A Network-based Localized Mobility Solution for Distributed Mobility Management

Fabio Giust*†, Antonio De La Oliva†, Carlos J. Bernardos†, Rui Pedro Ferreira Da Costa‡
* Institute IMDEA Networks, Spain
E-mail: fabio.giust@imdea.org
† Universidad Carlos III de Madrid, Spain
E-mail: {aoliva, cjbc}@it.uc3m.es
‡ Alcatel-Lucent Bell Labs, France
E-mail: rui_pedro.ferreira_da_costa@alcatel-lucent.com

***Abstract*—Internet traffic has increased steeply in recent years, due in great part to social platforms and peer-to-peer networks. In addition, users' wireless access represents an ever-growing portion of such demand, thus posing a paradigm shift in the flow of Internet information, for which most deployed architectures are not prepared for.**

This evolution in user traffic demand is tackled by a different approach for IP mobility, called Distributed Mobility Management, that is focusing on moving the mobility anchors from the core network and pushing them closer to the users, at the edge of the network.

The work presented here copes with the distributed approach, describing a novel solution for network-based localized mobility support in a flat architecture without central mobility anchors. It leverages PMIPv6 standard, but it is intended to overcome most of the issues in current centralized architectures, by splitting the control plane from the data plane and distributing them throughout the access networks.

I. Introduction

The number of mobile subscribers and the amount of traffic produced by them is experiencing a huge growth in these years, and the trend is not likely to stop. Users are now familiar with hand-held devices capable of accessing data services through wireless technologies, and the widespread penetration of Internet-based applications designed for such terminals is considerably raising the demand of mobile Internet connectivity. Accordingly, 3G USB modems are less expensive now, operators are more prone to offer flat rates for data connections, and WiFi hotspots are made available in the public transportation systems of many cities in the developed countries, thus contributing for an almost 100% coverage.

In parallel, recent mobile architectures as WiMAX and EPS are intended to be IP-based both for data and voice communications, triggering a real need to optimize IP protocols for mobile networks.

IP mobility management plays a key-role in providing the "always-on" and ubiquitous service envisioned by future technologies. Unfortunately, the IP mobility protocols standardized so far have not met the expectations regarding deployment success, being proprietary customized solutions in place instead. Nevertheless, the mobility management schemes standardized by IETF for IPv6 networks are extensions or modifications of the well known Mobile IPv6 protocol, (MIPv6) [1], such as Proxy Mobile IPv6, (PMIPv6) [2].These protocols usually handle operations at a cardinal point, the mobility anchor, following a centralized approach. Unfortunately, as this node is usually far away from the edge and deep into the core network, the centralized approach results in burdening the anchor with data forwarding and control mechanisms for a great amount of users.

In order to address such issue, the solution proposed in this work leverages on the *Distributed Mobility Management* paradigm, currently under discussion in the IETF [3]. It basically develops the concept of a flatter system, in which the mobility anchors are placed closer to the user, distributing the control and data infrastructures among the entities located at the edge of the access network. There are fundamentally two main approaches being researched now: one aimed at making Mobile IPv6 work in a distributed way, and another one doing the same exercise for Proxy Mobile IPv6. The idea behind our design is to re-use what was defined in PMIPv6, i.e. data structures, messages format, functionalities of the entities involved, presenting what are the changes required to make PMIPv6 work in a distributed way, either completely or partially, and trying to give the most complete description of those working schemes.

The rest of the paper is organized as follows. In Section II we summarize how MIPv6 and PMIPv6 work, and highlight the drawbacks of the centralized solutions; Section III is devoted to introduce the distributed paradigm and our proposal of a flat architecture based on PMIPv6, and, finally, we conclude the paper in Section IV.

II. Background and Motivation

A. Centralized Mobility Management Overview

Most of current mobility management solutions derive from **Mobile IPv6**, (**MIPv6**) [1], the first mobility protocol standardized by the IETF for IPv6 networks. MIPv6 enables global reachability and session continuity by introducing the Home

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7-ICT-2009-5) under grant agreement n. 258053 (MEDIEVAL project). This work has been also supported the Ministry of Science and Innovation of Spain under the QUARTET project (TIN2009-13992-C02-01).

Agent (HA), an entity located at the Home Network of the Mobile Node (MN) which anchors the permanent IP address used by the MN, called Home Address (HoA). The HA is in charge of defending the HoA's reachability when the MN is not at home (i.e., where the HoA is not topologically valid), and redirecting received traffic to the MN's current location. When away from its home network, the MN acquires a temporal IP address from the visited network - called Care-of Address (CoA) - and informs the HA about its current location by sending a Binding Update (BU) message. An IP bi-directional tunnel between the MN and the HA is then used to redirect traffic from and to the MN.

The efforts towards network-based mobility management resulted in the standardized **Proxy Mobile IPv6**, (**PMIPv6**) [2], developed as an enhancement of MIPv6. In PMIPv6, the home agent is replaced by the Local Mobility Anchor (LMA): it is in charge of routing packets in uplink and downlink containing the IPv6 prefixes assigned uniquely to MNs on a per user basis, the Home Network Prefix (HNP), and it stores the MNs' mobility sessions information. PMIPv6 evolved from MIPv6 by relocating relevant functionalities for mobility management from the MN to a network node, called the Mobile Access Gateway (MAG), which is the first IP hop and default gateway seen by the terminal. In PMIPv6 indeed, mobility is transparent for MNs: the network learns through standard terminal operation, such as Router and Neighbor Discovery [4], about MN's movements and coordinates routing state information using Proxy Binding Update (PBU) and Proxy Binding Acknowledgment (PBA) messages. The former is sent by MAGs to the LMA to indicate the MN's location, and the latter is sent as a response to ensure that the procedure succeeded. The LMA stores a Binding Cache Entry (BCE) containing the MN's identifier, its prefix and the serving MAG's address, called Proxy Care-of address (Proxy-CoA). Users' traffic is encapsulated between the LMA and the Proxy-CoAs. The set of deployed MAGs and the corresponding LMA forms the Localized Mobility Domain, in which mobility is completely transparent to the IP stack of the MNs.

B. Limitations of centralized mobility management solutions

Even if centralized approaches have been fully investigated in the last decade, several limitations inherent to the centralized nature exist, as reported in [3]:

- **Sub-optimal routing.** Data traffic always traverses the central anchor, regardless the current geographical position of the communication end-points and the possible presence of a shorter path. This scenario is worsened when the recipients are close to each other but the anchor point is far from both.
- **Scalability problems.** Links around the anchor and the node itself have to be provisioned to cope with all the subscribers' traffic and control messages. This is a big issue in terms of scalability and network design, as the number of mobile users keeps on growing.
- **Reliability.** Centralized anchoring points (i.e., HAs and LMAs) represent a potential single point of failure that can crucially damage the access conditions for a large amount of users.
- **Lack of fine granularity on the mobility management service.** Currently the mobility service is provided on a per user basis, that is, user's communications are treated as a whole. A finer granularity should be allowed, thus session continuity is granted only for those IP flows that really require it, and eventually it is possible to save signaling if an MN does not need mobility at all.

III. Description of the solution

A. Distributed Mobility Management

The limitations of centralized architectures encourage to explore a novel approach for mobility management that should be *dynamic* and *distributed*. It should be dynamic in order to offer mobility support at a per-flow granularity. It should be distributed to avoid control and data packets traversing a centralized mobility anchor.

For this purpose, the IETF is working on an analysis document [5] that proposes the guidelines for the development of distributed solutions, that can be divided into two main categories: *i)* partially distributed, that basically consists on removing the data path constraint towards the anchor, but maintaining a centralized control plane, and *ii)* fully distributed, that consists on eliminating any centralized role in the architecture.

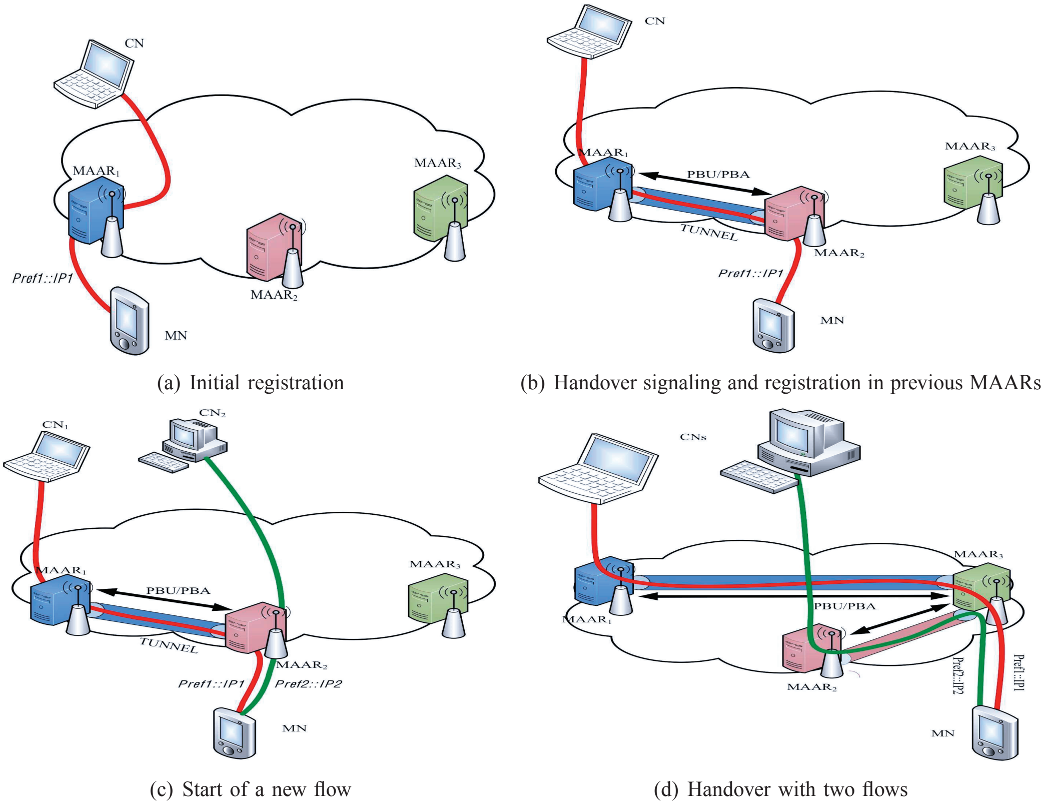
In next subsection we address how to develop a DMM scheme based on PMIPv6, describing first a fully distributed approach and showing that considerable changes need to be introduced, also with several extra requirements on the mobile node. Therefore, a partial distributed solution is presented too, stressing the lighter intervention on the original protocol.

Nevertheless, both variations share the same data forwarding plane, briefly summarized here: a MAG (that we call Mobility Anchor and Access Router, MAAR) assigns IPv6 prefixes on a per-MN basis from a prefix pool that belongs to that MAAR only and that are anchored (i.e., topologically correct). In this way, an MN receives a prefix, and therefore configures an IP address, per each visited MAAR. A MAAR anchors the flows started using the prefixes it advertises, so it acts as plain IPv6 router (i.e., it does not encapsulate packets) when the flow is addressed to an MN in its network. Otherwise it establishes a tunnel with the MAAR currently serving the MN.

We introduce next the terminology related to the roles that the MAAR may play for each IP flow traversing it:

- **Anchor MAAR (A-MAAR)** is referred to the MAAR that advertised the prefix used in the communication/flow.
- **Serving MAAR (S-MAAR)** is referred to the MAAR where the MN involved in the communication/flow is attached to.

It should be noted that this separation is only conceptual and applied on a flow basis: the same MAAR can play simultaneously different roles for the different flows. With respect to the PMIPv6 semantic, an A-MAAR acts as an LMA, and the S-MAAR as a MAG.



(a) Initial registration

(b) Handover signaling and registration in previous MAARs

(c) Start of a new flow

(d) Handover with two flows

Fig. 1. Fully distributed PMIPv6

B. Fully distributed approach

In the fully distributed approach, the LMA and the MAG are collapsed into the MAAR. In other words, MAARs are not only responsible of the forwarding of flows established with their prefixes in the manner described before, but they also maintain and handle the mobility sessions for those flows. The next use case example clarifies the working scheme of the proposed design.

An MN enters in the domain and attaches to MAAR$_1$, which advertises Pref$_1$. This prefix is used by the MN to configure an IP address and to start a communication with a remote Correspondent Node (CN), see Fig. 1(a).

When a handover occurs, MAAR$_2$ learns that the MN has attached and advertises a new prefix, Pref$_2$. Then, it sends a PBU messages to MAAR$_1$, that is anchoring MN's flows, and, upon the corresponding PBA reception, a tunnel is established between them, hence the flows can be redirected through it, as shown in Fig. 1(b). It is worth noticing that, in this juncture, MAAR$_1$ is the A-MAAR for the flow, while the S-MAAR is represented by MAAR$_2$; in addition, MAAR$_2$ becomes the A-MAAR and S-MAAR for the flow started by the MN using the new address (see Fig. 1(c)). Moreover, upon a new movement of the MN to a new access network, MAAR$_3$ assigns a third prefix and updates the MN's location to its former S-MAARs, i.e. MAAR$_1$ and MAAR$_2$, by means of PBU messages. Again, when the PBAs are received, tunnels are set up and the old communications are recovered, as depicted in Fig. 1(d).

The main issue in this mechanism is how a MAAR can differentiate between the first attachment to the network and subsequent handovers, that is, the MAAR, upon the MN's attachment, should be informed of the past MN's location to be able to contact the formerly visited MAARs. More than one solution is suggested here, each with advantages and disadvantages that should be evaluated looking for the best trade-off:

- **Broadcast PBU.** When a MAAR detects the attachment, it sends a broadcast request to all the MAARs and waits for a reply, that might be a void PBA in case the MN joined the network for the first time. It should be noted that sending the messages only to the neighbor MAARs is not enough to reconstruct the MN's past history, thus the procedure might result excessively long and introduces unnecessary signaling.
- **Terminal indication.** The MN explicitly sends the prefixes acquired previously when joins the new access network (e.g., as options in the Router Solicitation message). Although this is a fast and light procedure, it requires some capabilities on the host that are not always possible to rely on (or not even desirable), and might pose some security issues if a malicious MN misbehaves.
- **Automatic learning.** In [6] it is proposed that mobility capable access routers learn the previous MN's location by inspecting the source address of packets sent by the

MNs. That is, these routers store in an internal database the prefix pools belonging to the others, and, when an MN moves, upon receiving the first uplink packet, they check which is the access router the prefix belongs to and establish a tunnel with it. This requires little signaling but it may lead to an excessive delay if the MN does not send anything and the router has to explicitly request a packet.

- **IEEE 802.21 support.** Handover may follow a Make-Before-Break philosophy and integrate layer-2 and layer-3 mobility procedures within the same framework to assist and drive the handover. IEEE 802.21, Media Independent Handover Services, is a suitable protocol for this purpose.

Among the options listed above, we claim that IEEE 802.21 should be preferred as it builds a control plane infrastructure by which a handover is prepared, executed and completed in a controlled and assisted way, according to a Make-Before-Break philosophy. Next paragraphs are devoted to detail this approach.

The 802.21 services are provided by an entity called MIH Function (MIHF). It runs in the terminal and in the network nodes, it interacts with the network interfaces, with the layer 3 (or higher) mobility protocol and with the other MIHF peers, either local or remote. A Point of Service (PoS) is a MIHF in the network that can directly exchange messages with a peer MIHF installed in the terminal. A PoS instance serves one terminal, so multiple instances run in the same network entity. In our design, PoSs are implemented in the MAARs. PoSs maintain information about the access network entities as Base Stations, Access Points and similar nodes connected to the MAAR, by interacting with the MIHF running in them. These MIHF are called Point of Attachments (PoAs), and provide the access link to MNs, but they do not establish a control communication directly with them. According to IEEE 802.21 working scheme, a PoS learns when a handover for its MN is imminent (the details of how this is done are out of scope in this paper); therefore it queries resources availability in the surrounding PoAs by means of a message called `MIH_N2N_HO_Query_Resources.request`, sent to the corresponding PoSs connected to the desired PoAs. This message contains the current S-MAAR's (to be the A-MAAR) address, that is used later by the new S-MAAR to send the PBU message. Upon the list of candidate PoAs is filled with the requested information, the target for the handover is selected (either by the MN itself or by the PoS) and the corresponding PoS is notified about the decision. The MN can now move to the new PoA: the IP mobility procedure is triggered and the conclusion phase, with the old resources release, takes place.

This procedure works without changing the standard 802.21 primitives, when only one MAAR needs to be contacted with the PBU message (as in Fig. 1(b)). Conversely, when more than one MAAR is anchoring flows, as shown in Fig. 1(d), the current PoS needs to send to the candidate PoSs a list of all the past visited MAARs, and this requires a change in the format of the MIH_N2N_HO_Query_Resources.request primitive.

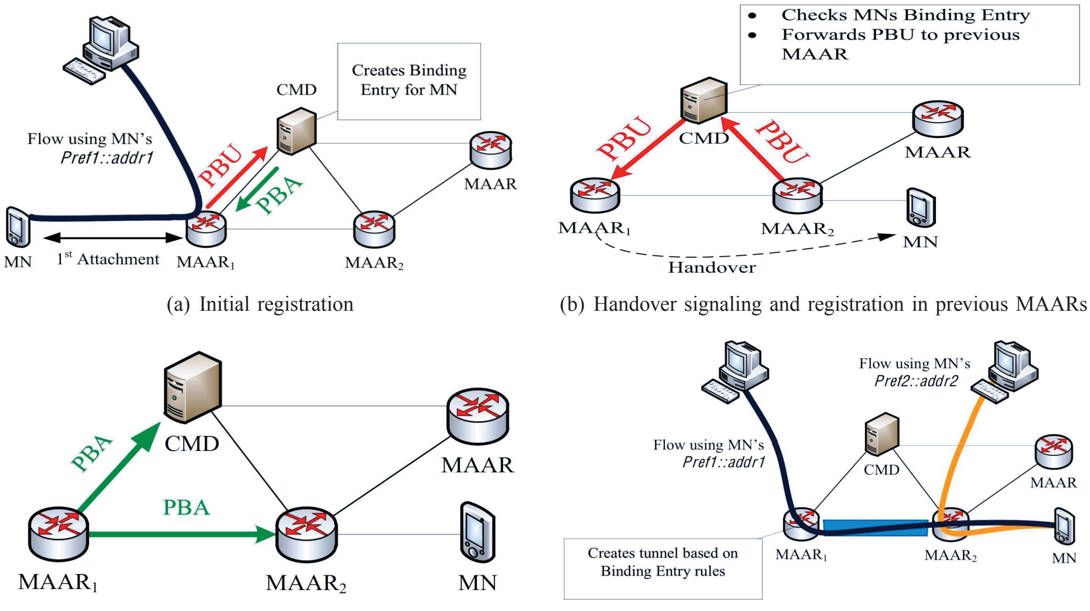
As shown in the above paragraphs, a fully distributed approach requires in all cases many interventions on the terminals, that might be not desirable, and, in the IEEE 802.21 case, also the implementation of a whole control infrastructure. For the purpose of reducing complexity and terminal requirements, we present in next subsection a partially distributed solution, which come at the cost of keeping the control plane centralized and only distributing the data plane (which is the most critical one).

C. Partially distributed approach

In this approach, most of the control plane burden is transferred to a centralized network node. According to this principle, in the proposed architecture an additional node is added, called Central Mobility Database (CMD), storing the mobility bindings for the MNs. It maintains the control functions of an LMA but it is relieved of the data forwarding plane since it is not traversed by users' data traffic.

The proposed approach works as follows. Upon MN attachment to MAAR_1, an IPv6 global prefix belonging to the MAAR's prefix pool is reserved for it (Pref_1). The prefix is sent in a PBU with the MN's Identifier (MN-ID) to the CMD that, since the session is new, stores a Binding Cache Entry containing as main fields the MN-ID, the MN's prefix and the MAAR_1's address (Proxy-CoA). The CMD replies to MAAR_1 with a PBA meaning that the MN's registration is fresh and no past status is available. MAAR_1 sends a Router Advertisement (RA) to the MN including the prefix reserved before, that can be used by the MN to configure an IPv6 address (e.g., with stateless auto-configuration). The address is routable at the MAAR, in the sense that it is on the path of packets addressed to the MN (see Fig. 2(a)).

When the MN moves from its current access, it associates to MAAR_2, which delegates another IPv6 prefix (Pref_2) and sends it to the CMD for registration. The CMD has already an entry for the MN, binding the MN-ID to its former locations, thus, it forwards the PBU to all the MAARs indicated as Proxy CoAs, in this case only MAAR_1 (as depicted in Fig. 2(b)). Upon PBU reception, MAAR_1 sends a PBA to the CMD to ensure that the new location has successfully changed, and another PBA to MAAR_2 containing the prefix anchored at MAAR_1 (see Fig. 2(c)), so that a tunnel can be established between them to recover the flow. Now packets destined to Pref_1 are first received by MAAR_1, encapsulated into the tunnel and forwarded to MAAR_2, which finally delivers them to their destination. In uplink, when the MN transmits packets with Pref_1, they are sent to MAAR_2, as it is MN's new default gateway, then tunneled to MAAR_1 which routes them towards the Correspondent Node. Conversely, packets carrying Pref_2 are routed by MAAR_2 without any special packet handling (as shown in Fig. 2(d)). For next MN's movements the process is repeated for the number of previous MAARs involved, that rises accordingly to the number of prefixes that the MN wishes to maintain. It should be noted indeed, that this design separates the mobility management at the prefix granularity, and it can be tuned in order to erase old mobility sessions when not required, while the MN is reachable through the latest



(a) Initial registration

(b) Handover signaling and registration in previous MAARs

(c) Acknowledgement from previous MAARs

(d) Data tunneling and use of new prefixes

Fig. 2. Partially distributed PMIPv6: message flow

acquired prefix. Advanced mechanisms might be implemented over this platform to achieve the aforementioned dynamic mobility activation, allowing, on the one hand, the MN to not cause mobility sessions to be created if the IP address continuity is not required for none of the prefixes. On the other hand, it results in a considerable messages proliferation, in the case all the prefixes acquired need to be kept reachable. Hence, a little variation is shown next, in order to limit the number of control messages sent. Let $n \geq 1$ be the number of prefixes an MN wishes to maintain reachable after an handover, the amount of PBU+PBA messages grows as $3n + 1$ (the first PBU + n more forwarded by the CMD + $2n$ PBAs). Let now the PBAs follow the same path as the PBUs. That is, once the CMD receives the first PBU from the new MAAR and forwards copies to the correspondent old MAARs, they only reply with a PBA back to the CMD, which in turn sends an aggregated PBA to the new MAAR, containing all the previous Proxy-CoAs. In this case the number of messages transferred grows as $2n + 2$, thus saving $n - 1$ packets. The drawback of this solution is the longer delay in the handover phase due to the mobility signaling. Indeed, as in the original solution a A-MAAR transmits the PBA message directly to the S-MAAR, the tunnels are established independently upon PBA reception at the S-MAAR, while in the modified version, the delay is bound to the PBA that takes the longest time to reach the CMD. The drawback can be mitigated introducing a timeout at the CMD, by which, after its expiration, all the PBAs so far collected are transmitted, and the remaining are sent later upon their arrival. Therefore, a trade-off should be evaluated between signaling overhead and handover delay.

IV. Conclusion

In this paper we have shown how a distributed mobility management solution can be achieved re-using existing concepts inherited by Proxy Mobile IPv6 protocols. The original architecture has been modified to adapt to a flatter architecture following two different approaches: partially or fully distributed.

Both solutions require decoupling the data and control planes, as the cardinal role of a centralized anchor is removed in the DMM philosophy. However, the description of such possible designs revealed that the the former requires a considerable intervention on the mobile node, that, to a further extent, can be equipped with an ancillary instance for handover control, as IEEE 802.21.

At last, we claim that the partially distributed is a lighter variant, because it maintains the advantages of a distributed data plane architecture, but does not introduce constraint on the terminal, nor dedicated control protocols, at the cost of keeping the control plane centralized.

References

[1] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775, June 2004.

[2] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6," RFC 5213, Aug. 2008.

[3] H. Chan, "Problem statement for distributed and dynamic mobility management," Internet-Draft (work in progress), Mar. 2011.

[4] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861, Sept. 2007.

[5] H. Yokota, P. Seite, E. Demaria, and Z. Cao, "Use case scenarios for Distributed Mobility Management," Internet-Draft (work in progress), Oct. 2010.

[6] P. Bertin, S. Bonjour, and J.-M. Bonnin, "A Distributed Dynamic Mobility Management Scheme Designed for Flat IP Architectures," in *New Technologies, Mobility and Security, 2008. NTMS '08.*, Nov. 2008, pp. 1–5.