

A Decoupling Approach for Distributed Mobility Management

Andréa Nascimento, Rute Sofia
SITI, Lusófona University
Building U, 1st Floor, Campo Grande 376
Lisboa, Portugal
{andrea.nascimento, rute.sofia}@ulusofona.pt

Tiago Condeixa, Susana Sargento
Instituto de Telecomunicações, University of Aveiro
Campus Universitário de Santiago, P-3810-193
Aveiro, Portugal
{tscondeixa, susana}@ua.pt

Abstract—This paper addresses the problematic of mobility management decomposition, as a first step towards the development of decentralized mobility management architectures. We propose and evaluate an IP mobility management solution based on the decoupling of data and control plane. We have carried out simulations, showing that the proposed approach can significantly reduce packet loss and reachability time for mobile nodes.

Keywords-Distributed mobility management; user-centric networks; flat networks.

I. INTRODUCTION

Currently, the most popular solutions for global mobility management rely on a centralized model, where a centralized and static element, the *Mobility Anchor Point (MAP)*, is responsible for keeping some form of association between previous and current whereabouts/identifiers of mobile nodes. MAPs control the signaling required to ensure smooth mobility between different points of attachment to the network.

Considering the way that Internet services are delivered and consumed nowadays, due to the widespread wireless technologies and increasing variety of user-friendly and multimedia-enabled terminals, there is a trend where the end-user has a particular role in generating, sharing, and controlling content as well as connectivity, based upon cooperation. These spontaneous environments are known as *User-centric Networks (UCNs)* [1] and rely on the notion that Internet users carry or own devices that may be part of the network. In these scenarios, centralized mobility approaches lead to poor network performance as the network nodes may easily appear/disappear from the network. Moreover, there are several scenarios that today require distributed mobility management, and where the available mobility management approaches are not suitable [2].

For these new networking paradigms, there is the need to: (i) better understand what exactly is meant by mobility management and which blocks compose a MAP; (ii) consider whether or not decoupling mobility management functionality into sub-blocks may assist in providing a distributed

mobility support; (iii) find out which is the best location for the placement of each of the mobility management sub-blocks throughout the network. Moreover, and focusing on UCNs, the Internet end-user is a network stakeholder; hence, it is possible that the user, through its terminal device, will be able to control mobility management functionality. Assuming, for instance, that the MAP may reside on an end-user device, then the period of time a mobility anchor point is available may vary frequently. This poses extra stress on seamless and centralized mobility mechanisms, which have to manage handovers more often.

To address this issue, and as follow-up of previous work [3][4], in this paper we present a proposal based on the splitting of control and data planes functionality having as benchmark *Mobile IPv6 (MIPv6)* [5]. Considering such approach, we have identified the issues that require further study [2], and proposed a decoupling approach. Our expectations with such decoupling are that it may allow a greater flexibility resulting in a performance improvement. Decoupling the functionality brings in the possibility of placing them, for instance, in different physical network devices, but it also raises several questions. A first question is whether or not there is a “best” location for each of the mobility management sub-blocks, and which is the trade-off between improvement and cost by doing such decoupling.

The paper is organized as follows. In section II we describe related work on mobility management. Section III provides a brief description on user-centric environments, that are the basis for our study. In section IV we describe the architecture of decoupling control and data planes on mobility management. In section V we provide a brief description of our experiments setup, and show our results. Section VI concludes this work.

II. RELATED WORK

The idea of distributing mobility management over the network has been the focus of intensive research in the last years, as observable from the *Internet Engineering Task Force (IETF) Distributed Mobility Management (DMM) Working Group* [6]. This effort is mainly related to the increasing amount of data traffic in wireless networks nowadays, as well as the trend on the evolution of networks

towards flatter architectures, making the current centralized available standards not suitable in the envisioned networks.

Having in mind the idea of flatter mobile network architectures, *Dynamic Mobility Anchoring* has been proposed [7] and further analyzed [8][9]. This approach addresses the concept of “flattening” by allowing a dynamic distribution of mobility management across the access nodes, in a way that is more suitable for both mobile nodes and anchor points. This is a tunneling based solution, which provides better performance as it takes into consideration both the location of mobile nodes and MAPs. Still, the mobility management functionality is kept together on a router in the access.

Chan described the main limitations of centralized mobility management approaches, e.g. usage of non-optimal routes, low scalability, the non-suitability for evolved network architectures and single point of failure [10]. The author discusses the location where a distributed mobility management approach could be deployed in the network.

The idea of splitting mobility functionality across different locations in the network is also not novel. Chan proposed the splitting of a mobility system into three logical functions: home network prefixes allocation; location management; mobility routing [11]. His approach is based on the *Proxy Mobile IPv6* [12] extension for MIPv6, and it is also proposed the usage of two mobility anchor elements, called *Home Mobility Anchor* and *Visited Mobility Anchor*. The main objective is to provide a system with mobility anchors distributed over different networks.

Sofia et al. [13] proposed an approach whose main objective is to separate control and data functionality in order to provide a more flexible mobility management framework, and to assist in developing non-centralized (e.g. distributed) mobility architectures. However, the authors do not present a proposal on how the communication between those separated elements could be performed, nor an analysis on which scenarios would benefit from such splitting.

Our work builds upon the work presented by Sofia et al. and we provide our own interpretation of how such control and data plane decoupling could be done, having as target a potential evaluation of the cost of such decoupling and whether or not it would be useful in dynamic environments such as UCNs. Next section presents our assumptions and mobility requirements of user-centric networks.

III. USER-CENTRIC NETWORKS

UCNs (previously coined as *User-provided Networks* [14]) relate to a recent trend based on spontaneous wireless deployments, where individual users or communities share subscribed access in exchange of specific incentives. In addition to the sharing of subscribed access, the Internet user role is augmented in UCNs given that: (i) the user becomes a producer/provider of specific services; (ii) the end-user devices can operate as network elements.

As some of those devices are carried by humans, they exhibit human movement patterns. Such patterns are not easily defined as they are based on individual users' routines and on users shared interests towards targets (e.g. locations, other users). For instance, MAPs may easily appear and disappear as they may also be placed in end-user devices.

Mobility management in UCNs is required to ensure adequate connectivity models and adequate network operation to support end-user expectations towards his/her roaming services. Considering the dynamics of user-centric networks and their self-organizing nature, it is crucial to attempt to develop end-to-end mobility management solutions more flexible than the ones existing today, as user-centric wireless networks are starting to heavily populate Internet fringes.

Due to the high variability of user-centric environments, our aim is to address ways to split mobility management functionality, as well as ways to “push” such functionality closer to the end-user, having in mind an optimization of mobility management in the context of those kind of networks. Placing the mobility management closer to the end-user can decrease the level of dependency on the access and core networks to increase the efficiency of mobility support.

IV. SPLITTING CONTROL AND DATA PLANES

According to Sofia et al. [13], one potential way to split data and control planes is to consider two different MAP roles, in contrast to most of the current approaches, which describe only one element. In their proposal, Sofia et al. consider two elements: the *Home Agent - Control (HA-C)* is the element responsible for the signaling plane and maintenance of binding cache entries; the *Home Agent - Data (HA-D)* is the MAP element responsible for actions such as encapsulation of data traffic to the new location of a *Mobile Node (MN)*. It is the HA-C that controls the signaling e.g., choice of HA-Ds to consider, or trigger the activation of tunnels between specific HA-D elements. This work considers one HA-C as example, but covers also the notion of several HA-Cs and several HA-Ds involved in the signaling.

The signaling procedure between the MN and the system is performed directly with the HA-C, which means that the MN does not communicate directly with any of the HA-Ds. When a MN performs a handover from its home network to a visited network and gets a new IP address, it sends a message to the HA-C with the association between its home address and its new address, and waits for an acknowledgment message. The HA-C will store this information, and it will trigger the establishment of a tunnel from the old location of the MN to the new one. To create this tunnel, the HA-C sends a message to the HA-D that is placed in the home network of the MN. Upon receiving this message, the HA-D will start to intercept all the messages destined to the MN's home address, and forward them to the MN's new location.

This tunnel will remain active until the MN inform all its *Correspondent Nodes (CN)* about its new location.

For our studies, we have considered this work and have extended it from a pragmatic perspective, within the context of UCNs. We address implementation details concerning format and message exchange between the control and data plane elements and analyzed the implications of this decoupling. For our own interpretation, we have considered two elements: the control plane element named MAP_C ; and the MAP_D , corresponding to the mobility anchor point responsible for the tunneling functionality. The MAP_C is composed of:

- **Identification database control:** corresponds to the mechanism that is applied to control the database identification.
- **Binding mechanism:** it is the signaling related to the device's register/update to the mobility system.
- **Handover negotiation:** the process taken when the device has its real-address changed, involving negotiation and signaling.

The MAP_D is responsible for the following aspects:

- **Device identification:** corresponds to the network identification for the MN.
- **Encapsulation/Tunneling:** it is the process of intercepting the packets destined to the *known-address*, encapsulating them with the *real-address*, and forwarding them.

In this work, we used MIPv6 as base protocol for the functional splitting. When decoupling control and data plane, the MN shall only exchange signaling information with the MAP_C . In our proposal, this signaling corresponds to *Binding Update (BU)* messages which contain information about the current IP addresses (*Care of Address, CoA*) and original IP address (*Home Address, HoA*) of the MN. The MAP_D role is to activate and deactivate tunnels as well as to encapsulate traffic, whenever a MN performs a handover. No context status is kept in MAP_D s. We have implemented a partially distributed approach, where several MAP_D s are deployed and controlled by one single MAP_C . The MAP_D , since it is responsible for data forwarding, it should be placed as close as possible to the MNs.

A. Communication Between MAP_C and MAP_D

The functionality placed in MAP_C and MAP_D is independent of each other as explained, but these elements need to communicate in order to perform their tasks. By decoupling functionality that up until now resided in a single element, we need to ensure that there is a way to perform robust and stable signaling.

The communication between MAP_C and MAP_D happens only when a MN performs a handover, and there is the need to create a tunnel from the MN's old location to the new one. To support that communication we consider two

new types of ICMP messages, *Tunnel Establishment Request (TER)* message and *Tunnel Establishment Acknowledgement (TEA)* message.

Assuming the handover situation where a MN gets a new CoA, after the regular BU process sent by the MN to MAP_C , the MAP_C sends a TER message to the old MAP_D . This message triggers the creation of a tunnel to the new MN's CoA; therefore, all packets addressed to the old address will be intercepted by the old MAP_D , encapsulated and forwarded to the new CoA. The MAP_D then answers to MAP_C with a TEA message. This tunnel established by the MAP_D is temporary, and it will remain active only until the binding update procedure is finished; after that, the data flow is again routed directly between MN and CN.

Assuming that a MAP_D realizes there is no traffic being sent to a specific MN for some time then it reacts by deactivating the tunnel. After the deactivation of a tunnel, the MAP_D does not store anymore any information about that MN. This reduces the need to keep status, and reduces overhead associated with data tunneling.

B. MAP Discovery

When a MN needs to send a BU message to a MAP_C , in order to register or update its real location, the MN should know the address of the MAP_C it should contact. It should also know the address of the MAP_D that is placed in the network it is currently attached to, to send this information to MAP_C . For the MAP_C address, considering that we are approaching a partially distributed architecture with only one MAP_C in the system, it is reasonable to think about configuring the MAP_C address manually in the MNs. For a fully distributed architecture, since it would have more than one MAP_C available, it would be needed to design a discovery mechanism, since the MAP_C would be chosen dynamically.

For discovering the MAP_D address, in our implementation, the MN assumes that the wireless router it is attached to has the role of a MAP_D . In scenarios where the router may not have the MAP_D functionality, a dynamic mechanism like the *Home Agent Address Discovery* [5] defined for MIPv6 standard can be applied. This mechanism describes the usage of an extra information sent by the routers in the *Router Advertisement* message, that is a flag to inform MNs and other routers if that router is acting as a *Home Agent* or not for that network. For a MAP_C to be able to contact the old MAP_D of a specific MN that performed a handover, we have determined that every message sent from a MN to MAP_C should contain information about the MAP_D , so the MAP_C stores this information in the binding cache entry for that MN.

C. Mobile Node Reachability

By decoupling the control and data plane functionality, the MAP_C can be placed anywhere in the network. This means

that there is a high probability for the MAP not to reside on the same network segment as the MN. From a MIPv6 perspective, one of the roles of the HA is to alert CN of the new whereabouts of the MN (CoA). During the temporary tunneling procedure, the MAP_C needs to offer ways for MN and CN establish a direct communication. Considering that the MAP_C has only control plane functionality, it could react in two ways:

- By sending a message to notify the CN of the current CoA address of the MN it is trying to contact. In this case, the CN will then create an entry in its binding cache and send the packets directly to the MN's CoA.
- By sending a message to the MN notifying that a CN is trying to communicate. In this case, the MN will perform the binding update procedure to inform the CN of its current location.

The first option has security issues, considering that the MAP_C will reveal the real current location of the MN to any node that requests for it. However, it is the easiest and faster way of establishing a direct communication between the CN and the MN in this case. The second option can take a longer time, since it needs two procedures until the CN gets the real location of the MN (one notification message is sent from the MAP_C to the MN, and then the MN can send a BU message to the CN with its real address). Another option, that would not incur in extra signaling, would be to place a MAP_D in the same network as the MAP_C , and make this MAP_D the default old MAP_D for when the MN makes its first register to the MAP_C . In our implementation, we used the first option, by creating a *Notification* (NOTIF) message, that is sent from MAP_C to CN containing the information about the binding between HA and CoA of the MN.

V. PERFORMANCE EVALUATION

To evaluate the proposal described in the previous section, simulations were carried out using the *network simulator version 2.33 (ns-2)* [15] with an extended version of the module *MobiWAN* [16] for MIPv6. Extensions to the *MobiWAN* module were implemented to make the mobility management agent available to be configured also in wired nodes, and to make it more flexible in the sense that more than one functionality can be configured in the same node at the same time (for instance, a node could be a MAP_C and MAP_D at the same time).

As evaluation parameters we have considered: *end-to-end delay*, *packet loss*, and *reachability delay*. The end-to-end delay was measured as the time a data packet takes from source until it reaches the final destination. The packet loss was measured as the percentage of data packets lost per flow. The reachability time is defined as the time interval that starts in the instant when a mobile node receives a new IP address, until the instant it receives the first data packet after

the communication break. All parameters were measured per flow and through all the flows in the simulation.

For the topology we have considered an IEEE 802.11b scenario with IPv6, with 6 wireless routers interconnected in two different ways as shown in Figure 1. The MAP_C element is located in a server, and the MAP_D s are configured as part of the routers for both scenarios. Our benchmark is MIPv6, and for this case we have placed the Home Agent in the same location as the MAP_C . In Figure 1(a) we consider an example for the most common topology available today from an operator perspective, where the MAP_C serves different wireless hotspots of a same operator, residing on the access backbone. While for the second topology (cf. Figure 1(c)) we assume a partially mesh scenario which stands for a potential example of a UCN, where now the MAP_C element is directly connected to some wireless hotspots.

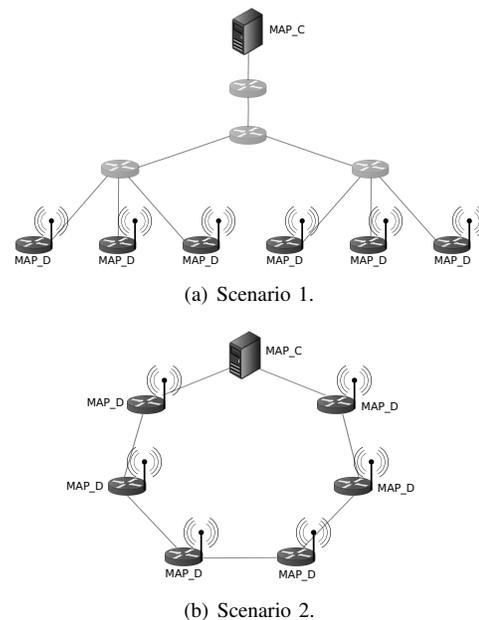


Figure 1. Evaluation topologies: (a) Scenario 1 - Operator based (b) Scenario 2 - UCN.

MNs have been configured with uniform speed randomly chosen from an interval between 0 and 5 m/s for a first round of experiments, and between 0 and 20m/s for a second round. The movement has been modelled with the *RandomWaypoint* model. For traffic we have simulated VoIP with codec G.711 bitrate, with *Constant Bit Rate (CBR)* flows with periods of speech and silence modeled with Exponential random variables with means 1.004s and 1.587s, respectively. The arrival of calls was modeled using a Exponential random variable with average 90, and the duration of the calls was modelled using a Pareto random variable with average 100 and shape -0.39. The flows were created and established between different pairs of mobile nodes randomly chosen,

for scenarios with 10 and 20 MNs in the network. We performed simulations with duration of 1 hour, and for each evaluated scenario and set of parameters, we have performed 50 repetitions. The results present means with a confidence interval of 95%.

A. End-to-end Delay Analysis

Figure 2 shows the achieved end-to-end delay for both scenarios. The first figure corresponds to Scenario 1 results, while the second one corresponds to Scenario 2 results.

Scenario 1 results show a higher end-to-end delay for the proposed approach when compared to MIPv6, being the highest when nodes move at lower speeds, and when the network is less congested. Still, end-to-end delay is always lower than 0.015 seconds. When looking at results from Scenario 2, the end-to-end delay is significantly lower as this relates to a scenario where the distance between MAP_C and MAP_Ds may vary. The increase in end-to-end delay relates to the need to establish tunnels between old and new MAP_Ds , and we believe that this delay increases when a MN performs a handover, proportionally to the distance between the old and new locations of the MN.

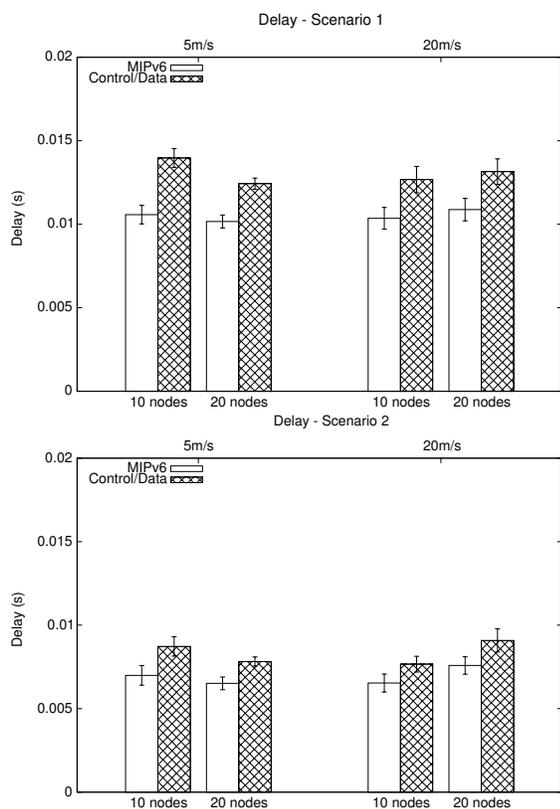


Figure 2. Delay.

B. Packet Loss

Figure 3 shows values obtained for packet loss, again where the first chart corresponds to results derived from

Scenario 1, and the second chart from results derived from Scenario 2.

Our proposed mechanism presents lower percentages of packet loss. This can be justified by the fact that, when a MN performs a handover, our proposal creates a tunnel from the old MAP_D to the new location of the MN, thus avoiding high packet loss during the CN binding process. As in MIPv6 the MN will be reachable again only after the CN binding procedure is finished, data packets sent meanwhile are lost.

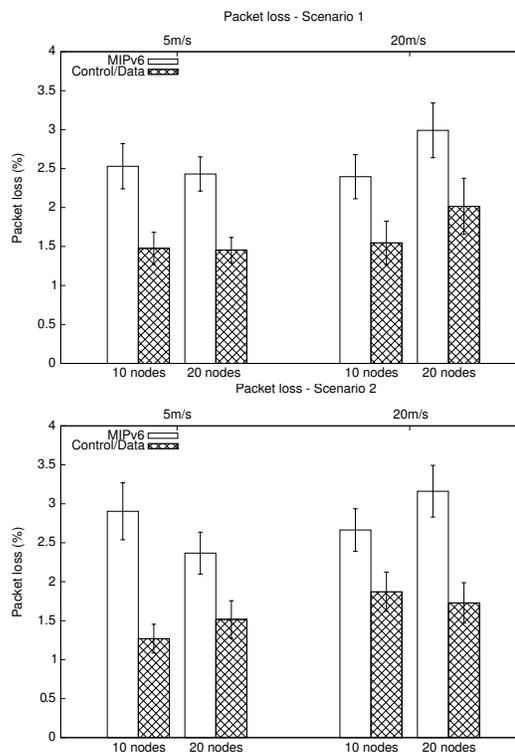


Figure 3. Packet loss.

C. Reachability Time

A third aspect we have addressed was reachability time for the MN. Figure 4 shows the reachability time, assuming the perspective of nodes that were receivers during handovers (cf. first chart for Scenario 1, and second chart for Scenario 2).

There is again a significant difference concerning making a node reachable after a handover. The MIPv6 reachability time varies between 7 and 8 seconds, while the reachability time for our approach was lower than 1 second. This significant difference can be justified again by the creation of tunnels between old and new MAP_D so that the time interval a MN is unreachable because it is still performing the CN binding update procedure gets considerably reduced. MIPv6 uses additional mechanisms to perform such update (namely, Home Test and Care-of Test), which implies in the

exchange of other messages before the MN can actually send a BU message to the CN, introducing an extra delay in the process.

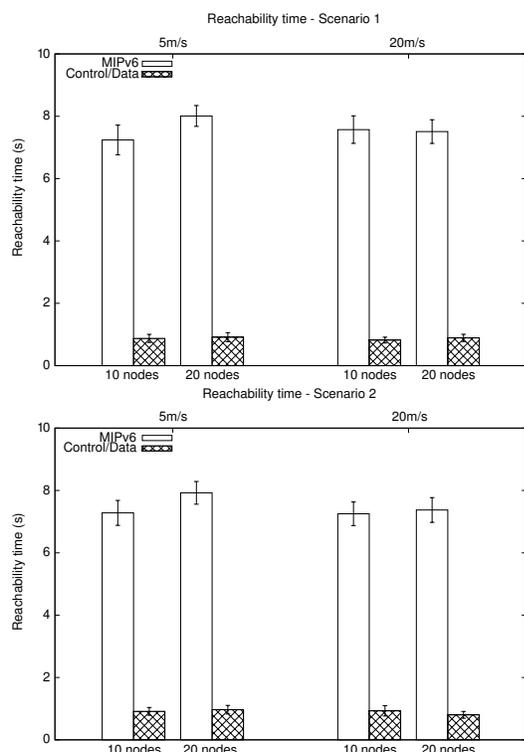


Figure 4. Reachability time.

VI. CONCLUSIONS AND FUTURE WORK

This paper addresses the problematic of distributed mobility management, based on previous work that we have developed. We propose a pragmatic decoupling of the MIPv6 control and data plane. We consider such approach as a first attempt to understand the trade-off associated to mobility management functionality decoupling. Based on ns-2 simulations, we have shown that this kind of decoupling can significantly decrease the packet loss and the time a MN remains unreachable during the binding update procedure.

As future work, we intend to continue the line of thought on decomposing mobility management functionality potentially reaching a level where a new architecture can be proposed. Also, we intend to go further on the decoupling of functionality by trying different approaches, such as the separation of handover and location management functions, and even more by considering basic functions of a mobility management system, as presented in [4].

REFERENCES

- [1] "EU IST FP7 ULOOP (User-centric Wireless Local Loop) Project," Gr. Nr. 257418, 2010-2013. [Online]. Available: <http://uloop.eu>
- [2] R. Sofia, A. Nascimento, S. Sargento, A. Matos, T. Condeixa, "UMM Project - D1: Use-cases," SITI, Lusófona University; IT, University of Aveiro, Tech. Rep., 2011.
- [3] A. Nascimento, R. Sofia, T. Condeixa, S. Sargento, R. Matos, "User-centric Mobility Management (UMM) Project," Terena Networking Conference 2011. (Poster), May 2011.
- [4] A. Nascimento, R. Sofia, T. Condeixa, S. Sargento, "A Characterization of Mobility Management in User-Centric Networks," in *Next Generation Wired/Wireless Networking, 11th International Conference, NEW2AN 2011, St. Petersburg, Russia, August 22-25, 2011. Proceedings*, 2011, pp. 314–325.
- [5] C. Perkins, D. Johnson, J. Arkko, *RFC 6275 - Mobility Support in IPv6*, IETF Std., 2011. [Online]. Available: <http://tools.ietf.org/html/rfc6275>
- [6] IETF, "Distributed Mobility Management (DMM) IETF Working Group." [Online]. Available: <http://tools.ietf.org/wg/dmm/>
- [7] P. Bertin, S. Bonjour, J.-M. Bonnin, "A Distributed Dynamic Mobility Management Scheme Designed for Flat IP Architectures," in *NTMS 2008, 2nd International Conference on New Technologies, Mobility and Security, November 5-7, 2008, Tangier, Morocco*, 2008, pp. 1–5.
- [8] P. Bertin, S. Bonjour, J.-M. Bonnin, "An Evaluation of Dynamic Mobility Anchoring," in *Proceedings of the 70th IEEE Vehicular Technology Conference, VTC Fall 2009, 20-23 September 2009, Anchorage, Alaska, USA*, 2009, pp. 1–5.
- [9] P. Bertin, S. Bonjour, J.-M. Bonnin, "Distributed or Centralized Mobility?" in *Proceedings of the Global Communications Conference, 2009. GLOBECOM 2009, Honolulu, Hawaii, USA*, 2009, pp. 1–6.
- [10] H. A. Chan, H. Yokota, J. Xie, P. Seite, D. Liu, "Distributed and Dynamic Mobility Management in Mobile Internet: Current Approaches and Issues." in *JCM*, 2011, pp. 4–15.
- [11] H. A. Chan, "Proxy Mobile IP with Distributed Mobility Anchors," in *Proceedings of GLOBECOM Workshop on Seamless Wireless Mobility*, 2010, pp. 16–20.
- [12] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, B. Patil, *RFC 5213 - Proxy Mobile IPv6*, IETF Std., 2008. [Online]. Available: <http://tools.ietf.org/html/rfc5213>
- [13] R. Sofia, A. Hof, S. Wevering, "Method for packet-based data transmission in a network having mobility functionality," European Patent 1,883,196, 2008. [Online]. Available: <http://www.freepatentsonline.com/EP1883196.html>
- [14] R. Sofia, P. Mendes, "User-Provided Networks: Consumer as Provider," *IEEE Communications Magazine*, vol. 46, no. 12, pp. 86–91, Dec. 2008.
- [15] "Network Simulator 2 - ns-2." [Online]. Available: <http://nsnam.isi.edu/nsnam/>
- [16] "MobiWAN for ns-2.33." [Online]. Available: <http://www.nicta.com.au/people/mehanio/nsmisc/ns-233-mobiwan-1.patch>