IEEE 802.11 Handovers Assisted by GPS Information

Julien Montavont
LSIIT(UMR7005) - ULP
bld Sebastien Brant
67400 Illkirch - FRANCE
Phone: +33(0)390-244-590
Email: montavontj@dpt-info.u-strasbg.fr

Thomas Noel
LSIIT(UMR7005) - ULP
bld Sebastien Brant
67400 Illkirch - FRANCE
Phone: +33(0)390-244-592
Email: Thomas.Noel@dpt-info.u-strasbg.fr

***Abstract*— IEEE 802.11 networks are now very common and are present in various locations. While roaming through access points, a mobile node is often required to perform a link layer handover. This mechanism causes user-interceptable connection loss and breaks in time-sensitive communication, especially if a network layer handover follows the link layer handover. Many solutions attempting to improve this process have been proposed but only a few use geolocation systems in the management of the handover. In this article, we present a new method to enhance both link layer and network layer handovers using geolocation information provided by a GPS system. The idea behind our algorithm is to predict the next mobile node point of attachment and the associated sub-network using the position of the mobile nodes. This method has been implemented using the new Mobile IP daemon for GNU/Linux operating system and evaluated through two scenarios.**

***Index Terms*— IEEE 802.11, Mobile IPv6, Fast Handover, Geolocation Assisted Handover**

I. INTRODUCTION

In the past few years, the IEEE 802.11 wireless technology (WIFI) has become very popular. With the increase of the WIFI throughput, real-time applications are now commonly used over this kind of network. However, the main issue when transmitting real-time traffic over WLAN is the handover process, happening when a mobile node (MN) moves outside the range of its current access point (AP). As the coverage of WIFI access points is relatively small, this process may occur often. Depending on the sub-network of the target access point, the link layer handover can be followed by a network layer handover. A common solution to manage this process is the Mobile IPv6 protocol [1]. The handover process introduces user-interceptable connection loss which is particularly serious in real-time communication. Many proposals focus on improving link layer handover [2], network layer handover [3] or both [4]. New schemes propose using a geolocation system to predict the next point of attachment for an MN. However, these solutions focus on cellular networks [5], heterogeneous wireless networks [6] or only try to improve the network layer handover [7]. In this article, we present a new method which enhances both link layer and network layer handover in IEEE 802.11 IPv6 networks using the GPS geolocation system. We have decided to use the GPS system because it is one of the easiest geolocation systems to set up and use, but other systems with similar characteristics could also be used. Each MN is equipped with a GPS receiver and sends its current position to a new network entity called a GPS Server. This host tracks movement of the MN and initiates the handover process whenever necessary. According to the position of an MN, it determines in advance its next point of attachment and sets the necessary parameters in order to minimise the time required by the handover process.

This document is organized as follows. First, we present a brief overview of the handover management using the IEEE 802.11 standard [8], the Mobile IPv6 protocol [1] and geolocation information [5], [6], [7]. Section III describes our handover enhancement using geolocation information, followed by a section about the algorithm implementation and scenario set up. Finally the measurement results and the conclusions are presented in sections V and VI.

II. RELATED WORK

This section describes the protocols currently used to manage layer 2 and 3 mobility over IEEE 802.11 wireless IPv6 networks. It aims to deal with the main issues regarding real-time communication requirements. In addition, we present solutions using geolocation information in the handover management.

A. IEEE 802.11 Standard Handover

A layer 2 handover (or link layer handover) refers to an MN physically changing its point of attachment. If an MN roams between two APs of the same sub-network, no routing issues occur and the MN can continue its ongoing communication. In the IEEE 802.11 standard [8], the layer 2 handover process is divided into three steps: discovery, authentication and association. These steps are illustrated in Figure 1.

During the discovery phase, an MN would broadcast Probe Request frames in order to discover surrounding APs over all of the IEEE 802.11 channels. The MN waits for MinChannelTime or MaxChannelTime (two different amounts of time defined in [8]) per channel according to Probe Response receptions. After scanning all IEEE 802.11 channels, it selects a target AP and starts the authentication procedure. This consists of

1-4244-0495-9/06/$20.00 ©2006 IEEE

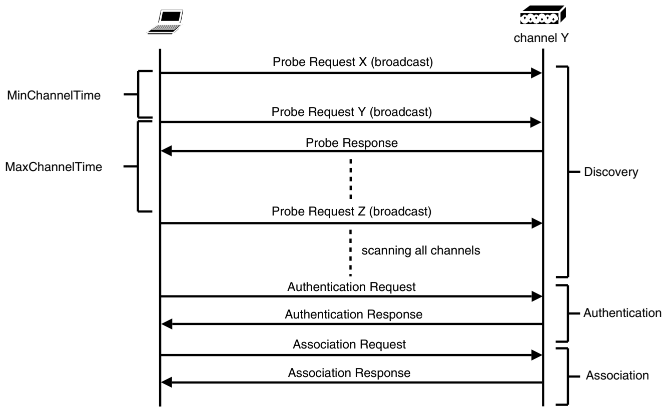

Fig. 1. Standard IEEE 802.11 Association

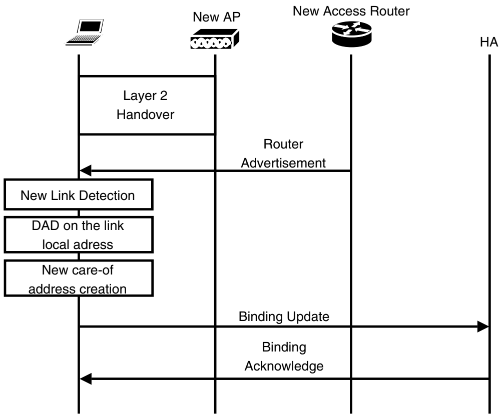

Fig. 2. Layer 3 Handover managed by the MIPv6 Protocol

the transmission of the MN's identity to the AP which may accept or reject the request using an Authentication Response. If the authentication is successful, the MN starts the association process. The layer 2 handover is complete when the MN receives an Association Response with a success status code.

The layer 2 handover, as described in the IEEE 802.11 Standard [8], takes too much time (between $58.74ms$ and $396.76ms$ according to the results found in [9]) for real-time communication to continue operating seamlessly. This is mainly due to the discovery phase which takes 90% of the layer 2 handover latency [9]: even if no access points are operating on a specific channel, the MN has to scan this channel and waits for MinChannelTime in vain. In addition to the layer 2 handover, the MN may be required to perform a layer 3 handover.

B. Mobile IPv6

The layer 3 (or network layer) handover refers to an MN roaming to a new AP in a sub-network other than the previous one. To resolve routing issues, the MN has to change its current IPv6 address. Without specific support, this change would affect ongoing communication. To manage layer 3 mobility over IPv6 networks, the Internet Engineering Task Force has defined the Mobile IPv6 protocol (MIPv6) [1]. After a successful layer 2 handover, the MN may be required to perform a layer 3 handover depending on the new AP's sub-network. The new link detection is based on the reception of Router Advertisement messages, which are periodically sent by access routers. After receiving such a message, an MN knows that it is in a new sub-network and performs a Duplicate Address Detection process [10] in order to verify the uniqueness of its link local address. Then, the MN can create a new care-of address using the IPv6 stateless address configuration mechanism [10] and sends a Binding Update to its Home Agent (HA) in order to update its binding. The layer 3 handover managed by the Mobile IPv6 protocol is illustrated in figure 2.

The layer 3 handover latency varies widely depending on the frequency of the Router Advertisement messages. Statistically, the longer the time between two consecutive Router Avertisements, the more it takes for the movement detection to be completed. According to [1], Router Advertisement broadcasts, when transmitted at the fastest rate allowed, must still be separated by at least 0.03 to 0.07 seconds. This allows for rapid new link detection, but consumes considerable bandwidth. In addition to the movement detection, the round trip time between the MN and its HA (needed to update the MN's binding) contributes to the total delay of the layer 3 handover. According to results found in [11], the layer 3 handover takes approximately $732ms$ to complete when using a local HA and access routers configured to send Router Advertisements between 0.05 and 0.15 seconds (minimum authorized values when measurements in [11] take place).

C. Geolocation in Handover Management

In the past few years, it has been proposed that geolocation of MNs be used for handover management in a wireless environment. The first usage of geolocation systems to assist mobility support was in cellular networks. In [5], the authors propose using the vector of the mobile node's movement and its velocity to resolve the well known ping-pong and far-away cell effects. This work is still in progress and the authors describe briefly the mechanism without providing any performance evaluation. Geolocation information can also be used in heterogeneous networks to select the most appropriate wireless technology for communication. In [6], the authors present a location assisted algorithm to manage handovers between WLAN and GPRS networks. A new network entity monitors movement of MNs and detects when an MN moves inside the coverage area of a WLAN. According to various parameters, such as velocity, direction and ongoing traffic of MNs, it can estimate if a handover is appropriate. The only mechanism for avoiding packet loss during handover is to forward data traffic over the two technologies until the end of the handover. The mobile IP protocol can also be improved using geolocation information [7]. Wireless sensors are placed at the border of AP's coverage areas and detect the movement of MNs. A wireless sensor can only detect the movement of an MN between two specific APs, and this allows the prediction of the next MN's foreign agent. When a sensor detects that an MN moves out of the coverage area of its current AP, the current MN's foreign agent informs the MN about this event, which allows the pre-registration of the new care-of address with the HA. Upon the reception of a pre-registration request, the HA creates a simultaneous binding and forwards the mobile node's data traffic to both care-of addresses. Although this mechanism allows the reduction of the layer 3 handover latency, it does not provide any solution to minimize the delay introduced by a layer 2 handover. In addition, this solution is designed for IPv4 networks and there is no guarantee that it works in IPv6 networks.

III. Algorithm Description

The idea behind our method is to use the position of the MN as a metric to improve both layer 2 and 3 handovers in IEEE 802.11 IPv6 networks. To achieve this purpose, we focus on reducing the layer 2 discovery phase and the layer 3 new link detection (two significant points introducing latency in the handover process as previously described). Many efficient proposals make the reduction of the delay introduced by large network distance between MN and HA possible using a hierarchical architecture ([12] and [13]). We simulate such a framework by considering in the following that the GPS Server, MNs and the associated HA are quite close in terms of the number of hops.

Each MN is equipped with a GPS receiver and periodically sends to a new network entity known as the GPS Server, a Location Update message (LU), which contains its coordinates. The GPS system is passive and only requires a GPS receiver device and enough line of sight satellites to operate. A GPS receiver estimates an MN's position at every second, with an accuracy of 10 meters. Handover management is controlled by the network. The GPS Server is a stand alone server and is located inside a network domain. It determines the next access point for each MN according to its location and initiates the handover process whenever necessary (see section III-B). The selection of the next MN's access point prior to the handover allows the configuration of all the required parameters on the MN to reduce the layer 2 discovery phase and the layer 3 new link detection.

A. GPS Server Update

The GPS Server maintains a list of domain access points including some information such as the AP's coordinates (latitude and longitude), the IEEE 802.11 channel on which the AP is operating, the AP's Service Set Identifier (SSID) and the IPv6 prefix. These parameters are statically configured.

At each position check (performed every second when using the GPS geolocation system), an MN records its current coordinates and compares them to the previous ones in order to determine if it has moved. The distance between two points is calculated using the Haversine formula. It assumes a spherical Earth and ignores ellipsoidal effects but remains particularly well-conditioned for numerical computation even at small distances. Let us denote the previous and the current coordinates of an MN as $(lat_1, long_1)$ and $(lat_2, long_2)$ respectively. Let us also denote the latitude separation with Δ_{lat} and the longitude separation with Δ_{long}, where angles are in radians, and R the Earth's radius ($R = 6,371km$). The distance d between the two points is calculated by the formula:

$$\begin{aligned} haversin\left(\frac{d}{R}\right) &= haversin(\Delta_{lat}) + \cos(lat_1) \\ &\quad \times \cos(lat_2) \times haversin(\Delta_{long}) \end{aligned}$$

where the Haversine function is given by:

$$haversin(\delta) = \sin^2\left(\frac{\delta}{2}\right)$$

Let h denote the $haversin(d/R)$. One can then solve for d either by simply applying the inverse Haversine or by using the arcsin (inverse of sine) function:

$$d = R \times haversin^{-1}(h) = 2R \times \arcsin(\sqrt{h})$$

If d is greater than 1 meter, we consider that the MN has moved and has to send to the GPS Server a LU message, which includes the identity of its current AP and its current coordinates.

B. Handover Management

Once the GPS Server receives a LU message, it checks the distance between the MN and its current AP. We define the G threshold, which corresponds to the maximum distance between an MN and its associated AP where the MN is not considered to be close to the border of its AP's coverage. If this distance is below the G threshold, the MN is still in the range of its current AP and is not required to perform a handover. Otherwise the MN is going out of the coverage area of its current AP and the GPS Server has to determine a target AP for the pending handover. Among all APs in the GPS Server list, the GPS Server chooses the closest one to the MN as the target AP. If the target AP is not the same as the current one, the GPS Server sends a Handover Initiate (HI) message to the MN. The HI message contains the target AP's 802.11 channel, SSID, and sub-network IPv6 prefix. According to the measurements found in [4], we decided to set the G threshold to 50% of the AP's range.

Upon the reception of an HI message, the MN starts a handover process and tries to associate to the target AP using the relevant information included in the HI message. As the MN already knows its handover destination, it only sends a Probe Request over the target AP's channel and waits for the target AP's specific Probe Response. Upon the reception of a Probe Response from the target AP, the MN moves forward in the handover steps and starts the authentication process. Therefore, the discovery phase consists only of one Probe Request / Probe Response exchange with the target AP, which significantly reduces the layer 2 handover latency. The MinChannelTime and MaxChannelTime values are no longer used except for the error cases (see section III-C).

After successful association, the MN checks if the HI message includes a new IPv6 prefix implying that the target

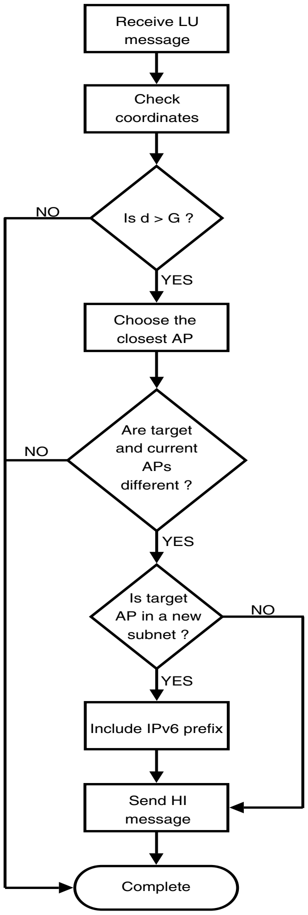

Fig. 3. Handover Management on the GPS Server

AP is in a sub-network different from the previous one. Therefore, the new link detection is no longer based on Router Advertisement reception. Upon successful association to the target AP, the MN can configure a new care-of address using the IPv6 prefix included in the HI message and immediately sends a Binding Update (BU) to its Home Agent (HA). After the reception of a successful Binding Acknowledgement, the handover is completed and the MN can continue its ongoing communication. The handover management is illustrated in Figure 3.

C. Error cases

Different errors may occur, which may affect the algorithm performance. First, geolocation systems estimate positions with an error which depends on their accuracy (10 meters for the GPS system). This geolocation error may affect the AP determination. The GPS Server may select an AP which is out of the MN's range or initiate the handover process while the MN has not crossed the *G* threshold. In such cases, an MN would wait for the expiration of a timer (i.e. MinChannelTime or MaxChannelTime) before considering that the target AP is out of range. If there is at least one responding AP over the target AP channel, the MN tries to associate to it. Otherwise, it switches its channel and starts a new scan as described in [8]. After a successful association, the movement detection is performed using Router Advertisement messages as described in [1].

Other significant errors refer to the unavailability of the GPS Server or of the geolocation system. In both cases, if the GPS Server does not respond to an MN for some reason (e.g. the GPS Server is down, the MN using the GPS system enters a building), the MN stays associated to its current AP until it moves out of its range. As the GPS Server did not provide any target AP, it starts a standard layer 2 handover [8] followed by a layer 3 handover managed by [1] whenever necessary.

Therefore, when errors occur, we expect performance to be similar to what we have with [8] and [1].

IV. Experiment Set Up

In order to evaluate our proposal, we implemented it and set up a testbed composed of three access points, three laptops and two desktop computers. The access points are 802.11b Cisco AP 1200 devices. The desktops are a 3.5Ghz and a 3Ghz Intel P-IV units with 512 MB of RAM. The laptops are 900MHz Intel P-III units with 256 MB of RAM. All of them run the Debian GNU/Linux operating system.

The first desktop computer is the GPS Server. The second one is the Home Agent running the release candidate 3 of the new MIPv6 daemon (mipv6-2.0-rc3 [14]). The first laptop is equipped with a WIFI PCMCIA card managed by the MADWIFI driver [15], a GPS receiver PCMCIA card and runs the same MIPv6 daemon version as the Home Agent. Finally, the two other laptops have two PCMCIA cards each and are used for traffic sniffing.

A. Implementation

As our scheme improves both layer 2 and 3 handovers, we have modified the MADWIFI driver and the MIPv6 daemon.

The MADWIFI driver modification is quite simple and consists of two parts. The driver's default behavior is to reset the card after each command modifying a parameter (such as SSID, WEP key, etc). As the card reset takes a significant amount of time, we decided to reset the card only when we configure the channel. When a handover is required, we set the SSID first, and the channel which implies a card reset. The second part addresses the IEEE 802.11 discovery phase. During this stage, the MN broadcasts a Probe Request which includes a specific SSID over the configured channel. Once the MN receives a Probe Response corresponding to this SSID, the MN moves forward in the driver state machine and starts the authentication process. Therefore, the MinChannelTime and MaxChannelTime values are only used in error cases as described in section III-C.

In regard to the MIPv6 daemon modification, we add a new socket to handle the GPS Server's HI messages. Once an MN receives an HI message, the MIPv6 daemon intercepts it and launches the layer 2 handover using IOCTL routines to supply layer 2 information to the MADWIFI driver. When the layer 2 handover is complete, the MIPv6 daemon acts as if

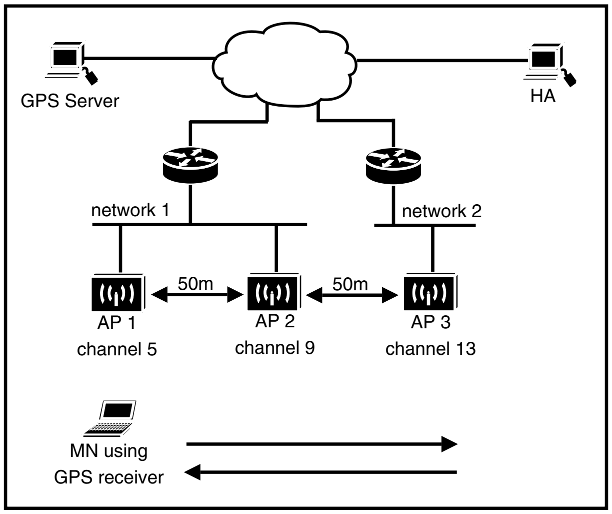

Fig. 4. Experiment Scenarios 1 and 2

it has received a new Router Advertisement (RA). This RA is included in the HI message and is similar to the ones sent by access routers over the target link. Therefore, the MIPv6 daemon configures a new care-of address and sends a BU to the HA.

In addition, we developed a light weight daemon which monitors coordinates from the GPS receiver and sends them to the GPS Server.

B. Evaluation Scenarios

This section describes two scenarios that we use to evaluate our protocol. As the GPS system does not work inside buildings, our scenarios take place in an outdoor environment. Note that any other geolocation system with at least the same characteristics can be used instead of the GPS system (e.g. indoor geolocation systems). Scenario 1 refers to a walking user moving outdoors. It involves 3 access points and one MN. They are placed on a line at a distance of 50 meters from each other. The MN roams between two different sub-networks. AP1 and AP2 are in sub-network 1, and AP3 is in sub-network 2. The channel allotment is illustrated in Figure 4. The MN movement is rectilinear and starts from AP1 to AP3 and goes back in the reverse direction. Therefore, the MN performs four layer 2 handovers and two layer 3 handovers. The HA and the GPS Server are quite close to the two sub-networks (in terms of the number of hops), which simulates a hierarchical architecture. The scenario 1 is illustrated in Figure 4.

The second scenario tries to evaluate our algorithm's behavior when the mobile node receives data traffic. Scenario 2 uses the same topology as scenario 1. The GPS Server acts as a responding peer and emits a continuous G.711 [16] audio flow. Packets are 1280 bits long and are sent every $20ms$ (i.e. G.711 codec specification). The data flow is generated by the MGEN tools [17]. Note that the Route Optimization mechanism [1] is not used in our experiments.

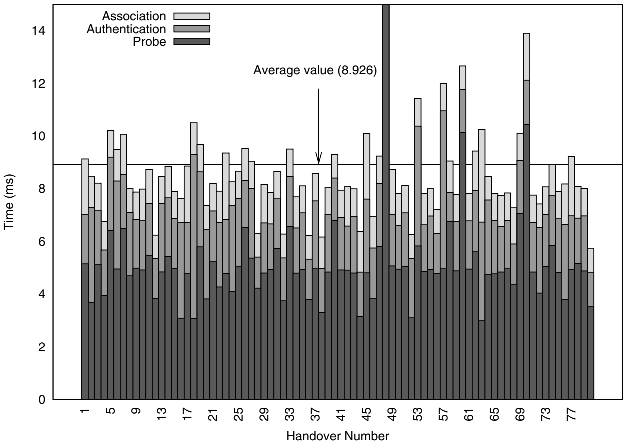

Fig. 5. Layer 2 Handover Latencies

V. Experiment results

The results shown in Table I, Figures 5 and 6 are obtained by using the sniffers previously described and the Ethereal tool [18]. Figure 5 shows layer 2 handover latencies for scenario 1 (i.e. without data flows). This scenario has been played 20 times including a total of 80 layer 2 handovers. It appears that our scheme allows the layer 2 handover to be completed in $8.926ms$ on average. This time is drastically reduced in comparison to the standard layer 2 handover latency (varying between $58.74ms$ and $396.76ms$ according to the results found in [9]). This is mainly due to the time required by the discovery phase. In our proposal, MNs scan only the channel of the target AP and move forward in the handover process upon the reception of a target AP Probe Response without waiting for MinChannelTime or MaxChannelTime values. Adjacent APs sending Beacon frames and MNs trying to associate with surrounding APs explain the fluctuations seen in the results. The large standard deviation presented in Table I is mainly due to the handover number 48 case, where the layer 2 handover takes $48.153ms$ to complete. If we do not consider this measurement, the standard deviation would be close to $1ms$. For the confidence interval shown in Table I, 95% of the layer 2 handover latencies are included in the interval $[7.905, 9.947]$.

When the MN moves between AP2 and AP3, the layer 2 handover is followed by a layer 3 handover. In order to evaluate our protocol performances concerning the layer 3 handover latency, we focus on three different events: the end of the layer 2 handover, Binding Update (BU) transmission and Binding Acknowledgement (BACK) reception (see section II-B for more details). Upon the reception of the BACK, the layer 3 handover is complete. Figure 6 shows the different times at which these events occur. The average values presented in Figure 6 and on line 2 of Table I are related to 40 handovers between AP2 and AP3. As we can see, our proposal allows the layer 3 handover to complete in $27.334ms$ on average. Although new link detection takes place immediatly after the end of the layer 2 handover, it still requires approximately

TABLE I
RESULTS FROM SCENARIOS 1

Handover Layer	Number of Handovers	Average values of Connection Loss Time (ms)	Std. Deviation (ms)	Confidence Interval (ms)
layer 2	80	8.926	4.660	$\Delta = 1.021$
layer 3	40	27.334	4.734	$\Delta = 1.467$

$14.035ms$ to the MIPv6 daemon to carry out the necessary operations before sending the BU message (parses RA included in the HI message, creates a new care-of address, updates the tunnel end-point at MN side, etc). The BACK message arrives approximately $4.724ms$ after the BU transmission. This means that the HA is quite close to the MN in terms of the number of hops, which simulates a hierarchical architecture. According to the mesurements found in [11], our protocol significantly reduces the layer 3 handover latency in comparison to the classical Mobile IPv6, where the layer 3 handover takes approximately $732ms$ to complete. This is mainly due to the time required by the layer 2 handover and the new link detection mechanism. Although the frequency of Router Advertisement is set between 0.05 and 0.15 seconds in [11], this step still requires too much time to complete. In addition, a high frequency of Router Advertisements consumes considerable bandwidth. By contrast, our protocol allows fast new link detection without introducing a significant load on the wireless link. For the confidence interval shown on line 2 of Table I, 95% of the layer 3 handover latencies are included in the interval $[25.867, 28.801]$.

Finally, Figure 7 illustrates the impact of layer 2 and 3 handovers on the data reception in the second scenario. Each dot represents the reception of a packet, at the time indicated on the Y-axis. The second scenario has been played 20 times. Based on our measurements on the time needed for the layer 2 and 3 handovers to complete ($8.926ms$ and $27.334ms$ respectively), we have expected one lost packet in the worst case (i.e. during layer 3 handover), when using G.711 data flow. However, in reality one packet is lost during layer 2 handover and two during layer 3 handover (on average). This is certainly due to the time required to send the target AP's parameters to the MADWIFI driver followed by the reset of the card.

Note that none of the errors presented in section III-C have been evaluated in our experiment. This is explained by the simplicity of our evaluation scenarios, where movement is rectilinear and continuous. We are planning to evaluate more precisely our protocol through more realistic scenarios.

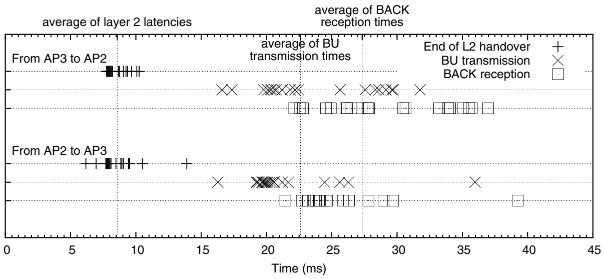

Fig. 6. Layer 3 Handover Events - each point represents one event

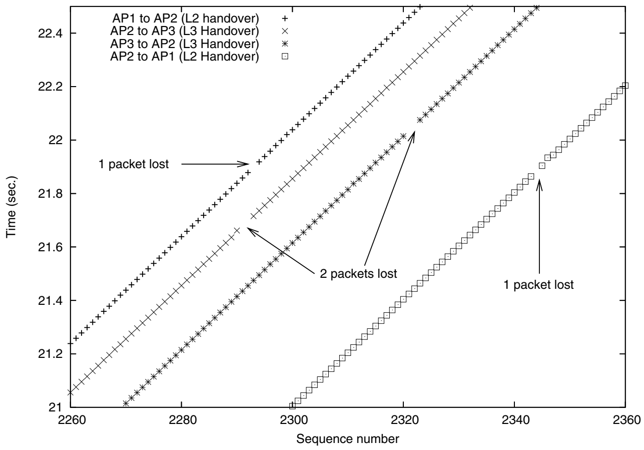

Fig. 7. Impact on Flows

VI. Conclusions and Future Work

In IEEE 802.11 networks, an MN moving out of the range of its current AP is forced to go through a sequence of procedures to regain network connectivity. This sequence is referred to as link layer or layer 2 handover. Depending on the new point of attachment sub-network, an MN may be required to perform a network layer (or layer 3) handover to resolve routing issues. Those mechanisms, as described in [8] and [1], involve connection loss and breaks, which are especially serious in time-sensitive communication. In this article, we present a new scheme to enhance both layer 2 and 3 handovers using GPS information. Our solution focuses on the reduction of the time required by the layer 2 discovery phase and the layer 3 new link detection, which are two of the significant points introducing delays in the handover process. This proposal has been implemented using the new MIPv6 daemon for GNU/Linux operating system [14] and the MADWIFI driver [15]. It has been evaluated through two scenarios.

Results presented in section V show that our proposal drastically reduces the layer 2 and 3 handover latencies to $8.926ms$ and $27.334ms$ respectively, which seem acceptable for real-time communications. As we use GPS information to predict the next MN point of attachment, we can send the target AP's parameters (SSID, channel, sub-network IPv6 prefix) prior to the pending handover. By this means, the layer 2 discovery phase only consists of one Probe Request / Probe Response exchange with the target AP. In addition, an MN knows when it moves to a new sub-network immediatly after the end of the layer 2 handover. Therefore, it can configure a valid care-of address and update its binding to the Home Agent without waiting for Router Advertisement receptions. By this means, the frequency of Router Advertisements can be reduced in order to save bandwidth.

Our future work in this area is to evaluate more precisely our proposal through more realistic scenarios. We expect to benefit from the Louis Pasteur University WLAN network deployment to extend our performance studies to large scale experiments and more error cases, such as those described in section III-C. In addition, we plan to design an extended next point of attachment selection, in order to reduce the potential impact of the geolocation error on the algorithm. We also plan to extend our protocol to allow MNs to switch between several geolocation systems (e.g. indoor and outdoor). Currently we are considering the security aspects of our solution in order to prevent malicious users to act as a fake GPS Server or to take the identity of a valid user when corresponding with the GPS Server.

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] J. Montavont, N. Montavont, and T. Noel, "Enhanced schemes for L2 handover in IEEE 802.11 networks and their evaluations," in *Proceedings of the IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'05)*, September 2005.

[3] C. Perkins and K.-Y. Wang, "Optimized smooth handoffs in Mobile IP," in *Proceedings of IEEE International Symposium on Computers and Communications*, July 1999, pp. 340 – 346.

[4] N. Montavont and T. Noel, "Anticipated Handover over IEEE 802.11 Networks," in *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2005)*, August 2005.

[5] S. Kyriazakos, D. Drakoulis, and G. Karetsos, "Optimization of the Handover Algorithm based on the Position of the Mobile Terminals," in *Proceedings of Symposium on Communications and Vehicular Technology*, October 2000.

[6] J. Pesola and S. Pokanen, "Location-aided Handover in Heterogeneous Wireless Networks," in *Proceedings of Mobile Location Workshop*, May 2003.

[7] V. Bahety and R. Pendse, "Scalable QoS Provisioning for Mobile Networks using Wireless Sensors," in *Proceedings of IEEE International Conference on Wireless Communications and Networking (WCNC'04)*, March 2004.

[8] 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.

[9] 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.

[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, "Stronger Interaction between Link Layer and Network Layer for an Optimized Mobility Management in Heterogeneous IPv6 Networks," *Elsevier, Pervasive and mobile computing*, Accepted for publication.

[12] A. Valko, "Cellular IP: A New Approach to Internet Host Mobility," *ACM SIGCOMM Computer Communication*, vol. 29, pp. 50 – 65, January 1999.

[13] S. Pack, M. Nam, and Y. Choi, "A Study on Optimal Hierarchy in Multi-Level Hierarchical Mobile IPv6 Networks," in *Proceedings of the 47th annual IEEE Global Telecommunications Conference (GLOBECOM'04)*, November 2004.

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

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

[16] "ITU-T Recommendation G.711. Pulse Code Modulation (PCM) of Voice Frequencies," 1972.

[17] PROTEAN research group, "The Multi-Generator Toolset (MGEN), http://pf.itd.nrl.navy.mil/mgen/."

[18] G. Combs and al., "The Network Protocol Analyser Ethereal, http://www.ethereal.com."