Thorough Empirical Analysis of the IETF FMIPv6 protocol over IEEE 802.11 networks

Emil Ivov
LSIIT (UMR7005)
Louis Pasteur University
Strasbourg, France
Email: Ivov@dpt-info.u-strasbg.fr

Julien Montavont
LSIIT (UMR7005)
Louis Pasteur University
Strasbourg, France
Email: montavontj@dpt-info.u-strasbg.fr

Thomas Noel
LSIIT (UMR7005)
Louis Pasteur University
Strasbourg, France
Email: Thomas.Noel@dpt-info.u-strasbg.fr

***Abstract*— The development of network technologies such as wireless LAN have made it possible for users to benefit from Internet connectivity almost anywhere and at anytime. In order to improve user experience, the IETF has defined the Mobile IPv6 protocol which allows mobile nodes to maintain their communication uninterrupted while roaming across various IPv6 subnets. However, the mechanisms that this protocol defines may cause undesired connection disruption and/or substantial packet loss which may significantly degrade the quality of real-time media streams. To address the problems of handover latency, the Mipshop IETF working group has adopted and developed the FMIPv6 protocol. Previous analysis, including some of our own, has shown that the protocol is particularly efficient in reducing both the duration of handovers and the number of lost packets. However, these previous works are often based on theoretical studies and simulations, or do not analyze all FMIPv6 operations and features. We therefore present in this article a thorough experimental evaluation of FMIPv6 over wireless LANs. This article extends our previous work by evaluating all aspects of FMIPv6: predictive handovers, reactive handovers, and network-initiated handovers. We also evaluate the benefits of FMIPv6 on layer 2 only handovers. All experiments are conducted with unmodified versions of the popular conferencing and video streaming applications Gnomemeeting and VLC.**

***Index Terms*— Fast Handovers for Mobile IPv6, wireless LANs, IEEE 802.11, Seamless Handover**

I. INTRODUCTION

Due to its relatively low deployment cost, easy device set up, and increasing performance, Wireless LAN (also known as Wi-Fi) is becoming very attractive and arguably one of the most popular ways for connecting to the Internet. In the same time, given all the maintenance and development problems associated with Network Address Translation (NAT), deployment of IPv6, the new version of the Internet Protocol, is gaining speed and is likely to soon become the default way for communicating through the Internet. To provide mobility support in IPv6 networks, the Internet Engineering Task Force (IETF) has defined the Mobile IPv6 (MIPv6) protocol [1] which is now adopted as a standard. Although MIPv6 allows a node to maintain its ongoing communication with remote hosts while roaming across different IPv6 subnets, it suffers various limitations which prevent it from being adopted in large scale production deployments. Among these is the protocol's poor support (or lack thereof) for rapid and seamless handovers. In addition, mechanisms introduced by underlying wireless technologies may also increase the overall delay experienced during handovers. As a result, when moving from one subnet to another, a mobile node may experience connection and/or packet loss that may seriously degrade the quality of real-time media streams delivered to the user. During the last few years, there have been many optimizations addressing the handover latency related to the MIPv6 mechanisms. Among these is the Fast Handovers for Mobile IPv6 (FMIPv6) protocol [2] which has recently been standardized by the IETF community. Numerous existing studies [3], [4] (including some of our own [5], [6]) evaluating FMIPv6, have shown that it could be very efficient for achieving seamless handovers. However, these previous works are often based on theoretical analysis and simulations, or do not cover all FMIPv6 use cases and features. We do believe, however, that many of the reasons why handover latency occurs are closely related to implementations and operating system specifics that are most often ignored in simulators and even more so in theory. Furthermore, simulation models tend to over simplify the characteristics of the wireless link. They often ignore the effects of interference and the complexity of propagation effects. These are the reasons why in this article, we have completed an entirely empirical study based on real experiments that evaluate all aspects of FMIPv6 over wireless LANs. The evaluation extends our previous work [5], [6] to a significantly larger set of test cases. This analysis uses the new FMIPv6 Open Source Implementation Suite from fmipv6.org [7] for the GNU/Linux operating systems. We will be concentrating on the impact that the FMIPv6 handover has on a video stream flow (sent by the VLC application, http://www.videolan.org) and on a video conferencing session (established with the Gnomemeeting application, http://www.gnomemeeting.org).

The rest of the article is organized as follows: First, we describe the FMIPv6 protocol itself, as well as some of the reasoning followed by the authors during the experimentation. Section III goes through a description of the testbed and scenarios that we used for our experiments. All our results are presented in Section IV. Finally, Section V concludes the article with a summary of results and potential next steps.

II. FAST HANDOVERS FOR MOBILE IPV6

A. Problem Statement

Although the mechanisms defined in the IEEE 802.11 standard and MIPv6 allow a MN to move across different AP-s and IPv6 subnets, the procedures accompanying such terminal mobility are often a cause for cuts and delays in ongoing communication. According to results presented in [8], the L2 handover would generally take between $58.7ms$ and $396.7ms$ to complete. Approximately 90% of this time is spent discovering candidate AP-s as the MN has to scan several radio channels before being able to select new AP-s to connect to [8]. The following MIPv6 handover could also add to the overall connection loss time. First, the minimal allowed delay between two consecutive Router Advertisements (which are used for movement detection) is between 0.03 and 0.07 seconds [1]. Therefore, in many cases, a MN would have to wait for approximately 0.05 seconds before detecting L3 subnet change. Then, after configuring its new IPv6 address and before actually being able to use it a MN needs to perform Duplicate Address Detection (DAD) in order to verify the uniqueness of this address. DAD may take several seconds to complete (approx. $1.5s$ for RFC default values) [9], [10]. Finally, the delay necessary for updating the binding with the HA depends on the Round Trip Time (RTT) between the MN and the HA plus the time necessary for MIPv6 implementation there to create and setup the corresponding tunnel. Considering all of the above, total connection loss time during a handover would often be longer than 1 second [11]. It is true that in many cases such a delay would be acceptable for certain applications (e.g. mail, web browsing), however delays or cuts above $150ms$ would generally cause user perceptible quality degradation in real-time communications such as VoIP or VoD [12].

B. Protocol Description

To address the problems described in the previous section, IETF has proposed the Fast Handovers for Mobile IPv6 (FMIPv6) [2] protocol which is one of the most promising solutions reducing the duration of the L3 handover and minimizing the number of lost packets. The idea lying behind this protocol is providing a MN with all properties of the IPv6 subnet it is going to move to, before it has actually done so. FMIPv6 also makes it possible for the next Access Router (AR) (the one the MN is moving to) to buffer all MN bound packets that arrive while the MN is disconnected because of the handover.

FMIPv6 defines a way for a MN to request from its current AR all L3 parameters of the IPv6 subnets behind neighboring AP-s. Note that FMIPv6 does not propose any specific method for discovering candidate access points and previous work on the subject points this as one of its major weaknesses [5], [6]. The default method proposed in [13] (i.e. periodically performing a scan over one or more 802.11 channels) may cause severe packet loss that gratly outweighs the benefits of the following seamless handover.

After discovering nearby APs the MN sends a Router Solicitation for Proxy Advertisement (RtSolPr) to its current AR. Upon reception, the AR resolves all AP identifiers included in the RtSolPr to subnet-specific information. Next, the AR sends back a Proxy Router Advertisement (PrRtAdv) containing one or more AR-AP tuples which include the requested information. Note that a MN may send a RtSolPr at any time.

FMIPv6 also defines semantics that allow the optimization of the handover itself through two distinct modes: Predictive Handover and Reactive Handover. Whenever possible, a MN would perform predictive handover as this allows it to fully benefit from all FMIPv6 optimizations. The reactive handover mode is most often employed in cases where a node has unexpectedly lost connection with its current AR or AP. Predictive and reactive modes are illustrated in figures 1 and 2 of the FMIPv6 protocol RFC [2].

1) Predictive Handover: If a MN is able to detect (e.g. through the use of link layer information) the need of a handover it could send a Fast Binding Update (FBU) to its current AR. This message contains MN's current CoA and the AR that the MN is planning to switch to (referred to as NAR for Next Access Router). At that point the PAR (Previous Access Router, previously referred to as the current AR) sends to the NAR a Handover Initiate (HI) message containing the identity of the MN (link layer address, current CoA and, if known, desired next CoA). The NAR confirms (or rejects) the handover with a Handover Acknowledge (HACK) message that may provide further NAR specific details. Upon HACK reception, 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 Neighbor Advertisement (FNA) message which is supposed to update respective neighbor cache entries on the NAR so that it could stop buffering MN's packets and complete handover signaling.

2) Reactive Handover: This mode basically represents the case where a MN 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 L2 handover has completed and is usually encapsulated in the FNA. NAR then forwards that FBU to PAR and PAR starts tunneling packets.

Note that FMIPv6 also defines a mechanism for an AR to initiate a handover - a network-initiated handover, initially previewed for purposes like load sharing. In this case, the AR sends an unsolicited PrRtAdv to the MN including the AP-AR tuple that the MN has to switch to. Upon reception of such an unsolicited PrRtAdv, a MN has to immediately start a predictive handover.

III. EXPERIMENTATION

This section describes the testbed and scenarios that we were using when evaluating the FMIPv6 protocol.

A. The Testbed

Our testbed is composed of three AR-s, three AP-s, one MN and one correspondent node. It also contains three IPv6 subnets. A top AR provides IPv6 Internet connectivity to the rest of the testbed and is also configured as a HA. AP1 is connected to one of the AR-s while AP2 and AP3 are connected to a second AR and hence a different IPv6 subnet. (Readers may refer to http://fmipv6.org/testbed.pdf for a visual representation of our testbed.)

All devices are running the GNU/Linux operating system, except for the AP-s which are 802.11b Cisco AP 1200 devices. The HA is running the new MIPv6 daemon for the GNU/Linux operating system (MIPL-2, http://mobile-ipv6.org). The MN and the AR-s (AR1 and AR2) are using the FMIPv6 Open Source Implementation Suite (http://fmipv6.org) which is based on MIPL-2. We are also using a legacy IPv6 node connected to the Internet from a point outside the testbed.

Due to the nature of the FMIPv6 protocol, the fmipv6.org implementation requires fine grained control over the behaviour of the wireless card so that it could be efficient. We have therefore equipped the MN with a 3Com 802.11 a/b/g PCMCIA wireless card managed by the MADWiFi driver (http://madwifi.org). MADWiFi is an open source GNU/Linux kernel device driver for wireless LAN chipsets from Atheros. Our driver modifications address the periodic scans (occurring prior the RtSolPr/PrRtAdv exchange), the link-layer triggers (required for predictive handover) and the L2 handover itself.

It is important to note that for all testing we have been using an fmipv6.org specific feature that allows all scanning to be executed on a secondary wireless interface. This allows us to resolve (or rather circumvent) FMIPv6's inability to provide the Link-Layer details of candidate Access Points, such as their ESSID and frequency. By using a secondary interface for all scanning, communication through the primary interface is never interrupted. The candidate access point discovery problem is described in more detail in [5], [6].

B. Evaluation Scenarios

In order to provide a complete and consistent evaluation of FMIPv6, we have run two distinct series of tests in two different scenarios. In Scenario 1, the MN would move between AP1 and AP2 and it would therefore also change its IPv6 subnet, moving from AR1 to AR2. This case represents a very common FMIPv6 demonstration scenario. Although FMIPv6 is often presented as a L3 solution, we have decided to evaluate its performance when optimizing L2 only handovers. In Scenario 2 the MN would therefore be moving between AP2 and AP3, without changing an AR in the process. See http://fmipv6.org/testbed.pdf.

For both scenarios we have executed a series of tests in which the MN is in the process of having a video conversation with its correspondent (both hosts are equipped with a webcam and use the Gnomemeeting). We have also run a second series of tests in which the MN receives a video stream from the correspondent node (using VLC). In the video stream case, data is encapsulated and sent through the Real-time Transport Protocol (RTP) [14]. RTP packets have an average length of approximately 1336 bytes and are sent every $30ms$. Gnomemeeting also uses RTP only this time the audio and video are sent in separate flows. Average audio packet size is approximately 70 bytes and packets are sent every $30ms$. Video packets have an average size of 950 bytes and are sent every $160ms$.

In Scenario 1, we perform an evaluation of all use cases defined by the FMIPv6 protocol: predictive handovers, reactive handovers and network-initiated handovers. Note that the FMIPv6 RFC does not define network-initiated handovers as a separate mode of operation but we have decided to run separate tests for them as we believe, that due to their possibility to allow for network mobility control and load sharing, they might represent special interest to many readers and potential implementors. With the tests run for Scenario 2, we analyze the influence of predictive handover on L2 handover in order to evaluate the benefits of buffering (or lack of it) while performing L2 only handovers.

IV. Performance Evaluation

Results presented in this section are obtained by running the Ethereal tool (http://www.ethereal.com) on the MN, the correspondent node and the AR-s. Every distinct test case (e.g. a Gnomemeeting conversation during a Predictive Handover in Scenario 1, or a VLC streaming session in a reactive L2 only handover) was evaluated through a series of at least ten consecutive test runs, which gives us a total of 13 cases and more than 130 test handovers. Results shown on all following figures correspond to the test run whose L2 handover time represents the median for all L2 handovers in the corresponding test case.

Figures 1 and 2 present the results for FMIPv6 predictive handovers in Scenario 1. We can see, that the predictive mode allows MN-s to not lose even a single packet while performing the handover. Triggered by a link-layer event, the MN initiates the handover by sending to its current AR (i.e. the PAR) a FBU containing the identity of the AR that it plans to switch to (i.e. the NAR). After exchanging the HI and HACK messages with the NAR, the PAR starts tunneling packets bound to MN and routes them to the NAR. It then sends a FBACK back to the MN. Upon reception of the FBACK, the MN starts the L2 handover and associates with the new AP. Once on the NAR's link (after approximately $20ms$), the MN sends a FNA requesting the NAR to deliver all packets that it has buffered so far. At this point data packets are still being forwarded to the PAR by the HA and are therefore tunneled to the NAR through the FMIPv6 tunnel until the MN sends a BU from the NAR's link allowing the HA to update the corresponding binding cache entry. After the MIPv6 BU/BACK exchange, the HA starts forwarding data packets directly through the NAR. The tunnel between the PAR and the NAR has been active (i.e. has been forwarding packets) for $190ms$ (Figure 1) and $134ms$ (Figure 2) on average, with respectively 5 and 12 packets forwarded through the FMIPv6 tunnel. The lifetime of the tunnel between PAR and NAR is

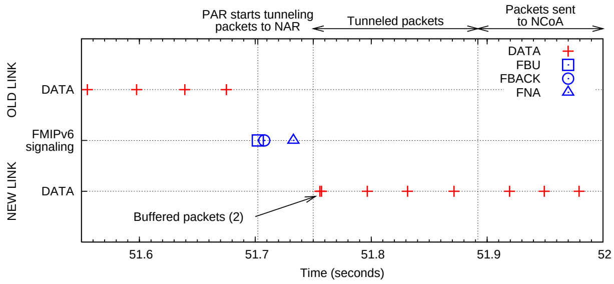

Fig. 1. Impact of FMIPv6 Predictive HO on a video stream

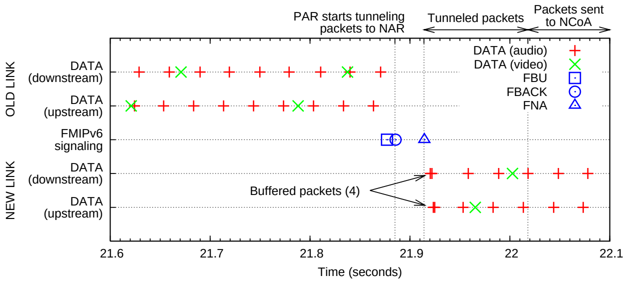

Fig. 2. Impact of FMIPv6 Predictive HO on a video conference stream

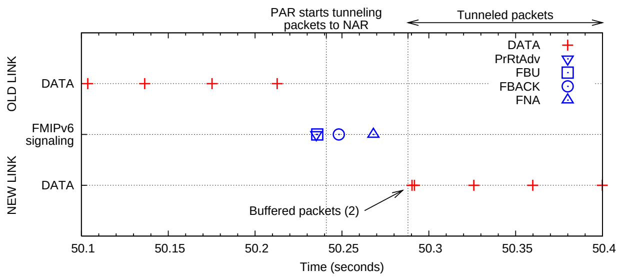

Fig. 3. Impact of FMIPv6 network-initiated HO on a video stream

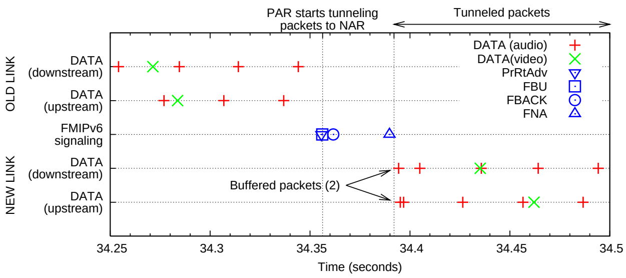

Fig. 4. Impact of FMIPv6 network-initiated HO on a video conference

specified by the MN at the transmission of the FBU. During our experiments, we configured the MN to request the minimal allowed tunnel lifetime, i.e. 4 seconds according to [2]. Note that the MN is not required to perform DAD before sending BU (see Section **??**) as its NCoA is negotiated prior to the handover (through the FMIPv6 exchanges) which explains the relatively short interval between the end of the L2 handover and the transmission of the BU. We observe small delays during the reception (or transmission) of buffered packets, but these delays have no impact on the user application (neither on the MN nor on the correspondent node). As a result, the test runs for these cases are an example of completely seamless FMIPv6 predictive mode handovers.

Results related to FMIPv6 network-initiated handovers in Scenario 1 are shown on Figures 3 and 4. As expected we can see that FMIPv6 performance here is very similar to that observed during predictive handovers. In fact, the only thing that differentiates a network-initiated handover from a predictive one is that it is up to the PAR to select a destination AP and NAR for the MN, as well as the fact that the handover would start right after the MN has received the PrRtAdv (Figures 3 and 4). As a result FMIPv6 network-initiated handovers, like those initiated by the MN, are in most cases completely seamless. Note that packets sent directly to MN's NCoA (after the BU/BACK exchange between the MN and the HA) are not shown for readability reasons.

Figure **??** shows the performance of FMIPv6 reactive handovers for Scenario 1. When a MN fails to anticipate a handover and does not sent a FBU from the PAR's link, it would perform a FMIPv6 reactive handover. To trigger reactive handovers, we manually shutdown the current AP of the MN (hard reset). As we can see from Figure **??**, it takes respectively $1.324s$ in the VLC test case and $1.497s$ when testing with Gnomemeeting for the wireless device

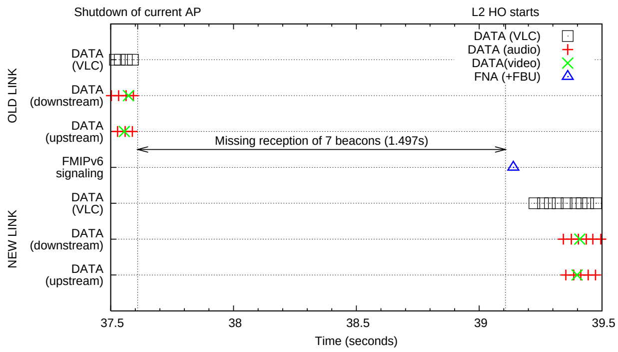

Fig. 5. Impact of FMIPv6 Reactive Handovers

(network card) to detect the link failure and starts the L2 handover. In wireless LANs, a link failure is generally detected through the Beacon messages that AP-s send periodically. After missing more than a certain number of such beacons a node would assume that it has lost its connection with the corresponding AP. Missing a single beacon frame does not necessarily imply connection loss as the event may be due to a link layer collision. The probability for such collisions to occur in wireless LAN-s is quite high, and wireless devices would generally wait a certain amount of time before giving up on the connection. When configured with default parameters the MADWifi driver would wait for 7 times the default beacon retransmission interval before reporting that the link is down. In our testbed, the AP-s are configured to send Beacon every $200ms$ which explains the amount of time necessary for the MN to initiate the L2 handover.

After completing the L2 handover, the MN sends a FNA and a FBU from its new link. The rest of the handover is not different from a predictive one (see Section II). All packets sent between the shutdown of the AP and the reception of

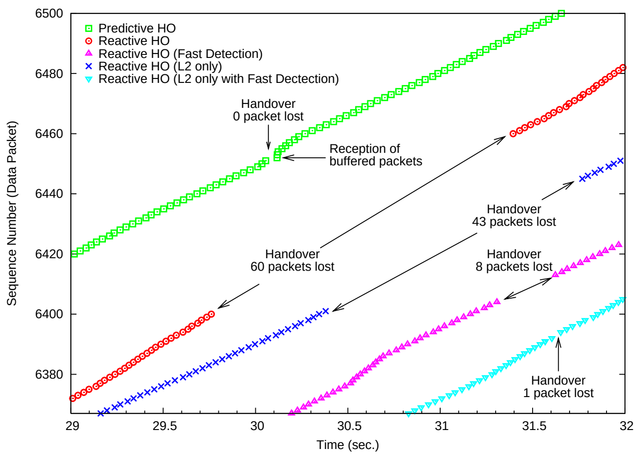

Fig. 6. FMIPv6 Reactive Handover performance

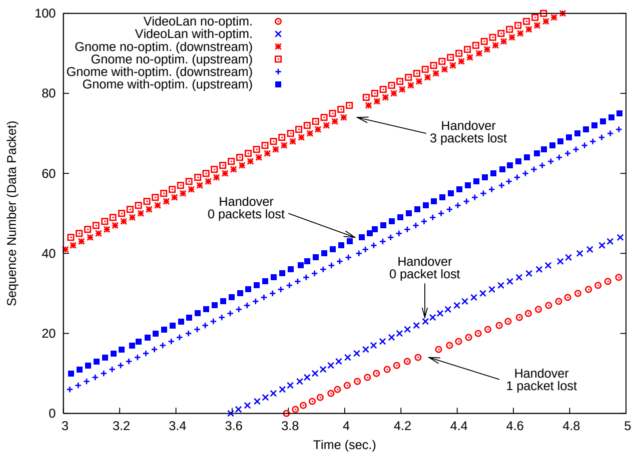

Fig. 7. Benefits of FMIPv6 for IEEE 802.11 handovers (scenario 2)

the FBU on the PAR are lost as there has been no buffering or tunneling (the PAR would tunnel packets for the MN only after receiving the FBU from the MN). During these delays, the MN has lost 60 packets for VLC and 130 packets for Gnomemeeting (Figure **??**) which leads to a relatively long interruption of media flows. For the Gnomemeeting tests the 130 packets can be divided to: 57 audio and 11 video packets (downstream) and 51 audio + 11 video packets (upstream).

As the lower performance of the FMIPv6 reactive handovers is mainly due to the slow link failure detection mechanism of the wireless device, we have decided to further evaluate them in some slightly different conditions. Instead of simply unplugging the AP we shut it down through its user interface. In this case the AP sends a Disassociation Request to the MN which is now able to immediately detect the link failure and start a L2 handover. Through the rest of this section we refer to this test case as “Fast Detection”. We also include this test case when evaluating reactive handovers in Scenario 2 (i.e. when the MN only performs a L2 handover). The results have been obtained in the same circumstances as already explained (i.e. using VLC and Gnomemeeting). Figure 6 shows the results of these additional experiments. Each dot represents the reception of a data packet, at the time indicated in the X-axis. We have included the results for the FMIPv6 predictive handover in Scenario 1 as a reference. During a Fast Detection test, the MN loses an average of 8 packets, which is more than with a predictive handover, where there were no lost packets, but considerably less than the 60 packets of the standard reactive case. Using such fast detection of the link failure, the MN could initiate the handover sooner and therefore reduce the duration of the overall procedure. However, such amount of lost packets still introduces short flow interruptions in the video. In the case of a L2 only reactive handover, for standard and Fast Detection schemes, we have lost respectively 43 and 1 packets while the MN has been disconnected.

In Scenario 2 we analyze the benefits of the FMIPv6 predictive mode when the MN only performs a L2 handover. Due to the fact that FMIPv6 is often presented as a network layer solution, some might argue that using it in a Wi-Fi deployment that only uses a single subnet might be unnecessary. Therefore in order to demonstrate the benefits of FMIPv6 in L2 only handovers, we have completed a series of tests in such an environment. Figure 7 shows the results from these experiments for both video streaming and a videoconference. Each dot represents the reception (or transmission) of a data packets at the time indicated on the X-axis. We can see that if the MN does not send a FBU for a L2 handover (the case is referred to as no-optim. on the figure) it loses 1 packet in the video streaming use case and 3 packets for the videoconference use case. This relatively small number of lost packets is explained by the short duration of the L2 handover. When using the information previously discovered through periodic scanning and the RtSolPr/PrRtAdv exchange, takes approximately $20ms$ to complete, as the MN could immediately send and receive packets after the L2 handover has completed. However, with even as few as 1-3 lost packets the MN would still experience noise in audio and video rendering for both VLC and Gnomemeeting applications. When applying the mechanism of L3 predictive handovers (cases shown as with-optim. on the figure), the MN does not lose a single data packet because of the buffering at both the MN and the NAR (which is also the PAR in this scenario). Despite the short delays observed in the reception (or transmission) of buffered packets, the handover remains seamless for the user in both the VLC and the Gnomemeeting use case. In other words it appears that a MN can earn a lot in terms of performance when using FMIPv6 even for L2 only handovers. Finally, Tables I and II present the average values for all results related to our experiments.

V. Conclusions and Future Work

The rapid deployment of wireless technologies have raised new user expectations such as the ability to always connect to the Internet as well as being able to communicate while on the move. The Mobile IPv6 protocol is an IETF standard which allows Mobile Nodes (MN) to remain reachable while roaming through several IPv6 networks. The constantly improving characteristics of wireless technologies, and wireless LANs in

TABLE I

RESULTS FOR EXPERIMENTS RELATED TO VLC

Handover	Average HO Proc. Length (ms)	Average Packet Loss	Average Packet Buffered	Average User Experience
Predictive	33.2	0	1.3	No interruption
Predictive (L2 only)	19.3	0	0.5	No interruption
L2 no opt.	18.5	0.6	0	Short flow interruption
Network-Initiated	32.6	0	1.7	No interruption
Reactive	1518.9	52.6	0	Long flow interruption
Reactive (Fast det.)	248.4	8.3	0	Long Flow Interruption
Reactive (L2 only)	1442.3	45.1	0	Long Flow Interruption
Reactive (L2 only + Fast det.)	20.0	0.8	0	Short Flow Interruption

TABLE II

RESULTS FOR EXPERIMENTS RELATED TO GNOMEMEETING

Handover	Average HO Proc. Length (ms)	Average Packet Loss	Average Packet Buffered	Average User Experience
Predictive	36.6	0	2.3	No interruption
Predictive (L2 only)	20.9	0	0.4	No Interruption
L2 no opt.	17.7	1.7	0	Short flow Interruption
Network-Initiated	28.9	0	1.6	No Interruption
Reactive	1542.3	127.7	0	Long Flow Interruption

particular, have made possible the transmission of high quality media streams. Such communications are generally very time-sensitive, i.e. transmission delay may seriously degrade the quality of the content that is being streamed. To address this requirement, the Mipshop IETF working group has defined the Fast Handovers for Mobile IPv6 (FMIPv6) protocol. The idea lying behind FMIPv6 is to anticipate the next attachment point of a MN prior to the actual handover. This way, packets sent to a MN while it is performing a handover are forwarded from the previous access router to the next, and are then buffered until the MN is ready to receive them. In this article, we have completed an entirely empirical study based on real experiments that evaluate all aspects of FMIPv6 over wireless LANs.

Results presented in Section IV show that the performance of FMIPv6 could vary a lot depending on the employed FMIPv6 operation mode. Predictive mode could be very efficient with as few as 0 lost packets during a handover and no user-perceptible cuts or delays at all. After successfully anticipating a movement event, every packet sent while the MN is performing handover is forwarded to and buffered by its Next Access Router (NAR). Once the MN is connected to its new subnet, the NAR could directly deliver buffered data packets. This mode is clearly the main advantage of FMIPv6 for achieving seamless handovers. Network-initiated handovers, that we analyzed in order to get an idea on the possibility of using the protocol for purposes like load sharing, provide performance that is virtually the same as with predictive handovers. Not surprisingly, the FMIPv6 reactive mode is less impressive and when MNs are not able to anticipate a handover it only allows to limit the damage to some extent. As we can see from the results, the performance of reactive handovers is directly related to the time that is necessary for the wireless device to detect link failure. The longer this detection lasts, the more it is going to take for reactive handover to complete, and since no buffering has been started, all packets delivered during this period of time are lost. We have shown, however, that FMIPv6 reactive mode could still reach relatively good performance if we have a relatively fast link failure detection mechanism (see Figure 6). Note, that this problem is not related to FMIPv6 and is also an issue with the standard Mobile IPv6 protocol.

Finally, we have seen that FMIPv6 (predictive mode in particular) could also reduce the number of lost packets during a layer 2 only handover (i.e. when the new access point of a MN is located in the same IPv6 subnet as the old one). Depending on the layer 2 handover latency and the rate at which data packets are sent (respectively every $20ms$ and $30ms$ on average in our experiments), the MN is likely to lose a substantial number of packets while performing a layer 2 only handover without any optimizations. FMIPv6 allows a MN to request packet buffering from its current access router prior to initiating its layer 2 handover, by simply notifying it that it is planning to move to another one of its access points (i.e. the PAR is the same as the NAR). In other words, while presented as a layer 3 specific solution, FMIPv6 could just as easily be used for layer 2 only handovers.

In conclusion, FMIPv6 is particularly well suited for achieving seamless handovers over wireless LANs: predictive and network-initiated handovers allow a MN to not lose data packets and even with reactive handovers we are still able to achieve better handover performance than we could hope to have in comparison to the standard Mobile IPv6 protocol. However, as already pointed out in our preliminary evaluation of FMIPv6 [5], the discovery of candidate access points (occurring prior to handovers) may still introduce large delays in ongoing communications. Without specific support, these stages make the seamless handover purpose meaningless. For the purposes of the evaluation presented in this article, we have been using the possibility to configure the fmipv6.org MN to use a secondary wireless interface for scanning and thus avoid interrupting traffic on the primary one. Future work on the subject should include investigation of alternative, more optimal and energy efficient methods for achieving non-interruptive candidate access point discovery.

REFERENCES

[1] D. Johnson, C. Perkins, and J. Arko, "Mobility Support in IPv6, Internet Engineering Task Force Request for Comments (RFC) 3775," June 2004.

[2] R. Koodli (Ed.), "Fast Handovers for Mobile IPv6, Internet Engineering Task Force Request for Comments (RFC) 4068," June 2005.

[3] J. Kempf, J. Wood, and G. Fu, "Fast Mobile IPv6 Handover Packet Loss Performance: Measurement for Emulated Real Time Traffic," in *Proceedings of the IEEE Conference on Wireless Communications and Networking (WCNC'03)*, March 2003.

[4] X. Perez-Costa, M. Torrent-Moreno, and H. Hartenstein, "A Performance Comparison of Mobile IPv6, Hierarchical Mobile IPv6, Fast Handovers for Mobile IPv6 and their Combination," *SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 4, pp. 5–19, 2003.

[5] E. Ivov and T. Noel, "An Experimental Performance Evaluation of the IETF FMIPv6 Protocol over IEEE 802.11 WLANS," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'06)*, vol. 1, April 2006, pp. 568–574.

[6] J. Montavont, E. Ivov, and T. Noel, "Analysis of Mobile IPv6 Handover Optimizations and Their Impact on Real-Time Communication," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'07)*, March 2007, pp. 3244–3249.

[7] E. Ivov and M. Andre, "The FMIPv6 Open Source Implementation Suite, http://fmipv6.org."

[8] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 93–102, 2003.

[9] T. Narten, E. Nordmark, and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6), Internet Engineering Task Force Request for Comments (RFC) 2461," December 1998.

[10] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration, Work in Progress, Internet Engineering Task Force draft-ietf-rfc2462bis-08.txt," May 2005.

[11] N. Montavont and T. Noel, "Handover Management for Mobile Node in IPv6 Networks," *IEEE Communication Magazine*, vol. 40, no. 8, pp. 38–43, 2002.

[12] International Communication Union (ITU), "Transmission Systems and Media, Recommendation G.714: One-way Transmission Time," 1996.

[13] P. McCann, "Mobile IPv6 Fast Handovers for 802.11 Networks, Internet Engineering Task Force Request for Comments (RFC) 4260," November 2005.

[14] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications, Internet Engineering Task Force Request for Comments (RFC) 3550," July 2003.