Handover Performance with HIP and MIPv6

Petri Jokela, Teemu Rinta-aho, Tony Jokikyyny, Jorma Wall, Martti Kuparinen, Heikki Mahkonen, Jan Melén, Tero Kauppinen, Jouni Korhonen*

NomadicLab, Ericsson Research
Oy L M Ericsson Ab
Jorvas, Finland
<firstname>.<lastname>@ericsson.com

*) Emerging Technologies and Innovations
TeliaSonera
Helsinki, Finland
jouni.korhonen@teliasonera.com

***Abstract*—Mobility management in the current Internet is designed to work with Mobile IPv4 and, when IPv6 is available, with Mobile IPv6. These solutions are based on the current architecture in the Internet, where the IP address represents both the locator and the identifier of the node.**

In the IETF, identity and location information separation has raised a lot of discussion and new ideas have emerged to separate these. Host Identity Protocol is one candidate that can be used for this separation. It introduces also a new way of handling mobility management taking advantage on the mentioned identity and location separation.

Keywords: Mobile IPv6, Host Identity Protocol, mobility management

I. INTRODUCTION

In the early days of the Internet, hosts were big and clumsy and remained in fixed locations. The development in technology has brought us light and small hosts that are easy to move. In order to maintain active connections to other hosts, the mobile host must be able to handle movements and inform other communicating parties that it has changed the topological location in the network. The first standardized mobility management scheme for the Internet was Mobile IPv4 (MIPv4) [1].

During the IPv6 [2] standardization, the mobility management was designed in parallel with the base specification. The aim was to provide an integrated mobility management scheme, Mobile IPv6 (MIPv6) [3]. In IPv4 this was not possible as the need for mobility came much later than the basic IPv4 protocol, thus MIPv4 became an add-on feature.

One problem with the current Internet architecture is that the IP address is used both for describing the topological location of the host and, at the same time, to identify the host. The Host Identity Protocol (HIP) [11] is one proposal to solve this semantic overloading of IP addresses. HIP introduces a new name space, the Host Identity name space, where the host identities are cryptographic. The location information, i.e. the IP address, is used only for routing purposes, not to identify the host. The resulting architecture provides a simple, yet secure, way to provide mobility and multi-homing for end-hosts.

From the operator's perspective the mobility management should not consume too much network resources. The amount of signaling should stay on a decent level. On the other hand, the end-user is not directly interested in the amount of signaling, but merely on packet loss and delays that directly affect the experienced quality of service. Naturally, the signaling may cause an increase in the bill, and that concerns also the end-user.

A mobile host may travel between different access networks, operated by different operators and even between networks of different technology. When the access technology changes, e.g. between WLAN and GPRS, the handover is said to be a *vertical handover*. In addition to this, the mobile host may make handovers between IPv4 and IPv6 networks.

In this paper, we compare the handover performance between Mobile IPv6 and HIP based mobility management in a heterogeneous IPv6 network environment. Our mobile host moves between WLAN and GPRS networks, and the mobility management is handled purely on layer 3 without any performance optimizations such as HMIPv6 [4] or FMIPv6 [5].

II. MOBIILTY MANAGEMENT

The purpose of mobility management is to keep host's communication context active while moving in the network. The mobility is transparent to the application level, although the QoS of the connection may change and affect applications. The mobile host may make a handover inside one access network, between different access technologies, or even between different IP address realms.

A. Mobile IPv6

Mobile IPv6 [3] allows a mobile node to move from one link to another without changing the mobile node's *home address*. The movement of a mobile node away from its home link is transparent to transport and higher-layer protocols. They use only the home address of the mobile node to identify it.

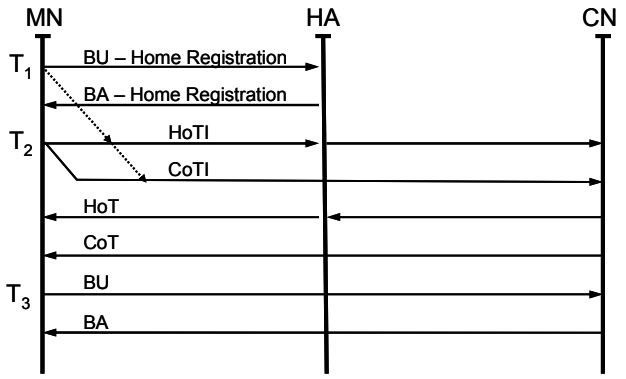

Figure 1 Complete Mobile IPv6 signaling procedure

A mobile node, attached to some other than its home link, obtains a care-of address that belongs to the particular foreign link. The association between the home address and the care-of address is maintained by the Mobile IPv6 protocol. A mobile node maintains at least a "home registration", i.e. the home agent of the mobile node has the binding between the home address and the care-of address of the mobile node. Packets destined to the home address are captured by the home agent and further tunneled to the mobile node using its care-of address.

If both communicating hosts support Mobile IPv6, the mobile node may start a route optimization procedure. During the procedure, the peer learns the binding between the home address and the care-of address. Further packets from the peer node use the care-of address of the mobile node as the destination address.

Home registration and route optimization require signaling each time a mobile node moves and changes the care-of address as well as when the bindings are about to expire. All bindings have lifetimes which depend on the lifetimes of the home address and the care-of address. A return routability procedure is required before a binding update message can be sent. This procedure provides a reasonable level of confidence that the mobile node is addressable at the claimed care-of and home addresses.

B. Host Identity Protocol

1) HIP – Separation of Namespaces

If you are asked a question "Who are you?" and you respond with your home street address, you are not actually answering the question. Nonetheless, the question is answered in an analogous way in the current Internet. When a host is identified, the IP address, providing the topological location of a node in the Internet, is given as the answer.

In real life, if you have to prove your identity and the asking person is unsure, you show your ID-card. Respectively, if you are asked to give your address, you will give the street address providing your (home) location. Using this analogy in the Internet, the host identity and location information must be separated from each other. HIP provides one possible solution for decoupling the location from the identity.

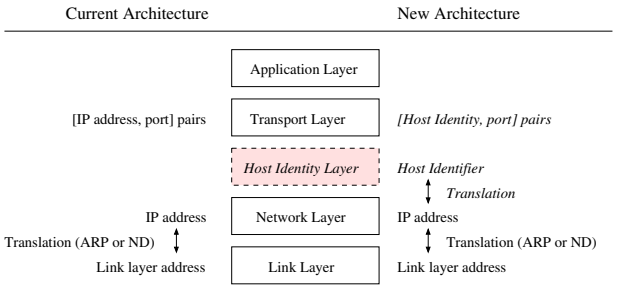

Figure 2 New Host Identity Layer

Each HIP enabled host has identities, one or more, long-term or short-term, that can be used to identify it in the network. In HIP, the identifier is the public key of a public-private key pair. When the host possesses the private key, it can prove that it actually "owns" this identity that the public key represents. It is like showing an ID-card.

Each host can generate short-term keys to be used only for a short time. These are handy when it is not necessary for the node to be identified with the same identity later. For example, buying books from a bookstore may be a long-term relationship, while once contacting a server that may collect user profiles may be considered to be a short-term action where the long-term identity is not wanted to be revealed.

The HIP Host Identity (HI), being a public key, is not practical in all actions; it is somewhat long. In HIP, the HI is represented with a 128-bit long Host Identity Tag (HIT) that is generated from the HI by hashing it. Thus, the HIT identifies a HI. Since the HIT is 128 bits long, it can be used for IPv6 applications directly as it is exactly the same length as IPv6 addresses.

When HIP is used, the upper layers, including the applications, do not see the IP address any longer. Instead, they see the HIT as the "address" of the destination host. The location information is hidden at a new layer, introduced between the Transport and Network Layers (See Figure 2). The IP addresses no longer identify the nodes; they are only used for routing the packets in the network while the HI is used as the identity. Mapping between identities and locators is done at the new layer.

2) Establishing Connection

HIP defines a base message exchange containing four messages, a four-way handshake. During the message exchange, the Diffie-Hellman procedure is used to create a session key and to establish a pair of IPsec ESP Security Association (SA) between the nodes.

Figure 3 shows the four-way handshake. The negotiating parties are named as the *Initiator* and the *Responder*. The Initiator begins the negotiation by sending an I1 packet, basically containing the HITs of the nodes participating in the negotiation. The destination HIT may also be zeroed, if the Responder's HIT is not known by the Initiator.

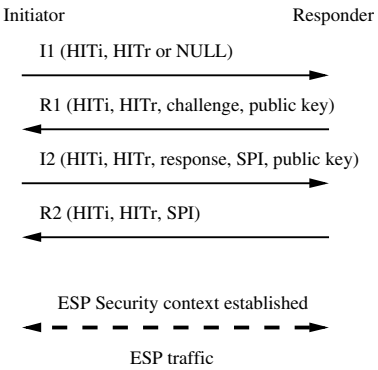

Figure 3 HIP four-way handshake

When the Responder gets the I1 packet, it sends back an R1 packet that contains a puzzle to be solved by the Initiator. The protocol is designed so that the initiator must do most of the calculation during the puzzle solving. This gives some protection against DoS attacks. The R1 initiates also the Diffie-Hellman procedure, containing the public key of the Responder together with the Diffie-Hellman parameters.

Once received the R1 packet, the Initiator solves the puzzle and sends the response cookie in an I2 packet together with an IPsec SPI value and its encrypted public key to the Responder. The Responder verifies that the puzzle has been solved, authenticates the Initiator and creates the IPsec ESP SAs. The final R2 message contains the SPI value of the Responder.

3) HIP Mobility Management

With HIP, the separation between the location and the identity information makes it clear that the packet identification and routing can be cleanly separated from each other. The host receiving a packet identifies the sender by first getting the correct key and then decrypting the packet. Thus, the IP addresses that are in the packet are irrelevant.

A HIP mobile node, moving in the network, may constantly change the point of attachment to the Internet. When the connection point is changed, also the IP address changes. As in MIPv6, this changed location information must be sent to the peer nodes. The DNS system is too slow to be used for constantly changing location information. Therefore, there must be a more stable address that can be used to contact the MN. This more stable address is the address provided by the Forwarding Agent (FA); the FA forwards the connection initialization messages to the current location of the mobile node.

The HIP Mobility and Multi-homing protocol defines a readdress (REA) packet that contains the current IP address of the mobile node. When the node changes location and IP address, it generates a REA packet, signs the packet with the private key matching to the used HI, and sends the packet to the peer node and to the FA.

When the peer node receives a REA packet, it must start an address verification process for the IP address that is included in the REA packet. The address verification is needed to avoid accepting false updates. It sends an Address Check (AC) packet to the address that was in the REA packet. When the node receives an AC that matches to the REA sent earlier, it responds with an Address Check Reply (ACR) packet. After the peer node has received the ACR packet, the address verification is completed and it can add the IP address as the location information of the mobile node.

Because the MN can move between networks using different IP address versions, the address received by the CN may also be from a different address family than the previous address.

The correspondent node may support only one IP address version. In this case, it must use a proxy node that represents it towards the mobile node, which gives the mobile node a virtual interface where the correspondent node can be reached from.

III. NETWORK SETUP

A. Mobile Nodes

The selected mobility management schemes operate on different platforms; HIP is implemented on FreeBSD and MIPv6 on Linux. There are certain differences between packet handling in the operating systems, creating some additional challenges for result comparison. Hardware platforms for both mobile nodes are identical.

Neither the mobile nodes nor the underlying network implement any handover optimization procedures, such as HMIPv6 or FMIPv6. Handover performance relies purely on the layer 3 signaling.

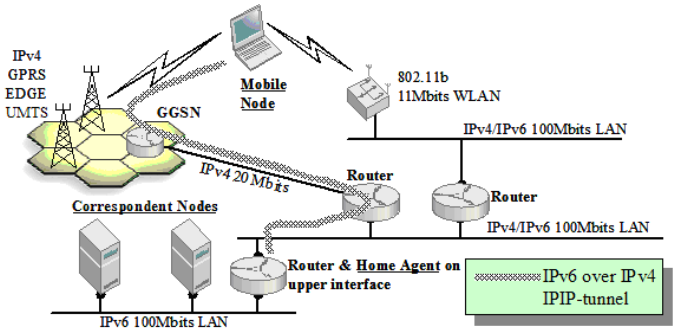

Figure 4 Test network setup

B. Network Setup

The HIP and MIPv6 handover measurements are carried out in our test network. Figure 4 illustrates an overall architecture of the combined GPRS and 11 Mbps 802.11b WLAN test network. The GPRS access is connected to the live GPRS network of TeliaSonera. Access to the test network is, though, restricted to a group of selected users. GPRS defines four forward error correction schemes [9]. In our tests, CS-2 is used as a forward error correction scheme, which will provide about 12 kbps data rate above RLC layer per timeslot. Our GPRS terminals are capable of 1 uplink and 4 downlink or 2 uplink and 3 downlink timeslots. The WLAN access is directly connected to the test network and does not contain any authentication mechanisms.

The test network has light background load, mostly normal router and other network nodes generated broadcast traffic. All

routers presented in the Figure 4 are Linux 2.4 based, except the Cisco 7200 router, which connects the test network to the GPRS core. All networking nodes in our test network have an IPv4/IPv6 dual stack.

C. Measurement Plan

The IPv6 connection over the GPRS is tunneled in an IPv6 over IPv4 IPIP-tunnel because no IPv6 capable GPRS access point is available. The tunneling is done for both the HIP and MIPv6 implementations. Tunnels are pre-configured so that no additional delay or signaling is caused due to the tunnel setup. It is also assumed that both WLAN and GPRS interfaces are active all the time. This effectively eliminates the delay caused by interface activation and network connectivity establishment. All tests will also be stationary, which allows us to avoid known GPRS cell-reselection related performance issues [1].

For background network traffic load generation we use the Iperf [8] tool for a bulk TCP flow. Our traffic profile includes a single bulk TCP flow from a server to the Mobile Node. All packets are captured from both interfaces with the Ethereal tool. From the collected packet dumps we extract the time stamps to evaluate both signaling and traffic related performance behavior.

IV. Results and Analysis

A. Measurement Notes

The measurement was performed using an automated script for both protocols. The script was run 50 times for both cases and results were collected using Ethereal.

The data was collected from the mobile node as planned to measure the mobility management performance from the end-user's point of view. Therefore, we did not make any measurements inside the mobile node for resolving processing times, neither did we make any measurements at the peer nodes. The following figures (Figure 5, Figure 6) show the times between packets leaving or arriving at the mobile node. The "processing times" shown in the figures might involve some activity besides the actual packet processing due to mentioned reason. Also the capturing of packets may have introduced some additional delays.

During measurements, it was noticed that the MIPv6 implementation had a bug in the vertical handover management. New interfaces were used for sending data before the binding acknowledgement was received from the peer node. Therefore, the TCP stream was split between two interfaces for a certain period of time. The implementation started to send TCP ACKs over the GPRS interface, while the incoming data stream still used the WLAN interface. This caused significant problems to MIPv6 signaling and affected the results.

In MIPv6 the mobile node must process 8 packets after changing the care-of address (see Figure 1) while the HIP mobile node only needs to process 6 packets (3 with the CN and 3 with the FA). However, in our measurements we did not include the FA updates because the delay on the data transfer is depending only on CN update completion. Due to this HIP performs better than MIPv6 on slow links.

B. Performance Analysis

In Figure 5 and Figure 6, we show how the average time distributes over different phases during the location updates.

The MIPv6 implementation bug caused TCP ACK sending via the GPRS access network after home registration had completed. The incoming data flow was near 5 Mbps, thus generating a fair amount of TCP ACKs that filled up the 12 kbps uplink of GPRS. This caused dropping and re-sending of MIPv6 signaling packets (CoTI, HoTI), resulting in worse average handover performance compared to a standard-conforming implementation. The HIP implementation, instead, used the new interface only after the signaling had completed and data from the peer node started to flow in using the new access network.

The results show that the average delay from the beginning of the handover until the recover of the TCP stream was 8.05 seconds for Mobile IPv6 and 2.46 seconds for HIP.

The delay for Mobile IPv6 consists of home registration, home registration processing, Return Routability test and binding update (see Figure 5).

The delay variance in the MIPv6 (Figure 7) case was mainly caused retransmission of MIPv6 signaling packets. The reason for this behavior was the bug in the MIPv6 implementation.

With HIP, the delay (Figure 8) consists of REA – AC time, AC processing and ACR – data time. (See Figure 6)

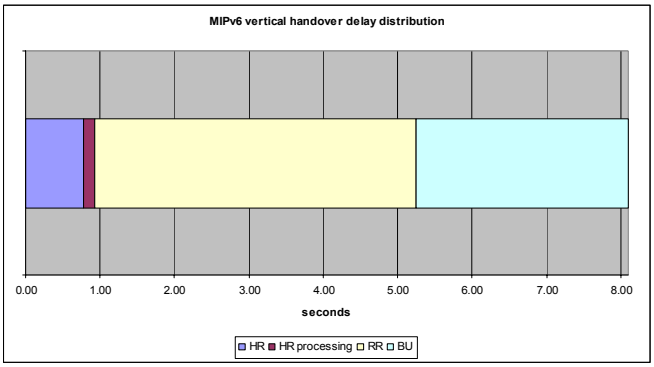

Figure 5 MIPv6 vertical handover delay distribution

V. Conclusions and Future Work

In theory, MIPv6 handover performance should be close to that of HIP (see Figure 9), being close to two round-trips from MN to CN. MIPv6 specification lets MN start home registration and RR simultaneously, but the implementation used here waited for the home registration to finish, before starting the RR procedure. Even with an optimal implementation the standard MIPv6 protocol involves more signaling packets than HIP. Although the number of round-trips would be the same, more processing at the network nodes would be required.

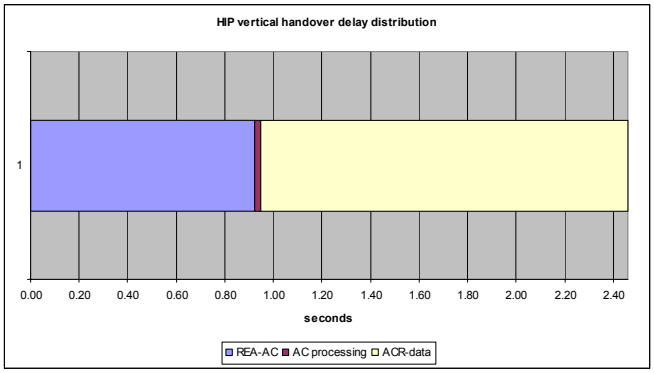

Figure 6 HIP vertical handover delay distribution

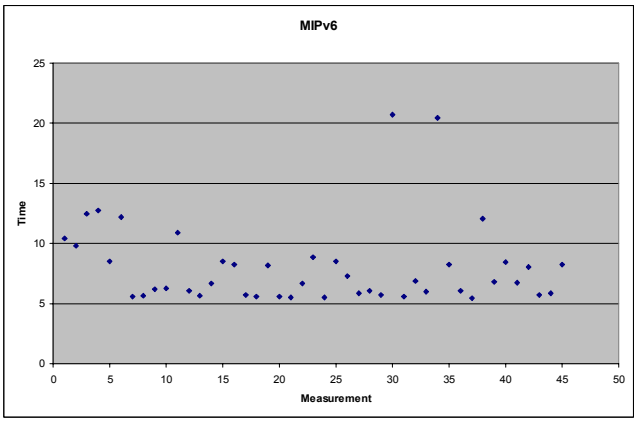

Figure 7 MIPv6 handover delays

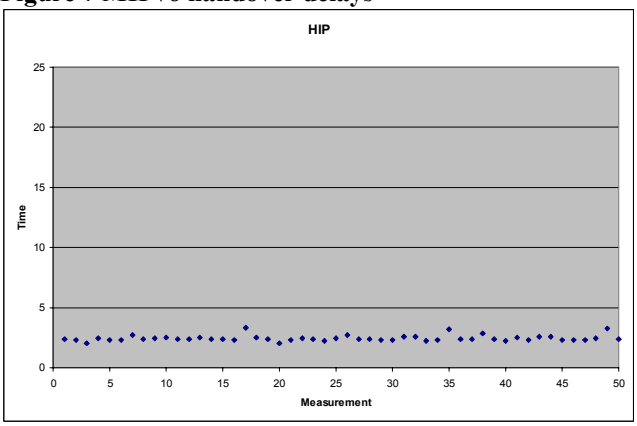

Figure 8 HIP handover delays

There is on going work in the IETF to optimize MIPv6 signaling and to minimize the amount of service disruption during layer 3 handover. Preconfigured binding management keys [6] could be used to replace the RR procedure for peer nodes that the MN communicates often with. This would reduce the handover delay roughly by half, as seen from Figure 5. Another proposal is to use credit-based authorization for early binding updates [7], where the new care-of address can be restrictedly used already before the RR procedure is finished. When running RR parallel to using the new care-of address for user traffic also roughly halves the delay caused by the handover. Another worked-on optimization is simultaneous bindings [10], where traffic gets n-cast for a short period to one or more locations where the MN in is expected to move. This should minimize the packet loss during the layer 3 handover.

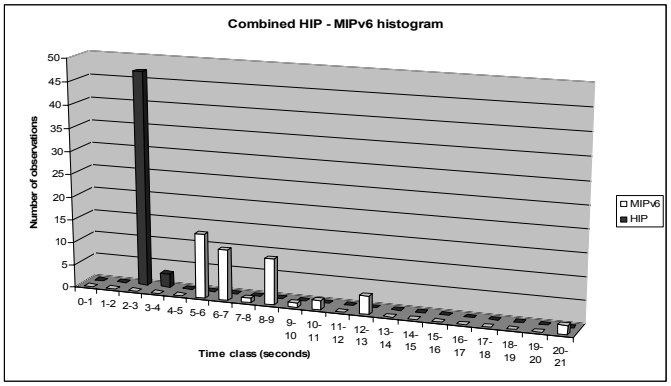

Figure 9 Comparison of delays

These optimization proposals are currently being worked on and a future work item for us will be to evaluate the performance of these and other future optimizations. Also comparing different implementations of basic MIPv6 could be useful to see how the implementation design affects the performance in different heterogeneous environments.

As a part of the future work we plan to run similar handover measurements using UMTS and IPv6 GPRS networks. The UMTS network would still be IPv4 based. Compared to the current GPRS network capabilities, throughput provided by UMTS is notably higher as well as latencies are smaller. IPv6 GPRS will allow us to discard the IPv6 over IPv4 tunneling and use native IPv6 traffic over the GPRS. We also plan to isolate and measure computational delay sources in both HIP and MIPv6 networking stacks. This will give us a better understanding how these two protocols behave in heterogeneous environments and where the real bottlenecks are located.

[1] C. Perkins, IP Mobility Support for IPv4. RFC 3344, 2002.

[2] S. Deering, R. Hinden, Internet Protocol, Version 6 (IPv6), RFC 2460, 1998.

[3] D. Johnson, C. Perkins, J. Arkko, Mobility Support in IPv6. RFC 3775, 2004.

[4] H. Soliman, C. Castelluccia, K. El-Malki, Hierarchical Mobile IPv6 mobility management (HMIPv6). draft-ietf-mipshop-hmipv6-01.txt, work in progress.

[5] R. Koodli, Fast Handovers for Mobile IPv6. draft-ietf-mipshop-fast-mipv6-01.txt, work in progress.

[6] C. Perkins, Preconfigured Binding Management Keys for Mobile IPv6. draft-ietf-mip6-precfgKbm-00.txt, work in progress.

[7] C. Vogt, J. Arkko, R. Bless, M. Doll, T. Kuefner, Credit-Based Authorization for Mobile IPv6 Early Binding Updates. draft-vogt-mipv6-credit-based-authorization-00, work in progress.

[8] Iperf Version 1.7.0, http://dast.nlanr.net/Projects/Iperf/.

[9] 3GPP, Radio Access Network; Channel Coding (Release 6), TS 45.003 v6.3.0 (2004-04).

[10] Malki K. E., Soliman H, Simultaneous Bindings for Mobile IPv6 Fast Handovers, draft-elmalki-mobileip-bicasting-v6-05.txt, work on progress.

[11] Moskowitz R., Nikander P., Jokela P., Henderson T., Host Identity Protocol, draft-ietf-hip-base-00.txt, work on progress.

[12] A. Gurtov, M. Passoja, O. Aalto, M. Raitola, Multi-Layer Protocol Tracing in a GPRS Network, In Proc. of IEEE Vehicular Technology Conference (VTC'02), September 2002.