An Experimental Performance Evaluation of the IETF FMIPv6 Protocol over IEEE 802.11 WLANs

Emil Ivov
Network Research Team
Louis Pasteur University
Strasbourg, France
Email: Ivov@dpt-info.u-strasbg.fr

Thomas Noel
Network Research Team
Louis Pasteur University
Strasbourg, France
Email: Thomas.Noel@dpt-info.u-strasbg.fr

Abstract— **Straightforward, transparent mobility management has now been available for some time with IPv6 through the Mobile IPv6 protocol. There are numerous MIPv6 implementations available and some have been successfully deployed and tested in real world scenarios. Yet, certain aspects of that mobility support, such as security, seamless handovers, and heterogeneous network technologies support are still being discussed and improved by working groups on the IETF and in various research communities. One such example is the continuing work on the "Fast Handovers For Mobile IPv6" protocol that aims to reduce the packet loss and latency inherent to the handover process. In this document we present a set of experiments with an implementation of that protocol and the resulting performance evaluation. All tests take place over IEEE Wireless LANs, and are performed using the fmipv6.org protocol implementation for the Linux operating system, that we have developed and contributed to the open source community. The paper, exposes an analysis of the provided results and brings out some issues, not currently addressed by the protocol, like for example connection and packet loss occurring during the IEEE 802.11 scanning procedure, and lack of more appropriate alternative mechanisms for discovery and selection of candidate access points. The purpose of the document is to provide evaluation material based on a link layer protocol that is (to our understanding) among the primary link layer protocols that FMIPv6 was built to work on. We find such an evaluation quite necessary for the pending optimisations of the protocol.**

I. Introduction

Massive proliferation of Internet connected mobile devices such as Laptops, PDAs and lately even mobile phones on one hand, and increasing accessibility and popularity of real-time services such as IP telephony and video conferencing on the other have turned IP mobility into a very hot topic. Users are now often able to get on-line while on the move and protocols like Mobile IPv6 [JPA04] are trying to aid them to continue using their network access uninterrupted while physically traversing different subnets and experiencing attachment point and network address changes.

Through the use of a static Home Agent (HA) entity and a Home Addresses (HoA) for MIPv6 enabled hosts, Mobile IPv6 masks node movement for correspondents. All packets sent to a mobile host (i.e. to its HoA) are intercepted by the HA and forwarded to the actual/current location of the node.

When, however, a Mobile Node (MN) reaches the physical border of its current wireless subnet and enters an area belonging to a new (different) one, it is forced to go through a sequence of procedures such as, link layer scanning for candidate access points, Stateless Address Autoconfiguration [TN98], Duplicate Address Detection [NNS98], and MIPv6 Binding Update, before it is able to actually regain connectivity. This sequence is often referred to as handover, and it generally involves connection disruption and considerable packet loss.

Handovers may involve different network technologies, both wired and wireless. A node may even perform a handover without being forced to do so (i. e. for a reason other than connection loss), in case a new link, offering better performance characteristics, has become available or because the network has requested it to do so for load balancing reasons. In the rest of this document we will mainly be considering handovers where both the previous and new networks are IEEE 802.11 [IEE99] Wireless LANs.

The "Fast Handovers For Mobile IPv6" (FMIPv6) [Koo05] protocol was designed with the goal to bring to a minimum the duration of the handover, also known as handover latency and assist an MN to rapidly recover communications.

In this document we provide a performance evaluation of the FMIPv6 protocol over IEEE 802.11 networks, using an implementation that we had previously developed and contributed to the open source community through fmipv6.org [IA05]. We try to expose the protocol's strengths and weaknesses as inferred from that evaluation so that implementors and protocol designers may aptly address them.

The rest of this paper is organized as follows. Section II briefly presents technologies inherent to FMIPv6 handovers, such as WLAN 802.11, Mobile IPv6 and FMIPv6 itself. In section III we give a summary of existing evaluations of the FMIPv6 protocol and show the need of a real, practical evaluation. Section IV goes through a description of the tesbed that we used for our experiments. In section V we present and analyse the results of our empirical studies and section VI concludes the article with a summary of results and potential next steps.

II. Technological Background

The FMIPv6 protocol was designed to work in conjunction with existing technologies that strongly influence its semantics.

1-4244-0270-0/06/$20.00 (c)2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2006 proceedings.

We therefore think that a short presentation of this technological background is due.

A. The Wireless LAN Handover

The IEEE 802.11 standard defines two major network topologies and modes of operation for wireless devices:

Infrastructure: Wireless devices are connected to a central entity called an Access Point (AP). Nodes communicate only with their corresponding AP and do not exchange messages directly.

Ad hoc: In this mode there is no central entity and nodes exchange messages directly.

During the rest of the document we will be concentrating on the infrastructure mode.

When a WLAN device needs to connect (associate) to an AP (either after power up, sleep mode, or simply upon entering an area covered by a new AP), it would first need to discover nearby APs, then select one and attach to it. To find out what APs are available in the region a node may either passively listen for Beacon Frames broadcasted by APs (passive scanning) or send Probe Request frames and wait for incoming Probe Response-s from APs. During this stage a node generally discovers all APs that it could potentially attach to as well as some key link layer characterstics such as their corresponding frequencies and ESSIDs.

Once the wireless device has determined which AP best responds to its selection criteria, it will go through the Authentication Process, which is the exchange of information between the AP and the station, where each side proves the knowledge of a shared secret.

When the station is authenticated, it will start the Association Process, which is the exchange of information about the stations and AP capabilities. Only after the association process is completed, a station is capable of transmitting and receiving data frames.

B. MIPv6 Basics

Mobile IPv6 [JPA04] allows devices to remain reachable while moving within the Internet topology.

When a node using MIPv6 moves to a foreign link, it creates a Care-of Address (CoA) topologically valid for its new location, and informs a preconfigured Home Agent (HA) for its movement by sending a message known as a Binding Update (BU). The HA creates a tunnel to the MN's new location, replies with a Binding Acknowledgement message (BA) and start redirecting packets bound for the MN's home address over that tunnel.

Basic MIPv6 techniques however do not address the need of a seamless handover which makes them insufficient for many real-time applications like VoIP.

C. Short FMIPv6 Presentation

The FMIPv6 [Koo05] protocol enables an MN to request information on neighboring AP's and the subnets behind them. To do this, a node sends a "Router Solicitation for Proxy Advertisement (RtSolPr)" message. This solicitation may contain the id of one or more APs (obtained from a link layer scan procedure for example), thus requesting subnet information corresponding to the AP (see Figure 1). It may also contain a wild card signifying a request for all nearby AP-AR couples (where AR stands for Access Router).

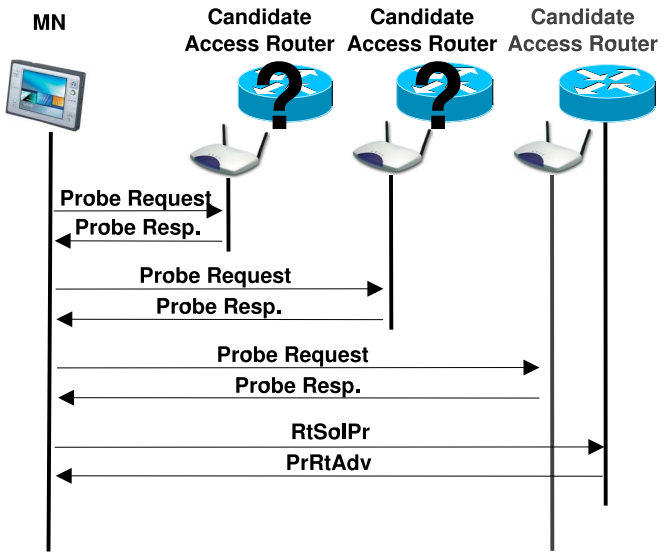

Fig. 1. FMIPv6 Protocol Operation. Candidate Access Router Discovery over IEEE 802.11.

The currently default access router responds with a "Proxy Router Advertisement (PrRtAdv)" resolving the specified AP identifier (or wild card). The information is in the form of an [AP-ID, AR-Info] tuple, where AR-Info is a set of ICMPv6 options that may be but are note limited to: AP link layer address, AR link layer address, AR subnet prefix and prefix length.

If a MN is able to detect (e.g. through the use of link layer information) the need of a handover it sends a Fast Binding Update (FBU) to it's current router (later referred to as PAR for Previous Access Router). This message contains MN's Care-of Address in PAR's network (PCoA) and the access router that the MN is planning to switch to (NAR). At that point PAR sends to NAR a Handover Initiate (HI) message containing the identity of the MN (link layer address, PCoA and, if known, desired NCoA). NAR confirms (or rejects for that matter) the handover with a Handover Acknowledge (HAck) message that may provide further NAR specific details. Once the HAck Received, PAR sends a Fast Binding Acknowledgement (FBAck) back to the MN which (in this particular case) receives it on PAR's link. The MN is then ready to actually switch links. Once on NAR's link it sends a Fast Neighbour Advertisement (FNA) message which is supposed to update respective neighbour cache entries on the NAR and completes handover signalling.

This type of handover, characterised by the fact that the FBAck is received by the MN while in PAR's network, is called by the FMIPv6 RFC [Koo05] a predictive handover (as it indicates that the MN could anticipate the procedure)

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2006 proceedings.

and is graphically presented by Figure 2.A.

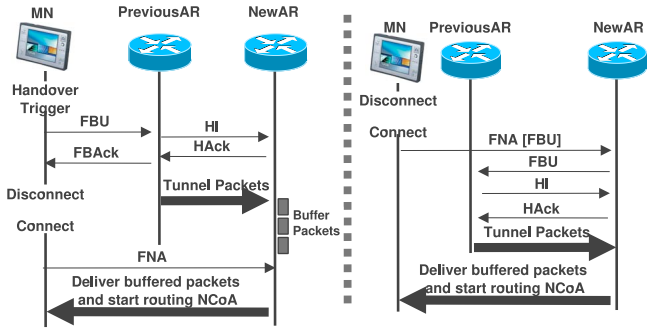

Fig. 2. FMIPv6 Protocol Operation. (A-Left) Predictive and (B-Right) Reactive modes

The FMIPv6 protocol also defines a reactive handover scenario which basically represents the case where a Mobile Node could not anticipate a handover so it was able to only react once it was already in progress (hence the name). In that case the FBU is sent from NAR's link after Layer 2 handover has completed and is usually encapsulated in the FNA. NAR then forwards that FBU to PAR, the HI/HAck message exchange follows as in the predictive case and PAR starts tunnelling packets. The reactive case is depicted by Figure 2.B.

Note that the predictive and reactive scenarios we just described though representative are not exhaustive for both modes (predictive and reactive). What officially distinguishes both types of handover and thus protocol operation is whether the FBAck was received on while the MN was still on PAR's link or once it had arrived on NAR's.

III. Related Work

To this date, numerous studies exist on the behaviour of FMIPv6. [KP01] is one of the early papers available on the protocol. Authors provide an evaluation based upon a proprietary experimental implementation and also propose a context transfer scheme (the latter being outside the scope of our current study). The paper was published a relatively long time ago (2001 - approx. 5 years) when still little was known about FMIPv6 and there were many uncertainties concerning its performance. It was therefore of considerable use and had a significant impact on following work in the IETF. Yet we find the experimentation setup somewhat unrealistic and results to be incomplete from a today's point of view. Handovers, for example are triggerred upon reception of a user issued command, and do not make use of L2 triggering or any other scheme. More importantly, experimentation results only include the length of the handover procedure but not the handover latency (the amount of time that connection was unavailable and packets were being dropped by the AR). Last but not least, there is no consideration of candidate AP discovery which we believe to be critical in the current state of the protocol (we talk about this in section V).

In [HH02] authors provide an analytical evaluation and comparison of the FMIPv6, HMIPv6 [SCMB], and MIPv6 protocols, based on default values provided by the respective RFCs and drafts. Given the analytical approach however, many of the practical aspects that we already mentioned (e.g. scanning, packet loss and etc) have been neglected or not fully taken into account.

Authors of [RH02] analyse a combination of FMIPv6 and HMIPv6 and present an evaluation with [ns2] in the case of a TCP flow. Both HMIPv6 and FMIPv6 are reported to reduce MIPv6 handover latency 7 and 15 times respectively. The paper also uncovers a problem with the superposition of the two handover optimisations that consists in packet disordering due to ARs being often closer to the HMIPv6 MAP than to each other. This problem seems to be tampering with TCP flows and thus hindering the maximum potential handover performance. We believe that the document is quite useful and provides valuable insight on properly combining both optimisations. Yet it doesn't seem to have the objective of providing detailed and realistic protocol feedback. Values for L2 handover delay for example are fixed to 20 ms (which might correspond to a certain link layer technology but this could hardly be the case of IEEE 802.11).

Another performance analysis of FMIPv6 is provided in [PC03]. Authors focus on protocol overhead, wrongful anticipation, and "eating up buffer space" in access routers, and study how these problems relate to the "sensitivity" of L2 Triggers. They show that these vary largely depending on how close in time is the link layer trigger event to the actual connection disruption. They also give and an optimal value for the time distance between this event and the disconnection (i.e. how long before the MN loses connection should it start its handover) and prove it analytically. We find, however, that this research is of little use for implementors and protocol designer since the exact moment in time that connection disruption happens is rarely (if ever) known in advance. We also believe that authors slightly overstate the problem with the protocol overhead since it is negligible compared to the data flows typical for VoIP, that are often used as a reference for handover performance evaluations.

Kempf, Wood and Fu provide in [KWF03] yet another evaluation based upon experimentation with a proprietary implementation. Experiments make use of a wired handover emulator configured for 40ms link layer handover length (i.e. connection is unavailable for 40 ms) and 10ms of packet delaying. The values were reportedly chosen to match those of 3G systems. Yet we find that this emulator adds a level of uncertainty and though useful to some extent from an analytical point of view - it (like all simulations and emulations) introduces some doubts as to the validity of the experimentation in real world deployments and usage of the protocol over wireless networks. Link layer triggers for example are (once again) configured to be sent at a predefined amount of time before the link layer handover is to occur. Access routers receive an instant link down trigger for mobile nodes belonging to their network which is not quite the case in reality

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2006 proceedings.

and especially not for IEEE 802.11 that have not been taken into consideration in this work.

IV. Testbed and Test Scenarios

The testbed that we have built and used both for the development of the implementation and for our experiments consists of three access routers a mobile node and a correspondent node (Figure 3). Two of the access routers, FMIP-AR1 and FMIP-AR2, both had a commercial AP connected on one of their wired interfaces. During the tests the mobile node FMIP-MN switched back and forth and performed handovers between those two routers. A third router FMIPNET was both serving the role of a home agent and interconnecting the rest of the testbed with the Internet. A correspondent FMIP-CN node, connected to the internet from a completely different IPv6 subnet in our campus, was used for generating flows destined to the FMIP-MN.

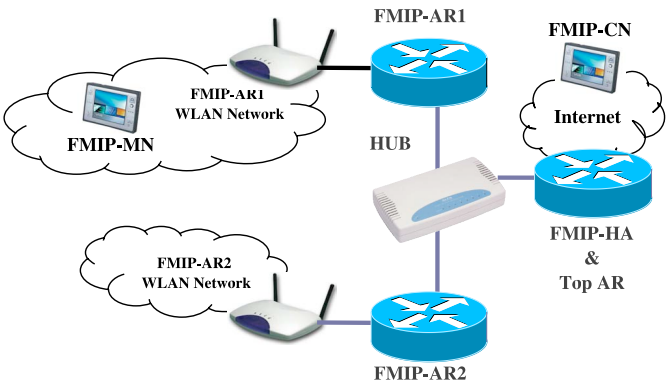

Fig. 3. The fmipv6.org Experimental Testbed

All three routers were running the Linux Operating System with the USAGI modified 2.6.8.1 kernel and the Quagga Routing Suite [QUA] (version 0.98.4). The routing protocol used both inside the testbed and on the interconnection link was RIPv3. FMIPNET was using the MIPL 2.0rc2 [mip] Home Agent implementation from the Helsinki University of Technology.

FMIP-AR1 and FMIP-AR2 were also running the fmipv6-ar router implementation from fmipv6.org [IA05].

FMIP-MN was running the Linux Operating System with the USAGI modified 2.6.8.1 kernel and was equipped with a single wireless interface using an Atheros chipset and the MADWiFi [MAD] driver. The MADWiFi version running on the MN had minor modifications that optimised its behaviour during handovers controlled by userland applications (i.e. the fmipv6 daemon) and thus allowed it to perform rapid L2 handovers (note that these modifications do not in the least deviate from standard IEEE 802.11 practices). FMIP-MN was also running the fmipv6-mn mobile node implementation from fmipv6.org [IA05].

The reason for choosing this exact configuration is because we believe it's widely spread and Wireless LAN deployments often include one or more of its components. We therefore believe that results obtained in this experimentation set are of high interest and should be considered when implementing and/or improving the protocol.

Fig 3 provides a diagram describing the testbed and the way it's deployed.

For all predictive testing, the mobile node (FMIP-MN) was beginning the experiment attached to one of the FMIPv6 routers (FMIP-AR1 or FMIP-AR2). The Tx Power on that router was then manually lowered through a Wireless Extensions [Tou] ioctl call. The fmipv6.org MN implementation uses link layer information provided by the MADWiFi driver through the Wireless Extensions package. Whenever quality dropped below a configurable threshold (which is what happened upon modification of transmission power on the AP) the implementation performed a handover and switched to the other router.

Reactive handovers were tested by turning down the wireless interface of the router, that FMIP-MN was currently attached to and thus forcing it to associate with the other one.

We are aware that these scenarios do not represent all possible FMIPv6 flows but we believe they are covering and put into use most key parts of the protocol semantics.

V. Evaluation Results and Analysis

A. Predictive Handovers

The predictive mode of operation of the FMIPv6 protocol is the one that best addresses handover issues and allows bringing connection disruption time and packet loss to levels that would satisfy most existing real time applications.

Figure 4 contain results from a (representative) experiment with one such predictive handover. Exactly 12.3 seconds after the beginning of the experiment the mobile node, by regularly scanning link layer quality, detects that it has crossed the predictive handover threshold and decides to begin a handover. It therefore sends the Fast Binding Update that we see as the first FMIPv6 message on Figure 4. In the experiment at hand, the FBU does not get immediately answered and the corresponding FBAck is only received after a retransmission of the FBU. This actually happens quite often since before sending an FBAck, the PAR needs to send a Handover Initiate message to the NAR, wait for a Handover Acknowledge and (in the case of fmipv6.org) create the forwarding tunnel so that it could be already operational once the MN has moved to NAR's network. Right after receiving the FBAck (at time 12.294), the MN starts an L2 handover which lasts for about 10 ms. After it arrives on the new link (at approximately 12.312s) it announces its arrival by broadcasting an ICMPv6 Neighbor Advertisement and sending a Fast Neighbor Advertisement to NAR. The FNA also allows routers to stop buffering packets and forward those that have been received through the tunnel prior to MN's arrival. In this case there is one such packet and it is the one immediately following the last packet received by the MN on PAR's link.

As shown on Figure 4 the MIPv6 implementation on the MN (MIPL v2.0 in our case [mip]) sends a Binding Update upon detection of MN's new address, modifies its tunnel to

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2006 proceedings.

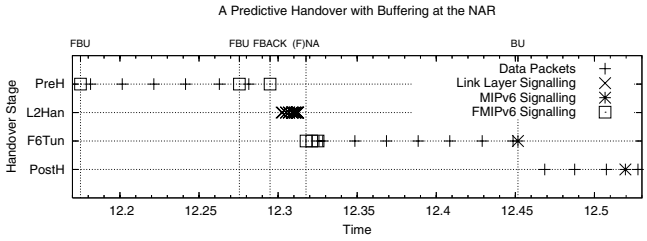

Fig. 4. Packet loss and handover latency for a predictive handover

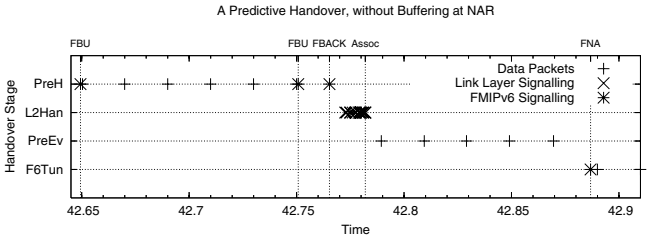

Fig. 5. Packet loss and handover latency for a predictive handover without buffering at NAR

match NAR's link and thus ends the usage of the FMIPv6 tunnel. In this experiment that has happened 133 ms after the FNA has been sent. The reason for this delay comes from the fact that in order to confirm movement, MIPL had to send a Router Solicition and wait for the corresponding advertisement, which is by the way purposefully delayed by the NAR (see [NNS98]) before sending the BU. The advantage that FMIPv6 offers in this case is the fact that an FMIPv6 implementation "knows" exactly what handover has taken place since it is the entity that has caused it and that controls it.

The time that the MN has suffered connection loss is equal to 10.42ms and there has been no packet loss as the one packet that arrived while the MN was out of reach, was buffered and later resent by the NAR.

B. Buffering Issues

An interesting phenomenon that we have observed during our predictive experiments is the fact that L2 triggered events may sometimes be delayed significantly and thus have an impact on protocol performance. The standard mechanism used for L2 trigger event delivery to userland in Linux based Operating Systems is through RTNETLINK sockets. When and under what conditions these are sent is currently a matter of driver behaviour and though there is intensive work on standardising them this is not currently the case. When we were using a Wireless LAN card managed by the HostAP [Hos] driver for example, the events received indicating L2 handover end were received more than 100ms later than the MN had actually associated with the new AP. In such cases the MN's (Fast) Neighbor Advertisement on NAR's link suffered significant delay. And in the cases where NAR was providing an insufficient amount of buffering (such as neighbour discovery's default 3 packets for example [NNS98]), packets were being dropped while they could have been received by the MN, had they been sent.

We therefore conducted a set of experiments with the host AP driver on the MN side and without any buffering being done by the NAR. One such experiment, represented on Figure 5, shows that the MN receives 5 packets before it gets notified of L2 attachment and loses 1 because of the lack of buffering on the NAR. Using buffering would have saved that one packet (provided the configuration at NAR had allowed it to store a sufficiently large number of packets) but would have delayed the other 5 by more that 110ms. It is difficult to say whether it would have been better for the MN to receive all packets saved by the buffering but delayed by the L2 trigger delay or rather get them immediately but lose one. We believe that the MN alone could determine whether short handover latency or low packet loss is more critical to it as that depends on the types of applications being run on the mobile node. Yet the FMIPv6 protocol does not provide MNs with a way to indicate their preference to routers and this is one of the issues that we'd like to resolve in future work.

C. Reactive Handovers

On figure 6 we see results from a reactive handover scenario. At a certain point of the experiment we turned down the accesspoint that FMIP-MN was associated with. The MADWiFi driver detects link loss through missed beacons. By default the number of beacons that need to be lost on a link for MADWiFi to declare it down is 7 which makes for at least a 700ms delay. We modified that number to 3 in order to achieve better performance for reactive handovers. Do notice that losing link layer connectivity generate a completely different event from the one caused by a drop in signal strength. Signal strength is measured upon received packets and if there is no AP to send packets, no signal strength would be measured and no link quality event generated. Thus there is no risk of wrongfully beginning a predictive handover and waiting for the FBU transaction to expire (700 ms with default values from [Koo05]) before initiating a reactive handover and thus losing time on a dead link.

2.289s after the beginning of the experiment or 375 ms after the last data packet received on the link the fmipv6.org daemon on the MN detected that the link was down and started an L2 handover to the access point that seemed to offer best quality during its last scan (results being stored by the FMIPv6 daemon itself). Right after it arrived on the new link the MN sent an FNA, to the NAR, followed by an FBU for the PAR.

Note that this is not really the standard protocol behaviour since the FMIPv6 [Koo05] protocol advises that FBUs in reactive handovers SHOULD be encapsulated in the FNA (as opposed to sending them separately). We didn't take this approach as it was causing problems with the MobileIPv6 stack on the router which ignored mobility header packets with the "next header" value different from NONE. From that point on, the handover continued in the manner seen in previous sections - PAR started tunnelling packets destined to

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2006 proceedings.

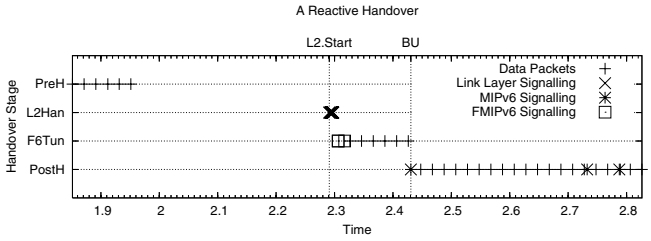

Fig. 6. Packet loss and handover latency during a reactive handover.

MN's PCoA to its NCoA and once MIPL detected the change as well, it modified the MIPv6 tunnel to point to NCoA. The handover has thus caused us a connection loss during approximately (less than) 343.53ms and caused us to lose 17 data packets.

D. Candidate Access Point Discovery

Figure 7 shows what we believe to be one of the currently most flagrant operational issues with the protocol. FMIPv6 does not provide a way for mobile nodes to discover Candidate (neighbor) Access Points. Sending a wild card RtSolPr to PAR requests a list of Access Point link layer addresses known to the AR but it would by no means tell the MN whether these APs are within its coverage or either of their link characteristics such as channel/frequency, ESSID and others. A mobile node has therefore no other choice but to perform a wireless scan and only then send an RtSolPr demanding information on the discovered packets. The length of the scanning procedure may vary from one implementation to the other but is generally considered to be the heaviest part of a Wireless LAN handover. In our testbed we performed tests with two kinds of WLAN cards - one that supported IEEE 802.11a/b/g standards and one that only worked for IEEE 802.11b/g. Those supporting only the "a" and "b" standards were by default performing the complete scan procedure over the 13 standard channels for an average of 2670ms. The cards that also supported the "a" standard in addition to the "b" and "g" required more time in order to "sweep" through 802.11a's frequencies which made for an average of 15266ms. After modifying the MADWiFi variant of MaxChannelTime (called by the implementors as "dwelltime") from its default MADWiFi value of 200ms to the minimum allowed 100ms, times for b/g and a/b/g cards dropped to 1368ms and 4728ms respectively.

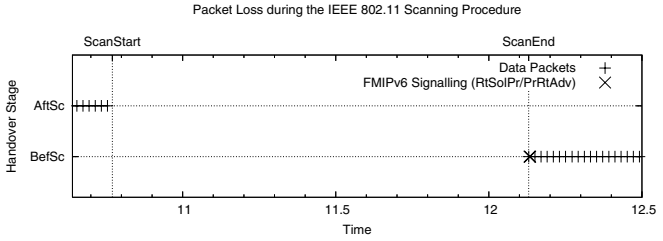

Fig. 7. Packet loss and latency during scanning for candidate APs.

Figure 7 represents the impact that the shortest of the above times has had on packet loss. In that particular example 69 packets have been lost during a frequency scanning, that took 1368ms.

Note that performing this scan at an arbitrary point of FMIPv6's execution not only has a significant impact on ongoing communications but is by no means a guarantee that at the time the MN initiates a predictive or reactive handover, the set of candidate APs discovered during the last scanning procedure is still valid.

E. Results Summary

Table I contains a summary of the results exposed in the previous subsections. One could easily see that predictive handovers are making for great handover performance by bringing packet loss and latency to a minimum. We do not provide a separate comparative "bare MIPv6" set since the purpose of this document is not to show the advantages of the FMIPv6 protocol over standard MIPv6. This has been already done far more than once in documents referenced in the III section. Furthermore, we believe that such comparisons are somewhat "unfair" as MIPv6 implementations do not generally provide any management of the handover destination selection process, the reason for that being the fact that MIPL's primary goal is to provide transparent mobility handling (i.e. avoid addressing and routing problems), and not protocol deficiencies. FMIPv6 on the other hand was designed to do just that. We have, however, included MIPv6 signalling so that the reader could could see the way both protocols interact.

Table I shows a summary of the losses during the different experiments and parts of FMIPv6 operation. The problem with candidate access point discovery is obvious - and losses in packet delivery caused by that problem greatly overpower whatever gains have been accomplished during the handover itself.

TABLE I
RESULTS FROM DIFFERENT HANDOVER SCENARIOS

Test Scenario	Conn LossTime	Pack Loss	Pack Buff	FMIP Tunnelled	Mvmt Detect
Predictive	10.42ms	0	1	8	21.07ms
Buffer Issues	9.35ms	1	0	5	104.80ms
Reactive	343.53ms	17	0	0	332.21ms
Scanning	1368.11ms	69	NA	NA	NA

VI. Conclusion and Future Work

We truly believe, and have shown, that FMIPv6 has the potential of bringing considerable optimisations upon the use of Mobile IPv6 for packet loss sensitive applications such as VoIP. According to experimentation conducted in our testbed the protocol considerably reduces (and in some cases even eliminates) packet loss (during handover) and handover latency to a level acceptable for real-time communications.

Yet there are still issues unadressed by the protocol that remain quite disturbing for media streaming, namely - candidate Access Point discovery. We have shown that the impact

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2006 proceedings.

of a wireless scan, conducted for lack of alternative ways to discover neighbouring APs, causes severe packet loss and lengthy connection disruption.

Another (rather minor, but still an) issue that we have demonstrated is the lack of a way for the mobile node to specify whether it wishes a New Access Router to buffer packets before the MN has had the chance of announcing itself on the new link.

Future work on the subject will be targeting exactly that candidate Access Point discovery problem and will include elaborating alternative, non-interrupting mechanisms. One possible way of doing so would be rendering possible parallel scanning and communication.This could be achieved through either the use of a secondary wireless interface, rapid alternation of frequencies on a single interface through temporal division of the medium access, or using multiple antennas on the wireless device.

We are also planning on extending the protocol with new IEEE 802.11 specific options that allow access routers to send to mobile nodes, all details that they might need for rapidly associating with a new Access Point, such as frequency, ESSID, and authentication info.

VII. Acknowledgement

We would like to thank Martin Andre from the Network Research Team at the Louis Pasteur University for codeveloping the fmipv6.org implementation with Emil Ivov and for valuable insight he provided during discussions, as well as questions he raised upon review of this article.

References

[HH02] Xavier Perez Costa Ralf Schmitz Hannes Hartenstein, Marco Liebsch. A MIPv6, FMIPv6 and HMIPv6 handover latency study: analytical approach, June 2002.

[Hos] HostAP. Host ap linux driver for prism22.53. http:hostap.epitest.fi.

[IA05] Emil Ivov and Martin Andre. The FMIPv6 Open Source Implementation Suite. http://www.fmipv6.org, 2005.

[IEE99] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Std 802.11, 1999.

[JPA04] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. RFC 3775 (Proposed Standard), June 2004.

[Koo05] R. Koodli. Fast Handovers for Mobile IPv6. RFC 4068, July 2005.

[KP01] Rajeev Koodli and Charles Perkins. Fast Handovers and Context Transfers in Mobile Networks, October 2001.

[KWF03] J. Kempf, J. Wood, and G. Fu. Fast Mobile IPv6 Handover Packet Loss Performance: Measurement for Emulated Real Time Traffic. 2:1230–1235, 2003.

[MAD] MADWiFi. Multiband atheros driver for wifi. http://madwifi.sf.net.

[mip] Mobile IPv6 for Linux. http://mobile-ipv6.org.

[NNS98] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). RFC 2461 (Draft Standard), December 1998.

[ns2] The Network Simulator ns-2. http://www.isi.edu/nsnam.

[PC03] Sangheon Pack and Yanghee Choi. Performance Analysis of Fast Handover in Mobile IPv6 Networks, 2003.

[QUA] QUAGGA. Quagga routing software suite. http://www.quagga.net.

[RH02] Hesham Soliman Karim El-Malki Robert Hsieh, Aruna Seneviratne. Performance analysis on Hierarchical Mobile IPv6 with Fast-handoff over End-to-End TCP, 2002.

[SCMB] Hesham Soliman, Claude Catelluccia, Karim El Malki, and Ludovic Bellier. Hierarchical mobile ipv6 mobility management (hmipv6).

[TN98] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. RFC 2462 (Draft Standard), December 1998.

[Tou] Jean Tourrilhes. The Linux Wireless Extensions and Wireless Tools.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2006 proceedings.