Evaluation of the Fast Handover Implementation for Mobile IPv6 in a Real Testbed

Albert Cabellos-Aparicio*, Jose Núñez-Martínez, Hector Julian-Bertomeu, Loránd Jakab, René Serral-Gracià, and Jordi Domingo-Pascual

Universitat Politècnica de Catalunya,
Departament d'Arquitectura de Computadors, Spain
{acabello,jnunyez,bertomeu,ljakab,rserral,jordid}@ac.upc.edu

Abstract. Fast Handovers is an enhancement to the Mobile IPv6 protocol, currently specified in an IETF draft, which reduces the handover latency. This can be beneficial to real-time applications. This paper presents a novel implementation of Fast Handovers and an analysis of the handover. Using a real testbed we study the handover latency and the provided QoS: analyzing the OWD, IPDV and Packet Loss before and after the handover. Finally we present a comparison between the Mobile IPv6 and the Fast Handovers handover.

1 Introduction

In the past years Wireless LAN (IEEE 802.11) [1] has evolved and become cheaper considerably. A great interest exists among users in being on-line without wires. In current Internet status, a user can be connected through a wireless link, but he cannot move. That's why IETF designed Mobile IP. This protocol, jointly with WLAN is able to provide mobility to the Internet. In other words, a wireless user with Mobile IP can move from one point of attachment to another without losing the network connections. That's because it will have a fixed IP address that will not change regardless of the location. The most critical part of this technology (WLAN + Mobile IP) is the handover. During this phase, the mobile node (MN) is not able to send or receive data, and some packets may be lost or delayed due to intermediate buffers. This is often unacceptable for real-time or streaming applications (i.e. VoIP).

According to the measurements performed in [2], the WLAN/IPv6/Mobile IPv6 handover takes about 2 seconds. This time is unacceptable for VoIP traffic. The IETF "MIPv6 Signaling and Handoff Optimization" working group has designed Fast Handovers for Mobile IPv6 (FMIPv6) [3] in order to speed it up. Fast Handovers' main goal is to reduce both the handover latency (the duration of the handover) and the packet losses to zero.

This paper presents a novel and unique implementation of Fast Handovers. Our implementation runs on Linux and, as far as we know, is the first public

* This work was partially funded by the MCyT (Spanish Ministry of Science and Technology) under contract FEDER-TIC2002-04531-C04-02 and the CIRIT (Catalan Research Council) under contract 2001-SGR00226

T. Magedanz, E.R.M. Madeira, and P. Dini (Eds.): IPOM 2005, LNCS 3751, pp. 181–190, 2005.

implementation of the FMIPv6 protocol [17]. Our goal is to study the FMIPv6 handover in a real testbed using passive and active measurements. We aim to study the handover latency and the provided QoS level analyzing important parameters such as One-Way-Delay, Inter Packet Delay Variation and Packet Loss. We apply the methodology explained in [2] to evaluate the performance of the protocol. We also compare the QoS parameters after and before the handover, we study the differences between the Mobile IPv6 and the FMIPv6 handover and finally, we evaluate if, as stated in [3] FMIPv6 is suitable for VoIP.

Several papers focus on the handover measurement, [4] studies the handovers of different mobility protocols using a simulator, [5] studies the handover latency without taking into account the wireless handover through a mathematical model, [6] studies the FMIPv6 performance through a simulator, [7] proposes a new algorithm to improve the handover latency of the WLAN/Mobile IPv6 handover and finally, [8] makes an empirical analysis in the 802.11 handover. Our paper goes further, analyzing a real implementation of the protocol in a testbed, studying the overall performance of the protocol, especially during the handover and comparing it with Mobile IPv6.

2 Wireless and Mobility Protocols Overview

2.1 IEEE 802.11

The Wireless LAN protocol [1] is based on a cellular architecture, where each cell is managed by a Base Station (BS, commonly known as Access Point or AP). Such a cell with the BS and the stations (STA) is called a Basic Service Set (BSS) and can be connected via a backbone (called Distribution System or DS) to other cells, forming an Extended Service Set (ESS). All these elements together are one single layer 2 entity from the upper OSI layers' point of view. APs announce their presence using periodic "Beacon Frames" containing synchronization information. If a STA desires to join a cell, it can use passive scanning, where it waits to receive a "Beacon Frame" or active scanning, when it sends "Probe Request" frames and receives a "Probe Response" frame from all available APs. Scanning is followed by the Authentication Process and if that is successful, the Association Process. Only after this phase is complete the STA capable of sending and receiving data frames. STAs are capable of roaming, i.e. moving from one cell to another without loosing connectivity but the standard does not define how it should be performed, it only provides the basic tools for that: active/passive scanning, re-authentication and re-association.

2.2 Mobile IPv6

Mobile IP was designed in two versions, Mobile IPv4 [9] and Mobile IPv6 (MIPv6) [10]. The protocol's main goal is to allow MNs to change its point of attachment to the Internet while maintaining its network connections. In other words, the mobile node has a special IP address (Home Address or HAd) that will remain unchanged regardless of the MN's location, moreover, the MN will use temporary IP Addresses (Care-of-Address or CoA) when connected to foreign networks (not its home network), however, it is still reachable through its HAd (using tunnels or with special options in the IPv6 header). A special entity (Home Agent or HA) manages MN's localization by binding the MN's CoA to MN's HAd.

MIPv6 has three functional entities: the Mobile Node (MN) which is any mobile device with a wireless card and the MIPv6 protocol, the Home Agent (HA) which manages MN's localization and finally the Correspondent Node (CN), a fixed or mobile node that exchanges data packets with the MN.

The protocol has four phases. Initially in the Agent Discovery phase the MN has to discover if it is connected to its home network or to a foreign one. IPv6 routers send periodically "Router Advertisements" including network prefix information. The MN will listen to those messages discovering at which network it is attached and will obtain a CoA if it is not in the home network. Next, in the Registration phase, the MN must register its CoA (where it is located) to the HA and CNs in order that they can bind it with the HAd. After this phase, Registration and Tunneling comes, the MN establishes tunnels (if necessary) with the HA and CNs in order to send or receive data packets. Notice that the CNs will still send packets to the same destination IP address (the HAd). The last phase is the Handover, the MN changes its point of attachment and it must discover in which network it is connected once again (Agent Discovery) and register its new CoA (Registration). During this phase some data packets can be lost or delayed due to incorrect MN location.

2.3 Fast Handovers

FMIPv6 is a MIPv6 handover enhancement that reduces the handover latency and stores packets delaying them instead of losing them. This is accomplished by allowing the MN to send packets as soon as it detects a new subnet link (IEEE 802.11 in our case) and delivering packets to the MN as soon as its attachment is detected by the new access router.

FMIPv6 has different operational procedures, for instance, in the "Predictive Handover" the MN discovers nearby APs using the IEEE 802.11 "scan" and then requesting all the important information related to the corresponding new access router. When attachment to an AP takes place, the MN knows the corresponding new router's coordinates including its prefix, IP address and MAC address. Through special "Fast Binding Update" and "Fast Binding Acknowledgment" messages the MN is able to formulate a prospective new CoA (without changing its point of attachment), this CoA must be accepted by the new access router prior to the MN movement. Once the MN has changed its point of attachment and it is connected to the new access router link, it can use its new CoA without having to discover the subnet prefix, it also knows the new access router MAC and IPv6 address, and hence this latency is eliminated. As soon as it is attached the MN sends a "Fast Neighbor Advertisement" announcing its presence. Moreover, the previous access router will tunnel and forward packets to the new care of address until the MN sends a "Binding Update" registering its new CoA to HA and CNs, hence, any packet is lost. The other FMIPv6 operational procedure is the "Reactive Handover" which is very similar to the previous one, however this is not supported by our implementation.

3 Fast Handovers Implementation

3.1 Overview

Our FMIPv6 implementation is written in C and runs on Linux Kernel 2.4.26, it enhances the Mobile IPv6 MIPL 1.1 [11] implementation and complies with the draft–ietf–mipshop–fast–mipv6-03.txt. The basics parts of the draft are implemented, some optional and error recovery parts are under development as future work. However the non-implemented parts do not affect the performance of the protocol, which is the paper's main goal. Our implementation also supports any wireless card (with Linux support) through the "Wireless Tools for Linux" [12].

3.2 Implementation Structure

This section describes the FMIPv6 implementation structure which is mainly divided into two modules:

– *fh–base:* This is a "dumb" module that runs into the kernel and interacts with the IPv6 module, the MIPL module and Netfilter. It receives commands from the user space.
– *fh–daemon:* This is a user-space daemon, interacts with the user, the wireless interface (through netlink) and actually implements the FMIPv6 protocol. It communicates with the "dumb" fh–base kernel module to perform the protocol operational procedures.

We have splitted the implementation into two parts, user–space and kernel–space. The FMIPv6 protocol is an ongoing work and, we can easily adapt the fh–daemon (running on user-space) without having to change the kernel part (the most difficult one).

3.3 Development Environment

Developing support for a new protocol for the Linux Kernel is not an easy task, especially if it has to interact with other modules. In order to have a productive development environment we used User-Mode-Linux (UML) [13]. UML provides a virtual machine that emulates a Linux Box. We recreated our real testbed using UML on a single physical machine, all the virtual machines had the same configuration than the real ones, we used the same network topology, the same kernel and software versions. IEEE 802.11 is not supported by UML, however we emulated the handover using IEEE 802.1 and we simulated movement between two switches. The IEEE 802.11 part of the implementation was only tested in the real testbed. With this development environment we were able to intensively test our implementation in an easy and affordable way. Only after the implementation was mature enough, we moved it to the actual testbed to test it and to measure the FMIPv6 handover.

4 Measurement Scenario

The testbed's main goal is to test the FMIPv6 implementation in a real scenario, evaluate its handover latency using passive measurements and measure the important QoS parameters using active measurements. The testbed is shown in Figure 1, all the machines are synchronized using NTP (Network Time Protocol) obtaining 1ms accuracy. See [2] for further details.

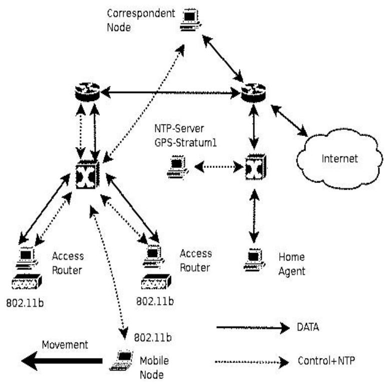

Fig. 1. Simplified measurement scenario

All the machines belonging to the testbed are using the GNU/Linux Debian Sid distribution, however the hardware depends on the role of each computer. Both access routers have two wireless cards, one for communicating with the MN and the other one to capture frames (passive measurements). Those cards have the Atheros Chipset (802.11b). The running kernel is 2.4.26 patched with the FMIPv6 implementation. The MN uses a wireless Cisco Aironet 350 card, the running kernel is also 2.4.26 patched with MIPL 1.1 and with the FMIPv6 implementation. Finally the HA and the CN use the 2.4.26 kernel patched with the MIPv6 software (there is no need of FMIPv6).

5 Methodology

We will apply the methodology depicted in [2] to evaluate the FMIPv6 handover.

5.1 Passive Measurements

The handover latency is the time spent during the handover. To compute it we developed a special tool "PHM" that monitors the signaling messages in both APs of our testbed. We capture all the packets sent or received by the wireless interface using Ethereal. The handovers are forced using special user-space wireless utilities for Linux [12]. When the MN has received the "Fast Binding

Acknowledgment" message it is ready to move to the new access router. At that point we force the wireless card to change from the old AP to the new one. Our FMIPv6 behaves as stated in [14]. As soon as our implementation detects the new link (using [12] once again) we send the "Fast Neighbor Advertisment" to announce the MN presence. Once the handover is finished and having the frames captured by Ethereal, PHM processes the signaling messages off-line providing the computation of the handover latency. Moreover PHM is able to differentiate between the different parts of the handover latency (Scanning, Authentication and Association for 802.11b). In fact, PHM is easily extensible to other mobility protocols and is able to compute the handover latency also for MIPv6.

5.2 Active Measurements

Using active measurements we intend to analyze the provided QoS at IP level. The basis of such tests is to generate a synthetic flow traveling through the network under test. The developed application to make such measurements is NetMeter [15] and we apply the methodology presented in [2] to perform the active measurements.

5.3 Evaluation of the FMIPv6 Implementation

For a good analysis of the handover, it is necessary to build up a good set of tests. In this paper we ran a set of 10 tests each 5 minutes long, from where we extracted a set of 40 valid handovers.

In order to evaluate the protocol and our implementation we used two different packet rates and sizes. Half of the tests had 64kbps traffic. This flow simulates with UDP the properties of VoIP traffic under IPv6. It sends 34 packets per second with 252 bytes of payload as stated in [16]. Due to the low rate needed for VoIP the other tests are done on a higher packet rate, so the impact of a different bandwidth can be studied. This flow (Data) sends 84 packets per second with a payload size of 762 bytes per packet. The paper's main goal is to analyze our FMIPv6 implementation, check if it works as expected and provide performance results, especially regarding its handover latency and the QoS parameters. To test our implementation under stress conditions requires having multiple MNs and APs which is very difficult to deploy in a real testbed. These kind of tests are left as future work and will be done using the UML infrastructure.

All the tests are from the CN to the MN. With FMIPv6, when the packets flow in this direction, the access routers must tunnel and buffer packets showing an interesting behavior. However, when the traffic source is the MN, there is no need to tunnel packets, just to buffer them on the MN (the FMIPv6 handover latency remains constant for both directions), that's why we focus on the CN–>MN direction.

6 Results

This section describes the results obtained from the tests discussed in the previous section.

6.1 Handover Latency

Figure 2 shows an instantaneous One-Way-Delay (obtained with NetMeter) where it is easy to see the handover. We can see that no packet is lost; regarding the delay we see a spike. This behavior is due to FMIPv6, while the MN is changing its point of attachment (from one AP to the other) the old access router is tunneling and forwarding packets to the new access router and the new access router, at the same time, it is buffering packets until the MN regains connectivity. So, FMIPv6 delays (buffers) packets instead of loosing them. The packets will be stored in a buffer while the MN's WLAN layer is disconnected; hence, this delay is equal to the 802.11 handover latency (see the numerical results).

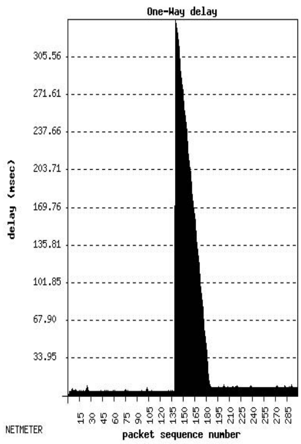

Fig. 2. FMIPv6 handover, instantaneous OWD (VoIP Traffic)

Once the handover is finished we can see that the delay is slightly higher than before, that's because the packets are being routed to the old access router and tunneled to the new access router, introducing an extra hop. When the MN sends a "Binding Update" to its HA and CNs the traffic will be routed directly to the MN. However the MIPL Mobile IPv6 implementation does not support this enhancement and it is not implemented. Table 1 shows the results obtained with our PHM application and are the numerical results of the FMIPv6 handover latency (results are in milliseconds).

Table 1. FMIPv6 handover latency (ms)

	Mean	Std.Dev.
VoIP Traffic	319.05	25.67
Data Traffic	330.34	29.22

The PHM tool shows that the FMIPv6 handover latency is equal to the IEEE 802.11 handover latency (as expected) computed in [2] and [5]. The rate and the packet size do not affect the handover latency.

6.2 QoS Parameters Analysis

Table 2 summarizes all the results obtained with NetMeter regarding the provided QoS level of the FMIPv6 handover. This is accomplished by taking 100 packets before the handover and calculating the OWD, the IPDV and the same after it.

Table 2. FMIPv6 One-Way-Delay and Inter Packet Delay Variation (ms)

		OWD (ms)		**IPDV** (ms)	
		Before	After	Before	After
VoIP Traffic	Mean	2.77	6.17	16.68	31.4
	Std.Dev.	1.40	4.52	16.4	37.0
Data Traffic	Mean	7.27	17.3	16.54	63.9
	Std.Dev.	2.73	17.1	22.41	65.0

These numerical results confirm that the delay is slightly higher after the handover due to the extra hop. They also show that the OWD remains constant before and after the handover for VoIP traffic. For longer packets (762 bytes) the OWD has variance after the handover (17ms of IPDV after the handover). [2] shows important QoS fluctuations in the Mobile IPv6 handover due to the wireless card. In a MIPv6 handover the wireless card decides to switch to a new access point regardless of the above layers, it changes its point of attachment when it detects a signal degradation [1], hence, the provided QoS is severely affected, especially for longer packets. In FMIPv6 the wireless card is forced (by the above layers) to switch from one AP to another one without having to wait until the signal degrades. The FMIPv6 OWD variance for long packets after the handover may be due to implementations issues, packets must be tunneled and forwarded, not just forwarded. However, the results provided in [2] shows that MIPv6 suffers from a higher variance than FMIPv6.

6.3 Fast Handovers vs. Mobile IPv6

Figure 3 shows a Mobile IPv6 handover [2] where we can clearly see the gap produced by the handover.

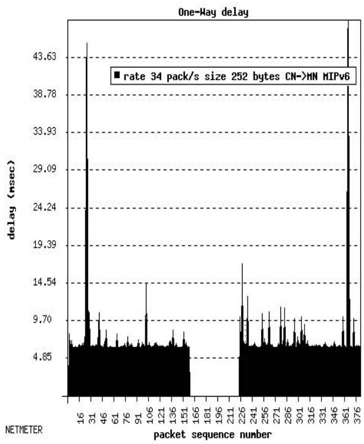

Fig. 3. MIPv6 handover, instantaneous OWD

During this period of time, the MN is not able to send or receive data, thus the packets are lost, while the FMIPv6 implementation does not lose any packet. Regarding the OWD/IPDV before and after the handover, as explained above, Mobile IPv6 suffers from a higher IPDV before the handover due to the decrease of the signal strength (especially for long packets). Finally, regarding the handover latency (the time for interruption), Mobile IPv6 has approximately 2 seconds [2] while in FMIPv6 is about 325ms. This handover latency produces packet losses in Mobile IPv6 that may be computed as the rate multiplied by the handover latency.

7 Conclusions

This paper presents a novel Fast Handovers implementation and analyzes through active and passive measurements the protocol handover in a real testbed. The analysis focuses on the handover latency and the level of provided QoS (OWD, IPDV and PL). Finally it compares the performance obtained between Mobile IPv6 and Fast Handovers in a real testbed.

The results obtained through passive measurements show that the FMIPv6 handover latency is equal to the WLAN handover latency, therefore, FMIPv6 reduces the IPv6 and MIPv6 handover latency to zero and is as fast as the WLAN handover. Active measurements show that, in FMIPv6 there is a light QoS degradation after the handover for long packets, whereas in MIPv6 the WLAN signal strength degrades and there is a severe OWD variance. Moreover, while MIPv6 loses packets, FMIPv6 delays them. In the worst case a packet is delayed a 'WLAN handover latency' (about 325ms) which is often acceptable for VoIP traffic.

The FMIPv6 protocol and our implementation achieve the expected goals. In [17] are the FMIPv6 implementation NetMeter and PHM Tool available under the GPL license. Also all the detailed results and several figures are available.

References

1. IEEE 802.11: Wireless LAN Medium Access Control and Physical Layer. (1997)
2. A. Cabellos-Aparicio, R. Serral-Gracià, L. Jakab, J. Domingo-Pascual: Measurement based analysis of the handover in a WLAN MIPv6 scenario Passive and Active Measurements 2005, Springer, LNCS 3431, pp 203-214
3. R. Kookl: Fast Handovers for Mobile IPv6 draft-ietf-mipshop-fast-mipv6-03.txt
4. N. Montavont and T. Noel: Handover Management for Mobile Nodes in IPv6 IEEE Communications Magazine (2002)
5. A. Mishra, M. Shin and W. Arbaugh: An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process Vol.33 ACM SIGCOMM Comp. Comm. Review
6. Marc Torrent-Moreno, Xavier Pérez-Costa, Sebastià Sallent-Ribes: A Performance Study of Fast Handovers for Mobile IPv6 28th IEEE Inter. Conf. on Comp. Networks
7. Marco Liebesch, Xavier Pérez Costa and Ralph Scmitz: A MIPv6, FMIPv6 and HMIPv6 Handover Latency study: Analytical Approach IST Mobile and Wireless Telecommunications Summit (2002)
8. Xavier Pérez Costa and Hannes Hartenstein: A simulation study on the performance of Mobile IPv6 in a WLAN-based cellular network Volume 40 issue I Computer Networks: International Journal of Telecommunications Networking (2002)
9. C. Perkins: IP Mobility Support for IPv4 RFC 3344 (2002)
10. D. Johnson, C. Perkins and J. Arkko: IP Mobility Support for IPv6 RFC 3775
11. HUT: MIPL Mobile IPv6 for Linux (online) *http://www.mobile-ipv6.org/* (2004)
12. Wireless Tools for Linux (online) http://www.hpl.hp.com/Jean_Tourrilhes/Linux/Tools.html
13. Jeff Dike: UML – User-Mode-Linux (online) *http://user-mode-linux.sourceforge.net*
14. P. McCann: Mobile IPv6 Fast Handovers for 802.11 Networks draft-ietf-mipshop-80211fh-04.txt
15. R. Serral-Gracià, A. Cabellos-Aparicio, H. Julian-Bertomeu, J. Domingo-Pascual: Active measurement tool for the EuQoS Project. IPS-MOME 2005 Workshop.
16. John Q. Walker, NetIQ Corp: A Handbook for Successful VoIP Deployment Network Testing, QoS and More (2002)
17. SAM - Advanced Mobile Services - Research Project MCyT (online) *http://sam.ccaba.upc.edu*