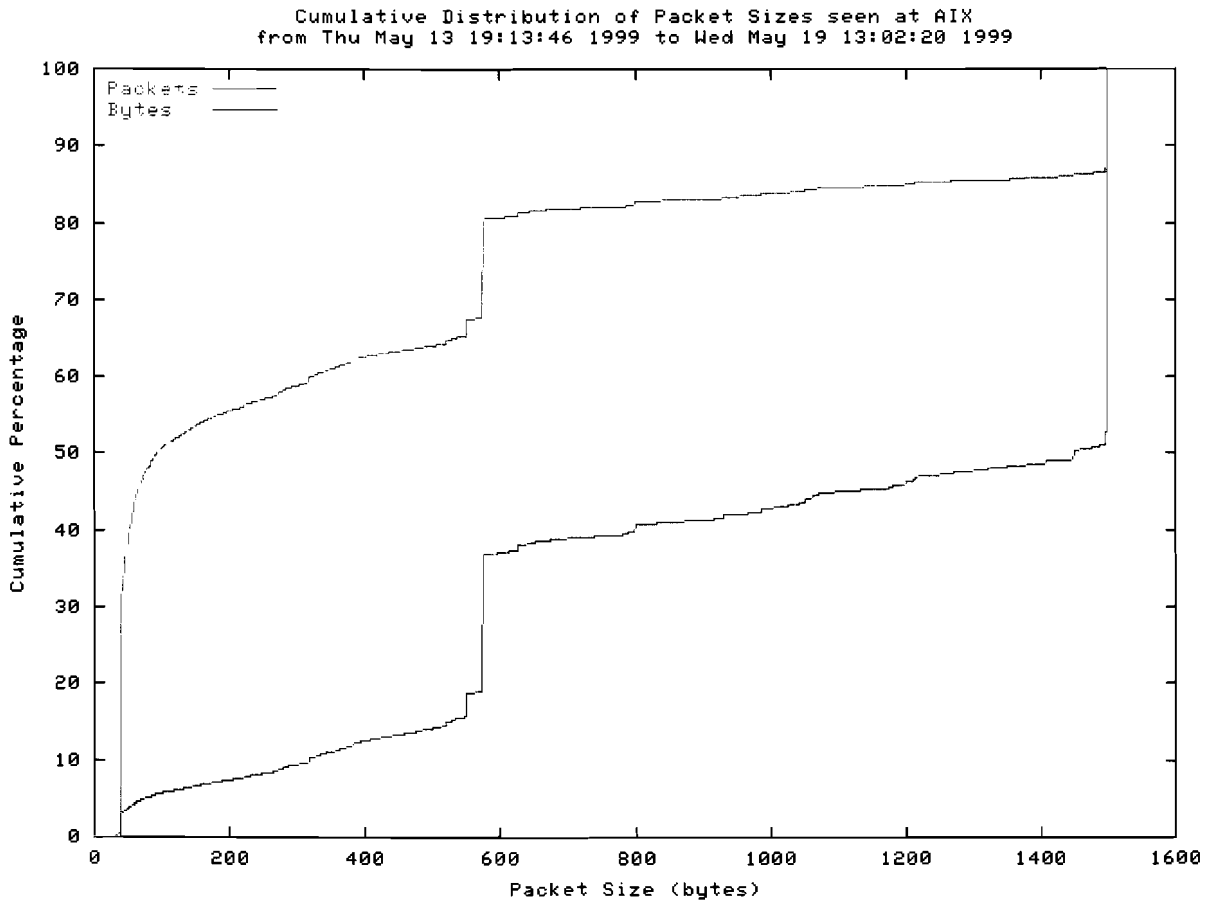


Fig 1: IP packet length distribution from 39 trace files captured between Thursday, May 13th 1999 at 19:14:36 PDT and Wednesday, May 19th 1999 at 13:02:20 PDT.

Statistics for the underlying packet length distribution:

Mean: 413 bytes, Standard Deviation: 509 bytes

Median: 93 bytes, Percentiles: 5th 40 bytes, 25th 40 bytes, 75th 576 bytes, 95th 1500 bytes

Number of Observations: 127 million packets [127710031 packets]

These numbers correspond to the red curve in the figure above.

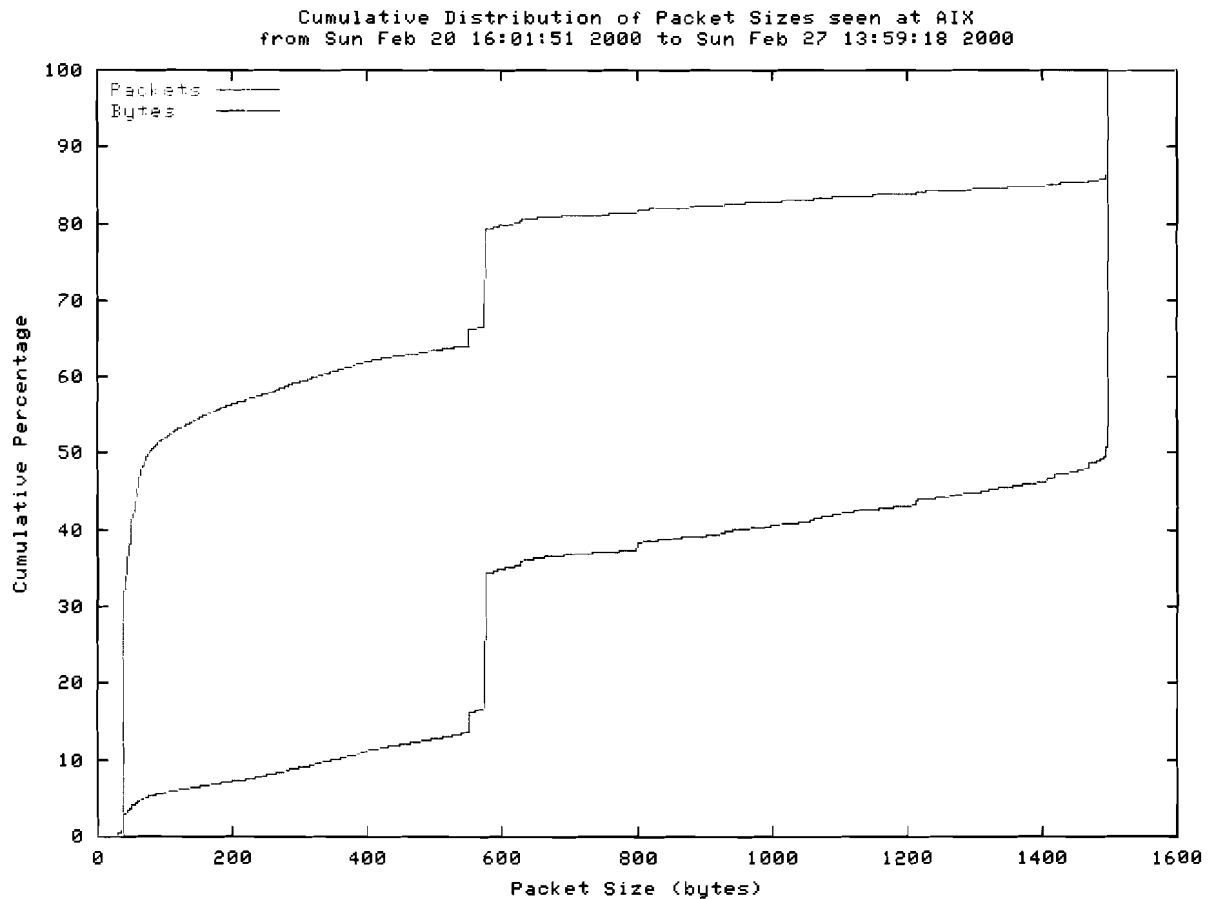


Fig 2: IP packet length distribution from 43 trace files captured between Sunday, February 20th 2000 at 16:01:51 PST and Sunday, February 27th 2000 at 13:59:18 PST.

Statistics for the underlying packet length distribution:

Mean: 420 bytes, Standard Deviation: 521 bytes

Median: 78 bytes, Percentiles: 5th 40 bytes, 25th 40 bytes, 75th 576 bytes, 95th 1500 bytes

Number of Observations: 84 million packets [84415871]

These numbers correspond to the red curve in the figure above.

The primary features of this distribution are all due to the way common TCP implementations divide a data stream into packets. Approximately 85% of the traffic in these traces is TCP, and a large proportion of this TCP traffic is generated by bulk transfer applications such as HTTP and FTP. Consequently, the majority of the packets seen are one of three sizes: 40 byte packets (the minimum packet size for TCP) which carry TCP acknowledgements but no payload, 1500 byte packets (the maximum ethernet payload size) from TCP implementations that use path MTU discovery, and 552 byte and 576 byte packets from TCP implementations that don't use path MTU discovery.

These two distributions are strikingly similar despite the fact that the second is based on data collected more than 9 months after the first. The second distribution has a slightly larger contribution from packets smaller than 100 bytes, but the difference is quite small. The following graph shows how the mean and median values vary over the entire duration of our study.

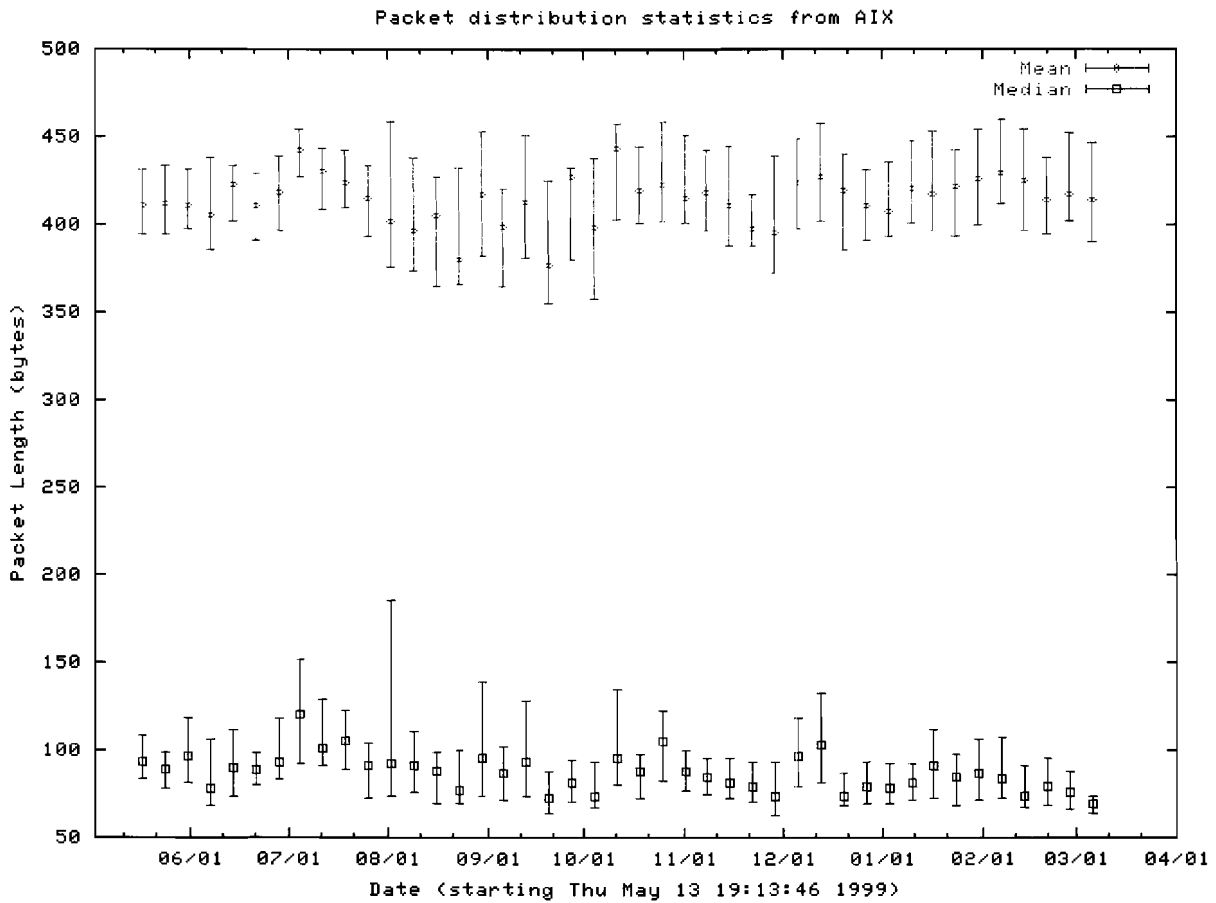


Fig 3: Mean and median packet length values for each packet trace collected between Thursday, May 13th 1999 at 19:14:36 PDT and Sat Mar 11 13:57:38 PST 2000. Values have been binned by week, and the median for each bin is plotted with first and third quartile error bars.

Appendix J Assignment

This appendix gives the original and the extended assignments of the graduation project.

Original Assignment

Page 1/2:

Assignment

“Integrating WLAN Solutions into a Macro Cellular Network”

Student: Tom van Sebille
Eindhoven University of Technology
Suggested period: 1 Dec. 2001 – 31 Aug. 2002

Ing. A.B.P. Jongen (Alfons) & R. Crutzen (Rob)
Libertel-Vodafone
Department: Technology Development
Customer Data Solutions
Gelissendomein 5
6229 GK Maastricht
Telephone: +31 (0)43 3557670 / +31 (0)6 54670670
& +31 (0)43 3557119 / +31 (0)6 54670739
E-mail: Alfons.Jongen@Libertel.nl
& Rob.Crutzen@Libertel.nl
Date: 19 Nov. 2001

Introduction

Libertel-Vodafone¹ in Maastricht is part of the global Vodafone Group Plc. It provides (mobile) telecommunication services. It has a GSM/GPRS network, which completely covers The Netherlands. It does not only offer voice services, but also data services. For the GPRS network, Libertel-Vodafone has built an IP backbone network. The department Technology Development (TD) investigates innovative services for the near future on a customer-focussed base. The section Consumer Data Solutions (CDS) concentrates on new and extended data services for consumers and corporates.

Background

The ownership of the IP backbone, as well as the developments in the field of wireless communication, allows Libertel-Vodafone to offer a new kind of radio access technology, Wireless LAN (WLAN). Libertel-Vodafone would like to investigate the technical aspects of offering this broadband radio access at certain areas, the so-called hotspots (airports, railway stations, conference rooms), at first. The idea is that their GPRS/UMTS users can have access to the Internet via the WLAN in those hotspots. Libertel-Vodafone could then function as the access provider for the users that have a GPRS/UMTS account and therefore stays in control of the customer relation.

Problem statement

The main problems are

1. the authorization of WLAN users, based on their GPRS account,
2. the seamless IP handovers between a GPRS connection and a WLAN connection and vice versa.

Assignment

The objective of the assignment will be the investigation of how these problems can be solved and implemented in a demo. The first problem requires extra attention, because it needs to be demonstrated in the live-and-operational Libertel-Vodafone network. This also involves consulting and gaining commitment from other Libertel-Vodafone departments. For the second problem, a non-live demonstration would suffice, which might be extended to the live network at a later stage.

¹ From 01 Jan. 2002 the company name will be Vodafone.

Page 2/2:

Time planning

- 1) Literature study and consulting of Libertel-Vodafone departments involved about (2 months)
 - a) the GSM network,
 - b) GPRS / UMTS,
 - c) WLAN,
 - d) and the IP backbone
 - e) radius-server (also expertise available at TU/e that can be used).
- 2) Make a concept how to handle the problem (2 months)
- 3) Implement it in a demo (3 months)
- 4) Writing the report (2 months)

Some points of special interest are:

- Authentication (AuC-HLR interfaces / functionality / configuration),
- access management (Authorisation),
- Accounting (billing),
- Security,
- WLAN Radio Planning in hotspot areas,
- dimensioning of required transmission capacity,
- Profile of the WLAN users (what data rates en quality do you want to support with WLAN network)

An extra feature would be adding the Bluetooth technology as an extra way to access the Internet.

Extended Assignment

Page 1/1:

Extended Assignment Tom van Sebille

Meeting notes of 25-03-2002

During the meeting at the 25th of March 2002, attended by Jacco Kwaaitaal from TU/e and Al Jøngen, Rob Crutzen and Tom van Sebille from Vodafone NL, dep. Customer Data Solutions (CDS), the following points were discussed:

- 1) The work Tom has done up to now;
- 2) The status of the GPRS-WLAN interworking pilot at Vodafone NL and the role of CDS in it;
- 3) It is decided to extend the assignment of Tom van Sebille.
Depending on the decision of the Vodafone NL Management Board Tom will:
 - A) Assist in the GPRS-WLAN interworking pilot with Ericsson as project coordinator. Points of special interest will be:
 - i) 3rd party roaming (RADIUS proxying; user profiles);
 - ii) security aspects ((mutual) authentication; encryption);
 - iii) handover solution, see;
 - iv) business model;
 - B) Retrieve knowledge from Vodafone Global (U.K.) which has been gained in other GPRS-WLAN interworking pilots (D2Vodafone-Cisco, OmnitelVodafone-Nokia). points of special interest will be:
 - i) security aspects ((mutual) authentication; encryption);
 - ii) handover solution, see;

Either way, Tom will investigate, develop, and implement a solution for seamless handovers between WLAN and GPRS. The solution will be found using Mobile IP. It will be a small-scale solution, not affecting the Vodafone NL network, since Vodafone Global has stated that this doesn't have a high priority.