This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2007 proceedings.

Analysis of Mobile IPv6 Handover Optimizations and Their Impact on Real-Time Communication

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

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

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

Abstract— **The Mobile IPv6 protocol is becoming one of the most common ways to support IPv6 mobility. Yet, despite its popularity the protocol still suffers various limitations which prevent it from being adopted in large scale commercial deployments. Among these is the protocol's poor support (or lack thereof) for rapid and seamless handovers. When moving from one subnet to another, a mobile node may experience connection and/or packet loss which may introduce noise and cuts in real-time media streams delivered to the user. Many different solutions to this problem have been presented and evaluated in various papers during the last few years but none of them have so far been selected as the standard Mobile IPv6 handover optimization. In this article we present experimental analysis of two optimization schemes and compare their performance to that of a standard Mobile IPv6 implementation. One of them, the Fast Handovers Mobile IPv6 protocol (FMIPv6) is being strongly supported by IETF working groups and is probably the one that, among all other handover optimization schemes, has most approached standardization. The GPS Handovers solution, on the other hand seems to be one of the few that have the maturity and completeness of the FMIPv6 protocol, and in the same time resolves problems that seem to be unaddressed by FMIPv6. All analysis in this paper is based upon experimentation which we believe is superior to simulation or emulation when dealing with a subject that is so heavily influenced by the implementation specifics of the accompanying technologies. Mobile nodes in all experiments are communicating via the IEEE 802.11 WLAN technology. All experiments are conducted with unmodified versions of popular conferencing and streaming applications Gnomemeeting and VLC.**

Index Terms— **Mobile IPv6, Fast Handovers for Mobile IPv6, Geolocation Assisted Handover, Seamless Handover.**

I. Introduction

Rapid development of wireless technologies such as Wi-Fi, have made it possible for users to communicate while on the move. Wireless LANs are being constantly improved and are already offering the throughput, reliability, and security necessary for the transportation of sensitive data and real-time traffic. The area covered by public and commercial WiFi hotpots is rapidly increasing which creates an environment that is very likely to additionally encourage user mobility.

Due to the relatively short coverage range of Wi-Fi access points (AP), a roaming mobile node (MN) is likely to connect to multiple APs. Every reconnection consists in a layer 2 handover procedure described in the IEEE 802.11 standard [1]. When all APs that a node connects to are located in the same IPv6 subnet, the MN is able to communicate immediately after the layer 2 handover has been completed. The time that it takes to complete this layer 2 handover procedure is inherently associated with connection loss and its duration may vary according to the wireless devices [2]. In addition, if the new AP turns out to be on a different IPv6 subnet, the MN has to update its current IPv6 address in order to be able to communicate. Without specific support, this requires reinitialization of all current communication sessions. A common solution to this problem is the Mobile IPv6 (MIPv6) protocol [3] which has lately been accepted as the Internet Engineering Task Force (IETF) standard for managing layer 3 handovers in IPv6 networks. The protocol however does not completely remove the latency introduced by the layer 3 handover. The overall delay accumulated during the layer 2 and layer 3 handovers is therefore likely to cause user-interceptable connection loss and cuts in time-sensitive applications such as streaming or real-time communication. During the last few years there have been many optimizations that address latency accumulated during the layer 2 or layer 3 (or both) parts of a handover in IPv6 wireless LANs. In this document we focus on the two that we find most promising - the Fast Handovers for Mobile IPv6 protocol (FMIPv6) [4] and the GPS Handover [5]. The FMIPv6 protocol appears to be the one that is most widely supported by the IPv6 research community and the only one that has been standardized by the IETF. We do believe however that as of the writing of this document FMIPv6, in an effort to remain a layer 3 only solution without any link layer dependencies, has left unresolved various issues related to handovers in a wireless LAN such as discovery of candidate APs for example. We feel, however, that they could be resolved in a manner similar to the one employed by the GPS Handover scheme. The analysis that we propose in this article is a necessary first step towards such an integration.

Many previous works in this field, including some of our own, have been based on results obtained by simulation [6] and/or emulation [7]. In this document we have chosen an entirely empiric approach that we find most suitable for the present analysis. When evaluating a handover optimization, it is particularly important to analyze user-perception and behaviour of real-world applications since they represent the very reason to optimize. It is quite delicate (and in most cases impossible) to do so properly when simulating or evaluating mobility. Furthermore many of the reasons why handover latency occurs in the first place are closely related to implementation, platform or operating system specifics that are most often ignore in simulators.

We will be concentrating on the impact that the handover latency has on a video streaming flow (sent by the Video LAN software [8]) and on a video conferencing session, established with the Gnomemeeting application [9]). We have also performed an evaluation of the standard MIPv6 protocol in order to provide a comparative analysis with both optimizations. In addition to the applications we are also using the new MIPv6 daemon for the GNU/Linux operating system [10] and the FMIPv6 Open Source Implementation Suite [11].

The rest of the document is organized as follows. First, in II, we briefly describe all evaluated protocols. Section III presents the testbed that we have set up and used for all experiments followed by section IV where we present and analyze obtained results. Finally, conclusions and future work are presented in section V.

II. Algorithm Descriptions

A. Mobile IPv6

The MIPv6 protocol [3] is based upon the notion of a dedicated node, known as a Home Agent (HA), located in the primary (home) network of an MN. While in the same subnet as its HA, an MN would communicate with its peers according to the standard IPv6 protocol. Once it moves to a foreign network, the MN would acquire a new locally valid care-of address (CoA) and notify its HA that its location has changed so that it could then start to relay traffic between the MN and its correspondents.

Changing the layer 3 subnet is quite often caused by a change in the physical network of an MN. In wireless LANs, this occurs when an MN roams to a new AP. The procedure for this layer 2 handover is described by the IEEE 802.11 standard [1] which defines a discovery, an authentication and an association phase. Once the association has been completed, the MN would need to discover the properties of the IPv6 subnet that the new AP is located in. Such properties are generally delivered in Router Advertisement (RA) messages that the local access router broadcasts periodically or in response of a Router Solicitation (RS). Once the MN receives an RA, it can determine whether it has moved to a new IPv6 subnet in which case the MN would have to configure a locally valid network address which most often achieved through the IPv6 stateless autoconfiguration mechanism [12]. Next, the MN would send its new CoA in a Binding Update (BU) to its HA so that it would update its location. Upon reception of a Binding Acknowledgement (BACK) with a successful status code, the layer 3 handover is complete.

B. Fast Handovers for Mobile IPv6

The FMIPv6 protocol [4], standardized by the IETF, aims to reduce the duration of the layer 3 handover and minimize the number of lost packets. The protocol defines a way for a MN to request from its access router the layer 3 parameters of the IPv6 subnets behind all neighboring APs. FMIPv6 also defines semantics that allow the optimization of the handover itself. If an 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 it's current access router. This message contains MN's current CoA and the access router that the MN is planning to switch to (NAR). At that point the PAR (previously refered as the current access router) sends to NAR a Handover Initiate (HI) message containing the identity of the MN (link layer address, current CoA and, if known, desired next CoA). 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 Neighbour Advertisement (FNA) message which is supposed to update respective neighbour cache entries on the NAR so that it could stop bufferring MN's packets and complete handover signalling.

The FMIPv6 protocol also defines a reactive handover scenario which basically represents the case where an 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 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.

C. GPS Handover

With the growing popularity of geolocation systems, new schemes using the position of the devices in the management of mobility start to appear. In [5], we have presented one such scheme which uses the GPS geolocation system in order to improve the layer 2 and 3 handover in IPv6 wireless LANs. The idea behind this protocol is to constantly monitor position of mobile devices and determine their next AP prior to a handover. The proposal is based on the MIPv6 protocol and introduces a new network entity called Mobility Controller (MC). The MC manages a database containing all details on deployed network entities. All APs are registered together with their operating channel, Service Set Identifier (SSID), MAC address, position (latitude and longitude), and IPv6 address and prefix of the default access router. All MNs are equipped with a GPS receiver and periodically send their coordinates to the MC which calculates the distance between the MN and its current AP. When this distance becomes larger than a preconfigured threshold, the MC would select for the MN a new AP, located closer to its new position and send a Handover Initiate (HI) message to the MN containing all information necessary for it to switch to the new AP including the necessary layer 2 (SSID and operating channel) and 3 (address of the default access router and IPv6 prefix) parameters. When an MN receives such a message it would initiate the handover procedure. The layer 2 handover is thus very reduced as the

MN no longer needs to scan through all channels in order to discover its new AP and could instead directly proceed with authentication and association. The layer 3 handover is also enhanced because the lengthy address configuration phase is completely removed (with the help of the HI's information and the Optimistic DAD procedure [13]) and it is possible for the MN to send its BU right after the layer 2 association.

III. Experimentation

This section describes the testbed and scenarios that we were using when evaluating the two optimization schemes and Mobile IPv6, and explains our choice of the wireless devices used by the MN.

A. The Testbed

Our testbed is composed of three access routers, two APs, one MC, one MN and one correspondent node (CN). The testbed contains three IPv6 subnets. The top access router provides IPv6 Internet connectivity to the rest of the testbed and is also configured as a HA. Each AP is connected to a different IPv6 subnet and access router. Figure 1 illustrates the testbed.

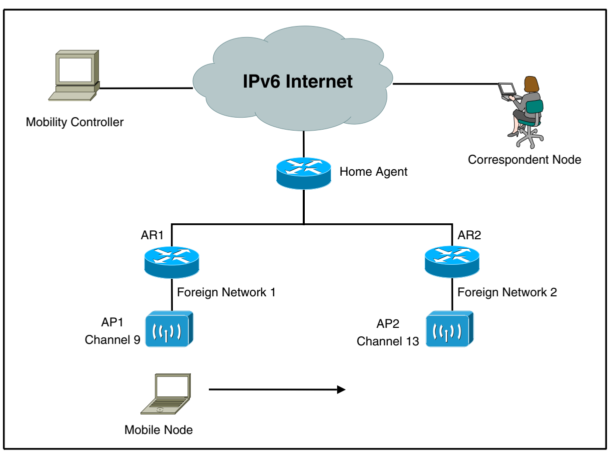

Fig. 1. Testbed used in the experiments

All network entities are running the GNU/Linux operating system, except for the APs which are 802.11b Cisco AP 1200 devices. The HA and the MN are both running the new MIPv6 daemon for the GNU/Linux operating system (MIPL [10]). For the evaluation of the FMIPv6 protocol, the MN and the access routers (AR1 and AR2) are also running the FMIPv6 Open Source Implementation Suite (FMIPv6.org [11]). For the evaluation of the GPS Handover protocol, the MN was running a modified version of the MIPL daemon [5].

During the evaluation of the standard MIPv6 protocol (and only then), we had configured all access routers to send RAs in intervals between 0.03 and 0.07 seconds as recommended by the MIPv6 specification [3] for performance reasons. As neither FMIPv6 nor the GPS Handover protocols depend on broadcasted RAs, we have set these intervals to their standard values [14], i.e. between 200 and 600 seconds, for all other experiments.

B. Selection of wireless devices

It is now well known that the time required by a layer 2 handover strongly varies according to the wireless devices [2]. In our first evaluation of the GPS Handover protocol [5], we have used a 3Com 802.11 a/b/g PCMCIA wireless card managed by the MADWiFi driver [15] that fits the GPS Handover specifications. Therefore, we have used the same wireless card and driver here.

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 retained the 3Com card and the MADWiFi driver for this implementation as well.

The MIPv6 protocol has been conceived as a reactive protocol and as such its implementations do not exert any control on the lower layers but rather wait for events that trigger various protocol procedures. In other words the decision for a link-layer reassociation is left to the driver, contrary to FMIPv6 and GPS Handover where the protocol itself is capable of starting it. Unfortunately, handovers controlled by the MADWiFi driver itself (as opposed to those controlled by upper layer entities) tend to be excessively lengthy, to an extent that would have made our MIPv6 experiments unrealistic, or at least - far from representing a common case.

We conducted a short evaluation comparing the previously mentionned 3Com and two Cisco 802.11b Aironet 350 PCMCIA cards with firmware versions: 4.25.30 (April 2002) and 5.60.17 (August 2005). Figure 2 presents the amount of time that a host experiences connection loss during a driver-managed handover with the corresponding card. The experiment was conducted while the node was receiving a flow of packets sent every $30ms$. As we can see, the 3Com card and the Cisco card using the 4.25.30 firmware cause longer intervals of connection disruption. The reason for this is that these cards scan all supported channels before associating with an AP. The 3Com card further lengthens the procedure by scanning 802.11a channels in addition to those supported by the 802.11b standard. The second Cisco card (firmware 5.60.17) completes its handover for an average of $33ms$ due to more advanced scanning algorithm. Therefore, we decided to use this card in all MIPv6 only experiments.

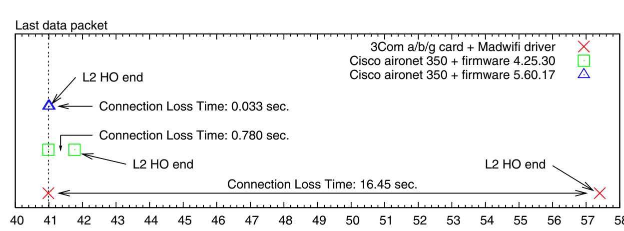

Fig. 2. Latency in layer 2 driver managed handovers

C. Evaluation Scenarios

We defined two scenarios for the evaluation of all protocols. In both of them an MN would move from AP1 to AP2 (see Figure 1). In the first scenario, the CN sends a video

TABLE I
RESULTS FROM SCENARIO 1 (VIDEO STREAM)

Protocols	Average L2 HO Proc. Length (ms)	Average Packet Loss	Average User Experience
MIPv6	22.4	3.67	Short flow interruption
FMIPv6	19.4	0	No interruption
GPS HO	19.4	1	Noise in audio

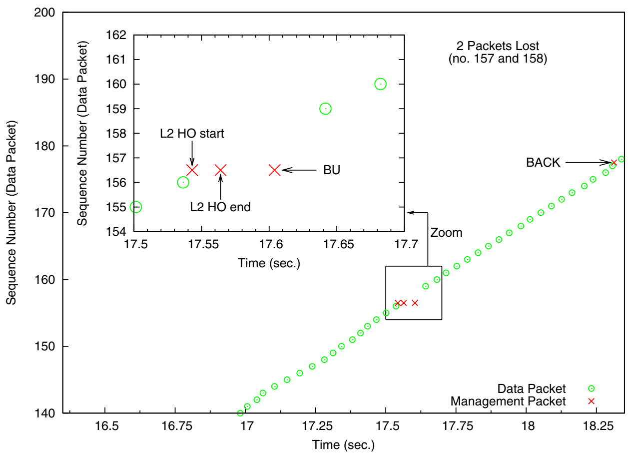

Fig. 3. Impact of the MIPv6 protocol on a Video Stream

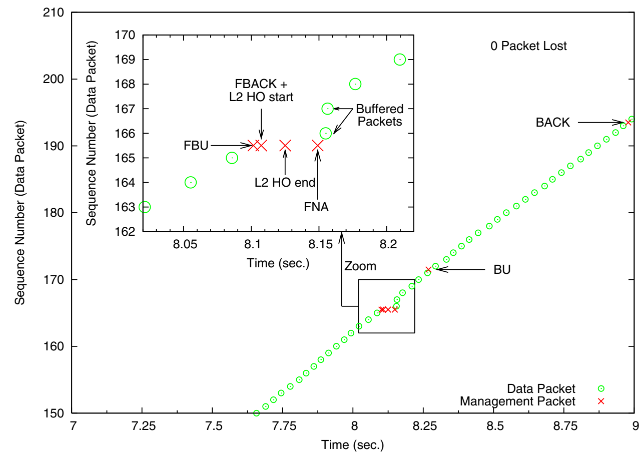

Fig. 4. Impact of the FMIPv6 protocol on a Video Stream

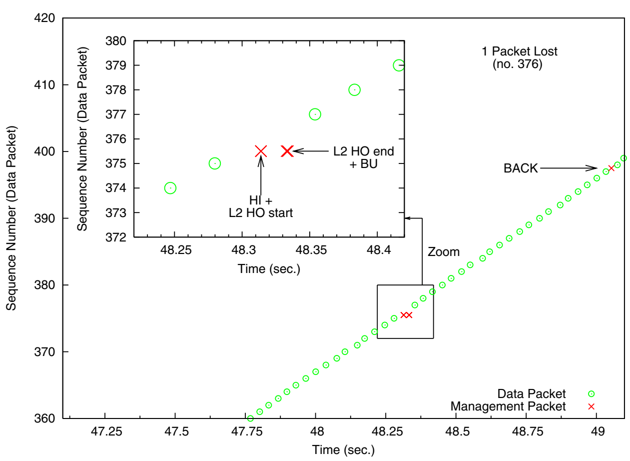

Fig. 5. Impact of the GPS Handover protocol on a Video Stream

stream to the MN using the VideoLAN application [8]. Data is encapsulated and sent in Real-time Transport Protocol (RTP) [16] packets, with an average length of approximately 1336 bytes and sent every $30ms$.

The second scenario includes a video conversation between the MN and a CN. They are both equipped with a webcam and use the Gnomemeeting software [9]. This application 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 $160ms$.

IV. PERFORMANCE EVALUATION

Results presented in this section are obtained by running the Ethereal tool [17] on both the MN and CN. We have also used additional wireless sniffers. For both scenarios, we have evaluated length of the layer 2 handover, number of data packets lost and user experience. Each scenario was run 10 times for every protocol.

Table I and Figures 3, 4 and 5 are related to scenario 1. We can see that time required by the layer 2 handover is approximately the same for all protocols. The FMIPv6 and GPS Handover protocols use a similar layer 2 handover procedure as for both of them the MN knows the parameters of its next AP prior to the handover and is able to associate with it without having to scan all supported channels first. The MIPv6 protocol achieves similar layer 2 performance because of the optimized handover management implemented by the Cisco card firmware (see Section III-B).

Concerning data reception, we can see that FMIPv6 protocol provides better results with an average of 0 packets lost per handover. This is due to the fact that FMIPv6 NARs buffer data packets destined for the MN until it arrives in their subnet. Upon reception of an FNA, buffering is stopped and buffered packets are delivered. This is clearly visible in Figure 4. Each dot represents transmission or reception of a packet at the time indicated on the X-axis. The sequence numbers indicated on the Y-axis correspond to RTP sequence numbers. The figure shows a delay in the reception of buffered packets but, because of application buffering, the delay remains undetectable for the application user making the handover completely seamless. Although the layer 2 handover is as fast as the one in the

TABLE II
RESULTS FROM SCENARIO 2 (VIDEO CONFERENCE)

Protocols	Average L2 HO Proc. Length (ms)	Average Packet Loss (Reception)	Average Packet Loss (Transmission)	Average User Experience
MIPv6	20.2	2.3 (audio) 0.4 (video)	2.3 (audio) 0.4 (video)	Short flow interruption
FMIPv6	22.5	0.3 (audio) 0 (video)	0.3 (audio) 0 (video)	No interruption
GPS HO	20.98	1.13 (audio) 0.13 (video)	0.88 (audio) 0 (video)	Noise in audio

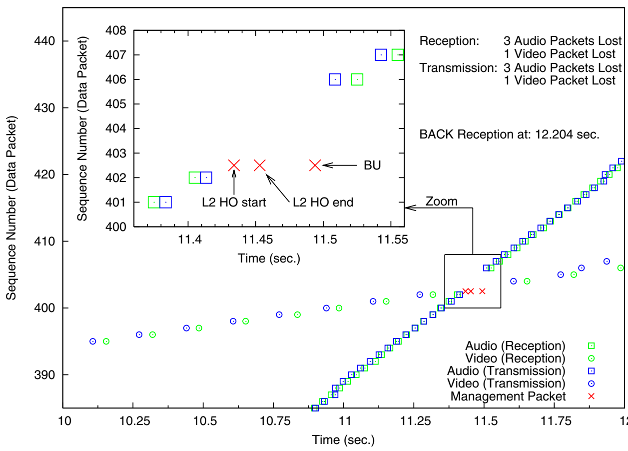

Fig. 6. Impact of the MIPv6 protocol on a Video Conference Stream

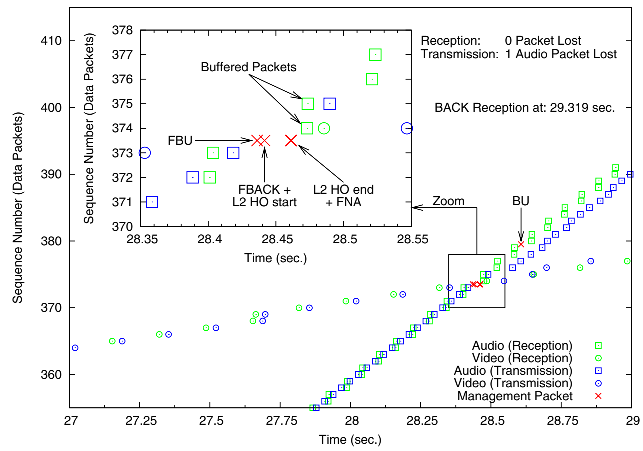

Fig. 7. Impact of the FMIPv6 protocol on a Video Conference Stream

FMIPv6 protocol and the BU is sent right after the completion of the layer 2 handover (see Figure 5), the MN loses a data packet in the GPS Handover protocol (1 packet lost on average). Until the Home Agent receives the BU, it goes on forwarding data packets to the previous location of the MN. These packets are neither buffered nor retransmitted and they never reach the MN. This packet loss may introduce noise perceptible for the user in the audio and/or video flows. The same issues are even more pronounced in the MIPv6 protocol tests. Due to slow movement detection mechanisms, the overall connection loss time is increased. Although the RAs are sent at the maximum frequency allowed by the MIPv6 specifications, the MN waits for approximately $50ms$ on average between the completion of the layer 2 handover and the transmission of a BU (see Figure 3). Therefore, the number of packet lost is increased up to an average of 3.67 per handover, introducing a short interruption in the rendered video stream.

The results related to the second scenario are presented in Table II and Figures 6, 7 and 8. In this scenario, the MN and the CN have established an audio video conversation. Therefore, the MN is both receiving and transmitting data packets. We can see that the results pertaining to reception are quite similar to those observed in scenario 1. Obviously, the MN loses less video packets than audio due to the different frequencies at which they are being sent (every $30ms$ for audio packets and every $160ms$ for video packets). The length variation of the layer 2 handover and the link detection mechanism explain the slight difference observed in the average packet loss values for the MIPv6 protocol between the scenarios 1 and 2.

Concerning transmission, the behaviour of each protocol

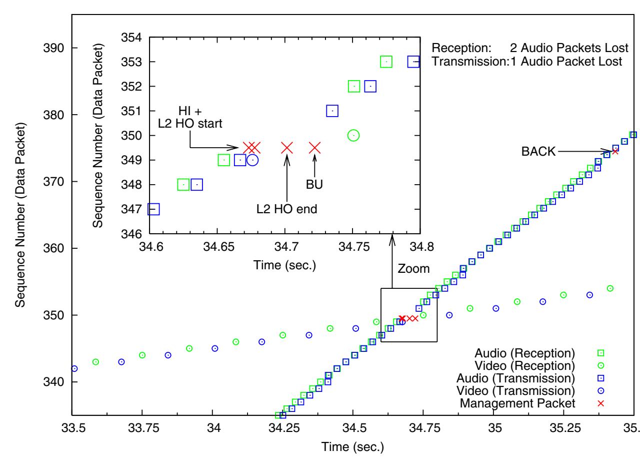

Fig. 8. Impact of the GPS Handover protocol on a Video Conference Stream

is close to those observed for reception. In the MIPv6 and GPS Handover protocols, this is due to the network distance between the MN and the HA. In our experiments, the MN can almost send data packets after the transmission of the BU (see Figures 6 and 8). In our testbed, RTT between the MN and the HA is negligible. Therefore, the HA starts forwarding data packets to the new location of the MN almost as soon as the MN starts using its new CoA. It is obvious that a larger delay to reach the HA will increase significantly the number of packets lost by the MN during reception. In the FMIPv6 case, the MN can still use its previous CoA to transmit data throughout the tunnel that it has established with the PAR. However, the MN has to perform periodically a complete scanning procedure in order to obtain the layer 2 parameters of surrounding APs (including the candidate AP). As discussed in Section III-B, the time required to scan all of the supported channels causes significant connection loss time. This problem is still left unresolved by the FMIPv6 specifications.

V. Conclusions and Future Work

Wide deployment of wireless technologies has allowed users to move while communicating. Moreover, the Mobile IPv6 protocol (MIPv6) [3] enables transparent movement across separate IPv6 subnets. Yet, the procedures accompanying such terminal mobility are often a cause for delays in ongoing communication. In this article, we have presented an analysis of such delays and their impact on time-sensitive communication. We have experimentally evaluated and compared performance of the MIPv6, the Fast Handovers for Mobile IPv6 (FMIPv6) [4] and the GPS Handover [5] protocols while the mobile node (MN) receives a video stream or participates in an audio video conversation.

The results presented in Section IV have shown that the MIPv6 protocol cannot achieve seamless handovers. Even with a fast layer 2 handover (which takes approximately $22.4ms$ in our experiments due to the selected wireless device and to the evaluation testbed) and when configuring the sending frequency of Router Advertisements at the highest possible rate, the MN still loses data packets whatever the video stream or video conference scenarios. The number of packets lost may make the handover process perceptible to the users (short flow interruption). In a real life environment in which the wireless device may be less efficient, the layer 2 handover latency may significantly increase this connection loss time [2].

The GPS Handover protocol permits the reduction of the number of packets lost as the MN can send the BU right after the completion of the layer 2 handover without requiring a high frequency of sent Router Advertisements. But the MN may still lose data packets because they are forwarded to its previous location while its binding is not up-to-date on the Home Agent. In our scenarios, a single lost packet may be enough to cause a user perceptible impact. In addition, a large network distance between the MN and the Home Agent will increase the time required to update the binding of the MN and subsequently will increase the number of packets lost in both standard MIPv6 and GPS Handover protocols.

Finally, the FMIPv6 protocol provides better results with no packet loss on average. Upon completion of the layer 2 handover, the MN can receive all data packets sent during the handover process and buffered by the new access router. We have also noticed that the small delays on the reception of such packets have no incidence on the perception of the users. During transmission, the MN can still use its previous care-of address to send data packets as long as the FMIPv6 tunnel is up. Therefore, the sending frequency of Router Advertisement and the RTT to the Home Agent have no repercussion on the FMIPv6 performance. The very moment where the MN is unable to receive or transmit is while performing the periodic full scanning procedure. Although the effective handover is seamless, the discovery of candidate APs occurring prior handovers still introduces large delays in ongoing communications. Without specific support, these stages make the seamless handover purpose meaningless.

Encouraged by the results presented here, we plan to combine the benefits of our previous solution (the GPS Handover) and the FMIPv6 protocol. The GPS Handover would provide a reliable method to identify the next access point of the MN. Moreover, this protocol also provides the layer 2 parameters of the next access point prior to the handover. By this means, the MN would always perform predictive handovers and fast layer 2 handovers. With such features, we have seen that the FMIPv6 protocol can achieve seamless handovers, even when using time-sensitive applications. Currently, we are investigating the modifications necessary for the FMIPv6 protocol to support the GPS Handover features.

References

[1] I. S. 802.11, “Wireless LAN Medium Access Control (MAC) and Physical (PHY) specifications : High Speed Physical Layer Extension in the 2.4 GHz Band,” 1999.

[2] 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.

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

[4] R. Koodli, “Fast Handovers for Mobile IPv6, Internet Engineering Task Force Request for Comments (RFC) 4068,” June 2005.

[5] J. Montavont and T. Noel, “IEEE 802.11 Handovers Assisted by GPS Information,” in *Proceedings of the 2nd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob’2006)*, June 2006.

[6] J. Lai, Y. Sekercioglu, N. Jordan, and A. Pitsillides, “Performance Evaluation of Mobile IPv6 Handover Extensions in an IEEE 802.11b Wireless Network Environment,” in *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC’06)*, June 2006.

[7] 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.

[8] A. Cellerier and Al., “VideoLAN - VLC media player, http://www.videolan.org.”

[9] D. Sandras and Al., “Gnomemeeting: an open source VoIP and video conferencing application for GNOME, http://www.gnomemeeting.org.”

[10] V. Nuorvala, H. Petander, and A. Tuominen, “Mobile IP for Linux (MIPL). http://mobile-ipv6.org.”

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

[12] 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.

[13] N. Moore, “Optimistic Duplicate Address Detection (DAD) for IPv6, Internet Engineering Task Force Request for Comments (RFC) 4429,” April 2006.

[14] 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.

[15] G. Chesson, M. Renzmann, and S. Leffler, “Multiband Atheros Driver for Wifi (MADWIFI), http://madwifi.org.”

[16] 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.

[17] G. Combs and al., “The Network Protocol Analyser Ethereal, http://www.ethereal.com.”