Experimental Evaluation of a Handover Optimization Solution for Multimedia Applications in a Mobile-IPv6 Network

Carlos Jesús Bernardos[1], Ignacio Soto[1], Jose Ignacio Moreno[1],
Telemaco Melia[2], Marco Liebsch[2], Ralf Schmitz[2]

[1]Universidad Carlos III de Madrid – Dep. Ingeniería Telemática
Av. Universidad, 30; 28911 Leganés, Madrid – España
Telephone: +34 916248859 Fax: +34 916248749
E-mail: {cjbc, isoto, jmoreno}@it.uc3m.es
[2]NEC Network Laboratories, NEC Europe Ltd., Heidelberg – Germany
E-mail: {melia, liebsch, schmitz}@ccrle.nec.de

Abstract. The EU IST project Moby Dick worked with the vision, shared by many other researchers, that Next Generation Networks will be based on IPv6 with mobility, Security and Quality of Service support. These networks will offer all kind of services, including multimedia ones with real-time requirements, traditionally offered by circuit switched technologies. The IETF is finishing the standardization of a solution for mobility in IPv6 networks: Mobile IPv6. Additional protocols are being discussed to improve the performance of Mobile IPv6 to support real-time traffic during handovers; one of these proposals is Fast Handovers for Mobile-IPv6. This paper analyzes experimentally – using real implementations for Linux O.S. - the performance of Mobile IPv6 and Fast Handovers for Mobile IPv6, to study if the performance is acceptable for multimedia applications. Both quantitative measurements and results of quality perceived by users of IPv6 multimedia applications are provided, showing that Fast Handovers for Mobile IPv6 approach is good enough even for multimedia applications with strict real-time requirements.

I. INTRODUCTION

The IST project Moby Dick [1] defined, implemented and evaluated an entirely IPv6-based architecture integrating support for Quality of Service (QoS), IP(v6)-based mobility management as well as Authentication, Authorization and Accounting (AAA). In this framework, the deployment of an IPv6 mobility management solution is of great importance, in order to be able to provide uninterrupted and low jitter real-time multimedia applications (e.g. real time audio and video streaming or VoIP) to the end-user, even in the case of handovers. Moby Dick combined this mobility management with QoS and AAA to offer a secure and QoS-enabled mobile communications platform.

Internet is changing in the last years. The number of mobile terminals is growing and the current Internet protocol (IPv4) was not designed taking into account terminal mobility. Actually, IPv4 was designed for static hosts, with a narrow relation between their network address and their physical location. Therefore, the IP address was configured statically for the particular network they were attached to.

In the last years some protocols for dynamic assignment of IP addresses to nodes joining a network segment (e.g. DHCP [2]) have been designed and deployed, but these solutions provide portability and not transparent mobility. By portability we mean terminal mobility that allows a host to change its location, but it requires stopping and restarting its upper layer connections (e.g. TCP). Transparent mobility allows a terminal to move among different networks without stopping any connection. IPv6 [3] provides portability because of its mechanisms for easy automatic address configuration, but not transparent mobility.

Mobile IPv4 [4] and Mobile IPv6 [5] are the protocols defined to provide support for reachability and transparent mobility in IPv4 and IPv6 networks.

In this paper we present an experimental study of the handover latency characteristics of Mobile IPv6 (MIPv6) [5] and Fast Handovers for Mobile IPv6 (FMIPv6) [6], based on real implementations of these protocols. This study provides both quantitative (handover latency figures under different network conditions) and qualitative (user evaluation of the protocols performance), using software implementations of the two approaches mentioned above. The main focus of the experiments was to evaluate the performance of each solution and investigate respectively if it is suitable for real-time multimedia applications.

II. RELATED WORK AND MOTIVATION

There are some previous analytical and simulation studies related to handover latency of different mobility management approaches. Some of them ([7], [8]) have also been done within the framework of the Moby Dick Project. The main conclusions of these studies were:

- Fast Handovers and Hierarchical [7] approaches reduce significantly handover latency (and therefore packet loss), compared with Mobile IPv6.
- Fast Handovers approach reduces handover latency during handover more than a Hierarchical approach.
- A combination of a Hierarchical approach with the Fast Handovers approach reduces latency even more than any of them alone.

Although the combination of Fast Handovers with the hierarchical approach showed slightly better results, within the framework of the Moby Dick project the Fast Handovers approach was deployed alone because the combination with the hierarchical solution adds a significant amount of network complexity, while it was not clear if the improvement was required for getting adequate performance, a question that we wanted to answer with the work presented in this article.

Other research projects have analyzed different IP mobility management schemes. IST WINE GLASS ([9], [10]) used Mobile IPv6 to handle IP mobility but it did not implement any solution to optimize local mobility at the IP layer. The efficiency in local mobility depended on layer-2 technologies and it was restricted to mobility within the same IP subnet.

IST BRAIN and MIND ([11], [12]) projects proposed their own local mobility management solution: BCMP (Brain Candidate Mobility Protocol). This protocol combines properties of the IETF hierarchical solutions (like HMIPv6) and IETF Fast Handovers solutions. The solution is composed by two types of elements with special mobility functionality: Anchor Points and Access Routers. Anchor Points are special routers that provide addresses to the visiting mobile nodes in a set of IP subnets, and tunnel packets to the mobile nodes. Access Routers provide access to the mobile nodes, terminate the tunnel from the anchor point and forward the packets to the Mobile Node (MN). When a mobile node moves from an Access Router to a new Access Router, the first Access Router tunnels packets to the new one during the handover, until the Anchor Point is informed that it must tunnel the packets destined to the mobile node to the new Access Router. The solution is quite efficient and according with [13] it can achieve a handover delay as low as 5 ms (this is only layer-3 handover delay, the layer-2 handover delay in the configuration used in the experiments was fixed to 150 ms). Nevertheless, the infrastructure required by this solution is quite complex: there are Anchor Points, Access Routers, tunnels configurations between Anchor Points and Access Routers and tunnels between Access Routers. Besides, to provide addresses to the mobile nodes, the IPv6 stateless address autoconfiguration mechanism cannot be used and the stateful one must be adapted because the addresses of the mobile nodes must belong to the Anchor Point and not to the subnet they are visiting.

In this paper we present results of handover latency obtained by experiments in a real scenario with real implementations, modifying the network characteristics using the NISTNET [14] emulator. This paper also presents qualitative results from tests with real users, in which they showed their opinion about the performance of the different mobility solutions, rating the quality of the reproduction of video and audio streaming in a mobile node.

The two approaches analyzed were basic Mobile IPv6 support compared to Mobile IPv6 with Fast Handovers enhancement. We wanted to confirm the analytical and simulation results in a real scenario. Also, we wanted to find out which mobility management solution was needed for real-time multimedia applications from the point of view of latency in handovers:

- Is basic Mobile IPv6 support sufficient?
- Does the Fast Handovers extension provide suitable performance?
- Is there a need for optimization, e.g. the combination of a Hierarchical approach with the Fast Handovers approach?

The IETF is working now in standardize hierarchical and fast handovers solutions as experimental protocols. This is because it is acknowledged that they are important solutions but much more practical experience with these protocols is needed to discover its properties and usefulness in particular circumstances. Even the combination of both proposals is considered a possibility (the IST BRAIN/MIND solution is an example in that line). But this combination means an important network complexity that should be avoided if possible. In fact the hierarchical solution is quite complex by itself (anchor points, tunnels for all the traffic of the mobile nodes). This paper explores if Mobile IPv6 alone or in combination with a Fast Handovers solution can provide a performance good enough for real-time applications (the applications that require more quality).

III. BACKGROUND

This section presents the mobility implementations used and the way in which we measured the handover latency in each of the approaches. A detailed description of the basic mobility solutions can be found in [5] and [6].

A. Mobile IPv6 (MIPv6)

Basic Mobile IPv6 support is provided by MIPL [15]. MIPL is an Open Source implementation of the Mobile IPv6 protocol for the Linux Operating System. The version we have used in this study is mipv6-0.9.1-v2.4.16, which is compliant to the Mobile-IPv6 Internet-Draft version 15.

The latency due to a handover using basic MIPv6 is proportional to the round-trip time necessary for a Binding Update message (BU) to reach either the MN’s Home Agent (HA) or a Correspondent Node (CN). Therefore, the interruption time starts when the MN leaves its old link (it does not listen anymore to its old or Previous Access Router-PAR-) and finishes when it receives the first packet- from its HA or a CN- via its New Access Router (NAR). Furthermore, the latency is depending on the detection of the disconnection from the old link. Basic Mobile IPv6 follows the ’break before make’ philosophy; i.e. after loosing the current connection the mobile stack must detect a new point of attachment. The standard way to discover a NAR is via the reception of a router advertisement (RA) and then reconfigure the end user device to be able to communicate on the new link. This simple movement detection scheme increases the handover latency, since the detection of the NAR takes place during the ‘disconnection time’.

B. Fast Handovers for Mobile IPv6 (FMIPv6)

Fast Handovers for Mobile IPv6 (FMIPv6) support [6] is an extension of MIPv6 that minimize handover latency by allowing the Mobile Node (or the Network) to decide and prepare the movement before actually performing it (‘make-before-break’ philosophy).

FMIPv6 consists of three phases: handover initiation, tunnel establishment and packet forwarding. The first phase – handover initiation – is started by some kind of L2 trigger (e.g. as a result of measurement of the signal strength and/or some information carried in beacon frames). The Mobile Node sends a Router Solicitation for Proxy (RtSolPr) message to the PAR indicating the MN’s desire of changing its point of attachment (i.e. to perform a fast handover to a NAR). This message contains the link-layer address of the NAR. In response to the RtSolPr message, the MN receives a Proxy Router Advertisement (PrRtAdv) message from the PAR, indicating – if the new point of attachment is known and belongs to a different Access Router – the new network prefix that the MN should use in forming the new Care-of Address (nCoA). With this information, the MN forms the nCoA and sends it to the PAR in a Fast Binding Update (F-BU) message, which is actually the last message sent before executing the handover. The MN receives a Fast Binding Acknowledgement (F-Back) either via the PAR or the NAR indicating that the binding has been successfully done.

The second phase – tunnel establishment – creates a bidirectional tunnel between the PAR and the NAR. To create this tunnel, two messages are sent. First, the PAR sends info about the current MN’s CoA (old Care-of Address – oCoA -) and the CoA it wants to use in the new link (nCoA) to the NAR, in a Handover Initiate (HI) message. The NAR responses with a Handover Acknowledgment (HAck) message, and the PAR then sets up a temporary tunnel.

Finally, the third phase – packet forwarding – takes place, allowing the MN to loose only packets due to the L2 handover. This is accomplished by the PAR and NAR forwarding packets sent by/addressed to the MN’s oCoA, while the MIPv6 registration phase is completed. When the MN arrives to the new link, it sends a Fast Neighbour

Advertisement (F-NA) message to indicate the NAR to start sending the packets the NAR has addressed to the MN.

C. Fast Handovers implementation (FHO) for Mobile IPv6

Fast Handovers implementation (FHO) provides an enhanced support to the basic MIPL scheme. FHO has been implemented within the framework of the Moby Dick project as a Linux kernel module. It provides main FMIPv6 functionality, with minor changes from the FMIPv6 draft03, but following its 'make before break' philosophy; i.e. preparation for the new connection is performed prior to the handover via the current link. The movement detection scheme is still based on the router advertisements, but it is enhanced to be 'network aware', i.e., router advertisements from surrounding Access Routers are stored and evaluated to initiate the fast handover.

There are some minor differences between the FMIPv6 specification and the FMIPv6 support provided by FHO, related to implementation issues. Basically, FHO does not establish a bi-directional tunnel (BT) between the PAR and the NAR. Instead, FHO starts a bicasting process: packets arriving at the PAR destined to the MN's oCoA are sent both to the old link and also to the new link, encapsulated in a packet destined to the nCoA.

FHO signalling is implemented as ICMPv6 messages, as depicted in the signalling flow chart shown in Fig 1. The Fast Handover process consists of three parts. It starts with the preparation phase, in which the communication between PAR and NAR takes place, e.g. to check available resources. The Fast Handover Execute (FHE) message initiates the second phase, where the fast handover is performed. And finally, in the third part, the MN connects to the new link and informs the HA and CNs about its new location by sending the respective Binding Updates.

Summarizing, when the MN notices a new link (by means of some triggering function, e.g. signal quality) and it wants to move to this link, it sends a Router Solicitation for Proxy (RtSolPr) message to its PAR, providing it the NAR address and its new CoA (nCoA). The Handover Initiation (HI) message is sent from the PAR to the NAR to indicate the process of MN's handover. If a NAR receives the HI message, it should test the proposed nCoA for uniqueness (QoS availability can be checked at this step), decide whether it is valid or not, and reply with a Handover Acknowledgement (HACK) message. Then the PAR sends a Proxy Router Advertisement (PrRtAdv) message to the MN.

One of the key points in the signalling flow is the communication between PAR and NAR, which is used for AAA and QoS signalling purposes within the framework of the Moby Dick architecture.

The second part of the signalling starts when the MN sends the Fast Handover Execute (FHE) message in order to inform the PAR that a handover will be executed, and to ensure the establishment of the bicasting with a temporary tunnel between the PAR and the nCoA. After creating the tunnel, the Fast Handover Acknowledgement (FHEACK) message is sent to both the old and the new CoA. The MN, as soon as it gets connectivity to the NAR, sends a Neighbor Advertisement (NA) message to the NAR. After receiving it, the NAR is aware of the MN and its link layer and link local addresses, as part of the standard IPv6 attachment procedure. Afterwards, the FHO module initiates the required Binding Updates to the HA and the CNs. During the tunnel lifetime the PAR sends all packets destined to the MN’s oCoA, to the oCoA and, using the tunnel, also to the nCoA. The packets are therefore duplicated during the lifetime of the bicasting tunnel. This increases the load of the network only for a short time and only in the wired part and not on the scarce wireless medium, in order to reduce the interruption: when the Mobile Node attaches to the new link, its data already arrives there.

The interruption time starts in this case when the MN leaves its old link and finishes when it is able to receive its first encapsulated packet via its NAR.

The Fast handovers solution presented in this paper was thought, in Moby Dick architecture, for intra-domain handover. The same solution can be applied to a inter-domain environment, but the solution would be more difficult to deploy, due to security and administrative concerns, because inter-AR signalling should be authenticated (as it is in Moby Dick prototype), and it is difficult to establish a security association between ARs belonging to different administrative domains.

IV. STUDIED SCENARIOS

This section introduces the main scenario used in the experimental study presented in this article.

In Fig 2 we can observe the scenario used for all the tests done in this study. The scenario consists of seven RedHat 7.2 Linux[1] 2.4.16, MIPL 0.9.1 machines. Four of them act as routers (R1, R2 and the Access Routers AR1 and AR2), one as HA, one as MN, and one as CN. This was part of the Moby Dick testbed at the UC3M.

One important point is that we needed the ability to modify the delay between the CN and the MN in order to evaluate how the handover latency is affected by network characteristics (the possible different scenarios of particular locations of MNs, CNs, and HAs). For this purpose we used the NISTNET emulator [14]. NISTNET allows a single Linux PC, set up as a router, to emulate a wide variety of network conditions (e.g. latency, jitter, packet loss, …). We were interested in the study of the handover latency modifying the network delay between the MN and a CN because this modification allows us to evaluate the effect of delays in the signalling traffic (e.g. due to the MN and the CN being far from each other). Other network characteristics, that do not have a special effect in mobility and that are also present in non-mobile computers, were not modified. NISTNET supports only IPv4 connections, so we had to set up an IPv6-in-IPv4 tunnel – between R1 and CN - for using it in our IPv6 scenario. In this set-up, every packet addressed to/sent by the CN traverses the IPv6-in-IPv4 tunnel, which allows modifying the network behaviour by changing the parameters of the NISTNET emulator running in R2. In the rest of the path – from R1 to the MN, packets are IPv6 native, so the tunnel inclusion does not affect to the overall test performance except for the small added delay due to IPv6-in-IPv4 tunnelling (the situation is not different from having an ATM transport or and Ethernet transport in the path, it is transparent to the IPv6 behaviour). Actually the IPv4 tunnel reflects deeply the current status of IPv6 networks in the Internet, with a lot of IPv4 clouds- in fact, is pretty possible that they will never disappear totally - connecting IPv6 native networks.

The MN's handover is performed between two WLAN cells. These cells have enough overlapping surface so there is no possibility for a MN of not being able to communicate with either of them. Notice that cell overlap is a requirement for seamless handover; the size of the overlap limits the possible speed of movement. Each WLAN cell belongs to a different IPv6 subnet, i.e., in our architecture an Access Point serves always as an Access Router.

It was a common believe of the Moby Dick consortium, that future network topologies will deploy the WLAN infrastructure mode. This mode, unlike the ad-hoc mode, allows efficient frequency use, with neighbour cells using different frequencies and not interfering with each other, while a mobile node, thanks to beacon frames, can discover other cells and execute handovers to them. However, the WLAN ad hoc mode has been chosen, because handover layer 2 latencies in 802.11b infrastructure mode were measured and they were too high (over 150 ms) to use this 802.11b mode for real time communications. This was independent of the particular equipment (in any case, it was too high, the figure given is the better one) and caused by the time needed for scanning alternative channels looking for candidate Access Points/Access Routers. Because of this, we adopted the described solution that allowed us to study the merits of the different layer 3 mobility approaches. Hopefully, the layer 2 handover latency problem of the IEEE 802.11b technology will be solved in that or other WLAN standard. It should be noted that, in a parallel work, an 802.11b modified driver is being developed and tested. In this modified driver, the Access Point informs its stations about surrounding neighbours APs (channel, MAC, IPv6 address), so the stations, when current signal weakness is detected, scan only in the channels in which there are APs. In this way, layer 2 latencies in 802.11b infrastructure mode are drastically reduced. Some preliminary tests with the FHO implementation have also been done with this scenario and the experiments performed show that the results presented in this article are also valid for 802.11b infrastructure mode with this optimization.

Therefore, the WLAN ad-hoc mode was deployed, including modifications to emulate infrastructure mode as described in the following.

The ARs and the MN are in the same 802.11b 'ad-hoc' network. The Layer-2 (L2) differentiation is provided by a modified WLAN driver, designed and implemented within the framework of the Moby Dick project. This driver is a modification of the Host AP driver developed by Jouni Malinen [16] and is only needed in the MNs.

[1] FreeBSD/KAME provides a better IPv6 support currently. On the other hand, application support is worse.

Because of the ‘ad-hoc’ network configuration, a MN receives frames from all the ARs that are within its coverage area. To provide a separation among sub-networks, the driver filters, by its L2 address, every frame that is not sent by the current AR (the AR that is serving the MN). So only those packets received from the current AR are delivered to upper layers. This includes that only the Router Advertisements sent by the current AR are delivered to the IPv6 stack of the MN. The driver also processes Router Advertisements sent by other Access Routers that, with the signal levels of the corresponding 802.11 frames, are delivered to a management software that executes the handover decision algorithm in the MN. When the management software module decides that a handover should be performed, it informs the FHO module of the handover parameters (i.e. MAC and IPv6 addresses of the old and new AR), required for the handover preparation and execution phases. The L2 handover is executed by the management software by informing the WLAN driver of the new current AR. This causes the driver to change the L2 frame filtering behaviour (frames actually delivered to the MN’s IPv6 kernel stack), from delivering all the frames sent by the old AR to delivering all the frames sent by the new AR.

Regarding Duplicate Address Detection (DAD), in the deployed scenario neither MIPL nor FHO performs DAD, assuming unique identifiers [17].

V. QUANTITATIVE TESTS

Test description

This section provides a description of the tests performed, as well as the tools used, the measurements done and the justification of the validity of the results.

We wanted to show how the different mobility management approaches behave under different network profiles. In this section we present quantitative results, giving handover latency (handover interruption time) figures for each of the mobility solutions in different circumstances. There are at least two different ways to measure handover

Therefore, Linux is a more suitable O.S. for deploying a Next Generation Network nowadays.

latency: (i) packet loss can be measured for a determined data stream and (ii) absolute measurements of time-stamps, added to the FHO module.

The idea of measuring the latency in terms of packet loss consisted in sending small packets using a high rate (using a small interval between two consecutive packets), so then we could approximate the handover latency by the multiplication of the number of packets lost, times the time interval between packets. Therefore we needed a tool that could send and receive numbered and time-stamped packets following a predefined small trigger. This measurement method provided us a clear idea of the handover latency perceived by both a MN and a CN that are communicating with each other. Ping6 is a tool that allows us to send sequence-marked small packets following a predefined trigger of values as small as 10 to 20 ms (without using it in 'flood' mode). This provides us a precision good enough in order to compare the performance of mobility management approaches to support real-time communications[2]. TCP based tools have not been used because this paper is focused in the characterization of handover delays introduced by different mobility solutions, and not in the interaction with TCP to measure, for example, the effect in the throughput seen by the applications. There are comparative, simulation-based, studies of different approaches and their behaviour with TCP [18], but the effect of mobility and TCP is outside the scope of this paper, although the figures of absolute handover latencies presented here could be used in future studies to see the effect in TCP applications.

Basically, the test consisted in using the ping6 tool to send packets from a CN to a MN, while the MN was moving from one foreign network to another. This experiment was performed repeatedly[3] and varying the network conditions.

We were interested in analyzing how MIPv6 and FMIPv6 solutions perform under different network delays. We have measured the handover latencies of both solutions with network delays (in each direction) from 0ms[4] to 500ms. By varying the added network delays, the impact of the effect of the delay in the signalling traffic can be evaluated. The range employed covers most practical situations: from having both peers

[2] There are analytical studies ([19]) that say that the maximum permitted interruption in a voice communication is about 50ms. Therefore, precisions below 20ms (which can be obtained with ping6) are good enough to show if a certain mobility management solution is able to support voice communications or not.

[3] The experiment was repeated 20 times under the same network conditions. This number has been shown to be enough in order to show that MIPv6 and FMIPv6 solutions were statistically different (calculating the p value of the t-test).

[4] We refer here to NISTNET added network delays.

(CN and MN) very close to each other (i.e. in the same network/domain) to CN located far from the MN (i.e. the path between them traverses congested links and/or some satellite or transoceanic link).

Fast movement detection is very important in order to lower handover latencies. The primary method for movement detection uses facilities of IPv6 Neighbor Discovery. A faster method could be based on introducing L2 stack interaction in movement detection, but this method would be L2-specific. Listening periodic unsolicited multicast Router Advertisement messages is the method most employed by Mobile IPv6 implementations.

Fast Handovers solution requires L2 interaction. Our implementation (FHO) uses Router Advertisements to detect access routers candidate to become the new access router. Also some information (sub-network prefix, IPv6 address) needed in the fast handover preparation and execution, is obtained from these Router Advertisements. But this is previous to the handover, and the change of point of attachment (layer 2 handover) is actively executed by the MN (by configuring the filtering function in the WLAN driver) when the MN already has the information for the configuration in the new sub-network (layer 3 handover). In this sense, we were interested also in analyzing how the Router Advertisement sending interval influences the handover latency, because the different impact in the two mobility management solutions.

To deploy the second measurement approach, time-stamps were added to the FHO source code. The measured disconnection time is the difference between the moment the MN leaves the PAR and the moment it re-connects to the NAR evaluated on the MN. This kind of measurement represents a very accurate granularity, because precision of the operations relies strictly on the CPU 64 bits register (TSC – timestamp counter register) and the measurement (i.e., logging of the time stamp) follows immediately the respective FHO primitives.

On the other hand, in this measurement, because it can be based only on the state in the MN, the end of the handover interruption time is defined by the instant of re-connection to the NAR. This is conceptually so in the deployed solution, because this is the instant that defines when the MN can continue its communications according to the Fast Handovers procedure. This procedure guarantees that in that instant the NAR is able to send the traffic to the MN in its new location/configuration. Nevertheless, experimentally, to be able to say that the interruption time in the communication between a MN and a CN has ended, we would need also to prove that there are not some unforeseen circumstances (e.g. a failure in the Fast Handovers procedure).

Therefore, the two kind of measurements given, one evaluating the time without receiving traffic and so less precise as it depends on the traffic, and the other evaluating the disconnection time in the MN by means of timestamps, complement each other to give a very clear idea of the performance of the mobility solutions. The first one is user-centric because it evaluates the interruption time caused by the handover in the communication between a MN and a CN, and it is pessimistic in the sense that real interruption time is lower than the measured value. The second one is implementation-oriented and it is more accurate, but real interruption time in the communication between a MN and a CN could be higher than the measured value.

In conclusion we studied MIPL and FHO performances, and how they are affected by the interval between Router Advertisements, and the network delay between the CN and the MN.

Results

Fig. 3 shows the handover latency of both MIPL and FHO implementations versus the network delay (in each direction) introduced by NISTNET emulator. Due to the finite precision introduced by the measurement method, two lines are shown for each mobility implementation, meaning the extremes of the mean handover latency. The handover latency is measured by counting the number of packets lost using very small ones sent very fast, but we cannot provide figures more accurate than the time interval used (i.e. if small packets are sent every 15 ms, and no packet is lost during a handover, it means that the actual handover delay introduced by the movement is some value between 0 and 15ms). Therefore, the lines shown in the figure are separated by the time interval used for the ping6 – about 15ms-.

In Fig. 4 we show the results obtained by modifying the Routing Advertisement interval. Two different intervals have been used: the minimum permitted in the Mobile IPv6 draft (MinRtrAdvInterval: 0.5 seconds, MaxRtrAdvInterval: 1.5 seconds) and one bigger value (MinRtrAdvInterval: 2.0 seconds, MaxRtrAdvInterval: 4.0 seconds).

Notice that these values are lower than the recommended ones in RFC 2461 [20] according to the modification proposed in the Mobile IPv6 draft[5].

For each of these values two experiments were done, one without adding network delay with the NISTNET emulator, and another adding a network delay of 500ms – representing a extreme situation with large network delay (i.e. MN and CN far from each other)- with the NISTNET emulator.

The time-stamp measurement, as described above, confirms that the latency for intra-technology 802.11b handover using the Fast Handovers enhancement is below 3 ms.

Comparison

We can observe in Fig 3 that MIPv6 (MIPL) handover delay is significantly dependent on the network delay existing between the MN and the CN which it is communicating with. This result is an expected one, because the interruption of a "conventional" handover is directly proportional to the round-trip time necessary for a binding update (BU) to reach the CN. Indeed, the results presented in this study confirm this strong dependence of the handover latency with network delay. On the other hand, FMIPv6 (FHO) handover delay is independent of the network delay (i.e. time required for the MIPv6 signalling to be completed and the binding associations to be refreshed), because of the 'make before break' philosophy and the bicasting process. It allows the MN to use its old CoA, while the "conventional" MIPv6 signalling takes place.

Moreover handover latency in the Fast Handover case is really low (even with the most pessimistic measurement method we can say that it is between 0 ms and 15 ms, and the optimistic one gives us values below 3 ms). Fast Handovers provides a solution for handovers that it is suitable for real-time multimedia applications (see note 2). It is also interesting to note that the handover latency of MIPL with no added network delay is larger than the one with FHO. The expected performance of an implementation of a HMIPv6 approach would be at best similar to the MIPL performance with 0 network delay, so these results also provide comparative results between the performance of FMIPv6 and other micro-mobility solutions like HMIPv6. Fig. 4 shows the big

[5] Moby Dick testbed is MobileIPv6 draft15 compliant, so the values used are the ones specified in this draft release. Later draft revisions have smaller values. Using these (smaller) values would lower the handover latencies obtained, but the goal of these tests was showing the influence of the Router Advertisement interval in the

influence of the Router Advertisement interval in Mobile IPv6 handover latency. This could be very important in links where the L2 technology has small capacity (like 802.11b), because the sending of a big amount of unsolicited Router Advertisements could waste significant amounts of shared bandwidth. On the other hand, Fast Handover solution is again independent of the interval between Router Advertisements. This is because, the new AR is discovered while the MN is using the previous AR, the decision to perform a handover to this AR is also done while being attached to the previous AR, and the MN executes the handover after the preparation phase. Once the handover is actually performed (i.e. L2 handover), packets reach the MN by means of the bicasting process (packets addressed to the MN's oCoA are delivered to the new MN's location, as they are encapsulated in a tunnel addressed to the MN's nCoA). Nevertheless, notice that, whatever is used for discovering the candidate new AR (beacon frames, frames with Router Advertisements), they use bandwidth, but if we use bigger intervals between them, in the Fast Handover approach we will finish with less time to execute the fast handover procedure to prepare the handover before loosing communication with the previous AR. This was not considered in our experiments in which the two WLAN cells had enough overlapping surface.

VI. QUALITATIVE TESTS

Test description

In the previous section we have described some quantitative tests and the results obtained. It seems evident that platforms based on MIPv6-only support (e.g. support provided by MIPL) are not suitable for real-time applications. On the other hand, FMIPv6 support (FHO) seems to be good enough to support this kind of services without users being aware of performance degradation due to mobility.

The quantitative results obtained in the previous section are quite good, but we wanted also to show how users perceive mobility, i.e., how real-time application performance,

handover latencies not to present absolute values. Therefore the results obtained are representative enough for our purposes.

as perceived by users, is affected by terminal mobility, using different approaches to manage it.

In this section we present the results obtained from user questionnaires filled by real users (students at University Carlos III of Madrid without any relation or knowledge about our work), in which they were asked about the quality of the reproduction of the same video in two different machines. One of them had MIPv6 support (MIPL) and the other FMIPv6 support (FHO). Both machines (the MNs) were executing handovers (forced by software in a way invisible to users, and without physical movement) from one network to another repeatedly, while playing UDP video + audio streaming sent by a CN (the testbed used is the one shown in Fig. 2, but using two different MNs). No buffering was employed, so packet loss due to handovers could be perceived by users.

VideoLAN [21] is the video streaming application used in the tests (see Fig. 5).

In these tests, NISTNET emulator was used to add a 500ms network delay in order to simulate the existence of a real network between the CN and the MNs.

In the user questionnaires, users had to give a score (from 1 to 5, 5 being the best) to the perceived quality in the video reproduction. Values below 3 meant unacceptable quality (the user would not pay for this service). The video employed was a trailer of "Ice Age" movie (duration: 2'20'', bitrate approx: 468.58 kbps, resolution: 320x176). Handovers were performed continuously every 40 seconds.

Results

The results obtained from the questionnaires of 25 users are shown in Fig 6. 95 % confidence limits are also shown in the figure.

Comparison

Results from user questionnaires confirm the quantitative results. FMIPv6 approach (FHO) is able to support real-time multimedia applications with users not being aware of the mobility of the terminal. MIPv6 (MIPL) mean value is below 3 (as stated in the questionnaires, a value of 3 means unacceptable quality- the user would not pay for this service-). Therefore, users interested in real-time applications would not pay a network operator offering mobility support based on MIPv6 (users consider the quality perceived with MIPL not good enough). Fig 6 shows a big difference in how the users score the performance of an exigent real-time application under mobility and using two different approaches of mobility management. It is clear that users prefer FHO support of mobility over MIPL support, with a big difference gap between the solutions. The FMIPv6 approach (FHO) obtains a score near 5, while, on the other hand, MIPv6 (MIPL) approach obtains a score near 2.5, which means that users are not satisfied with the performance provided by MIPv6 solution when a real-time multimedia application is involved.

VII. SUMMARY AND CONCLUSIONS

This study has presented results from some experiments involving two different mobility management approaches: Mobile IPv6 and Fast Handovers for Mobile IPv6. With the latter approach, packet loss is reduced to almost L2 handover loss (in the presented implementation this is only the disconnection time for the re-configuration of the interface).

Quantitative results have shown that FMIPv6 can be used to offer the performance in handovers needed by real-time multimedia applications. Other micro-mobility solutions have not been taken into account based on some simulation based studies, as already mentioned in section II, that showed that that we could expect better results using FMIPv6 than using HMIPv6. But a question remained about if it would be necessary to combine both solutions to get even better performance. Notice that this is not a desired solution because the added complexity in the network. The results obtained in this paper show that FMIPv6 support is enough to provide the performance required by real-time multimedia applications, as shown by both qualitative and quantitative results.

On the other hand, measured MIPv6 handover latencies are quite big. Moreover MIPL handover latency is dependent on the network delay whereas FHO handover latency is independent of it.

Router Advertisement interval also affects MIPL performance whereas FHO is not affected by it at all. Sending Router Advertisements (RA) with a high frequency would mean a high load to links with low bandwidth shared technologies (e.g. 802.11b). Moreover, high RA rates do not ensure small handover latencies. Small RA intervals cause the MN to be aware of its movement sooner, so RA interval effect is more important in “local” scenarios, in which the MN is close to the CN. In “remote” scenarios (MN and CN are some hops far), the predominant effect is the network round-trip time due to the need to complete MIPv6 signalling. These effects are clearly shown in Fig. 4 and 5. On the other hand, FMIPv6 is neither affected by network round-trip time nor by the RA interval, due to its “make-before-break” philosophy and the bicasting process, which allows the MN to receive packets delivered to its old location (i.e. addressed to its oCoA), by the PAR tunnelling them to the MN’s nCoA (i.e. delivering them to the new MN’s location).

Latest versions of Mobile IPv6 drafts have included the Return Routability Procedure due to security reasons. This procedure has to be completed before using the Route Optimization (sending the Binding Update). Therefore this makes even more important to be able to use the oCoA while completing this procedure and the Mobile IPv6 signalling in order to reduce handover latencies. This can be done with FMIPv6 support. Our experiments did not consider explicitly the Return Routability Procedure, but its use only means some added delay in the registering process and, so, its effect can be analysed by studying the results of the NISTNET added delay between the MN and the CN.

User perception of both mobility solutions agrees with quantitative results. Also according to user answers, FMIPv6 is able to support terminal mobility better, by reducing the amount of disruptions, for real-time applications, whereas MIPv6 support has been poorly scored.

Both the quantitative and qualitative results lead us to believe that it is not needed to improve the performance of the Fast Handover approach with a combination with a Hierarchical approach, at least from the point of view of handover latency. Fast Handovers approach is good enough even with exigent real-time multimedia applications, as it has been experimentally demonstrated by measuring the disruption time due to movements (i.e handover latencies) and qualitatively evaluating - by using real testers - the performance of applications.

VIII. ACKNOLEDGEMENTS

This work has been supported by the EU project IST-2000-25394 Moby Dick.
We would like to thank the contribution of all partners in the Moby Dick project for their contributions and discussions.

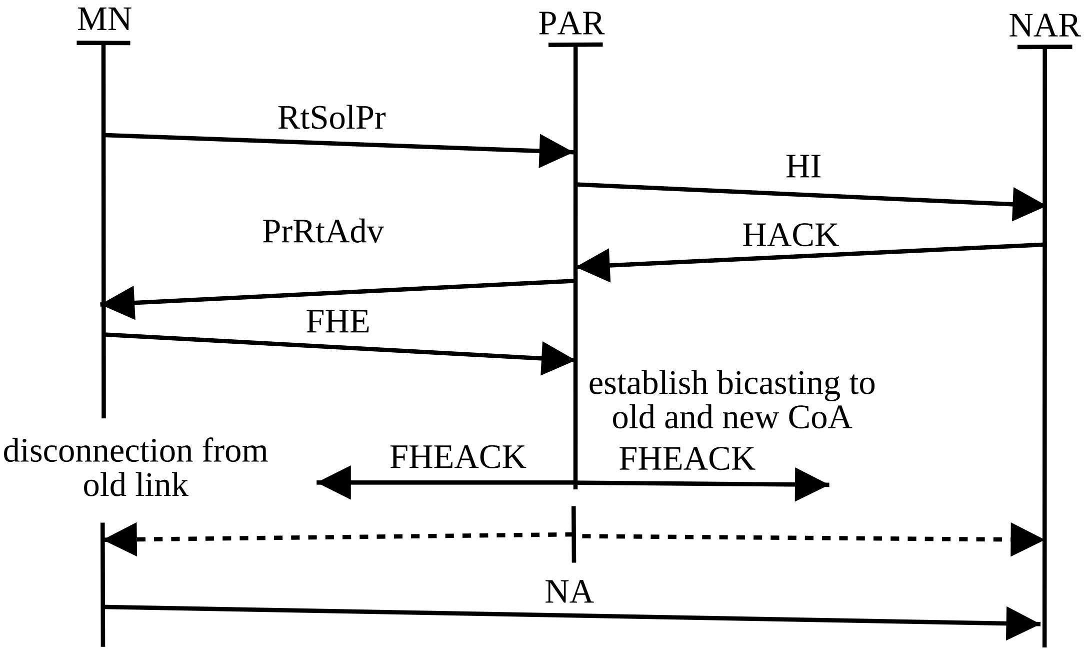

MN
PAR
NAR
RtSolPr
HI
PrRtAdv
HACK
FHE
establish bicasting to old and new CoA
disconnection from old link
FHEACK
FHEACK
NA

Fig. 1. FHO signalling

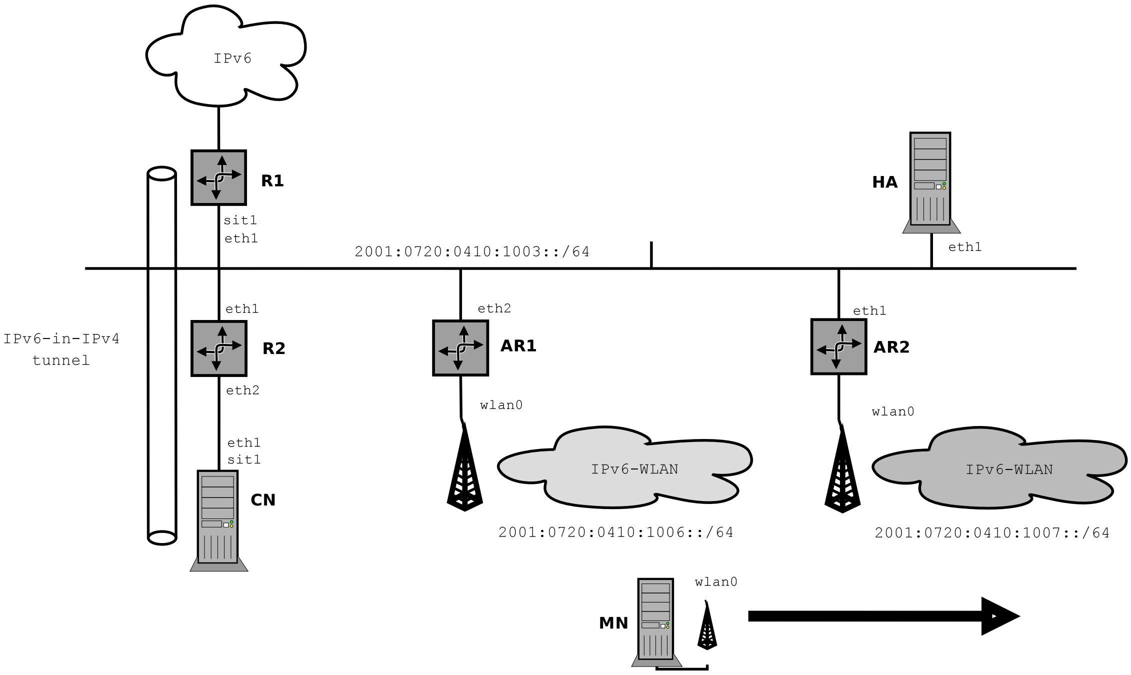


Fig. 2. Studied scenario

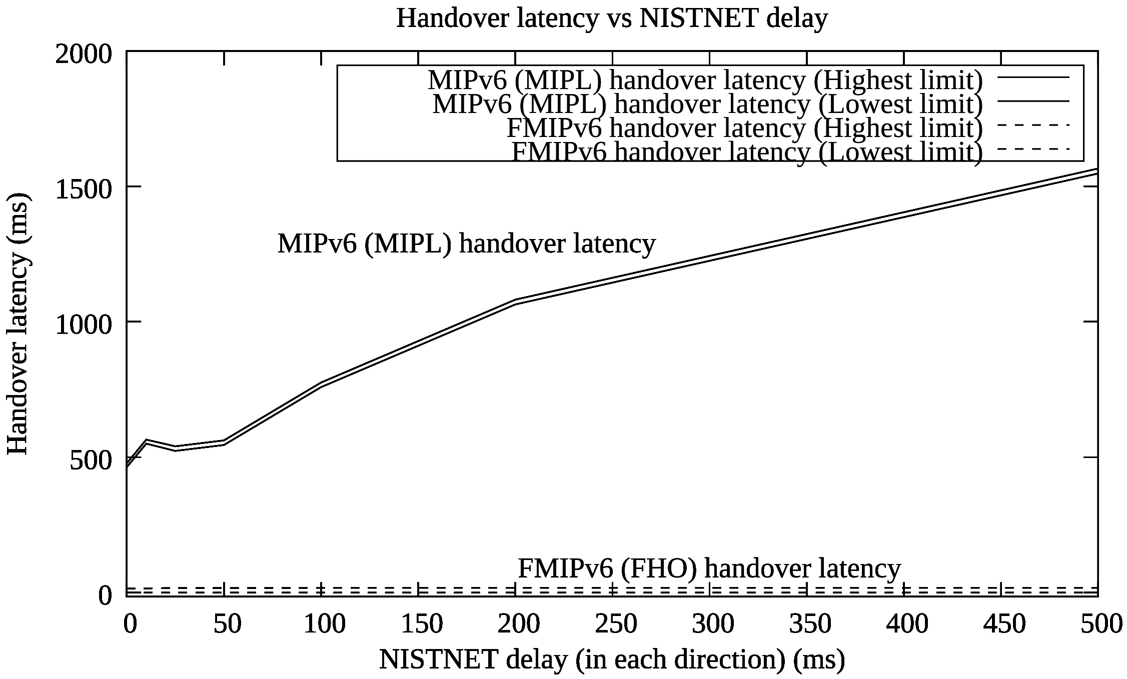


Fig. 3. MIPL and FHO handover latencies versus NISTNET network delay

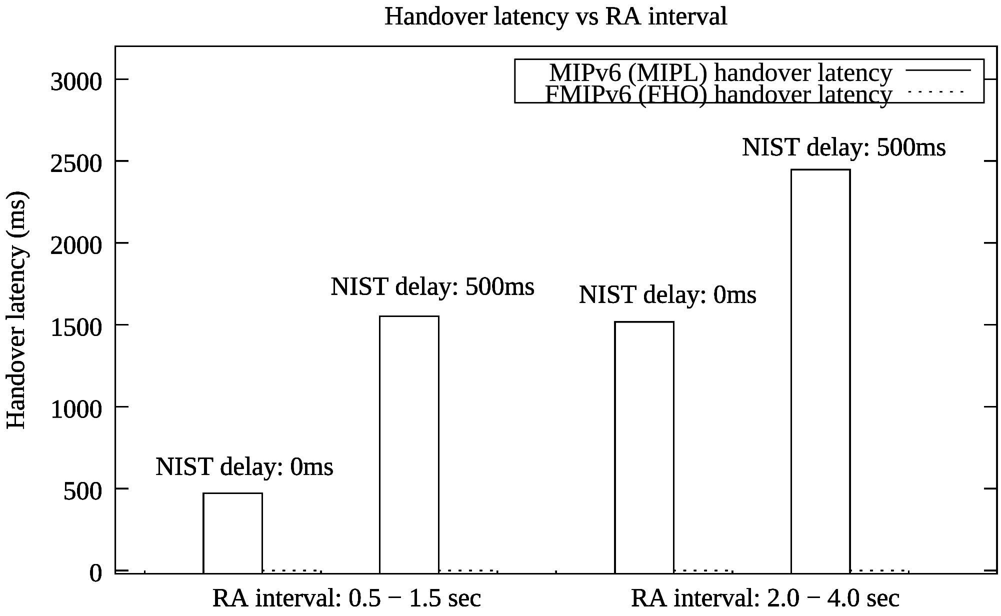
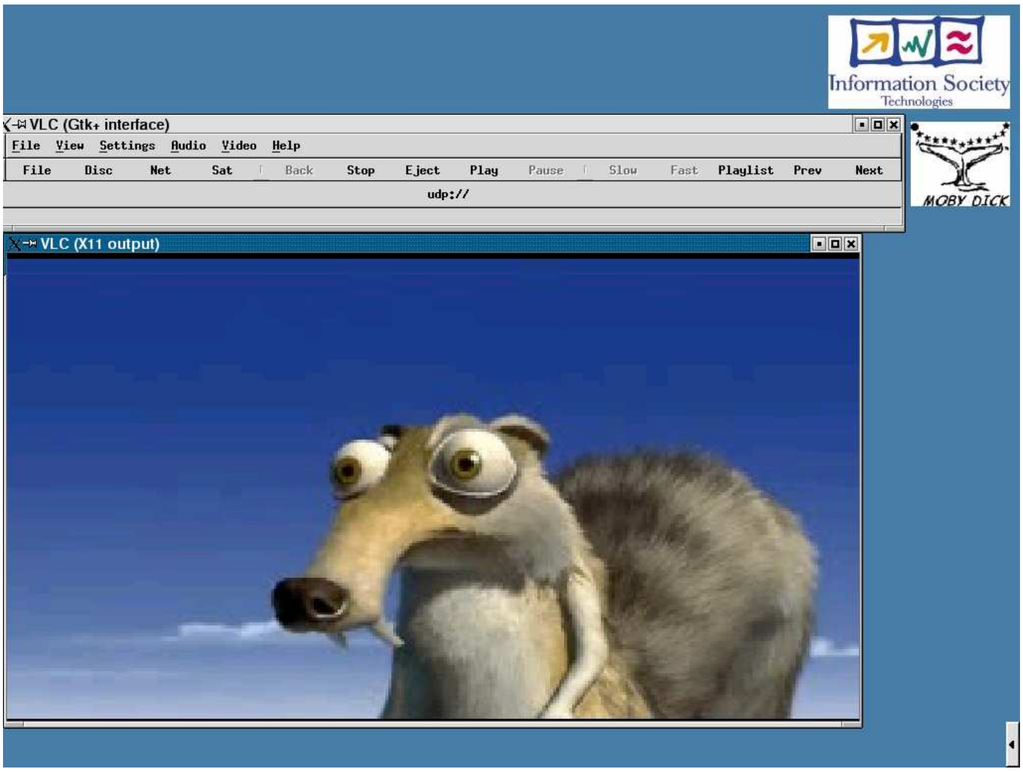


Fig. 4. Handover latency vs Router Advertisement interval



Information Society
Technologies
VLC (Gtk+ interface)
File View Settings Audio Video Help
File Disc Net Sat Back Stop Eject Play Pause Slow Fast Playlist Prev Next
udp://
MOBY DICK
VLC (X11 output)

Fig. 5. VIDEOLAN playing UDPv6 streaming video

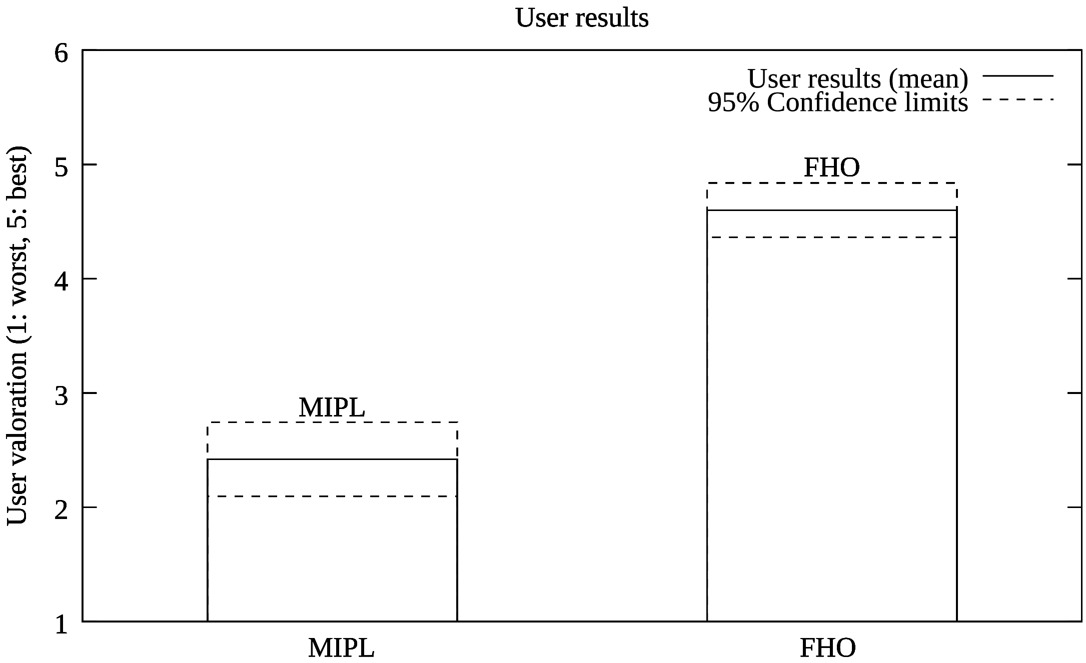


Fig. 6. User results

REFERENCES

[1] IST Project Moby Dick: *Mobility and Differentiated Services in a Future IP Network*, IST-2000-25394, 2000, see: http://www.ist-mobydick.org (2003-09-08)

[2] R. Droms, *Dynamic Host Configuration Protocol, RFC 2131,* March 1997

[3] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998

[4] C. Perkins, *Mobility Support in IPv4, RFC 3220,* January 2002

[5] D. Johnson, C. Perkins and J. Arkko, *Mobility Support in IPv6, draft-ietf-mobileip-ipv6-24.txt*, Internet Draft, work in progress, 2003-07-02

[6] Rajeev Koodli, *Fast Handovers for Mobile IPv6, draft-ietf-mobileip-fast-mipv6-06.txt,* Internet Draft, work in progress, 2003-03-01

[7] X. P. Costa, R. Schmitz, H. Hartenstein and M. Liebsch, A MIPv6, FMIPv6 and HMIPv6 handover latency study: analytical approach, I*ST Mobile & Wireless Telecommunications Summit 2002, Thessaloniki, Greece, June 17-19, 2002; pp. 100-105.*

[8] X. P. Costa and H. Hartenstein, A Simulation Study on the Performance of Mobile IPv6 in a WLAN-Based Cellular Network¸ *18th International Teletraffic Congress ITC, Berlin, Germany, August 31 - September 5, 2003*

[9] C. Janneteau (ed.), Deliverable D09 IST-1999-10699 WINE GLASS: Specification of Wireless Internet Architecture and Validation Plan. September 2001. Available in http://wineglass.tilab.com

[10] S. Recker (ed.), Deliverable D15 (Phase 2) Evaluation of Wireless Internet Testbed and Research Results, March 2002. Available in http://wineglass.tilab.com

[11] D. Wisely and E. Mitjana, Paving the Road to Systems Beyond 3G – The IST BRAIN and MIND Projects, *Journal of Communication and Networks, Special issue*, December 2002.

[12] C. Keszei, N. Georganopoulos, Z. Turanyi and A. Valko, Evaluation of the BRAIN Candidate Mobility Management Protocol, *IST Global Summit, Barcelona*, September 2001.

[13] E. García et al, Deliverable D6.4 IST-2000-28584 MIND, MIND Trial Final Report, November 2002. Available in http://www.ist-mind.org/deliverables.htm.

[14] NIST Net, network emulation package, see http://snad.ncsl.nist.gov/itg/nistnet/ (2003-09-08)

[15] MIPL: Mobile IPv6 for Linux, see http://www.mipl.mediapoli.com/ (2003-09-08)

[16] J. Malinen, Host AP driver for Intersil Prism2/2.5/3, see http://hostap.epitest.pi (2004-03-21)

[17] M. Bagnulo, I. Soto, A. Garcia-Martinez and A. Azcorra, Avoiding DAD for Improving Real-Time Communication in MIPv6 Environments, *PROMS 2002: 7th International Conference on Protocols for Multimedia Systems*, pp. 73-79. Lecture Notes in Computer Science, 2515. Ed. Springer-Verlag. Oct. 2002.

[18] R. Hsieh and A. Seneviratne, A Comparison of Mechanisms for Improving Mobile IP Handoff Latency for End-to-End TCP, *Mobicom 2003*, San Diego, California, USA, September 14-19 2003

[19] M. Karam and F. Tobagi, Analysis of the Delay and Jitter of Voice Traffic Over the Internet, *IEEE Infocom 2001*; April 2001

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

[21] VIDEOLAN: OpenSource Video streaming solution for every OS, see http://www.videolan.org/ (2003-09-08)