ARCADA

Mobile Data Communication based on Host Identity Protocol (HIP)

Jonny Mattsson

Degree Thesis

Information Technology

2010

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informationsteknik
Identifikationsnummer:	2968
Författare:	Jonny Mattsson
Arbetets namn:	Mobil datakommunikation baserad på Host Identity Protocol (HIP)
Handledare (Arcada):	Göran Pulkkis
Uppdragsgivare:	Arcada – Nylands svenska yrkeshögskola

Sammandrag:

Traditionell TCP/IP datakommunikation erbjuder ingen mobilitet eftersom IP-adressen används till både identifiering och lokalisering av noder. Flera protokoll har utvecklats eller är under utveckling för att göra datakommunikation mobil. Detta examensarbete ger en översikt över de vanligaste mobilitetsprotokollen samt en detaljerad beskrivning av Host Identity Protocol (HIP). Den praktiska delen består av olika mobilitetstest med applikationer som görs mobila genom att de körs över HIP samt av färdigställande av en Flash-animation av HIP.

Host Identity Protocol är ett av IETF(Internet Enginering Task Force) utvecklat experimentellt protokoll, som erbjuder möjligheten till mobil datakommunikation genom identitetsbaserad adressering som bygger på den publika nyckelns kryptografi. Detta leder till att IP-adressen endast används till att lokalisera noder i nätet och således bibehålls en förbindelse mellan två datorer trots att IP-adressen byts.

Målsättningen med detta examensarbete var att testa hur väl mobiliteten fungerar tillsammans med applikationer som används dagligen. De flesta tester lyckades. Att alla inte lyckades beror främst på att HIP fortsättningsvis är i forskningstadiet och de realiseringar som finns är ännu rätt så instabila.

De viktigaste referenserna har varit IETF:s dokumentation samt några vetenskapliga publikationer.

Nyckelord:	HIP, Host Identity Protocol, Mobile IP, MOBIKE, Mobility, Multihoming, Public-key cryptography.
Sidantal:	52
Språk:	Engelska
Datum för godkännande:	22.11.2010

DEGREE THESIS	
Arcada	
Degree Programme:	Information Technology
Identification number:	2968
Author:	Jonny Mattsson
Title:	Mobile Data Communication based on Host Identity Protocol (HIP)
Supervisor (Arcada):	Göran Pulkkis
Commissioned by:	Arcada – University of Applied Sciences

Abstract:

Traditional TCP/IP data communication offers no mobility because the IP address is used for both identification and localization of network nodes. Several protocols have been developed or are being developed to make data communication mobile. This thesis provides an overview of the most common mobility protocols and a detailed description of the Host Identity Protocol (HIP). The practical part consists of various mobility tests with applications that are made mobile with HIP and the completion of a Flash animation of HIP.

HIP is an experimental protocol developed by IETF (Internet Engineering Task Force). HIP offers the possibility for mobile data communication by providing identity based addressing that is based on public key cryptography. This means that the IP address is only used to locate nodes in the network and therefore a connection between two HIP nodes is not interrupted even if one or both hosts get a new IP address.

The aim of this study was to test how well daily used applications work when they are made mobile with HIP. Most of the tests were successful. All tests did not succeed mainly due to the fact that HIP is still on the research stage and the implementations that exist are quite unstable. The main references have been IETF documentation and some scientific publications.

Keywords:	HIP, Host Identity Protocol, Mobile IP, MOBIKE, Mobility, Multihoming, Public-key cryptography.
Number of pages:	52
Language:	English
Date of acceptance:	22.11.2010

CONTENTS

Figures

Abbreviations

CoA	Care-of-Address
DCCP	Datagram Congestion Control Protocol
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DSA	Digital Signature Algorithm
ESP	Encapsulation Security Payload
FA	Foreign Agent
FQDN	Fully Qualified Domain Name
HA	Home Agent
HIP	Host Identity Protocol
IDE	Integrated Development Environment
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
LAN	Local Area Network
MIP	Mobile Internet Protocol
MN	Mobile Node
MOBIKE	Mobile Internet Key Exchange
mSCTP	Mobile Stream Control Transport Protocol
NAT	Network Address Translation
NEMO	Network Mobility
RR	Resource Record
RSA	Rivest-Shamir-Adleman algorithm
SA	Security Association
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network

Definitions

Diffie – Hellman Key Exchange

A cryptographic protocol that can produce a shared secret key between two hosts over an insecure public network.

HMAC

A HMAC is a message authentication code that in HIP is calculated over an entire signaling packet excluding the SIGNATURE parameter.

Host Identity (HI)

A Host Identity is represented by the public part of a public/private key pair, presently RSA or DSA. Must be unique.

Host Identity Tag (HIT)

A HIT is a 128 bit value calculated as a cryptographic hash from a Host Identity.

Internet Key Exchange (IKE)

A protocol used with IPSec that creates a Security Association (SA). Uses Diffie-Hellman key exchange and public-key techniques (RFC 2409). IKEv2 is an updated version of IKE (RFC 4306).

Internet Protocol Security (IPSec)

A protocol suite that uses authentication and encryption for securing IP communication. IPSec works between the network and transport layer and therefore applications do not need to be redesigned for IPSec to use it (RFC 4301).

IP tunnel

An IP based communication channel that is used to transport other protocols by encapsulating their packets.

Local Scope Identifier (LSI)

A 32 bit value that can be used as an IPv4 address.

Security Parameter Index (SPI)

A numeric value that is chosen by each host when setting up an ESP Security Association for combining the right data packet with the right SA.

1 INTRODUCTION

This BSc thesis contains an overview of mobility protocols with focus on Host Identity Protocol (HIP). A mobility protocol in this thesis is defined as a protocol that can make traditional TCP/IP communication mobile. The main goal of this thesis is to examine mobile preferences of HIP by doing practical tests on common scenarios such as chatting and video streaming. This thesis work started as a part of the WISEciti research project. WISEciti stands for Wireless Community Services for Mobile Citizens and is a research project between several research institutions and companies (WISEciti Project, 2010)

1.1 Background

Traditional TCP/IP communication doesn't offer the possibility for the user to become mobile. In a traditional network architecture the IP address is used both to identify a communicating host and to locate the host's network position. If a host is attached to a new IP address, all communication is interrupted. Mobility protocols such as Mobile IP and HIP are designed to solve that problem. By introducing a new namespace, Host Identity namespace, and an new layer in the TCP/IP stack HIP separates the locator and identifier from each other (Moskowitz & Nikander, 2006). Mobile IP solves the mobility problem by using two IP addresses.

1.2 Aims and goals

The main goal of this thesis is to describe different mobility protocols that provide mobile data communication. One of the protocols, HIP, is described in detail and practical tests to evaluate its mobility preferences are performed. All three HIP implementations which have been developed until the writing of this thesis are evaluated.
The aim of this report is to show how data communication can be mobile, especially data communication that is used by common people on a daily basis.

1.3 Structure of thesis

This thesis contains of two parts. The first part is a theoretical part that describes different mobile protocols with a detailed description of HIP. The second part is a practical part that contains all the mobility tests that are performed with HIP. This part shows how the tests are performed, which settings are used, and contains results and illustrations of the tests.

2 OVERVIEW OF MOBILITY PROTOCOLS

A mobility protocol can be defined as a protocol that gives hosts in a network the possibility of moving around and switching network location or completely moving to another network without any interruption for running applications. Mobile IP (MIP), Mobile Internet Key Exchange (MOBIKE), Host Identity Protocol (HIP), Network Mobility (NEMO), Mobile Stream Control Transport Protocol (mSCTP), Datagram Congestion Control Protocol (DCCP), and Session Initiation Protocol (SIP) are examples of this kind of protocols. NEMO is based on Mobile IPv6 and offers mobility features for networks. mSCTP is the mobile version of SCTP and can be used for mobility management at the transport layer. DCCP works at the transport layer. SIP is a signaling protocol for controlling multimedia sessions. SIP works at the session layer. This thesis gives a brief overview of MIP and MOBIKE and a detailed description of HIP.

2.1 Mobile IP

Mobile IP, MIP, is a standardized protocol developed by Internet Engineering Task Force (IETF). The main purpose of MIP is to offer a mobile device the possibility of moving from one network to another without changing its IP address (C. Perkins, 2002). MIP for IPv4 is defined in RFC 3344 and for IPv6 in RFC 3375. MIP is already in commercial use. For example Cisco and Birdstep offer mobile VPN solutions that are based on MIP.

A Mobile Node (MN) is able to have two IP addresses, a home address which is permanent and a care-of-address (CoA) which is used when a MN is visiting another network. A MIP network solution consists of two specific hardware routers, a Home Agent (HA)

and a Foreign Agent (FA). A HA stores information about CoAs of Mobile Nodes having their home addresses in the network the HA is located in. A FA stores information about visiting Mobile Nodes in the network it is located in. A FA also delivers CoAs.

In MIP a node that communicates with a MN always sends all packets to the permanent home address of the MN. When the mobile node moves to another network the following happens. The MN sends to the FA a registration request containing e.g. home address, Home Agent and care-of address. The FA relays the message to the HA. The HA sends a registration reply back to FA where it appears if the request was granted or denied. As the last step the FA informs the MN about the outcome of the request. The communicating host is still transmitting its packet to the MN:s permanent address but the HA redirects these packets by encapsulating them with a new IP header, the CoA of MN, and sends them through an IP tunnel (Fig. 1). When the MN is transmitting, the packets go directly to the IP address of the communicating host. The MN uses its home address as the source for the IP packets.

The main difference between MIPv4 and MIPv6 is that in MIPv6 there is no FA. In MIPv6 route optimization is fully integrated. A communicating host can communicate directly with the MN without sending data packets through the HA.

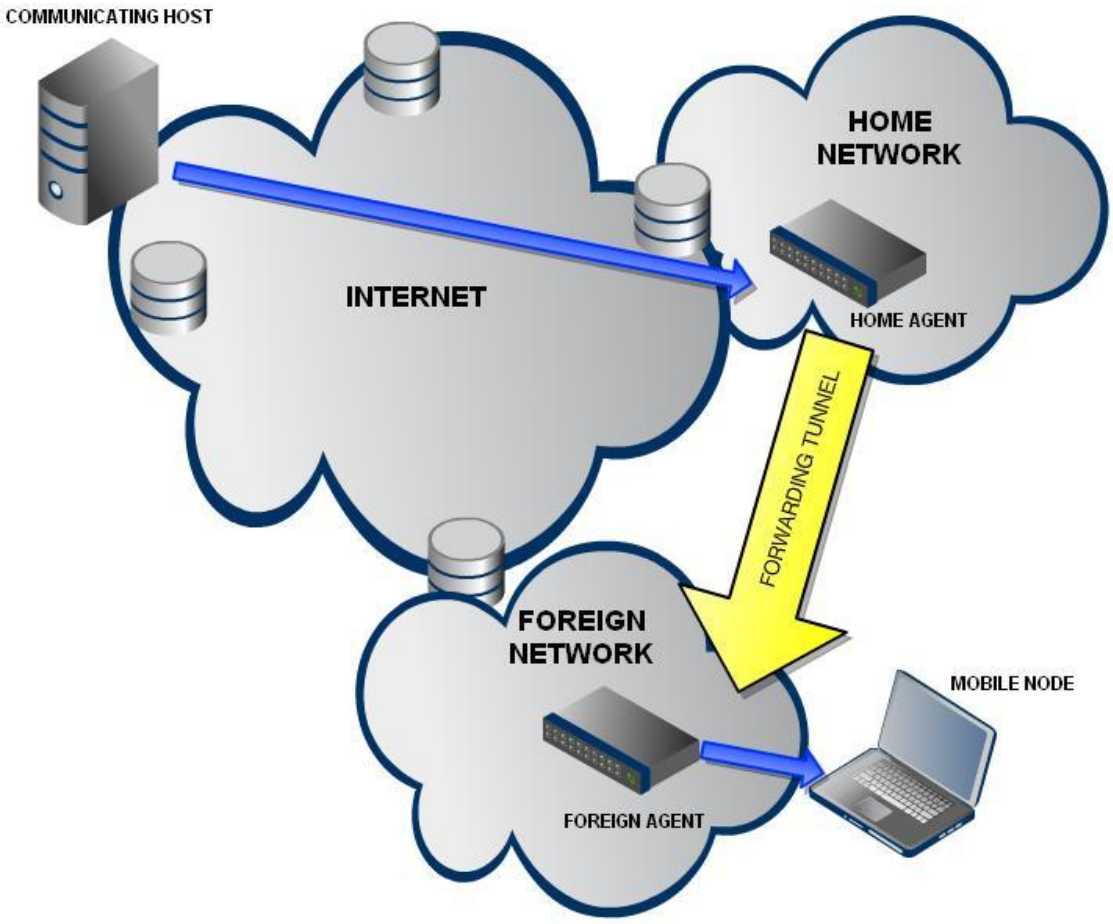

Figure 1. Packet routing from a communicating host to a Mobile Node that has moved to a foreign network.

2.2 MOBIKE

The IKEv2 mobility and multihoming protocol, MOBIKE, is an extension to IKEv2. MOBIKE enables the use of IKEv2 when a host has multiple addresses (multihoming) or when an IPSec host changes its IP address (mobility) (Kivinen, Tschofenig, 2006).

IKEv2 is designed to use the IP addresses of communicating peers to create the IKE security associations (SAs) and tunnel mode IPSec SAs. Therefore if a peer changes its point of attachment to the Internet, new IKE SAs and IPSec SAs need to be created. That is not recommended for several reasons, for example if authentication is based on user interaction such as entering a code from a token card (Eronen, 2006). Even without user interaction it is not recommended to often create new IKE SAs because the process involves expensive calculations.

MOBIKE solves this problem by working on the top of IKEv2. . The solution is based on a mechanism that updates the IP addresses of existing IKE SAs and IPSec SAs.

3 HIP IN DETAIL

3.1 Architecture

The TCP/IP stack (Fig. 2) uses the IP address as both locator and endpoint identifier. When a host gets a new IP address it will be recognized as a new host and therefore all communication with other hosts will be interrupted. HIP redesigns this stack by separating the locator and identifier from each other (Fig. 2) The IP address is still used as a locator but as identifier a public key of a public/private key pair is used. By using cryptographic keys HIP offers encryption and authentication by default.
When communicating over HIP the socket is bound to the HI instead of to an IP address. Therefore HIP offers mobility and multihoming to a very low infrastructure cost.

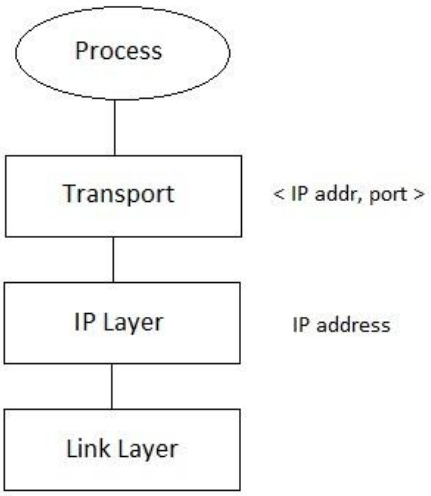

Figure 2. The TCP/IP stack.

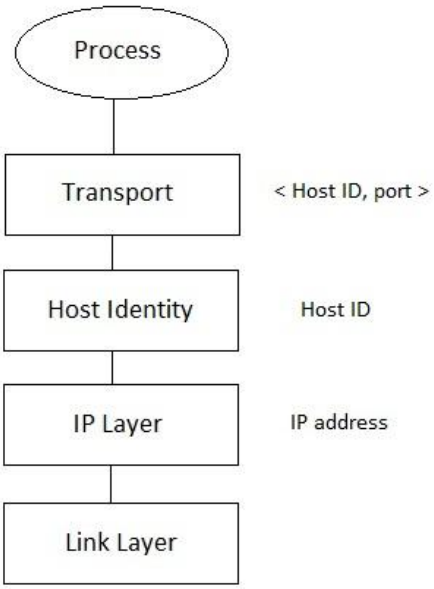

Figure 3. New HI layer in the TCP/IP stack.

3.2 Base Exchange

HIP Base Exchange (Fig. 4) is a four-way handshake between two hosts (Moskowitz & Nikander, 2008). The host that starts the Base Exchange is called initiator and the other host is called responder. The first packet, I1, is used to trigger the exchange and the three others, R1, I2, R2, are used to generate a session key based on Diffie-Hellman key exchange.

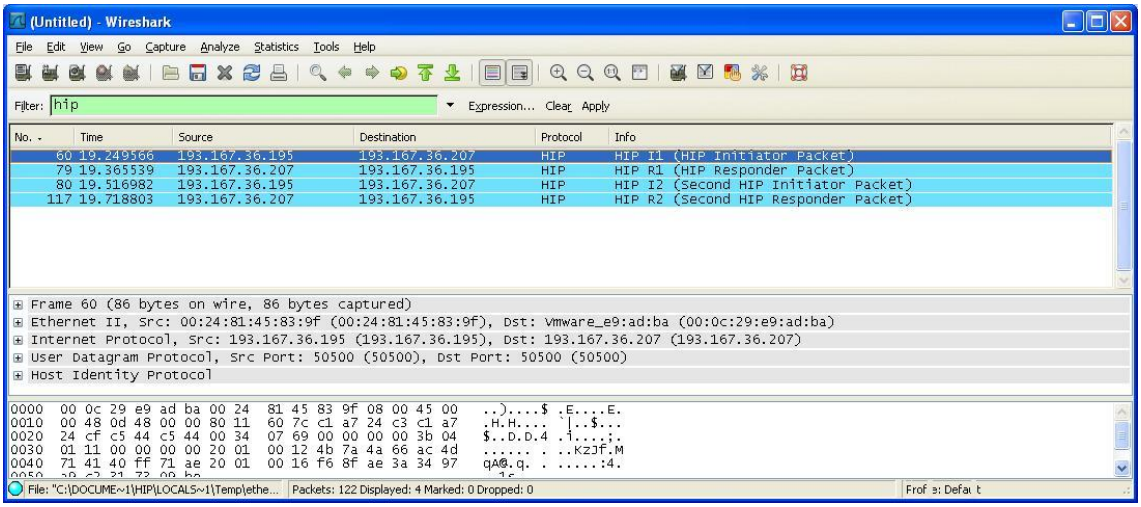

Figure 4. HIP Base Exchange captured with Wireshark.

3.3 HIP signaling

3.3.1 I1

I1 is the initiator packet that is sent from the host that tries to make a HIP connection to the other part. The I1 packet (Fig. 5) contains the initiator´s HIT and, if known, the responder´s HIT. If the responder´s HIT is unknown it has the value NULL and opportunistic mode is used. Opportunistic mode is not recommended to use as it is more vulnerable to Man-in-the-Middle attacks.

If a Rendezvous Server is used, the I1 packet is sent to the Rendezvous Server instead of to the responder. The Rendezvous Server looks up the IP address of the recipient and transmits the I1 packet to the responder.

After an I1 packet is sent, a timer is started at the initiator. If the R1 packet is not received before a chosen timeout the initiator retransmits the I1 packet and restarts the timer.

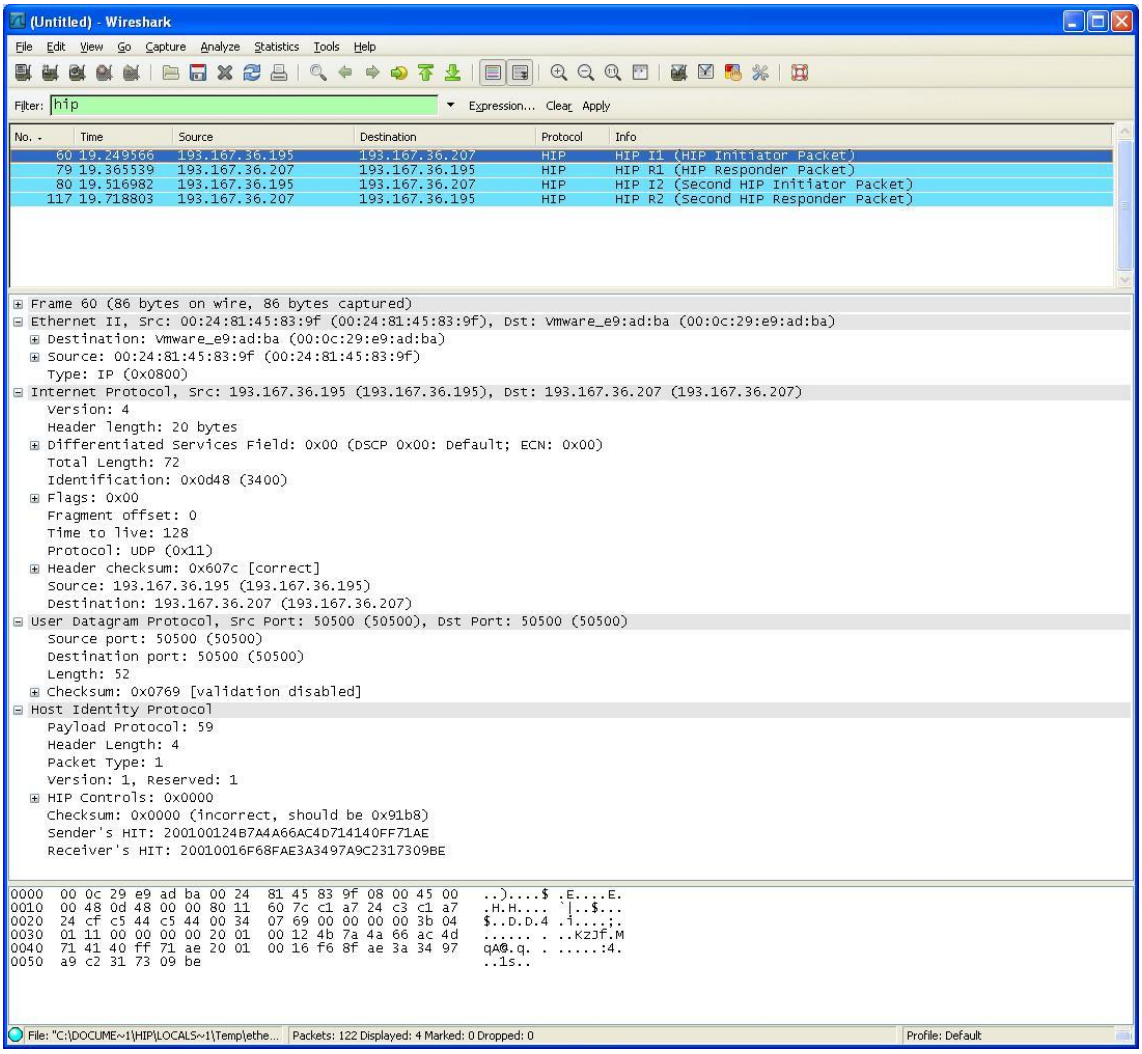

Figure 5. Data fields of an I1 packet captured with Wireshark.

3.3.2 R1

R1 is the packet that the responder sends, as a response to the I1 packet, to the initiator. A R1 packet (Fig 6) contains a Diffie-Hellman value, a cryptographic puzzle and the responder's public key. The responder uses its private key to sign the packet. The cryptographic puzzle contains a random number and a difficulty.

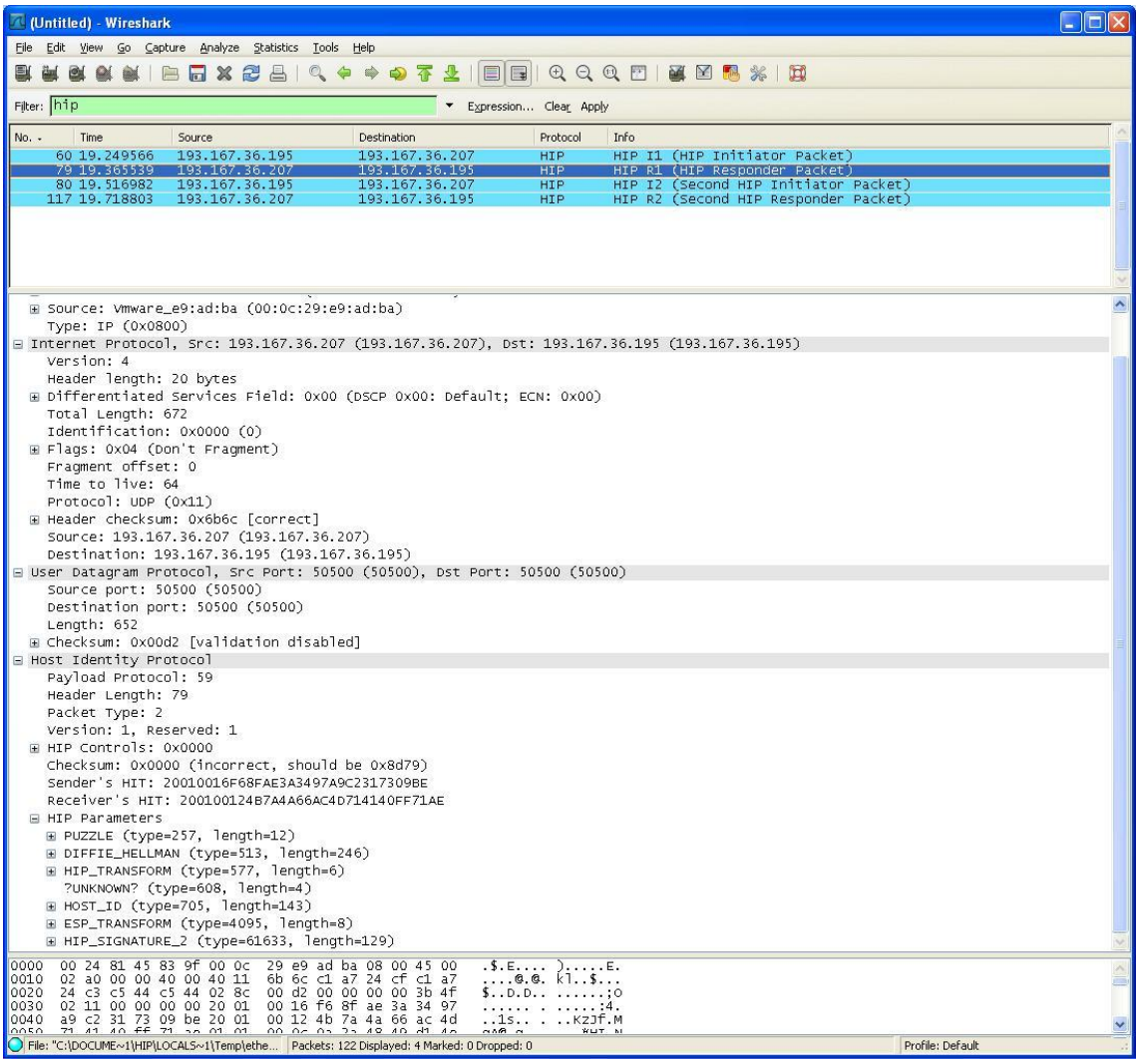

Figure 6. Data fields of a R1 packet captured with Wireshark.

3.3.3 I2

I2 is the second packet from the initiator and is a response to the R1 packet. An I2 packet (Fig. 7) contains a solution to the cryptographic puzzle and Diffie-Hellman values. A hashed message authentication code (HMAC) is included in the packet and is used as an additional protection against attacks. The packet is signed before the transmission.

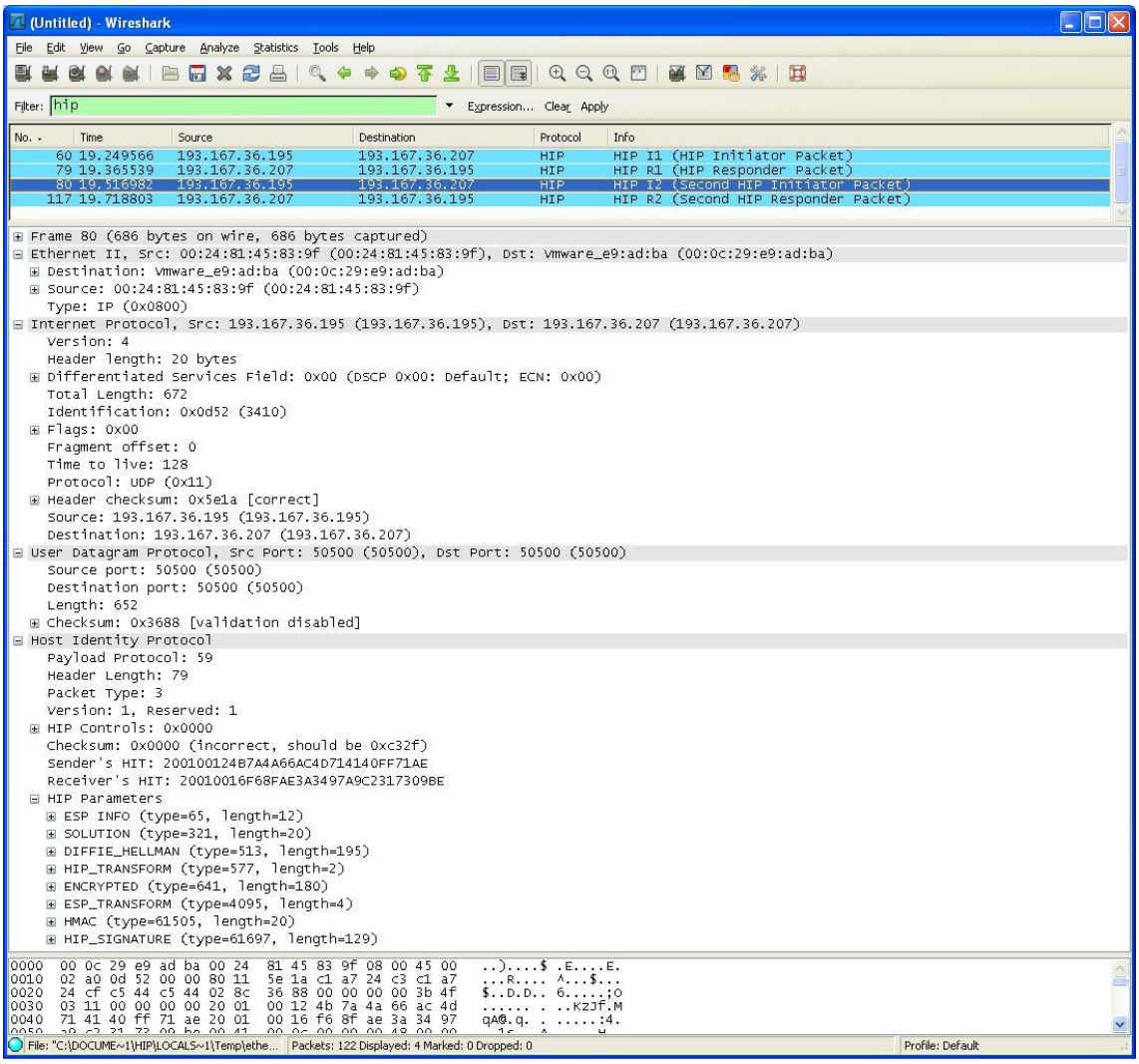

Figure 7. Data fields of an I2 packet captured with Wireshark.

3.3.4 R2

R2 is the second packet (Fig. 8) from the responder and is a response to the I2 packet. It contains a HMAC and a HIP_SIGNATURE. R2 is the fourth and the last packet in HIP Base Exchange. If the puzzle is solved and all parameters are correct, a HIP connection is established and data can flow between the two hosts.

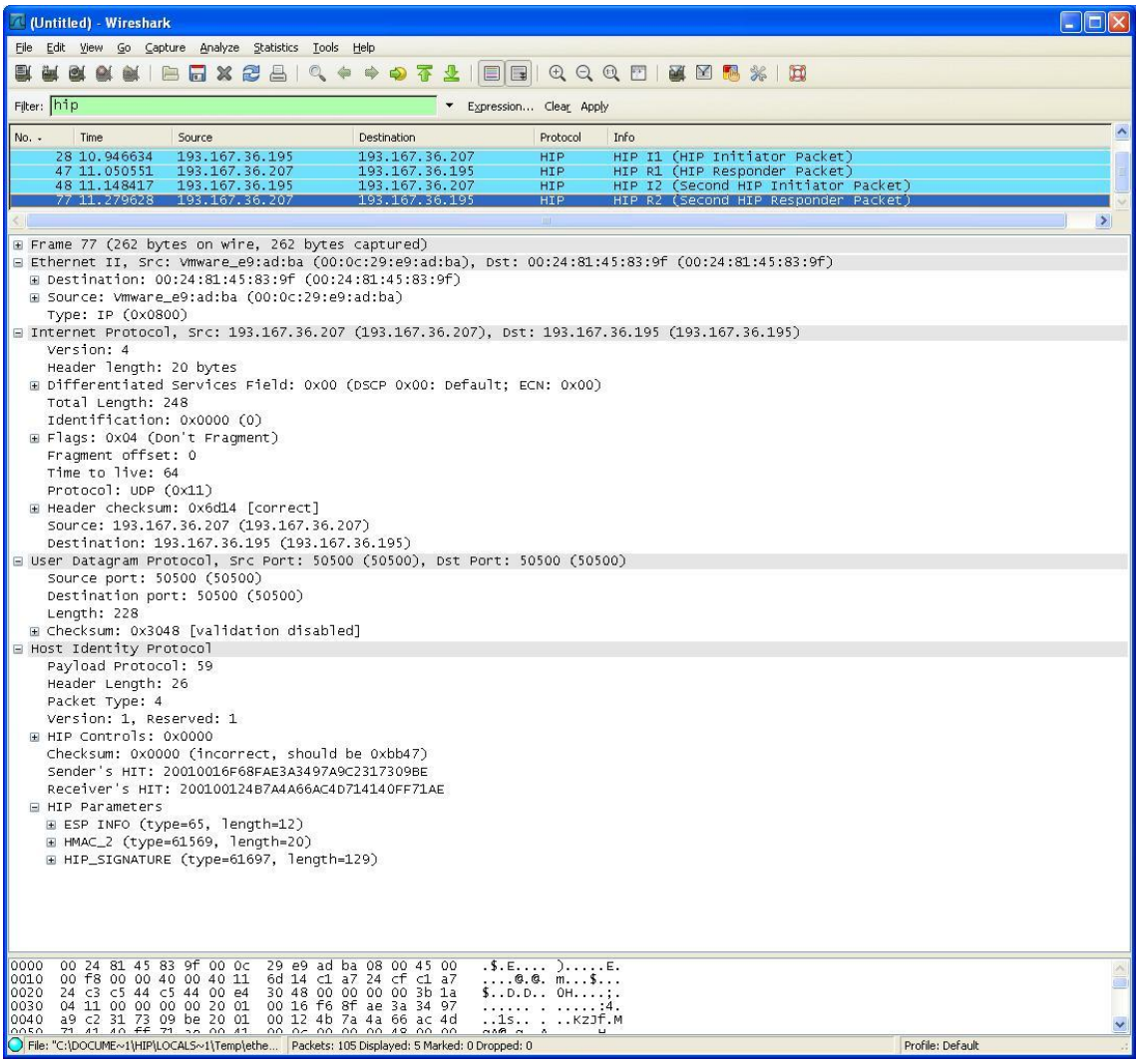

Figure 8. Data fields of a R2 packet captured with Wireshark.

3.4 Other HIP control packets

UPDATE

An UPDATE packet is used to send information about the HIP association to the other part. When a host changes its network location an UPDATE packet is sent to the other part containing the new IP address. If an UPDATE packet contains a SEQ parameter the responder needs to confirm the UPDATE with an ACK parameter.

NOTIFY

A NOTIFY packet is used to inform the other host about protocol errors and negotiation failure. A NOTIFY packet is a pure information packet.

CLOSE

A CLOSE packet is used to terminate an existing HIP association. A CLOSE packet contains a HMAC and a HIP_SIGNATURE.

CLOSE_ACK

A CLOSE_ACK packet is sent in response to a CLOSE packet and confirms the shut-down of a HIP association. A HIP_SIGNATURE is included for verifying its validity.

3.5 DNS extension

To continue using human-friendly domain names with HIP all DNS servers need to be modified. According to HIP DNS extension (Nikander & Laganier 2008) a HIP Resource Record (RR) will be stored in DNS. A HIP RR (Fig. 9) contains a HI, HIT and optionally one or several domain names of Rendezvous Servers. Because a HIT has the same length as an IPv6 address it can be stored in an AAAA record.

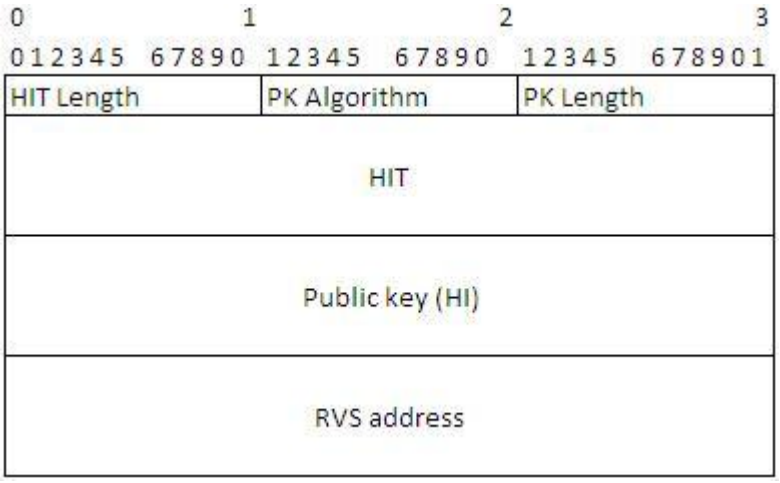

Figure 9. HIP data as it is stored in a HIP Resource Record in DNS.

3.6 ESP protection

For protection of user data HIP uses Encapsulated Security Payload (ESP) (Jokela, Moskowitch & Nikander, 2008). During Base Exchange a pair of Security Associations (SAs) is established, one SA in each direction, by modifying parameters in the HIP packets.

The responder begins the negotiations by adding a new ESP_TRANSFORM parameter to the R1-packet. An ESP_TRANSFORM message contains information of which ESP transform suites a host is prepared to use. Transform suites are limited to six and are stored in preferred order (Fig. 10)

The Initiator responds to the R1 packet by selecting one of the suggested transforms and adds that information in an ESP_TRANSFORM parameter in the I2 packet. An ESP_INFO containing an own chosen Security Parameter Index (SPI) value is also included in this packet. The responder responds to the I1 packet by including its own chosen SPI value in the R2 packet.

The SPI is used to locate right SA for received packets.

Suite ID	Value
RESERVED	0
AES-CBC with HMAC-SHA1	1
3DES-CBC with HMAC-SHA1	2
3DES-CBC with HMAC-MD5	3
BLOWFISH-CBC with HMAC-SHA1	4
NULL with HMAC-SHA1	5
NULL with HMAC-MD5	6

Figure 10. Supported ESP transform suites.

3.7 Rendezvous server

The Rendezvous Extension for HIP is used to reach a HIP mobile node that often changes its IP address (Laganier J, Eggert L 2008). Updating a host´s current IP address to DNS is very slow. Instead the FQDNs of a host´s Rendezvous Servers are stored in DNS in a HIP RR (chapter 3.3 DNS Extension).

A Rendezvous Server (RVS) is used in two ways, in client registration at client start up and in establishing a HIP connection with another node through itself. Every time when

a mobile node changes its IP address, the node sends that information to the RVS. If a HIP association already exists between the Mobile Node and the RVS the new IP address is delivered to the RVS with UPDATE packets. Otherwise a HIP Base Exchange with specific registration parameters is executed between the mobile node and the RVS. HIP Base Exchange through a RVS is basically a way to set up a HIP connection with a mobile node that has an IP address that is unknown to the initiator. The only difference from a Base Exchange between two nodes that know each other's IP addresses is that the first packet, I1, is sent to the RVS and the RVS sends it to the responder. After that the following packets, R1, I2 and R2, are transmitted directly between the nodes.

3.7.1 NAT traversal

A Rendezvous Server (RVS) is also used when a host is behind a NAT and HIP packets therefore are not forwarded to the HIP host. By establishing an outgoing connection from the host to the RVS the RVS can pass on related packets to the responder.

3.8 Mobility and Multihoming

3.8.1 Client mobility

The simplest mobility scenario is when two hosts are communicating with a single SPI pair between them. If one of the hosts changes its IP address, e.g. is moving to another subnet or is renewing its DHCP lease, the other hosts needs to be notified of this change so that it can update its HIT-IP mapping. This notification is made by a sequence of 3 UPDATE packets with different parameters (See 3.2.1). The first UPDATE packet contains ESP_INFO, LOCATOR and SEQ parameters. If no re-keying is needed the old SPI value and the new SPI value are the same as the existing SPI value in the ESP_INFO parameter. The LOCATOR parameter contains the new IP address and the SEQ parameter is just a simple sequence number that is incremented by one before each UPDATE packet is sent. The second UPDATE packet contains, besides ESP_INFO and SEQ, an ACK and an ECHO_REQUEST parameter. The ACK parameter is just a simple acknowledgment and the ECHO_REQUST contains a nonce which is used for per-

forming address verification. The third and last UPDATE packet contains only an ACK and an ECHO_RESPONSE parameter.

If re-keying is needed the first UPDATE packets contains a new SPI value and an index in the key material for generating a new ESP session key. The second UPDATE packet contains the sending host´s new SPI and the third UPDATE packet is the same that is used when not re-keying. Another way to re-key is to use the DIFFIE-HELLMAN parameter in the first and the second UPDATE packets for generating a new shared secret that will be used to generate the new keying material.

If a RVS is in use, the same UPDATE packet communication is realized between the mobile host and the RVS.

3.8.2 Simultaneous mobility

A scenario when two communicating hosts frequently and simultaneously change their point of attachment to the Internet would cause a problem if normal UPDATE packets were used. A solution to this problem has been presented and for an implementation based on OpenHIP. This solution has also been tested (Hobaya F, Gay v, Robert E 2009)). However it is not revealed whether the solution worked.
The proposed solution includes an extended use of a Rendezvous Server that updates the HIT-IP mappings on both mobile hosts.
A basic scenario with simultaneous mobility could be the following: two hosts that both have their own Rendezvous Server are communicating. Both hosts move to another location and get new IP addresses. Both hosts send an UPDATE packet to their own RVS and to the last known location of the other host. Since both hosts have a new IP address the packets are forwarded to wrong locations which results in a disconnection. According to the proposed solution the UPDATE packet is sent to the other hosts RVS instead of directly to the host. The RVS relays the packet to the right location. Because a client always informs its RVS of its new location after a mobility event, the RVS always knows the current location. Therefore simultaneous mobility is possible when the UPDATE packet is sent to the others host´s RVS instead of directly to the host. This solution requires modification to the software that is used by the clients and the RVS.

3.8.3 Multihoming

The Host Identity Protocol supports multihoming (Nikander P, Henderson T, 2008). A host is multihomed when it has multiple network interfaces or IP addresses. The benefit of multihoming is that it increases the reliability of network applications.
A simple HIP multihoming scenario is when a new IP address is added to a host, e.g. if a wireless host connects to LAN through a network cable while it is still connected to the WLAN. The data still flows through the WLAN interface, until a UPDATE packet is received with a LOCATOR parameter that has the IP address that is used with the LAN interface and a preferred value that indicates which interface is to be used. If the LAN interface is preferred, the other host updates its HIT-IP mapping. Otherwise the WLAN interface is used until the next UPDATE packet is sent.

3.8.4 Network mobility

Host Identity Protocol can also be used to make a network mobile (Melen J, Ylitalo J, Salmela P, Henderson T, 2009). A network in this context is a network with mobile nodes (MN) and mobile routers (MR). Such a network could be located on trains or buses where the entire network, including the Mobile Nodes and the Mobile Routers, can change its point of attachment to the Internet. Host Identity Protocol based Mobile Router (HIPMR) is a draft that describes the HIP extension that gives a network mobile features. According to the draft, the HIP nodes that are clients of a HIP Mobile Router give the MR permission to signal UPDATE packets on their behalf.
The described network has two different components, MN and MR. A scenario with a mobile network starts from the MN. The MN can find a MR by monitoring incoming beacons that all Mobile Routers are sending to inform all network nodes about their existence. After a suitable MR is found a Base Exchange is initiated by the MN. By using the HIP registration extension, the MN registers itself to the MR as a client of its routing service and delegates to the MR the authority to signal UPDATE packets on its behalf. A network mobility event happens when the MR changes its point of attachment to the Internet. If a new IP address is delivered to the MR, then packets from MNs to peers outside the network flow correctly but packets from other peers to the MNs are delivered to wrong locations. Because the MR has been authorized to signal UPDATE packets, the MR digitally signs UPDATE messages on behalf of MNs. This is possible be-

cause a MN shares a portion of its symmetric key space and therefore the MR is capable of computing HMACs of UPDATE messages. After a successful update event the communication can continue even if the entire network with the MR and all the Mobile Nodes have changed their point of attachment to Internet.

3.9 Implementations

Presently there are three major implementations of HIP, OpenHIP that is developed by the Boeing Company, InfraHIP that is developed by Helsinki Institue of Information Technology, and hip4inter that is developed by Oy LM Ericsson Ab.

3.9.1 OpenHIP

OpenHIP is an open source HIP implementation developed by Boeing Company. OpenHIP supports following operating systems: Linux, BSD, Mac OS X, and Windows XP, Vista and 7. OpenHIP uses an XML library for configurations files, supports SHA, MD5, HMAC hashing and 3DES, BLOWFISH, AES encryption (Boeing Company, 2010)

Using OpenHIP

After OpenHIP is installed an icon is placed in Windows task bar.

When right-clicking on the icon a menu appears. From the menu the user can reach the configuration files.

hip.conf (Fig. 11) is auto generated when installing and contains all HIP options.

```
<?xml version="1.0" encoding="UTF-8"?>
<hip_configuration>
  <cookie_difficulty>10</cookie_difficulty>
  <packet_timeout>10</packet_timeout>
  <max_retries>5</max_retries>
  <sa_lifetime>900</sa_lifetime>
  <send_hi_name>yes</send_hi_name>
  <dh_group>3</dh_group>
  <dh_lifetime>900</dh_lifetime>
  <r1_lifetime>300</r1_lifetime>
  <failure_timeout>50</failure_timeout>
  <msl>5</msl>
  <ual>600</ual>
  <min_reg_lifetime>96</min_reg_lifetime>
  <max_reg_lifetime>255</max_reg_lifetime>
  <hip_sa>
    <transforms>
      <id>1</id>
      <id>2</id>
      <id>3</id>
      <id>4</id>
      <id>5</id>
      <id>6</id>
    </transforms>
  </hip_sa>
  <esp_sa>
    <transforms>
      <id>1</id>
      <id>2</id>
      <id>3</id>
      <id>4</id>
      <id>5</id>
      <id>6</id>
    </transforms>
  </esp_sa>
  <disable_dns_lookups>no</disable_dns_lookups>
  <disable_notify>no</disable_notify>
  <disable_dns_thread>yes</disable_dns_thread>
  <enable_broadcast>no</enable_broadcast>
  <disable_udp>no</disable_udp>
</hip_configuration>
```

Figure 11. Content of the hip.conf configuration file.

The *my_host_identities.xml* (Fig. 12) contains information about the own Host Identity i.e. the public and private key. It also contains the HIT, which is calculated upon the public key, and a LSI.

```
<?xml version="1.0" encoding="UTF-8"?>
<my_host_identities>
  <host_identity alg="RSA" alg_id="5" length="128" anon="no" incoming="yes"
 r1count="10">
    <name>jonnym-1024</name>
    <N>C465749D177F4CF180626818EA33440575CDA95D2888OF1C03BB9C866CAB99196807
       FOA33A7B1B08A65E01887C3591B92884D135B79EA0E9CF48126607E1927690EB5937
       E1E2B20E89E269E2E868CFBDFC2C27F12337C69388CB18E90BEA5F8C3C678EFF00E0
       A3AB4FEC5641AC88B4112B8165F95CE8CC0A06A93882E81248FF</N>
    <E>010001</E>
    <D>0327D228D800CE9EAEBE6607C8738C6B3E0A50E75348645DC1DD98D53C17C1BCFBC3
       ECE4E282A9DB88B8C7F3867361E504B6A599E45A63217E981CDB90D2DA049069E2B3
       DEFBE2538FAE4644A2D223960551FFBC742A2717117F6D007ED12407FB174633CBAC
       F23FB721EB37D670366D659ED35C8FBA1B5CAA070108D65ACD89</D>
    <P>FEBCE61F2BEF9058CD17DA125B18C07509E21267F9FCC9904EF31F1801DD817A0A2C
       4D77C9D916370A1C38D9A07CCA0D636EDD625F72E8CE76EA7D7762B3DF15</P>
    <Q>C55E8EDEECF40758612D47635F2CF3C62C6398F36D27B866845C1BDF2F1799F8273A
       B2CD813E075646DFD50474166695A38D115E90DE67C933EDA303E3C8ECC3</Q>
    <dmp1>CD897F28219C2F5CE746DA86BB0812A99CAAD36D1FD0EE95C88DF445BBA38D47F
          A135B024F5420CF9211C9711743F907AA76169CBC250C6122A3FB4E2BCEB42D
    </dmp1>
    <dmq1>3DEAFC47536EFB1EEEDD9597C734030618C6E624F8098E001B660A186A5DAD1FF
          CAE9D897B124A9AF68122D2D3A873665E78D30A27C89E26B0F3CC405C259CA1
    </dmq1>
    <iqmp>9F9636265D0A799A363AAF20D5A8418687EA101846D6A90B866B3E277C04DB4BF
          E08A61540D0CEF4F655D6B0F82A3E8BECC68FCAC99FC446D51DB85B25AA1866
    </iqmp>
    <HIT>2001:12:4b7a:4a66:ac4d:7141:40ff:71ae</HIT>
    <LSI>1.255.113.174</LSI>
  </host_identity>
</my_host_identities>
```

Figure 12. Content of the my_host_identitites.xml file. Content of this file should not be shared in public.

known_host_identities (Fig. 13) contains information about other known host identities. It is used to locally map a HIT or LSI with a IP address when not using a DNS. Therefore a host´s HIT, LSI, IP and optionally one or several RVS names are stored in this file.

```
<?xml version="1.0" encoding="UTF-8"?>
<known_host_identities>
  <host_identity alg="RSA" alg_id="5" length="128" anon="no" incoming="yes">
    <name>karlsson-1024</name>
    <addr>192.168.1.101</addr>
    <HIT>2001:0016:23f2:11a1:a72d:0ec0:0780:1bb9</HIT>
    <LSI>1.110.30.145</LSI>
  </host_identity>
  <host_identity alg="RSA" alg_id="5" length="128" anon="no" incoming="yes">
    <name>wim.hip.arcada.fi-1024</name>
    <addr>193.167.36.210</addr>
    <HIT>2001:1b:ebc4:4ac4:afbb:919d:41b:daa4</HIT>
    <LSI>1.110.30.146</LSI>
  </host_identity>
  <host_identity alg="RSA" alg_id="5" length="128" anon="no" incoming="yes">
    <name>i3.hip.arcada.fi-1024</name>
    <addr>193.167.36.209</addr>
    <HIT>2001:1b:f9e4:bcf5:75a6:bfdf:316e:c4a9</HIT>
    <LSI>1.110.30.148</LSI>
  </host_identity>
<host_identity alg="RSA" alg_id="5" length="128" anon="no" incoming="yes">
    <name>i1.hip.arcada.fi-1024</name>
    <addr>193.167.36.207</addr>
    <HIT>2001:16:f68f:ae3a:3497:a9c2:3173:9be</HIT>
    <LSI>1.110.30.128</LSI>
  </host_identity>
<host_identity alg="RSA" alg_id="5" length="128" anon="no" incoming="yes">
    <name>i2.hip.arcada.fi-1024</name>
    <addr>193.167.36.208</addr>
    <HIT>2001:16:f68f:ae3a:3497:a9c2:3173:9be</HIT>
    <LSI>1.110.30.118</LSI>
  </host_identity>
<host_identity alg="RSA" alg_id="5" length="128" anon="no" incoming="yes">
    <name>hipserver.mct.phantomworks.org-1024</name>
    <HIT>2001:14:4dcd:2a09:74a:caee:2a0:ec4a</HIT>
    <LSI>1.230.120.200</LSI>
  </host_identity>
<host_identity alg="RSA" alg_id="5" length="128" anon="no" incoming="yes">
    <name>MyVirtualMachine-1024</name>
    <addr>192.168.0.17</addr>
    <HIT>2001:10:c2fa:ebe7:8dd2:1136:6821:73d4</HIT>
    <LSI>1.230.120.168</LSI>
  </host_identity>
</known_host_identities>
```

Figure 13. Content of the known_host_identities.xml file.

3.9.2 InfraHip

InfraHip is an open source HIP implementation developed by HIIT, Helsinki Institute of Information Technology. Only supported operating systems are those that are based on Linux 2.6.

Using InfraHip

The HIP daemon of InfraHip can be started with the following command: */usr/sbin/hipd.*

The host´s own HI and HIT are stored in */etc/hip*. If DNS is not in use, known HIT´s and IP addresses can be stored in */etc/hip/hosts* or mapped by the command *hipconf add map [HIT] [IP]*

3.9.3 Hip4inter

Hip4inter is a HIP implementation developed by Nomadic Labs at Oy LM Ericsson Ab. Hip4inter is only supported by FreeBSD 5.4 and 6.0 (HIP for BSD Project documentation, 2005).

Using Hip4inter

Hip4inter is very similar to Infrahip. It can be started by the command */usr/sbin/hipd.* HITs of known hosts can be stored in */etc/hosts.* The own HI and HIT are stored in: */etc/hip*

3.10 HIP on Symbian

Symbian is a widely used operating system for smartphones. A group of researchers in Helsinki Institute for Information Technology (HIIT), Helsinki University of Technology and University of Helsinki have ported both HIPL and OpenHIP for execution on a phone with Symbian S60 3rd edition OS (A. Khurri, D. Kuptsov, A. Gurtov, 2009) The following subchapters describe the porting process and the performance of HIP on three different Symbian phones.

3.10.1 Porting process

To be able to port the source code of HIPL and OpenHIP to Symbian without larger modifications, an Open C SDK plug-in for S60 3rd edition SDK was used. The Open C plug-in offers possibility to run many standard C functions and therefore it was an important component for avoiding a large modification of the source code.

During the porting process most compilation errors were data type conversions (A. Khurri, D. Kuptsov, A. Gurtov. 2009). For the HIPL project a few header files were up-

dated. The OpenHIP project was more suited for porting to Symbian, no system header files needed to be updated. Other porting issues were different errors that appeared due to differences in Linux and Symbian emulator compilers and memory alignment errors.
Both HIPL and OpenHIP are written in C and contain many platform dependent features. Therefore the ported version of HIPL does not fully support ESP encapsulation. For the OpenHIP a userspace alternative (PFKEY protocol and SADB) was used and therefore ESP encapsulation is fully supported.

3.10.2 Performance

Performance of HIP on Symbian was tested on 3 Nokia phones that are based on Symbian S60 3rd edition (Khurri, 2009). For the newest one of the tested phones, Nokia E51, a Base Exchange with a server over a WLAN varied between 1.68 and 3.17 seconds. A base exchange between two phones varied between 3.49 and 6.71 seconds. OpenHIP was slightly faster than HIPL.
A key pair creation with 512 bit key length took 4.90 seconds for a DSA key and 0.51 seconds for a RSA key. Creation of a 2048 bit key took 389.99 seconds for DSA and 40.73 seconds for a RSA key.

4 PRACTICAL MOBILITY TESTS BASED ON HIP

4.1 Mobile video streaming with VLC

The purpose of this test was to make a video stream mobile. VLC 1.0.3 player was used as both streaming server and client. All three implementations, OpenHip, InfraHIp and Hip4Inter, were first tested separately and then "crosstested". In all tests the server part was a virtual machine in the hip.arcada.fi domain. All clients were attached to the hip.arcada.fi network through a network wire. Mobility was tested by pulling out the wire and switching to a wireless network in the arcada.fi domain.

4.1.1 Openhip

The stream was started by choosing UDP as protocol and the client´s LSI as address. The stream can also use the HIT instead of the LSI. The graphical interface of VLC was used to start and receive the stream. The test was successful (Fig. 14)

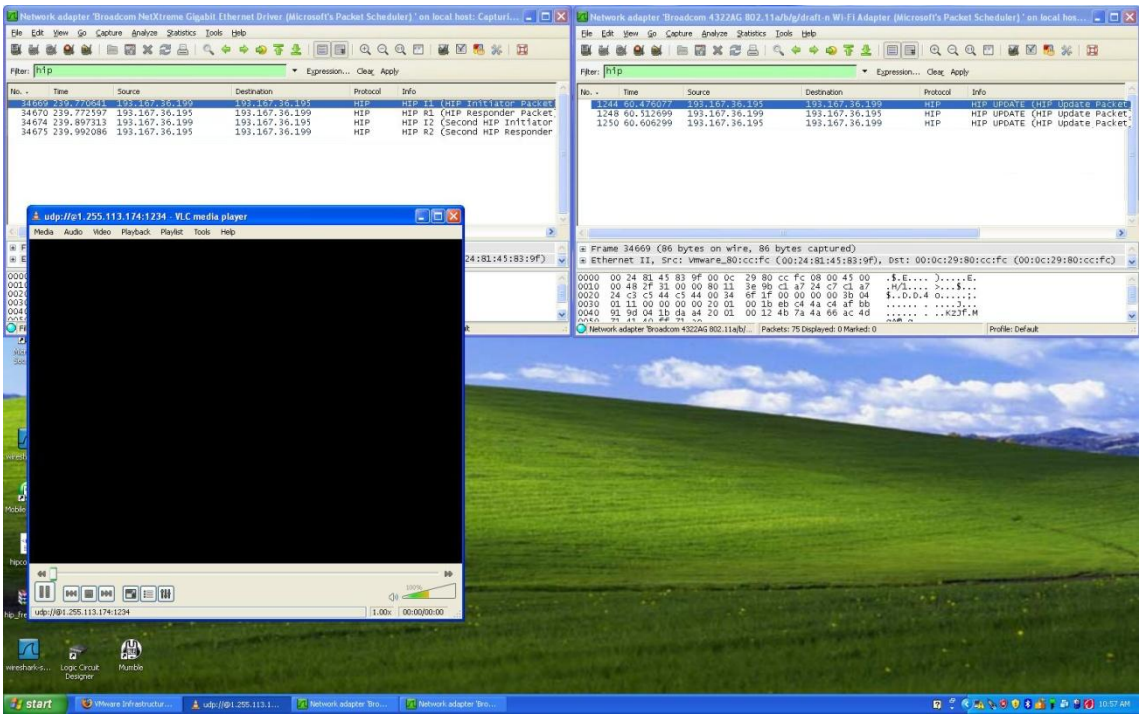

Figure 14. Successful mobile video stream using OpenHIP and VLC.

4.1.2 InfraHip

The stream was started from the command line with following command:

vlc–vvv[FILENAME.AVI]--ipv6--sout

'#std{access=udp,mux=ts,dst=[2001:0012:4b7a:4a66:ac4d:7141:40ff:71ae]:1234}'

The stream was received with the following command:

```
vlc -vvv 'udp://@[::]:1234'
```

The test was successful.

4.1.3 hip4inter

The stream was started from the command line with following command:

vlc–vvv[FILENAME.AVI]--ipv6--sout

'#std{access=udp,mux=ts,dst=[2001:0012:4b7a:4a66:ac4d:7141:40ff:71ae]:1234}'

The stream was received with the following command:

```
vlc -vvv 'udp://@[::]:1234'
```

The test was successful.

4.1.4 Crosstesting

All three implementations were also crosstested. Basic streaming worked on every combination but after a mobility event the only combinations that were able to continue the stream were those who had OpenHIP as client (Fig. 15). In all three implementations the software has been developed based on the same protocol description but it is still a possibility that it has been implemented in a different way. Especially the mobility section is a critical part of the software.

Implementations	Openhip (client)	Infrahip (client)	Hip4inter (client)
Openhip (server)	X	0*	0*
Infrahip (server)	X	X	0*
Hip4inter (server)	X	0*	X

Figure 15. Result of mobile video stream tests. "X" = successful, "O" = unsuccessful, "" = streaming on HIP works but not mobility.*

4.2 Mapping a network drive with Expandrive

Expandrive is a software company in Cambridge, MA. Expandrive has developed network mapping software that has the same name as the company: Expandrive. Expandrive can be used to map network drives on FTP/FTPS/SFTP/SSH-servers.

An Ubuntu 9.04 virtual machine in the hip.arcada.fi-domain was used as server and a laptop with Microsoft XP professional was used as client. A shareware version of Expandrive v1.8.3 was installed on the client computer.

The network drive was mapped by its LSI-address (Fig. 16)

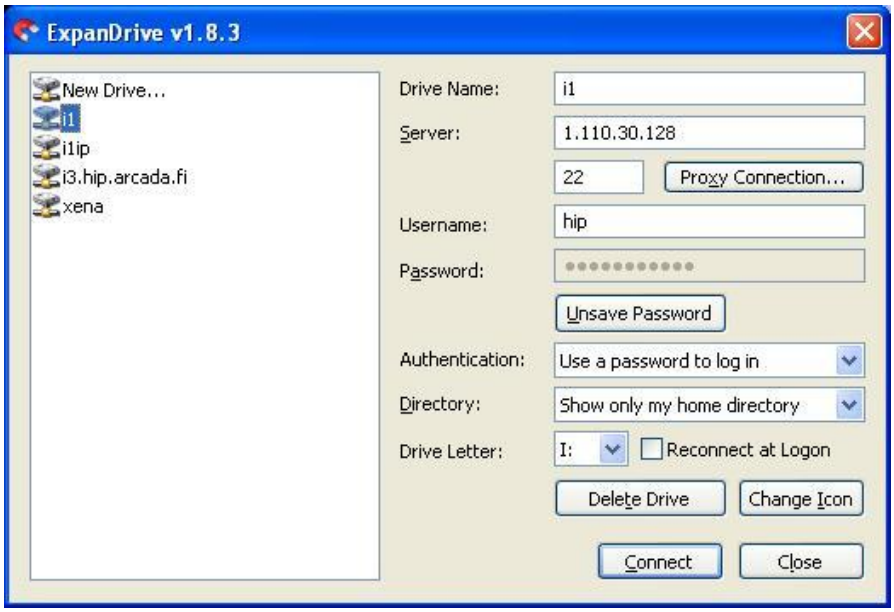

Figure 16. Settings showing the LSI address that is used to map a network drive.

The network drive was mapped successfully (Fig. 17). Mobility was tested by streaming a video from the mapped unit and switching from a cable based connection to a wireless connection. The mobility worked fine (Fig. 18).

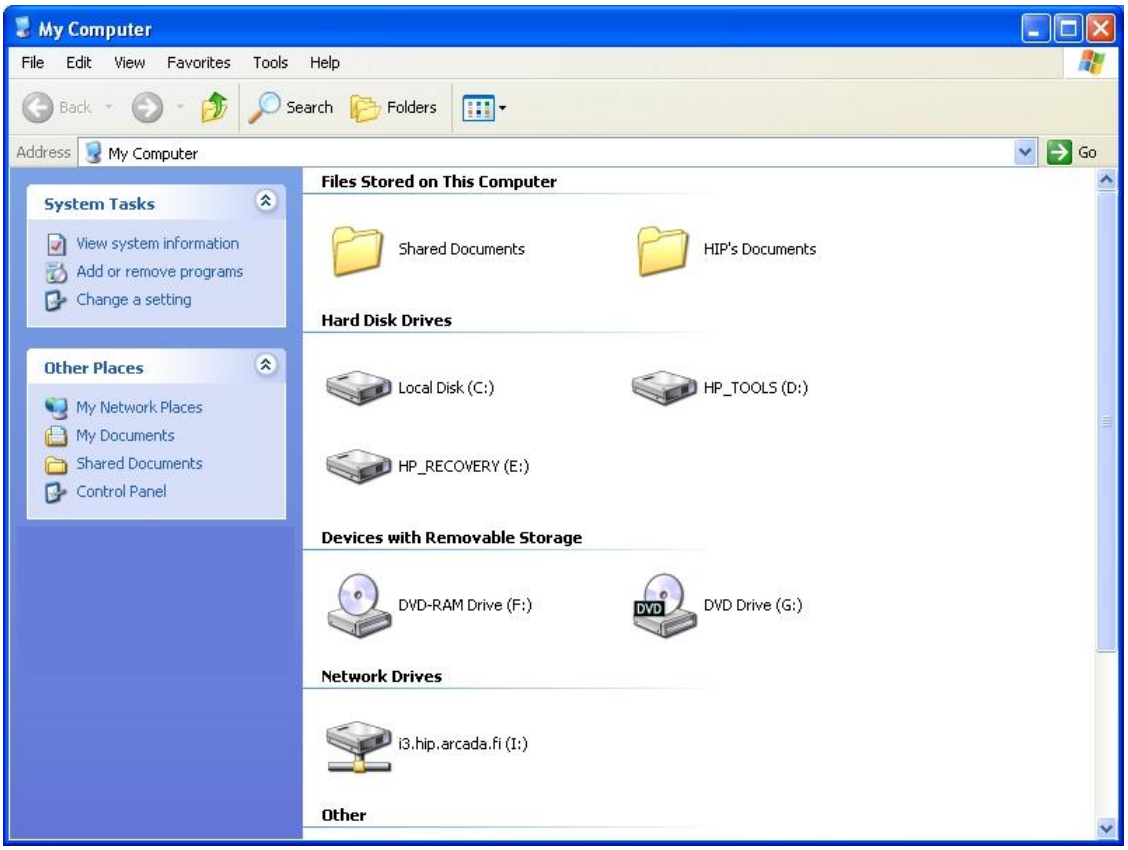

Figure 17. A successfully mapped network drive based on a HIP connection.

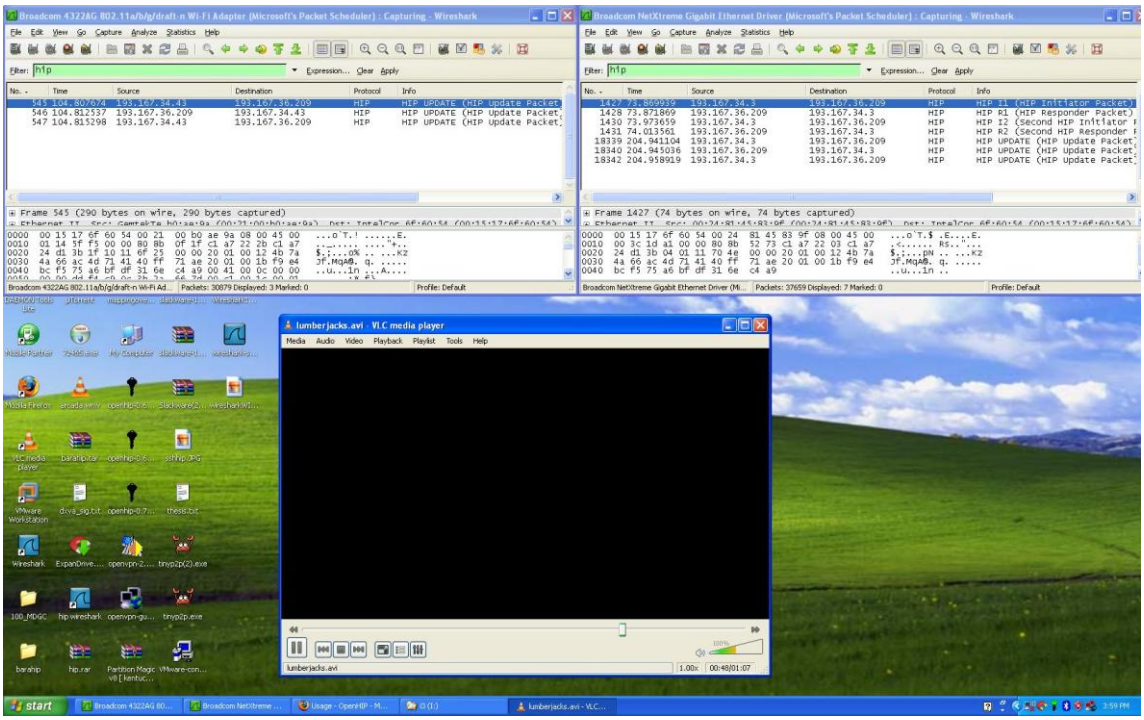

Figure 18. Screen capture of VLC streaming video and Whireshark captions from both used interfaces.

4.3 Text chat with a P2P chat application

The purpose of this test was to show how a chat can continue uninterrupted if it is based on a HIP connection. The chat application that was used in this test is a simple messaging application developed by Jonas F. Jensen. P2P chat is released under the GNU GPL license.

The test was performed on two computers that were in the same LAN. Both computers had Windows XP SP3 as operating system and OpenHIP as HIP software. A HIP connection was established by using LSI (Fig. 19) Mobility was tested by pulling out the wire of one of the clients and switching to a wireless network. The test was successful (Fig. 20).

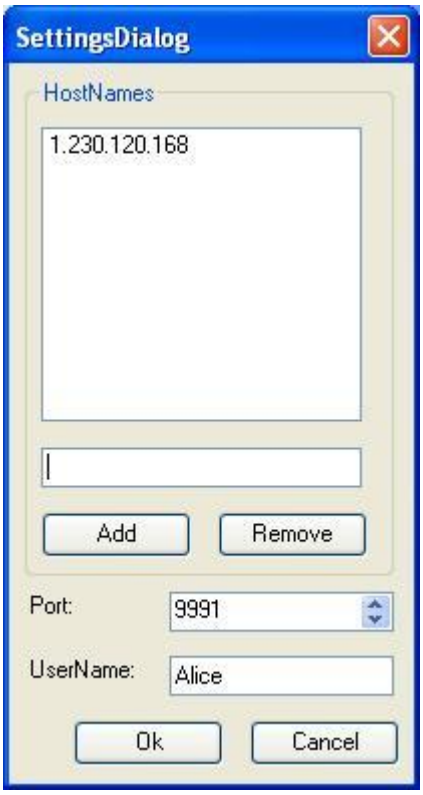

Figure 19. Settings from a P2P chat application showing that the communication is based on a LSI address.

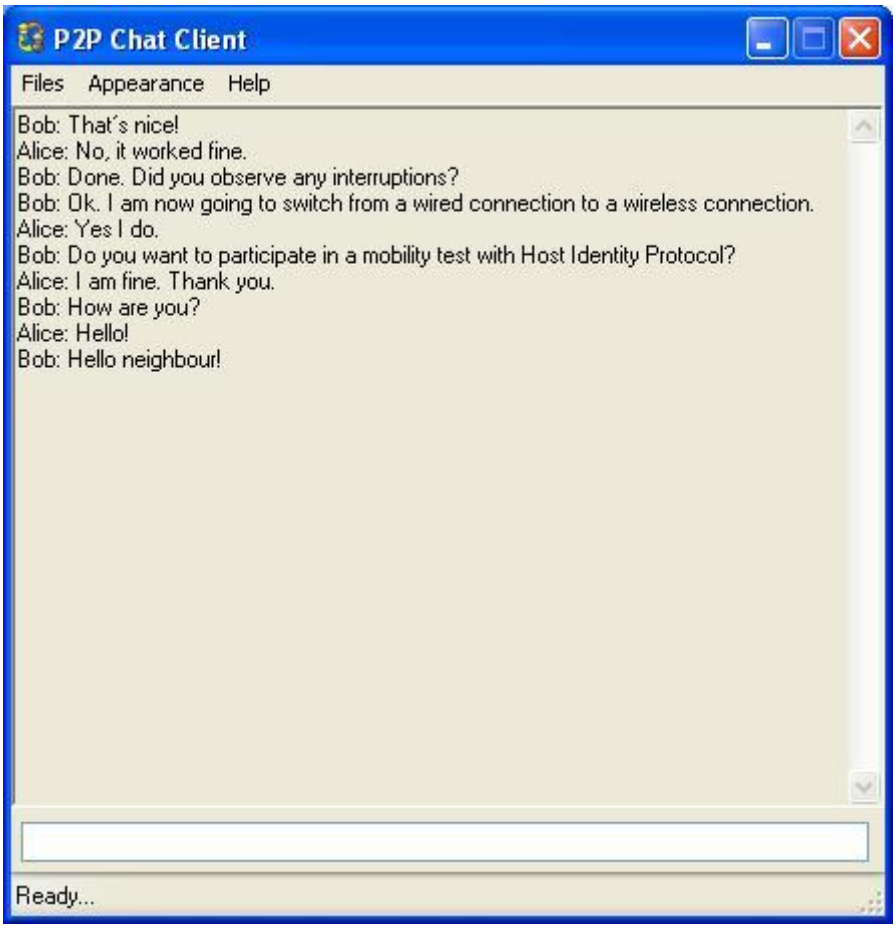

Figure 20. Screenshot from a P2P chat application that shows how the chat continues uninterrupted even if one of the clients gets a new IP address.

4.4 Video chat with yawcam

The purpose of this test was to show how a video chat can continue uninterrupted if it is based on a HIP connection. Video streaming software used in this test was yawcam and a webcam was used as the video source. Yawcam is a webcam application written in Java by Magnus Lundvall. It supports streaming to http, to ftp or to a file.

The test was performed on two computers in the same LAN. Both computers had Windows XP SP3 as operating system and OpenHIP as HIP software. The streaming host used the yawcam software and the streaming output was set to its LSI address. The receiver captured the stream by browsing to the sender´s LSI address and a specific port in a web browser. In this case Mozilla Firefox was choosed as web browser and the stream was captured with both IP address and LSI address of the sender. Mobility was tested by pulling out the wire of one of the clients and switching to a wireless network.

The test was successful. The stream that was based on the LSI address continued uninterrupted and the stream that was based on the IP address was interrupted (Fig. 21 & 22)

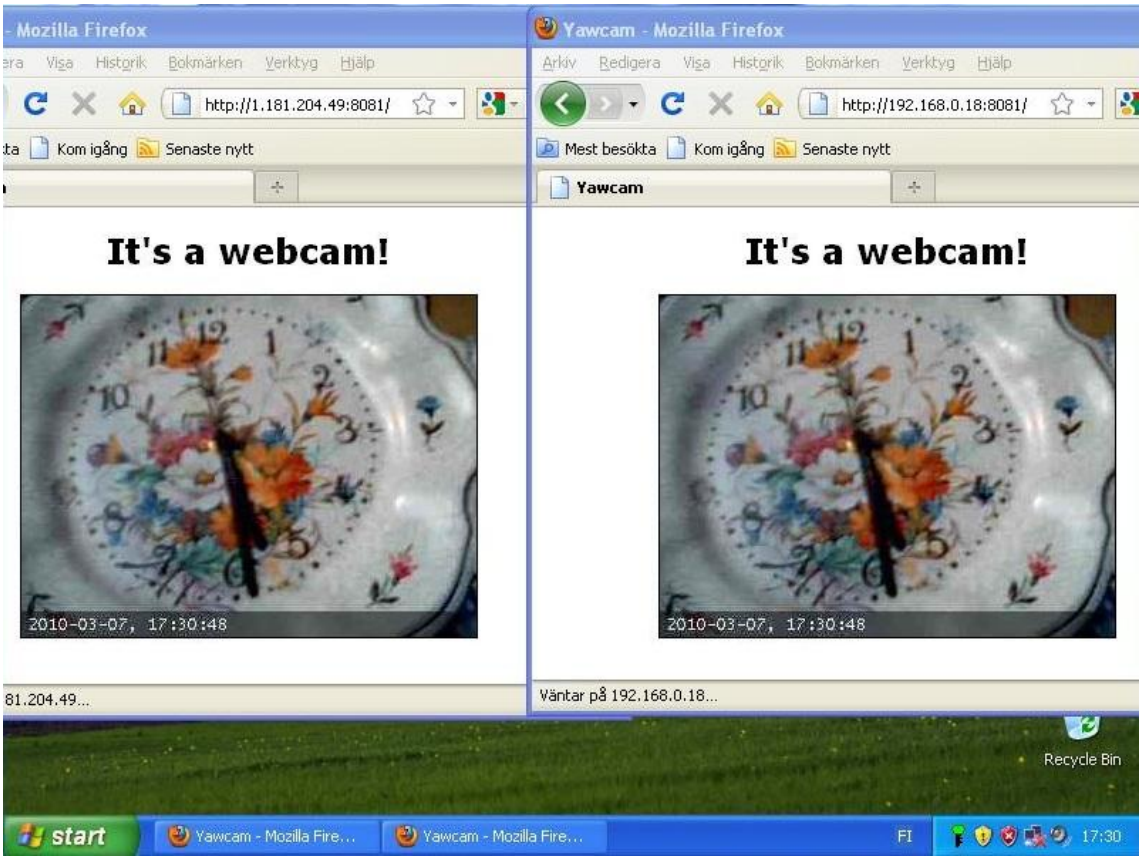

Figure 21. Two ways of streaming from the same source. The left one is based on the LSI address and the right one on the IP address.

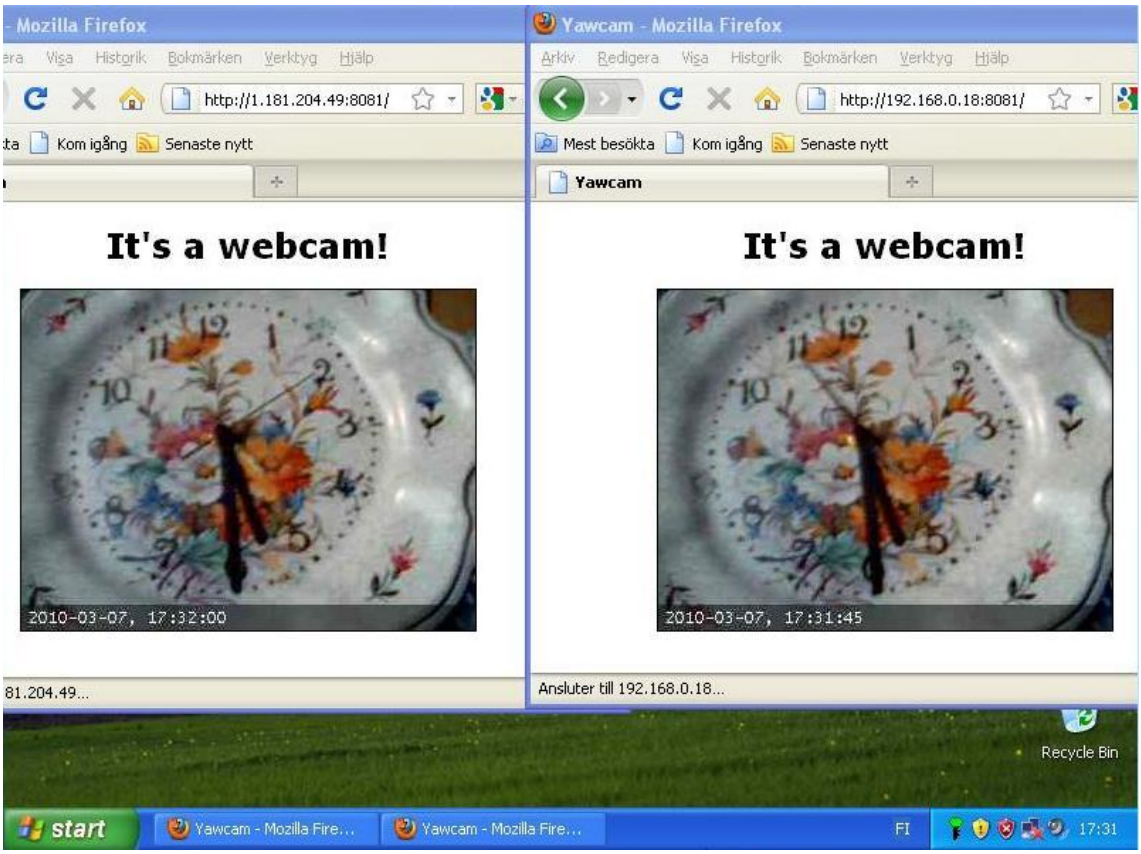

Figure 22. The left stream based on LSI continues uninterrupted and the right stream based on the IP-address is interrupted.

4.5 HIP on Symbian

The mobility of HIP on Symbian was planned to be tested as a part of this thesis. HIIT provided two binary files, both the HIPL and the OpenHIP version that they had ported. Unfortunately the signing of the binary files, with the IMEI-code of the phone that the application were planned to be performed on, failed. HIIT also provided source code for the HIPL project. After trying to compile the source code it turned out that it would have needed an unknown amount of modification. That would be outside the purpose of this thesis.

If someone would like to continue the work of Symbian of HIP, it should be ported to work on Symbian^3 instead of Symbian S60, because Nokia does no longer support S60. Symbian^3 is completely redesigned and porting the HIPL and OpenHIP projects to a Symbian^3 supported application may be a very complex task.

5 FLASH ANIMATION ON HIP MOBILITY

As a part of the WISEciti project a flash animation demonstrating HIP has been designed by BSc Laura Bergström, Arcada. The flash animation contains 5 scenes:

- Plain TCP/IP communication
- Plain HIP/TCP/IP communication
- ESP HIP/TCP/IP communication
- Mobile HIP/TCP/IP communication
- HIP Multihoming

Laura designed the 3 first scenes and a small part of the fourth scene. As a part of this thesis the rest of scene 4 and scene 5 have been designed.
Both scene 4 and scene 5 contain of 3 networks and 2 nodes (Fig. 23) The Server Node (SN) is located in a network called website.fi. A Mobile Node (MN) switches between the Arcada.fi network and the Otherdomain.fi network. In both networks the MN is using a WLAN and HIP mobility is demonstrated by switching between the networks.

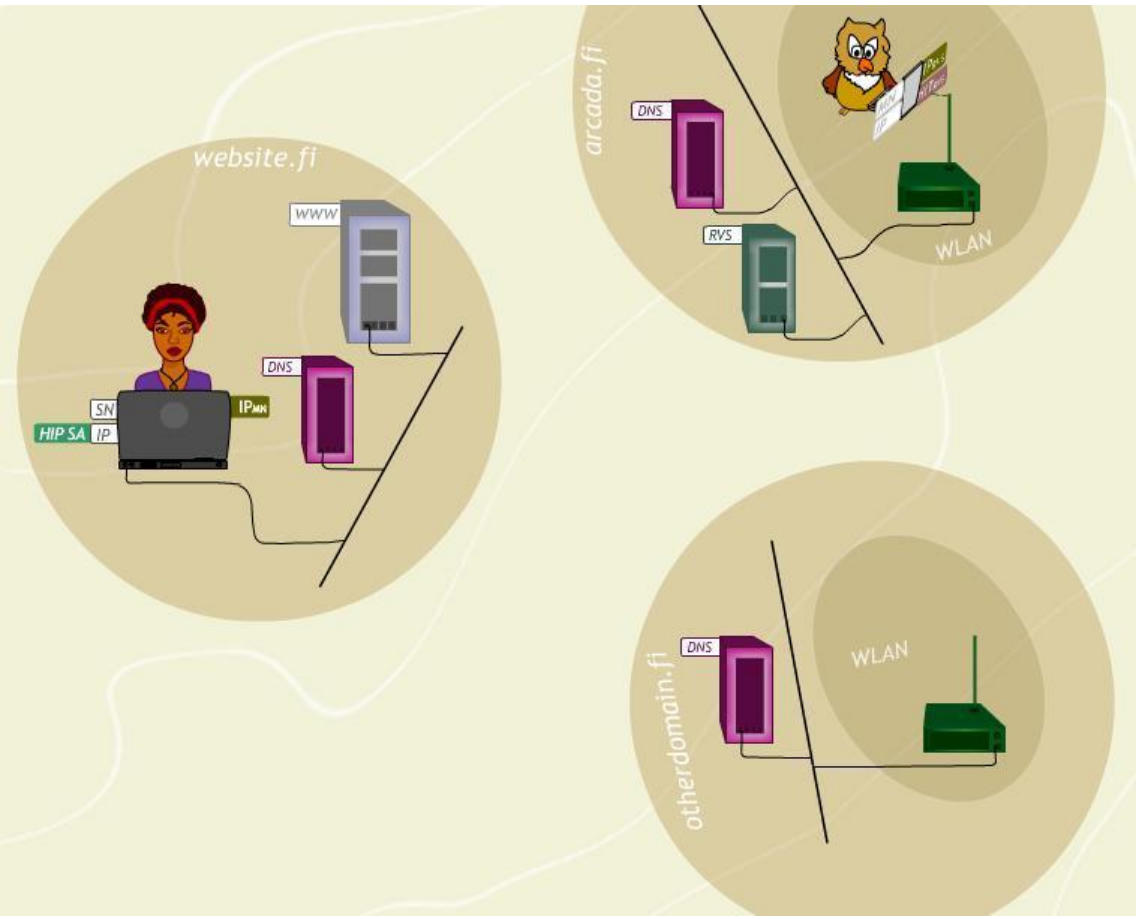

Figure 23. Network map of flash scenes 4 and 5.

Scene 4 is divided into 4 sub scenes.

- 4A: Rendezvous client registration
- 4B: Base Exchange with a Mobile Host through a RVS
- 4C: Update message to a Communicating Node
- 4D: Update message to a RVS

5.1 Scene 4A: Rendezvous Client Registration

Scene 4A shows the client registration to a RVS after the Mobile Node has entered Arcada.fi network. The scene shows detailed information of all 4 signaling messages (Fig. 24). After that there is a HIP association between the MN and the RVS located in Arcada.fi domain.

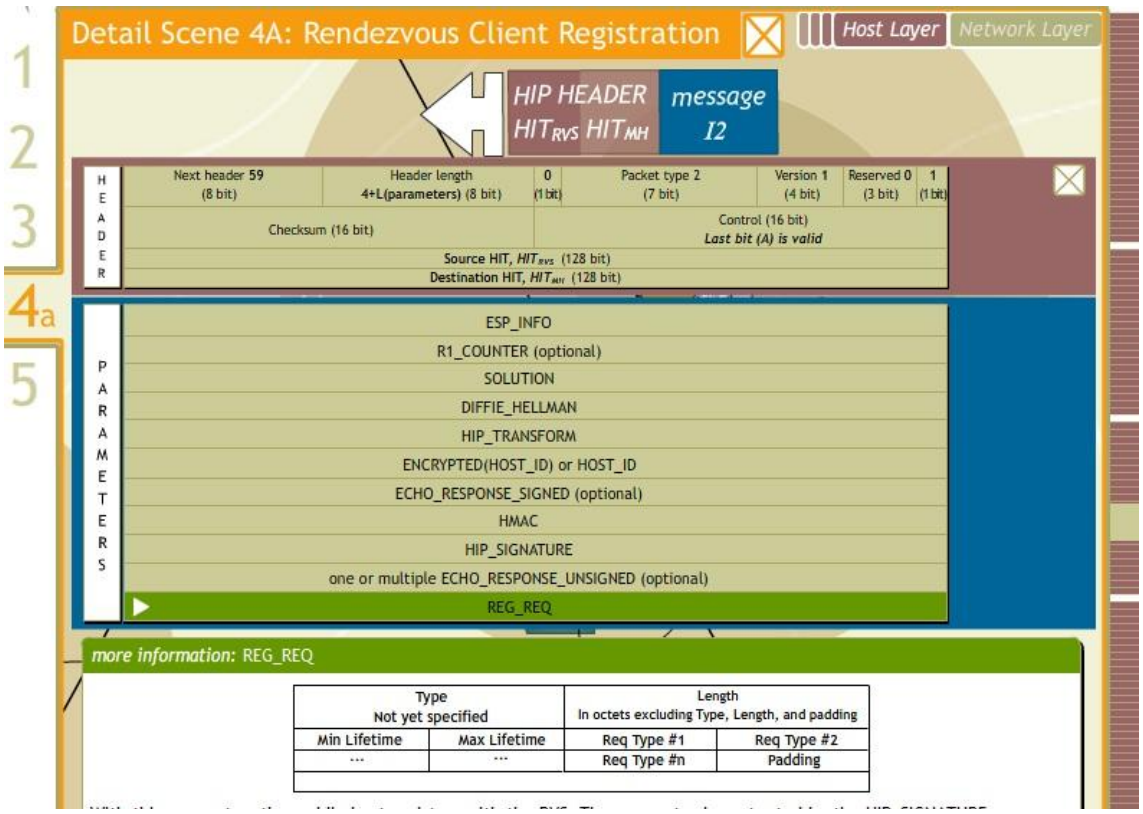

Fig 24. Detailed information of the I2 message when a Mobile Node is performing a RVS registration.

5.2 Scene 4B: Base Exchange with a MN through a RVS

Scene 4B shows the Base Exchange between the SN and the MN. Because the HIT of MN is unknown to SN, the Base Exchange is executed through the RVS that the MN is using. The HIT and IP of the RVS are received from the DNS. SN sends the I1 packet to the RVS (Fig. 25) that re-directs it to the MN. Because of the client registration in 4A, the RVS knows the HIT and IP address of the MN and is able to re-direct it to the right location.

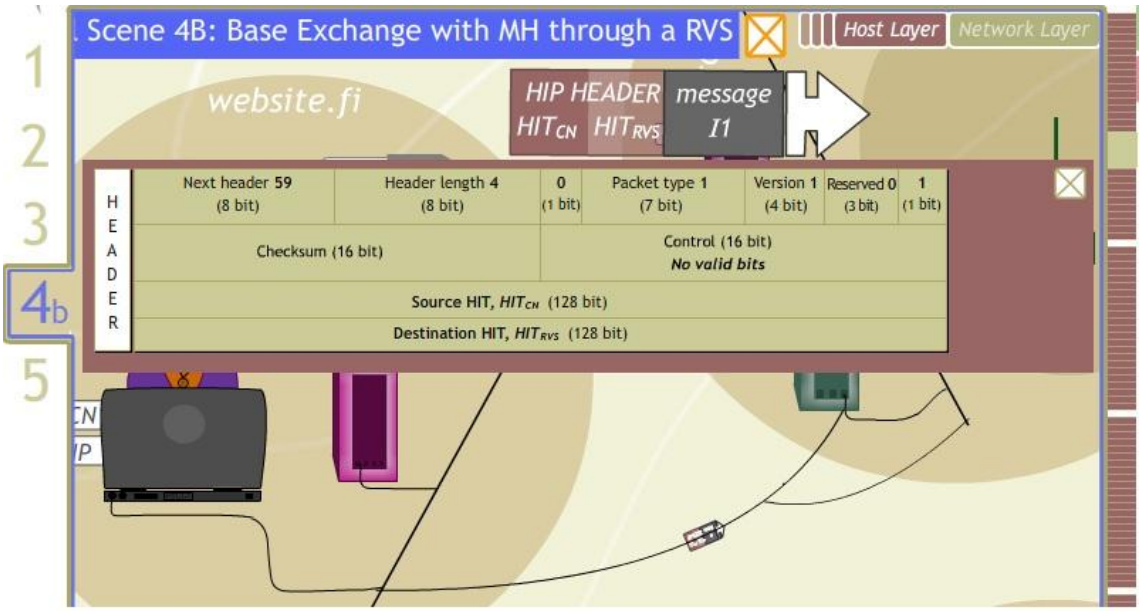

Fig 25. I1 packet is sent from SN to RVS because the HIT of MN is unknown to SN.

5.3 Scene 4C: Update message to SN

Scene 4C shows the UPDATE communication between the SN and the MN when the MN has moved from the Arcada.fi network to the Otherdomain.fi network. The scene shows detailed information of all the three UPDATE packets. Especially the LOCATION parameter is described in detail (Fig. 26).

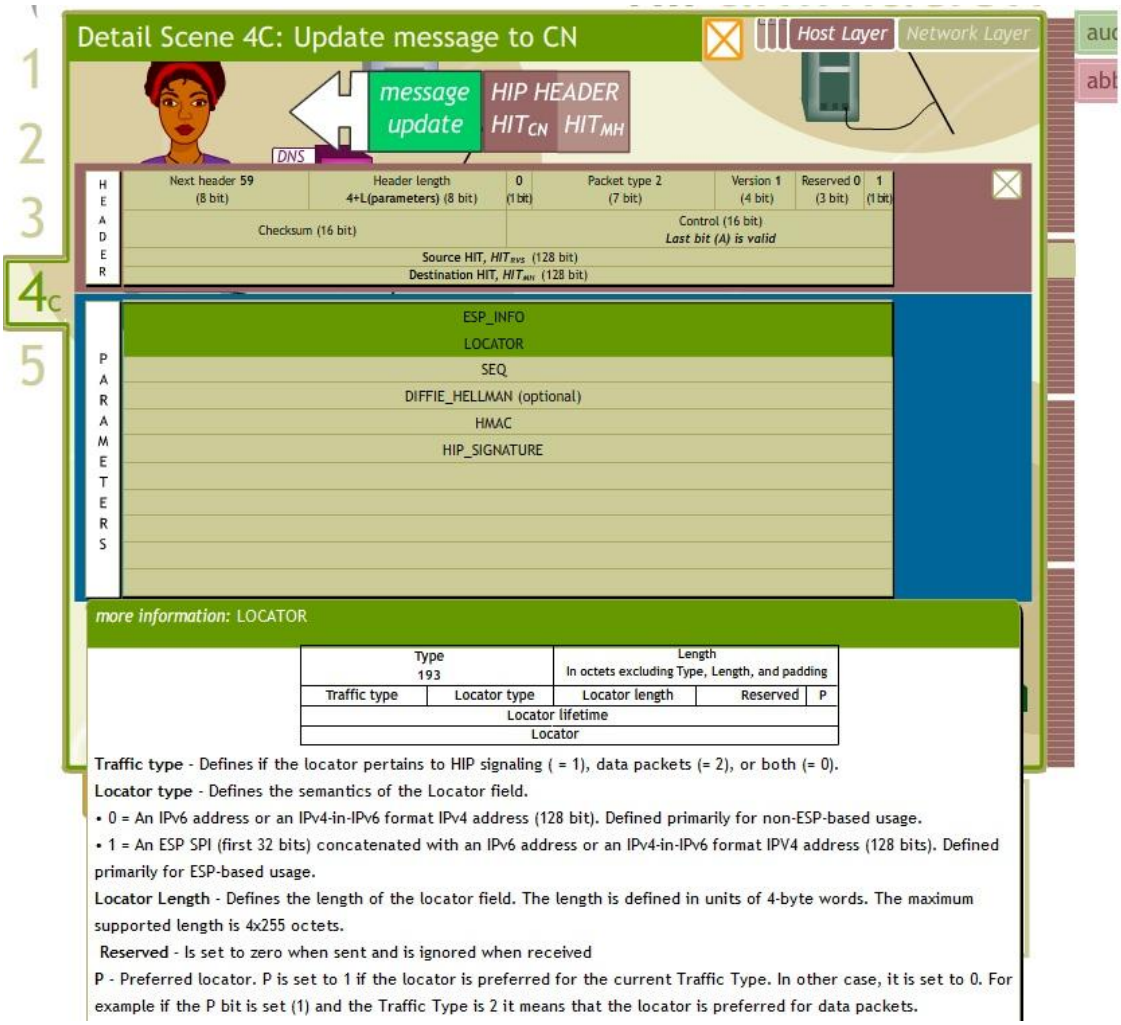

Fig 26. UPDATE packet from MN to SN with detailed information of the LOCATOR parameter.

5.4 Scene 4D: Update message to RVS

Scene 4D shows the same UPDATE communicating as in 4C but now it is executed between the MN and the RVS (Fig. 27).

Fig 27. UPDATE communication between MN and RVS.

5.5 Scene 5: Multihoming

Scene 5 has only a main scene. The scene begins when the MN is having a HIP association with the SN from a 3G network. The MN moves into the range of the WLAN in the Arcada.fi network. After the move and before any UPDATE packets, data is still transported through the 3G network (Fig. 28).

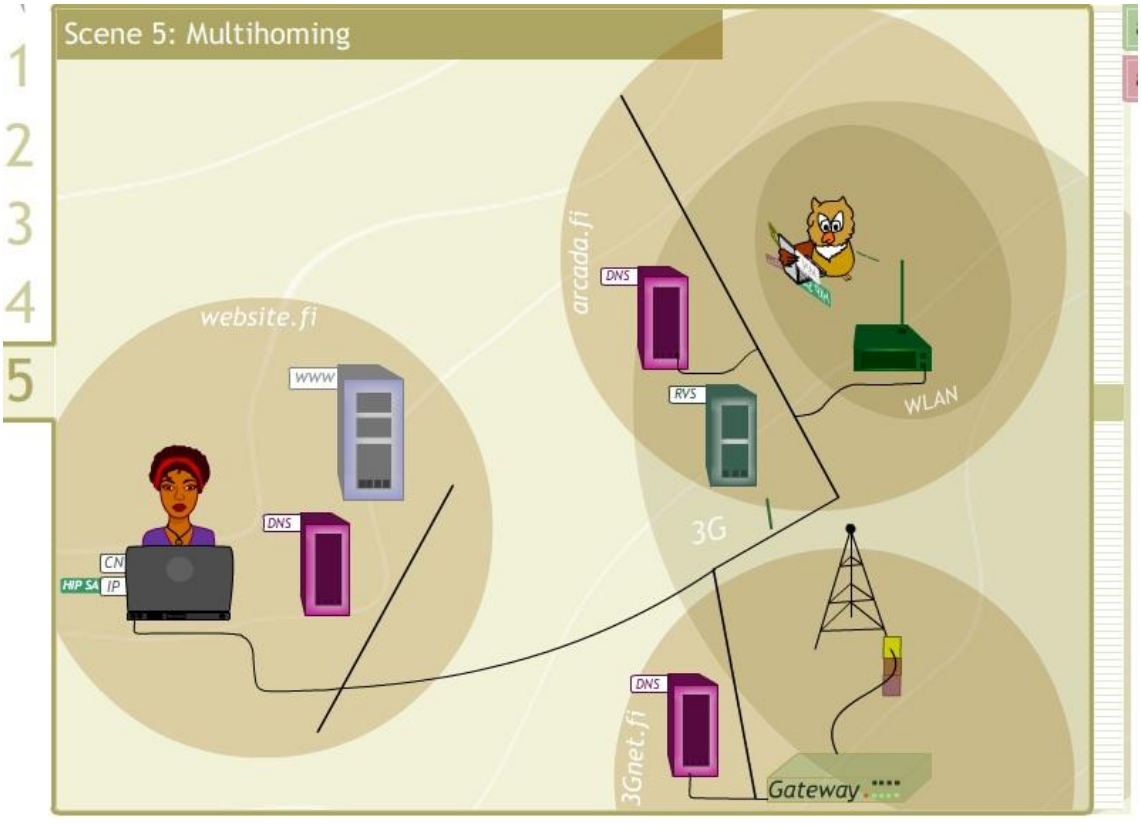

Fig 28. Data is transported to same location until SN receives an UPDATE packet from MN.

The scene shows the UPDATE communication with detailed information. The UPDATE packet from the MN contains a parameter that informs the SN which location address is preferred. After that data is transported to the new location of the MN (Fig. 29) even if it is still connected to the 3G network.

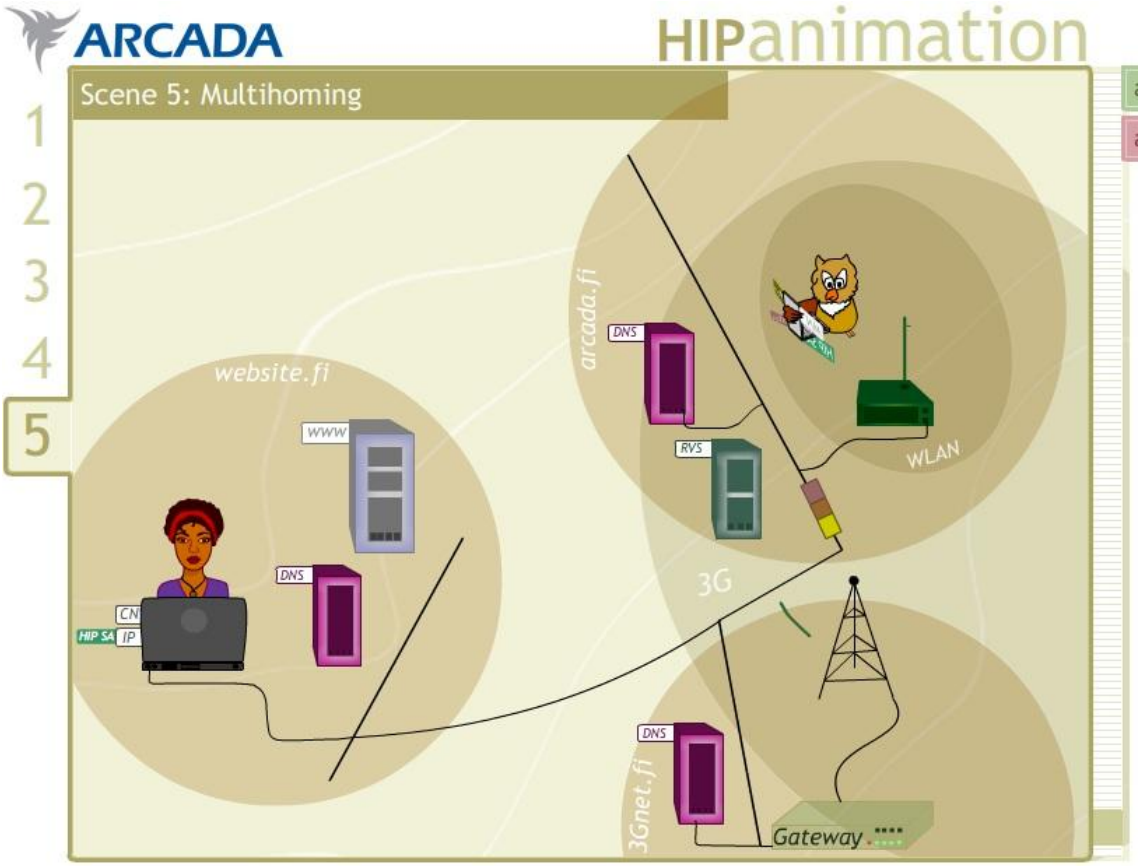

Fig 29. After an UPDATE event, SN transports data packets to the location that the multihomed MN prefers.

6 CONCLUSIONS

The society we live in is moving increasingly towards a mobile society where you evenly and continuously have to be connected to the Internet. Most of the data traffic today is transferred over connections that are not mobile. Great efforts are made to make all data traffic mobile. As this thesis shows, there are already commercial technologies (Mobile IP) as well as promising technologies such as Host Identity Protocol (HIP). The aim of this thesis is to evaluate the HIP from a user perspective. How do popular applications work when they are made mobile with HIP?

The practical part of this thesis is divided into two different parts. The first part was to evaluate the mobility of HIP with a video stream application, a text chat application, a video chat application, and a network drive mapping application. Due to the architecture of popular programs such as MSN messenger and Skype, lesser known applications were used. Some of the tests were done with all three HIP implementations and some only with one of the implementations when the tested applications were designed to work only on a computer with Windows as operating system.

None of the tests failed totally. Mapping a network drive, text chatting and video chatting worked perfectly with HIP. These tests had one thing in common, they were all tested only with one implementation (OpenHip or InfraHip) and both nodes used the same implementation of HIP. The mobile video stream test worked fine for all implementations when both parts of the communication used the same implementation and when the stream receiver used Openhip. Other combinations failed after a mobile event. The reason that some of the tests failed are probably at source level of the implementations.

As a part of the practical studies of this thesis, HIP on Symbian was planned to be tested. Unfortunately, it turned out that it would have required too much work even to get the source code that HIIT provided compiled.

The practical part of this thesis also included further development of a Flash animation. This was the most time-consuming part, although it doesn't take up so much space in the thesis report. This was mainly due to the fact that it is very hard to take over projects of other people. You don't know how they have planned to do the rest of the project and

you have to go through all what they have made to figure out what can be reused and so on. Another problem was that the project was developed with an earlier version of Actionscript and with an old IDE. The conversion to Adobe Flash CS3 caused that part of the Actionscript code was available only through the search function, even if you knew which button the code was linked to.

As a summary I think that HIP is a very strong candidate for future data communication. Theoretically, the advantages of HIP are not limited to the possibility of mobile communication, also the network security features make HIP a desirable candidate.
However, HIP is still at the research stage and the practical part of this thesis shows that there is a lot work to do to get it to work stable.
The fact that HIP is still on the research stage has made this thesis more challenging.
If a test has failed, is it because you are doing it wrong or is the problem associated with HIP implementations?
Also the theoretical part has been challenging as there is not yet so much publications on HIP. However, I am sure this will change much in the future because I believe that HIP is a very good candidate for future data communication.

REFERENCES

WISEciti Project 2010. Retrieved on 29 October from: http://www.cs.helsinki.fi/group/wiseciti/

Moskowitz, R & Nikander P. (2006). Host Identity Protocol (HIP) Architecture, IETF RFC 4423. Retrieved Mars 1, 2010 from http://tools.ietf.org/html/rfc4423

Moskowitz R, Nikander P, Jokela P, & Henderson T (2008). Host Identity Protocol. RFC 5201. Retrieved Mars 1, 2010 from http://tools.ietf.org/html/rfc5201

J. Laganier, DoCoMo Euro-Labs, L. Eggert (2008).Host Identity Protocol (HIP) Rendezvous Extension. RFC 5204 Retrieved July 30, 2010 from http://tools.ietf.org/html/rfc5204

Moskowitz R, Nikander P & Jokela P(2008). Using the Encapsulated Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP). RFC 5202. Retrieved Mars 1, 2010 from http://tools.ietf.org/html/rfc5202

Melen J, Ylitalo J, Salmela P, Henderson T (2009). Host Identity Protocol-based Mobile Router (HIPMR) draft-melen-hip-mr-02. Retrieved October 29, 2010 from http://tools.ietf.org/html/draft-melen-hip-mr-02

Nikander P, Henderson T (2008). End-Host Mobility and Multihoming with the Host Identity Protocol. RFC 5206. Retrieved August 14, 2010 from http://tools.ietf.org/html/rfc5206

Hobaya F, Gay V, Robert E (2009). Host Identity Protocol Extension Supporting Simultaneous End-host Mobility. Presented at Wireless and Mobile Communications, 2009. ICWMC '09. Fifth International Conference in Cannes, La Bocca on 23-29 Aug. 2009.

Nikander p, Laganier J (2008). Host Identity Protocol (HIP) Domain Name System (DNS) Extension. RFC 5205. Retrieved Mars 1, 2010 from http://tools.ietf.org/html/rfc5205

Laganier J, Eggert L (2008). Host Identity Protocol (HIP) Registration Extension. RFC 5203. Retrieved Mars 1, 2010 from http://tools.ietf.org/html/rfc5203

Perkins C (2002). IP Mobility Support for IPv4. RFC 3344. Retrieved July 30, 2010 from http://tools.ietf.org/html/rfc3344

Perkins C, Johnson D, Arkko J (2004). Mobility Support in IPv6. Retrieved July 30, 2010 from http://tools.ietf.org/html/rfc3775

Kivinen T, Tschofenig H (2006). Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol. RFC 4621. Retrieved July 30, 2010 from http://tools.ietf.org/html/rfc4621

Kent S, Seo K (2005) Security Architecture for the Internet Protocol. RFC 4301. Retrieved August 14, 2010 from http://tools.ietf.org/html/rfc4301

Harkins D, Carrel D (1998). The Internet Key Exchange (IKE). (RFC 2409). Retrieved July 30, 2010 from http://tools.ietf.org/html/rfc2409

Kaufman C (2005). Internet Key Exchange (IKEv2) Protocol. (RFC 4306). Retrieved July 30, 2010 from http://tools.ietf.org/html/rfc4306

Andrei Gurtov (2008) Host Identity Protocol (HIP) – Towards the secure mobile Internet. Chippenham: John Wiley & Sons Ltd. 295 s.

Khurri A, Kuptsov D and Gurtov A(2009). Performance of Host Identity Protocol on Symbian OS. Presented at Communications, 2009. ICC '09. IEEE International Conference in Dresden on 14-18 June 2009.

Khurri A (2009) Evaluating IP Security and Mobility on Lightweight Hardware. Licentiate Thesis, Helsinki University of Technology. Retrieved on August 28, 2010 from http://lib.tkk.fi/Lic/2009/urn100059.pdf

HIP for BSD Project documentation, HIP Implementation for FreeBSD. Retrieved July 30, 2010 from http://www.hip4inter.net/documentation/hip4bsd.pdf

InfraHip Project: Intro, 2010. Retrieved July 30, 2010 from http://infrahip.hiit.fi/index.php?index=about

OpenHIP project: Wiki, 2010. Retrieved July 30, 2010 from http://www.openhip.org/wiki/index.php?title=Main_Page

SOFTWARE USED IN THE PRACTICAL TESTS:

OpenHIP project: OpenHIP V0.7. Retrieved December 2009 from http://sourceforge.net/projects/openhip/files/

InfraHIP project: HIPL. Retrieved December 2009 from http://infrahip.hiit.fi/index.php?index=download

Hip4Inter project: HIP for FreeBSD. Retrieved December 2009. A version only available for WiseCiti-project participants has been used. A public version without RVS-support is available from: http://www.hip4inter.net/download/download.php

WireShark: Wireshark v0.99: Retrieved December 2009 from http://www.wireshark.org/download.html. A patch is needed to get wireshark to show HIP-packets and can be downloaded from http://sourceforge.net/projects/openhip/files/

Virtual Machines: VMware player V3.0.1: Retrieved December 2009 from http://www.vmware.com/products/player/

Yawcam: Yet Another webcam software V0.3.3. Retrieved Mars 1, 2010 from http://www.yawcam.com/download.php

P2P Chat application: P2PChat V0.9 Retrieved Mars 1, 2010 from http://sourceforge.net/projects/p2pchat/

VLC: VLC media player V1.0.2. Retrieved Mars 1, 2010 from http://www.videolan.org/vlc/

Ubuntu: Ubuntu Desktop V9.04. Retrieved December 2009 from http://www.ubuntu.com/desktop/get-ubuntu/download

Adobe Flash CS3 Professional. Retrieved Mars 1, 2010. http://www.adobe.com/products/flash/