

Efficient Micro-Mobility using Intra-domain Multicast-based Mechanisms (M&M)

Ahmed Helmy^x, Muhammad Jaseemuddin^{*}

Abstract

One of the most important metrics in the design of IP mobility protocols is the *handover* performance. Handover occurs when a mobile node changes its network point-of-attachment from one access router to another. If not performed efficiently, handover delays, jitters and packet loss directly impact and disrupt applications and services. With the Internet growth and heterogeneity, it becomes crucial to design efficient handover protocols that are *scalable, robust and incrementally deployable*. The current Mobile IP (MIP) standard has been shown to exhibit poor handover performance. Most other work attempts to modify MIP to *slightly improve* its efficiency, while others propose complex techniques to *replace* MIP.

Rather than taking these approaches, we instead propose a *new architecture* for providing efficient and smooth handover, while being able to *co-exist* and inter-operate with other technologies. Specifically, we propose an *intra-domain multicast-based* mobility architecture, where a visiting mobile is assigned a multicast address to use while moving within a domain. Efficient handover is achieved using *standard multicast join/prune* mechanisms.

Two approaches are proposed and contrasted. The first introduces the concept *proxy-based mobility*, while the other uses *algorithmic mapping* to obtain the multicast address of visiting mobiles. We show that the algorithmic mapping approach has several advantages over the proxy approach, and provide mechanisms to support it.

Network simulation (using NS-2) is used to evaluate our scheme and compare it to other major approaches to micro-mobility; hierarchical mobile IP and seamless handover. Results show that our scheme, although simpler, outperforms the other schemes in over 94% of the scenarios, and provides, on average, handover performance that is 1.23 to 3 times better, depending on topology and mobility. We also study multicast state aggregation and show that a simple *lossy bit-wise* aggregation scheme obtains aggregation gains ranging from 1.25 to 2. Our work provides extremely efficient, yet simple, schemes to achieve *notable improvement* in handover performance and *scalability*.

I. INTRODUCTION

The growth of mobile communications necessitates efficient support for IP mobility. IP mobility addresses the problem of changing the network point-of-

attachment transparently during movement. When the mobile node moves away from its current network point-of-attachment, *handover* is invoked to choose another suitable point-of-attachment. In such an environment, handover latency and mobility dynamics pose a challenge for the provision of efficient handover.

Several studies[1][8] show that Mobile IP (MIP)[3], the proposed standard, has several drawbacks ranging from triangle routing and its effect on network overhead and end-to-end delays, to poor performance during handover due to communication overhead with the home agent. Several micro-mobility approaches attempt to modify some mechanisms in Mobile IP to improve its performance[4][5]. However, as we will show, such approaches suffer from added complexity and, in general, do not achieve the best handover performance.

We follow a different approach to IP mobility using *multicast-based mobility (M&M)*. In such architecture, each mobile node is assigned a multicast address to which it joins through the access routers it visits during its movement. Handover is performed through standard IP-multicast join/prune mechanisms. Such approach, however, is not suitable for inter-domain IP mobility, for several reasons. First, the architecture requires *ubiquitous multicast deployment*, which is only partially supported in today's Internet. M&M should be designed for *incremental deployment*, and to allow co-existence with other IP mobility protocols. Second, the *multicast state* kept in the routers grows as the number of mobile nodes becomes larger. Hence, it is crucial to alleviate the state explosion problem using *state aggregation* techniques. Third, allocating a globally-unique multicast address for every mobile node requires a *global multicast address allocation* scheme, and wastes multicast resources. Furthermore, mobile nodes incur *security* delay with every handover, which may overshadow architectural mechanisms that attempt to reduce handover delays.

To alleviate these problems, we propose new schemes for *intra-domain* multicast-based micro-mobility that allow for incremental deployment. In this architecture, a mobile node is assigned a multicast address within a domain for use with *micro mobility*.

^x Department of Electrical Engineering, University of Southern California (helmy@usc.edu)

^{*} Nortel Networks (jaseem@nortelnetworks.com)

The allocated multicast address is *locally-scoped* (i.e., unique only domain-wide). This allows for domain-wide address allocation schemes. Packets are *multicast-tunneled* to the mobile node within the domain. The multicast address of a mobile does not change throughout its movement within the domain. This allows for lighter-weight security during handover, as it is used for micro-mobility (i.e., intra-domain).

In this paper we present two different approaches to multicast-based micro mobility, one approach is based on *mobility proxies* and the other based on a novel scheme for *algorithmic mapping*. We compare such approaches and show that algorithmic mapping provides a more scalable and robust approach, and we develop efficient, yet simple, mechanisms to realize it. Furthermore, we conduct extensive simulations to compare the handover performance of our approach to other main approaches to micro-mobility. Results show that our scheme provides, on average, handover performance that is 1.23 to 3 times better than the other approaches. We also show how multicast aggregation may be used to reduce state requirements in the routers.

The rest of the document is outlined as follows. Section II introduces multicast-based mobility. Section III provides overview of the intra-domain architecture, and discusses the proxy-based approach. Section IV describes the algorithmic mapping approach in detail. Section V gives evaluation and comparison results. Section VI discusses multicast state aggregation, while Section VII discusses related work. We conclude in Section VIII.

II. Multicast-based Mobility (M&M)

Performance during *handover* is a significant factor in evaluating performance of wireless networks. IP-multicast[27][2] provides efficient location-independent packet delivery. The receiver-initiated approach for IP-multicast enables receivers to join to a nearby branch of an already established multicast tree. Multicast-based mobility (M&M)[1][8] uses this concept to reduce latency and packet loss during handover.

In multicast-based mobility, each mobile node (MN) is assigned a multicast address. The MN, throughout its movement, joins this multicast address through locations it visits. Correspondent nodes (CN) wishing to send to the MN send their packets to its *multicast* address, instead of unicast. Because the movement will be to a geographical vicinity, it is highly likely that the join from the new location, to which the mobile recently moved, will traverse a small number of

hops to reach the already-established multicast distribution tree. Hence, performance during handover improves considerably. An overview of this architecture is given in Figure 1. As the MN moves, it joins to the assigned multicast address through the new access router (AR). Once the MN starts receiving packets through the new location, it sends a prune message to the old AR to stop the flow of the packets down that path. Thus completing the smooth handover process. In spite of its promise, we believe that many issues need to be addressed to realize multicast-based mobility in today’s Internet. These issues include scalability, multicast address allocation, multicast deployment and security.

Scalability of Multicast State. The state created in the routers en-route from the MN to the CN is source-group (S,G) state. With the growth in number of mobile nodes, and subsequently, number of groups (G), the number of states kept in the routers increases. In general, if there are ‘ x ’ MNs, each communicating with ‘ y ’ CNs on average, with an average path length of ‘ l ’ hops, then number of states kept in the routers is ‘ $x.y.l$ ’ states. Clearly, this *does not scale*.

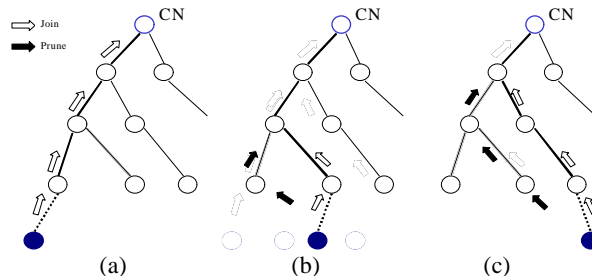


Figure 1. Multicast-based mobility. As the MN moves, as in (b) and (c), the MN joins the distribution tree through the new location and prunes through the old location.

Multicast Address Allocation. Inter-domain M&M requires each MN to be assigned a *globally-unique* multicast address. Using a global multicast address for each MN may be wasteful and requiring uniqueness may not be practical¹.

Ubiquitous Multicast Deployment. Inter-domain M&M assumes the existence of inter-domain multicast routing. We believe, however, that incremental deployment and interoperability should be an integral part of any architecture for IP mobility.

Security Overhead. Security is critical for mobility support, where continuous movement of mobiles is part of the normal operation. Such setting is prone to *remote redirection* attacks, where a malicious node redirects to

¹ Multicast address allocation is an active area of research [15]. We envision the number of MNs to grow tremendously.

itself packets that were originally destined to the mobile. The problem is even more complex with multicast, where any node may join the multicast address as per the IP-multicast host model. These security measures are complex and may incur a lot of overhead. If such measures are invoked with every handover, however, it may overshadow the benefits of efficient handover mechanisms².

To address the above issues, we propose a new approach for intra-domain multicast-based mobility.

III. Intra-domain Architectural Overview

In our intra-domain architecture, a mobile node is assigned a multicast address to which it joins while moving. The multicast address, however, is assigned only within a domain and is used for *micro* mobility. While moving between domains, an inter-domain mobility (e.g., Mobile IP) protocol is invoked. Several mechanistic building blocks are needed to realize our proposed architecture. First, when the mobile moves into a new domain it is assigned a multicast address. What is the address allocation scheme? Second, packets destined to the mobile are multicast-tunneled by an *encapsulator* to the mobile node. How are the *encapsulator(s)* selected and where are they placed? To answer these questions, we investigate and compare two different approaches: (1) A *proxy-based* architecture, and (2) *Algorithmic mapping* architecture.

A. Reference Architecture

We consider an IP network for a single domain, as shown in Figure 2. The network is connected to the internet through Border Routers (BRs). An Access Point (AP) is the radio point of contact for a mobile node. A number of APs are connected to an Access Router (AR). From the access router's point of view, each AP is a node on a separate sub-net. When a mobile moves from one AP to another without changing AR is an intra-AR handover case that can be specific to AR implementation and is not considered in this paper.

When a mobile moves into a new domain it is assigned a unicast address, called regional care of address (RCOA). It is also assigned a multicast care of address (MCOA). Address allocation and management is discussed later in this paper. When a mobile moves

from one AR to another, it is said to handover from old AR (AR_{old}) to new AR (AR_{new}). We use this terminology throughout the rest of the paper.

First, we shall describe the proxy-based approach and discuss the problems associated with it.

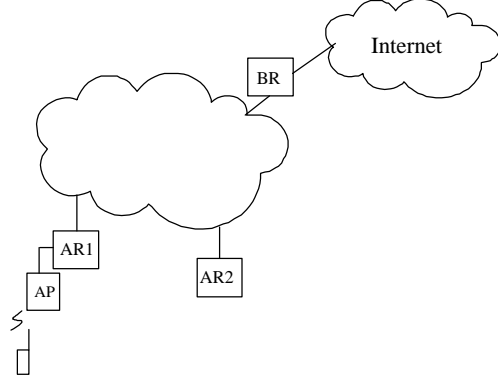


Figure 2. Reference Mobility Domain Network

B. Proxy-based Architecture

When a mobile node moves into a new domain, it contacts its access router (AR). The AR performs the necessary per-domain authentication and security measures, then assigns RCOA for the mobile node (MN). As shown in Figure 3, the AR then sends a *request* message to the mobility proxy (MP) to obtain a multicast address for the visiting MN. The request message includes the home address of the mobile node and its home agent's address. Upon receiving the request the MP performs two tasks. The first is to register its own address with the MN's home agent. The second task is to assign a multicast address for the visiting MN, send a *reply* message to the AR and keeps record of this mapping. The mapping is used for packet encapsulation later on.

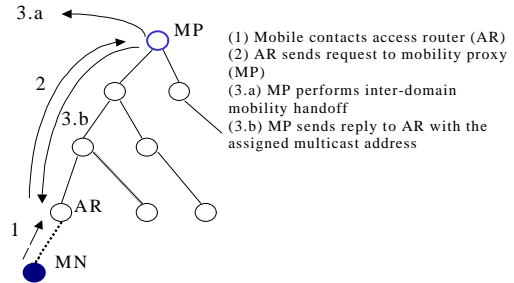


Figure 3. Event sequence as the mobile node moves into a domain.

Once this step is complete, the visiting MN joins the assigned multicast address (G). The joins are sent to the proxy-group pair (MP, G) and are processed as per the underlying multicast routing. The MN continues to

² Providing a comprehensive security solution for IP mobility is beyond the scope of this work. We believe, however, that our schemes relax security requirements during handover.

move within the domain using the same multicast address. The assigned multicast address is locally-scoped to the domain. Handover is performed using standard join/prune mechanisms and only lightweight intra-domain security is required in this case.

When packets are sent to the MN, they are forwarded to the MP using inter-domain mobility. The packets are then encapsulated by the MP, based on the mapping, and sent down the multicast tree to the MN. The MN uses the unicast RCOA for sending packets. To avoid single-point-of-failure scenarios multiple MPs are used. These MPs are typically placed at the border of the domain or at the center of the network³. An algorithm similar to [26] may be used for dynamic MP liveness and election mechanisms.

Several issues need to be addressed in the above architecture. First, the MPs need to maintain unicast-to-multicast address mapping for all visiting MNs. The scalability of such a scheme is of question. Second, complex robustness algorithms are needed to maintain MP liveness information, requiring initial configuration and setup. Third, the service disruption effect of MP failure is not clear. Since the MP registers its own address with the home agent and is used to encapsulate incoming packets, this introduces a *third-party-dependence* problem that is undesirable. In addition, MPs should run a multicast address allocation scheme to ensure collision-free address assignment.

To address these problems we propose a novel approach based on *algorithmic mapping* that obviates the need for explicit unicast-to-multicast mapping, and eliminates the need for complex address allocation.

IV. Algorithmic Mapping Architecture

We provide mechanisms for address management and duplicate address detection, and inter-AR handover.

A. Overview

When a mobile moves into a new domain it is assigned RCOA by the AR and the mobile performs *inter-domain mobility itself* (i.e., it registers the RCOA with its home agent). The AR *automatically infers* the multicast address (MCOA) for the mobile node from the assigned unicast address through a straight forward *algorithmic mapping*, described later in this section. The AR then triggers a Join message for MCOA to establish the multicast tree. Packets destined to the MN

are identified by the border router (BR) as being destined to a node on the m-subnet. The m-subnet is discussed below. The BR maps the destination unicast address to the *multicast address* and transmits the packets to the MN down the multicast tree. The serving AR changes the destination address from multicast to the unicast address. Since the destination address is modified twice within the network and restored to the RCOA by the AR, the packet does not cause security association violation at the mobile node.

This architecture provides several *advantages* over the proxy-based approach. It *avoids the third-party-dependence* on the MP. Moreover, since algorithmic mapping is used, *no explicit mapping is kept* or maintained by the encapsulator, which solves the mapping scalability problem and provides a more robust mechanism.

B. Address Management

The number of multicast addresses required is proportional to the number of mobile nodes in the domain. The scope of an MCoA is local to the domain where it is used. The IPv6 multicast address provides facility to define scope within the address [34]. Hence, in the rest of the paper we consider IPv6 address for both RCoA and MCoA.

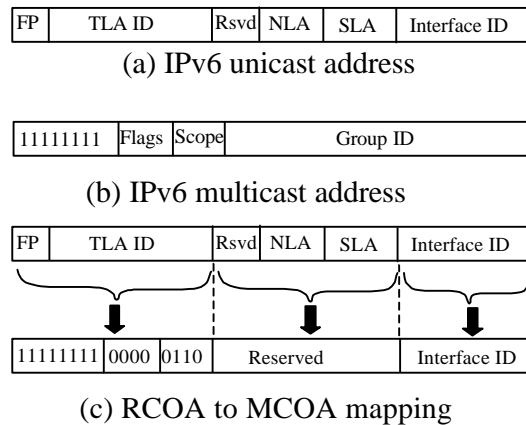


Figure 4. Algorithmic mapping

The standard IPv6 unicast and multicast address architectures [34] are shown in Figure 4 (a) and (b). We modify the group bits to include interface ID as the group ID, and the remaining suffix of the group ID is left reserved. The 64-bit interface ID address space is large enough for all the mobiles within a domain. Reserved bits are ignored by multicast routing. We also define a new scope: micro-mobility scope with value 0x6. The SLA is a 16-bit long field, used to create local

³ Network center are nodes with min(max distance) to any other node [28].

hierarchy and identify subnets [35]. A single subnet ID, identifying *m-subnet*, is set aside for RCoA. The *m-subnet* is a unique subnet that is characterized by the mobility where mobile nodes can use their RCoA to establish communication through any AR at the edge of the network. Hence, the *m-subnet* can be viewed as a logical subnet formed by all APs at the edge of the network. All ARs include the prefix for *m-subnet* in their router advertisements [39].

When a new MN moves into the domain, the AR replaces the $\langle \text{FP}, \text{TLA ID} \rangle$ of the RCoA with the multicast $\langle \text{FP}, \text{flag} (0000), \text{scope} (0110) \rangle$ values to form the MCoA. This provides a simple, yet very efficient and unique *algorithmic mapping*. A mobile node acquires RCoA on the *m-subnet* through either autoconfiguration or DHCP [36]. The autoconfiguration requires duplicate address detection (DAD) [37] on every subnet. In our scheme the mobile obtains RCoA and MCoA once it is connected to the network. We propose a scheme below that detects address duplication within the *m-subnet*, which is performed once at the AR during initial address assignment. The mobile afterward is able to move freely without running DAD at any other AR. When a mobile first connects to the network it performs a high latency inter-domain handover, hence duplicate address resolution latency is overshadowed by this handover latency.

C. Network-wide Duplicate Address Detection

Each Mobile user is assigned a globally-unique International Mobile Subscriber Identity (IMSI)[38]. MNs can form the interface identifier of IPv6 address from IMSI. They learn the network prefix for *m-subnet* from the ARs' router advertisements[39], and combine it with the interface identifier to form the RCoA.

Before configuring itself with this address, a mobile node runs Duplicate Address Detection (DAD) [37]. During DAD it sends Neighbor Solicitation (NS) message. The AR intercepts all NS messages for DAD. The LDAP directory server, as shown in Figure 5, keeps a list of interface identifiers of all mobiles in the network. The network control plane typically contains an LDAP server (or a similar central repository server). Hence, our mobility solution does not introduce a new component in the control plane. The LDAP directory servers provide reliability and robustness. The AR queries the LDAP directory server for the interface identifier extracted from the Target Address of the Neighbor Advertisement (NA) message. If the LDAP server returns an entry with the same identifier value, it

shows that a mobile is connected to the network with the same interface identifier value. The AR then returns NA message to the mobile indicating duplication. In case of no duplication, the AR updates the LDAP server with a new entry for the receiving interface identifier. Only when a MN leaves the network the related LDAP entry must be purged to avoid future conflicts if the MN revisits the network.

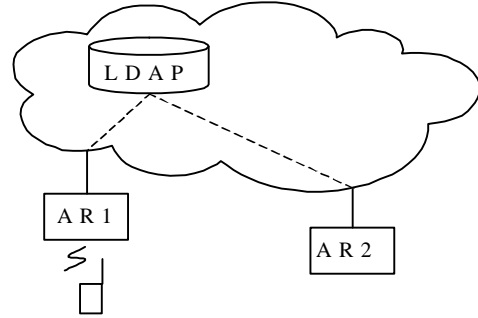


Figure 5. Network-wide DAD configuration

D. Inter-AR Handover within a Domain

When a mobile moves from one AR to another, a handover event takes place between the two routers. The handover involves path setup inside the network to redirect the incoming traffic flow to the new AR. Multicasting allows path setup to the new access router before the mobile is actually connected to it. Here, we introduce such a *proactive path setup* approach.

We define a set of adjacent access routers as the Control Access Router Set (CARset). The adjacency can be established based on the adjacency of the radio coverage area of the serving AR in case of cellular wireless network. The serving AR is called the Head of the CARset. Thus, there is a unique CARset defined for every AR. For example, in Figure 6 AR1 to AR7 constitute a CARset for AR1, which is the serving AR for the mobile. The mobile can move to any of the ARs in the CARset.

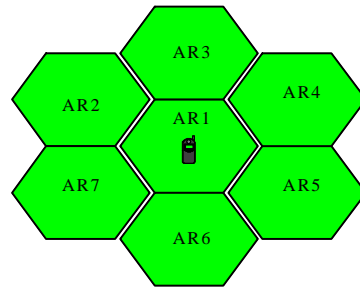


Figure 6. Control Access Router Set

A site-local multicast group address is given to each CARset, called CARset group address (CGA). Every AR that is a member of a CARset must join to the corresponding CGA, so that they can receive signalling from the serving AR. For example, in Figure 6, every AR surrounding AR1 become members of AR1s CARset (CGA1) by joining CGA1. Similarly, AR1 must also join six other CARsets corresponding to adjacent routers AR2 to AR7.

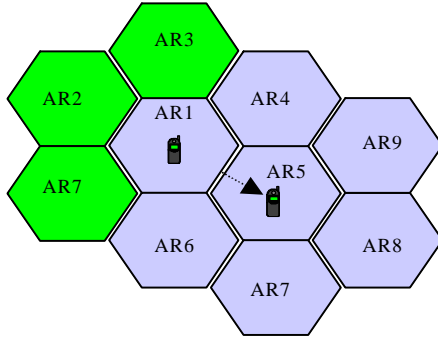


Figure 7. Handover across CARs

We define three new signaling messages as follows:

1. *J-message* causes the receiving router to *join* the multicast group identified in the message.
2. *L-message* causes the receiving router to *leave* the multicast group identified in the message.
3. *HO* message exchanged between the two routers involved in *handover*. Its parameter includes the mobile's RCOA and MCOA.

We explain the handover algorithm by using the example depicted in Figure 7. Consider the MN moving from AR1 to AR5. When connectivity is established between MN and AR5, AR5 multicasts a J-message <MCOA> to the members of its CARset (CGA5) requesting them to join the mobile's MCOA. It then sends HO <RCOA, MCOA> message to AR1 to initiate the prune process. When AR1 receives HO message, it multicasts an L-message <MCOA> to members of its CARset (CGA1) requesting them to leave the MCOA.

Although the ordering of (J => HO => L) messages ensures that L-message is initiated after J. The order of message reception, however, is not guaranteed to both CARsets. Depending on the order of arrival of J and L messages at an AR that is a member of both CARsets, it may leave the MCOA whereas it is supposed to have remained joined to that group.

To ensure proper operation in such scenario, we introduce the following mechanism. Each AR keeps its membership status in a 4-tuple <MCOA, Serving Router (SR), CGA, State> table. The table contains one row for every mobile roaming in a CARset of which the

access router is a member. There are two states defined: *Joined* and *Left*. The rules for updating the table specify that an AR only accepts L-message for a MCOA, if the source of the L-message matches the SR in the MCOA's entry (i.e., the AR has joined the MCOA on the request of the same SR)⁴. Otherwise, the L-message is discarded. The AR accepts all J-messages and creates/updates the related MCOA entry to include the source of the J-message (as the SR), CGA to the SR's CGA (as the entry's CGA), and the state to *Joined*.

Consider the example shown in Figure 7. Assume that the mobile's MCOA is MN1 and after power up in the domain it connects to AR1, which then multicasts a J-message to its CARset (CGA1). When AR4 receives the J-message, it creates an entry for the MCOA in *Joined* state as shown in Figure 8 (a). When the MN moves to AR5, AR5 becomes the new SR. Then AR5 multicasts a J-message to its CARset (CGA5) and then sends a HO message to the old serving router AR1.

Since AR4 is a member of both CGA1 and CGA5, it receives both J-message from AR5 and L-message from AR1. After receiving the J-message the table entry is updated as shown in Figure 8 (b). If received after the J-message, the L-message is discarded. Thus, AR4 remains joined to MN1. If received before the J-message, however, the L-message may cause AR4 to leave the MN1, which interrupts packet flow to AR4 until it receives J-message and joins the MN1 group. The outage may be minimized by delaying the leave operation. A simple scheme can be employed that periodically checks the table to leave the MCOA trees and purge all the entries that are in the Left state.

MCOA	Serving Router	CGA	State
MN1	AR1	CGA1	Joined
• • •	• • •	• • •	• • •

(a)

MCOA	Serving Router	CGA	State
MN1	AR5	CGA5	Joined
• • •	• • •	• • •	• • •

(b)

Figure 8. Table at AR4 (a): Table state when MN1 is connected to AR1 (b): Table state after MN1 moved to AR5

⁴ To account for lost L-message, or crash of the SR, a soft-state mechanism is used. SR sends periodic J-messages containing table changes (if any) and providing liveness.

V. Evaluation and Comparison

In order to evaluate the handover performance of our proposed micro-mobility scheme we have simulated it and conducted route analysis using the network simulator NS-2[17]. The evaluation metric used is the delay incurred to re-establish data reception during handover; when a mobile node connects to a new access router (AR_{new}) and disconnects from the old access router (AR_{old}). In addition, we compared our proposed scheme to two main approaches to micro-mobility; the hierarchical mobility approach[12][19][23][29][31], and seamless handover approach[5][22][33]. Hierarchical mobility defines two levels of foreign agents (FAs), as shown in Figure 9 (a). When a MN moves, it contacts its subnet FA, which in turn contacts its domain FA (FA_{domain}), thus localizing the registration to the current domain. The home agent (HA) is only contacted when moving from one domain to another⁵. Handover delay in such scheme is a function of the number of links (FA_{links}) from the new FA (or AR_{new}) to FA_{domain} .

In the seamless handover (SH) scheme, shown in Figure 9 (b), AR_{new} contacts AR_{old} to update the location of the mobile node. Packets are then sent/tunneled from AR_{old} to AR_{new} . The handover performance is a function of the number of links (SH_{links}) from AR_{new} to AR_{old} . In our multicast-based scheme, the handover latency is a function of the number of added links (L) from AR_{new} to the nearest point of the multicast tree. For example, in Figure 1 (b) $L=2$, and in (c) $L=3$. For simplicity, we assume that one AR is joined at a time, hence the results give an upper bound on handover delay in our scheme.

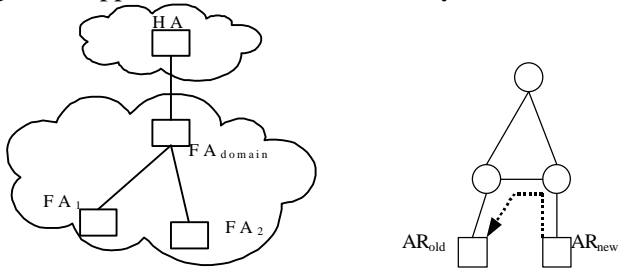


Figure 9 (a) Hierarchical MIP

(b) Seamless handover

We use various topologies for our evaluation, mainly focusing on intra-domain topologies. We use a mix of real-world, random and transit stub topologies (generated by GT-ITM[32]) with a range of 47 to 300

⁵ We shall only address quantitative differences in handover performance, and will not address qualitative differences concerning the need to configure the different levels of FAs or providing FA robustness mechanisms. These factors render our mechanism even more appealing, however.

nodes, as shown in Table 1.⁶ We use three different mobility models. The first (called *random*) allows the MN to visit any node in the topology randomly. The second (called *neighbor*) allows the MN to visit only directly-connected nodes in the next movement step. The third movement pattern (called *cluster*)⁷ allows the MN to connect randomly to only one of 6 nodes that are likely to fall within the same cluster as the MN⁸.

name	nodes	links	av deg	name	nodes	links	av deg
R-50	50	217	8.68	TS-100	100	185	3.7
R-100	100	950	19	TS-150	150	276	3.71
R-150	150	2186	29.15	TS-200	200	372	3.72
R-200	200	3993	39.93	TS-250	250	463	3.72
R-250	250	6210	49.68	TS-300	300	559	3.73
TS-50	50	89	3.63	ARPA	47	68	2.89

Table 1 Simulation Topologies. TS: transit stub, R: random, ARPA: original arpanet based on real data.

For every topology and movement pattern 10 simulations were run, with different random seeds for choosing the BR or FA_{domain} (i.e., 1080 simulations in total). Each simulation consisted of 100 movement steps. With every movement the number of links traversed by control messages to complete the handover was recorded; L for our scheme, AF_{links} for hierarchical MIP, and SH_{links} for seamless handover.

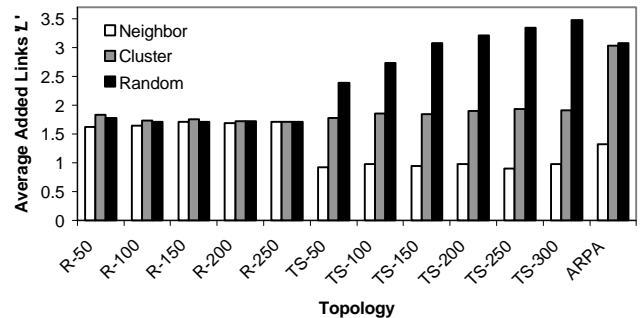


Figure 10. L for various topologies and movements

Results for handover performance of our scheme are shown in Figure 10. The average number of added links ' L ' during movement was found to be 2.49 for

⁶ Larger topologies were also simulated, for which results were even more favorable to our scheme.

⁷ In the transit-stub topologies, nodes were numbered sequentially in a cluster, so node numbering had geographical significance. We assumed a cellular system with 7 cell reuse architecture.

⁸ We believe these models give a reasonable mix for mobility. *Random* provides an upper bound on handover latency where the next AR maybe unrealistically far away. *Neighbor* establishes a lower bound on delay where the AR_{new} is directly attached to AR_{old} . *Cluster* gives an average case that is more realistic. Previous mobility models developed for geographical movement are inapplicable in our case as they do not model *network proximity*.

random movement, 1.28 for neighbor, and 1.91 for cluster movement. Overall, on average, control messages traverse 1.89 links to the closest point on tree.

For comparison, we define the ratios SH_{links}/L (or SH/L) and AF_{links}/L (or AF/L). Results for SH/L are shown in Figure 11. The average ratio was 1.47 (random), 0.84 (neighbor) and 1.38 (cluster), with an overall average of 1.23. It is interesting to note that the only instance where seamless handover outperformed our scheme is for neighbor movement over random topologies. This is due to the fact that $SH_{links} = 1$ always in this case. If we exclude the random topologies the numbers change to 1.77 (random), 1.01 (neighbor), 1.62 (cluster), with overall average of 1.47.

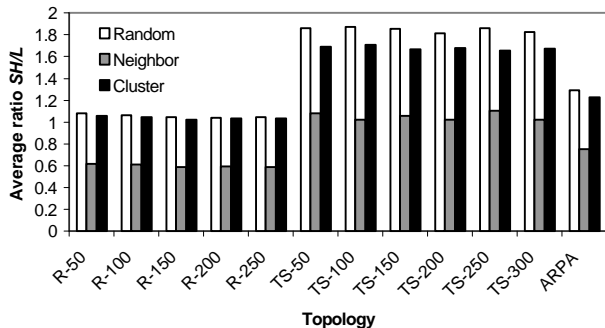


Figure 11. SH/L for various topologies and movements

For the FA/L ratio, the results pointed to a very clear advantage of our scheme. Averages obtained were 1.51 (random), 3.15 (neighbor), 2.06 (cluster), and 2.24 (overall average), as shown in Figure 12. Excluding random topologies these numbers become 1.82, 4.61, 2.78, and 3.07, respectively. A sum of these observations is given in Table 2.

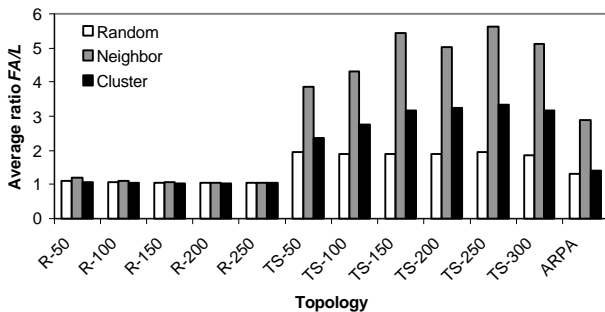


Figure 12. FA/L for various topologies and movements

In conclusion, results of our simulation show that the multicast-based approach outperforms the two other micro-mobility approaches considered in the study. Specifically, our approach achieved better handover

performance in over 94% of the simulation scenarios, on average outperforming the other approaches by ratios ranging from 1.23 to about 3 times.

FA/L		SH/L	
Avg	Avg w/o r	Avg	Avg w/o r
2.24	3.07	1.23	1.47

Table 2. Summary of Comparison Results ('w/o r' is the average if we exclude random topologies).

VI. Multicast State Aggregation

In addition to the architectural consideration we presented, the scalability of M&M approach can be improved by aggregating multicast states at routers. To our knowledge, this is the first work that addresses state aggregation in IP mobility, and one of the very few to address multicast state aggregation. We investigate several techniques for state aggregation. Aggregation used in unicast routing in the Internet is *prefix* aggregation⁹. Another kind of aggregation is the *bit-wise* aggregation¹⁰. Due to *continuous movement* of the mobiles it is hard to predict efficiency of the aggregation schemes. We analyze the two schemes using the *aggregation ratio*; number of states before aggregation to the number of states after aggregation. Figure 13 shows aggregation ratio for both techniques.

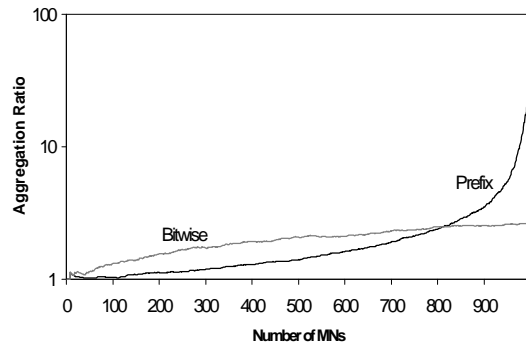


Figure 13. Bit-wise aggregation vs. prefix aggregation¹¹.

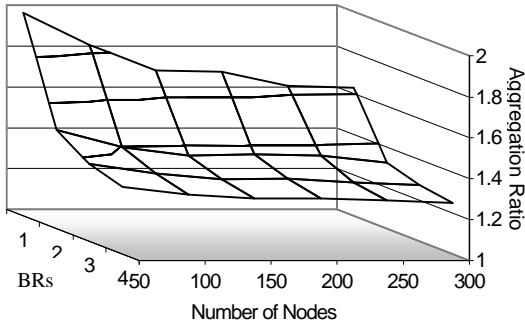
⁹ In prefix aggregation, two states can be aggregated only if they have the same unicast address prefix.

¹⁰ Bit-wise aggregation works with bits (e.g., entries 1 (binary '01'), and 3 (binary '11') can be aggregated as 'x1'), which, intuitively, provides more opportunity for aggregation. This may not always hold, however. For example, a sequence of {0,4,1,2,3} leads to 3 states in the bit-wise aggregation, whereas with prefix aggregation it leads to 2 states.

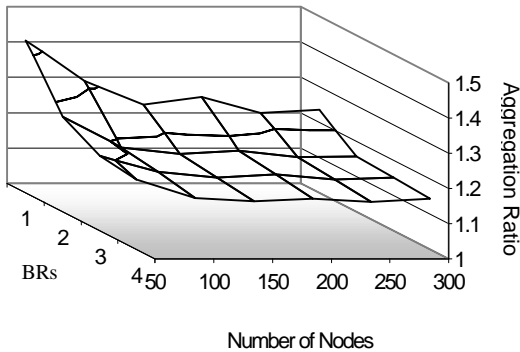
¹¹ Out of 1000 numbers, distinct numbers are chosen randomly until the whole population is covered. We obtained similar results with

It is interesting to note the cross over point at 80% population. Up to 80% of the population bit-wise aggregation outperforms prefix aggregation by a factor of 1.32. Hence, we choose to use bit-wise aggregation.

Another classification of multicast aggregation is perfect vs. lossy aggregation. Perfect aggregation compares all the fields of the multicast state, whereas lossy aggregation does not compare the outgoing interface list and may lead to extra network overhead (i.e., packets may be sent over a link that does not reach a receiver). We studied both bit-wise lossy and perfect aggregation. We have simulated topologies with 47 to 300 nodes, 1 to 4 BRs, and 250k MNs).



(a) Lossy Aggregation



(b) Perfect Aggregation

Figure 14. Aggregation ratio for lossy and perfect aggregation with various topologies and multiple BRs

We observed an obvious trend for states to be concentrated in a small number of nodes in the topology. In general, only 20% of the nodes had more than the average states per node, and around 60% of the nodes had 1% or less of the total number of MNs. This is a strong indication that state distribution across nodes is skewed, with highest aggregation ratios recorded in

other simulations. For in-sequence numbers (unlikely in our case) aggregation ratios are identical for both schemes.

the most concentrated nodes. Aggregation ratio averages are shown in Figure 14. The average aggregation ratio per node ranges from 1.25 to 1.99 for lossy aggregation and 1.14 to 1.43 for perfect aggregation. Evidently, lossy aggregation achieves better aggregation ratio. The trend is clear; for the same number of visiting MNs, as the number of nodes in the topology increases, the state concentration in the nodes decreases and the aggregation ratio decreases. Furthermore, as the number of border routers increase, the concentration of states in the nodes decrease and the aggregation ratio decreases. Also, the ratio increases as the average node degree decreases (i.e., number of links decreases, hence the concentration of states increases).

VII. Related Work

Several architectures have been proposed to provide IP mobility support. In Mobile IP (MIP)[3], every mobile node (MN) is assigned a *home* address and *home agent* (HA) in its home subnet. When the MN moves to another *foreign* subnet, it acquires a care-of-address (COA) through a foreign agent (FA). The MN informs the HA of its COA through a *registration* process. Packets destined to the MN are sent first to the HA, then are *tunneled* to the MN. This is known as *triangle routing*, a major drawback of MIP. *Route optimization* [4] attempts to avoid triangle routing by sending *binding updates*, containing the current COA of the MN to the correspondent node (CN). However, communication overhead during handover renders this scheme unsuitable for *micro* mobility. In[16] end-to-end IP mobility is proposed, based on dynamic DNS updates. When MN moves, it obtains a new IP-address and updates the DNS mapping for its host name. This incurs handover latency due to DNS update delays and is not suitable for delay bounded applications. Also, the scheme is not transparent to the transport protocol that is aware of the mobility.

In[10] the HA tunnels packets using a pre-arranged multicast group address. The access router, to which the MN is currently connected, joins the group to get data packets over the multicast tree. This approach suffers from the triangle routing problem; packets are sent to HA first and then to MN. Multicast-based mobility is proposed in [1] and [8]. Each MN is assigned only a unique multicast address. Packets sent to the MN are destined to that multicast address and flow down the multicast distribution tree to the MN. The CN tunnels the packets using the multicast address. This approach avoids triangle routing, in addition to reducing

handover latency and packet loss. The study in[1] quantifies the superiority of handover performance for multicast-based mobility over Mobile IP protocols. These schemes, however, suffer from several serious practical issues, including scalability of multicast state, address allocation and dependency on inter-domain multicast. We address these issues in our work.

Several approaches have been proposed for *micro* mobility[18]. The general approaches include mobile-specific routing, hierarchical approaches and seamless handover. Mobile-specific route approaches include cellular IP[20] and Hawaii[21]. A domain-gateway registers its address with the HA (this has similarities to our proxy-based approach) and forwards the packets to the MN. The MN's home address is used within the domain. These approaches need special signaling to update mobile-specific routes and require changes in packet forwarding and unicast routing in all the routers. In cellular IP[20], signaling is data-triggered to create paths by having routers snoop on the data packets. Hawaii[21] proposes a separate routing protocol and requires explicit signaling from the mobiles. In a way, these approaches attempt to create a distribution tree using extra routing entries for the mobile, similar to what multicast does. Our approach builds upon existing multicast mechanisms as opposed to re-creating them. Approaches based on seamless handover between old and new access routers, involve fairly complex signaling, buffering and synchronization procedures. Router-assisted smooth handoff in MIP[5], and edge mobility[22] belong to this category. Fast Handover [33] introduces fast tunnel set-up between AR_{old} and AR_{new} as soon as the layer 2 handoff is detected. The tunnel avoids packet losses caused by path set-up delay inside the mobility domain. In a way it is complementary to our multicast-based routing inside the mobility domain. Unlike fast handover, however, our m-subnet idea considers the edge of the network as a single subnet and allows mobile node to carry RCOA and MCOA across ARs, which reduces the handover latency. Approaches using a hierarchy employ a gateway per-domain and need to keep a location database to map identifiers into locations. This mapping suffers from scalability and robustness problems as was noted earlier in this paper. In[12] a hierarchy of foreign agents is created at the local, administrative domain and global levels. In[19] a multi-level hierarchy is used in which packets from the HA arrive at a root FA where they are tunneled to a lower level FA and then to the MN. Hierarchical MIP[29] builds a network of tunnels (overlay network) between FAs. [23][31] also use a

notion of mobility agent for localized handoff within a domain. We have shown in this paper that our multicast-based intra-domain mobility scheme clearly outperforms seamless handover and hierarchical approaches and is simpler.

Very little work has been done in the area of multicast state aggregation. Work in[24] proposes an interface-centric model for aggregation. This approach, however, benefits from having a large number of group members, which does not apply in our case. [25] studies strict, pseudo-strict and leaky *prefix* aggregations for wide-area multicast routing. We show, however, unlike most previous studies, that the *bit-wise* aggregation usually achieves better gains than prefix aggregation.

VIII. Conclusions

We have presented a novel approach to IP micro-mobility using intra-domain multicast-based mobility. Our approach solves major challenging problems facing the deployment of multicast-based mobility. In terms of multicast state scalability we note that the multicast state growth is $O(G)$ for the architecture presented in this study, as opposed to $O(S \times G)$ in[1][8]. Our novel algorithmic mapping scheme from unicast to multicast address ensures collision-free assignment by providing unique and consistent mapping throughout the network. This solves the address allocation problem and provides robustness and per-domain privacy as multicast packets are not forwarded out of the domain. In addition, we present a new proactive path setup scheme to improve handover performance. Our extensive simulations show that, on average, the number of added links ' L ' during handover is 1.89. In addition, our scheme clearly outperforms the other micro-mobility schemes studied in this paper in over 94% of the scenarios, obtaining, on average, between 1.23 and 3 times less handover delay. Furthermore, we provide a simple efficient scheme for multicast state aggregation. Our bit-wise lossy aggregation obtains between 1.25 and 2 aggregation ratio. Our work is the first to provide algorithmic mapping approach to multicast-based mobility. We are also the first to study, in details, the effects of multicast state aggregation in a mobility context.

References

- [1] A. Helmy, "A Multicast-based Protocol for IP Mobility Support", ACM Second International Workshop on Networked Group Communication (NGC 2000), ACM SIGCOMM-sponsored, Palo Alto, November 2000.

- [2] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, V. Jacobson, M. Handley, C. Liu, P. Sharma, "Protocol Independent Multicast – Sparse Mode (PIM-SM): Protocol Specification", *RFC 2362/2117 of IETF/IDMR*, March '97/'98.
- [3] C. Perkins, "IP Mobility Support", *RFC 2002, Internet Engineering Task Force*, October 1996.
- [4] C. Perkins, D. Johnson, "Route Optimization in Mobile IP", *Internet Draft, Internet Engineering Task Force*, February 2000.
- [5] C. Perkins and D. Johnson, "Mobility Support in IPv6", *Proceedings of MobiCom'96*, November 1996.
- [6] D. Johnson, C. Perkins, "Mobility Support in IPv6", *Internet Draft, Internet Engineering Task Force*, March 2000.
- [7] A. Myles, D. Johnson, C. Perkins, "A Mobile Host Protocol Supporting Route Optimization and Authentication", *IEEE Journal on Selected Areas in Communications*, vol. 13, No. 5, p. 839-849, June 1995.
- [8] J. Mysore, V. Bharghavan, "A New Multicasting-based Architecture for Internet Host Mobility", *Proceedings of ACM MobiCom*, September 1997.
- [9] J. Mysore, V. Bharghavan, "Performance of Transport Protocols over a Multicasting-based Architecture for Internet Host Mobility", *International Conference on Communications (ICC)'98*, vol. 3, p. 1817-1823, 1998.
- [10] S. Seshan, H. Balakrishnan, R. Katz, "Handovers in Cellular Wireless Networks: The Daedalus Implementation and Experience", *Kluwer Journal on Wireless Networks*, 1995.
- [11] H. Balakrishnan, S. Seshan, R. Katz, "Improving Reliable Transport and Handover Performance in Cellular Wireless Networks", *Proceedings of ACM MobiCom'95*, November 1995.
- [12] R. Caceres, V. Padmanabhan, "Fast and Scalable Handovers for Wireless Internetworks", *Proceedings of ACM MobiCom'96*, November 1996.
- [13] R. Caceres, V. Padmanabhan, "Fast and Scalable Wireless Handovers in Support of Mobile Internet Audio", *ACM Journal on Mobile Networks and Applications*, vol. 3, No. 4, December 1998.
- [14] A. Acampora, M. Naghshineh, "An Architecture and Methodology for Mobile-Executed Handover in Cellular ATM", *IEEE Journal on Selected Areas in Communications*, vol. 12, No. 8, p. 1365-1375, October 1994.
- [15] S. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, M. Handley, "The MASC/BGMP Architecture for Inter-domain Multicast Routing", *Proceedings of ACM SIGCOMM*, August 1998.
- [16] A. Snoeren, H. Balakrishnan, "An End-to-End Approach to Host Mobility", *Accepted to ACM MobiCom '00*, August 2000.
- [17] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, H. Yu, "Advances in Network Simulation", *IEEE Computer*, vol. 33, No. 5, p. 59-67, May 2000.
- [18] A. T. Campbell and J. Gomez IP Micro-Mobility Protocols, *ACM SIGMOBILE Mobile Computer and Communication Review (MC2R)*, 2001
- [19] E. Gustafsson, A. Jonsson, C. Perkins, Mobile IP Regional Registration, *Internet-draft, draft-ietf-mobileip-reg-tunnel-02*, March 2000.
- [20] A. Campbell, J. Gomez, S. Kim, A. Valko, C. Wan, Z. Turanyi, "Design, implementation, and evaluation of cellular IP" *IEEE Personal Communications*, Volume: 7 Issue: 4, Page(s): 42–49, Aug. 2000.
- [21] R. Ramjee, T. La Porta, L. Salgarelli, S. Thuel, K. Varadhan, L. Li, "IP-based access network infrastructure for next-generation wireless data networks", *IEEE Personal Communications*, Volume: 7 Issue: 4, Page(s): 34–41, Aug. 2000.
- [22] A. O'Neill, G. Tsirtsis, S. Corson, Edge Mobility Architecture, *Internet-draft, draft-oneill-ema-02.txt*, July 2000.
- [23] J. Kempf, P. Calhoun, C. Pairla, Foreign Agent Assisted Handover, *Internet-draft, draft-calhoun-mobileip-proactive-fa-01.txt*, June 2000.
- [24] D. Thaler, M. Handley, "On the aggregatability of multicast forwarding state", *IEEE INFOCOM 2000*, Page(s): 1654 -1663 vol.3, 2000.
- [25] P. Radoslavov, D. Estrin, R. Govindan, "Exploiting the Bandwidth-Memory Tradeoff in Multicast State Aggregation", *USC-CS-TR99-697*, Computer Science Department, USC, July 1999.
- [26] D. Estrin, M. Handley, A. Helmy, P. Huang, D. Thaler, "A Dynamic Bootstrap Mechanism for Rendezvous-based Multicast Routing", *Proceedings of IEEE INFOCOM '99*, New York, March 1999.
- [27] S. Deering, D. Cheriton, "Multicast routing in data internetworks and extended lans", *ACM Transactions on Computer Systems*, pp 85-111, May 1990.
- [28] E. Fleury, Yih Huang, P.K. McKinley. "On the performance and feasibility of multicast core selection heuristics", *7th International Conference on Computer Communications and Networks*, Page(s): 296–303, 1998.
- [29] H. Soliman, C. Castelluccia, K. Elmalki, L. Bellier, "Hierarchical Mobile IPv6", *Internet Draft, draft-ietf-mobileip-hmipv6-04.txt*, July 2001.
- [30] A. Misra, S. Das, A. Mcauley, A. Dutta, S. K. Das, "IDMP: An Intra-Domain Mobility Management Protocol using Mobility Agents", *I-Draft, draft-misra-mobileip-idmp-00.txt*, Jan 2000.
- [31] G. Dommety, "Local and Indirect Registration for Anchoring Handoffs", *I-Draft, draft-dommety-mobileip-anchor-handoff-01.txt*, December 2000.
- [32] K. Calvert, M. Doar, E. Zegura, "Modeling Internet Topology", *IEEE Comm*, p. 160-163, June 1997.
- [33] G. Dommety, et al, "Fast Handovers for Mobile IPv6", *Internet Draft, draft-ietf-mobileip-fast-mipv6-02.txt*, July 2001.
- [34] R. Hinden, S. Deering, "IP Version 6 Addressing Architecture", *Internet Draft, draft-ietf-ipngwg-addr-arch-v3-06.txt*, July 2001.
- [35] R. Hinden, M. O'Dell, S. Deering, "An IPv6 Aggregatable Global Unicast Address Format", *RFC 2374*, July 1998.
- [36] J. Bound, et al, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", *Internet Draft, draft-ietf-dhc-dhcpv6-19.txt*, June 2001.
- [37] S. Thomson, T. Narten, "IPv6 Stateless Address Autoconfiguration", *RFC 2462*, December 1998.
- [38] 3GPP, "3GPP TS Group Core Network; Numbering, addressing, and identification", *3GPP TS 23.003 V5.0.0 (2001-06)*, www.3gpp.org, June 2001.
- [39] T. Narten, E. Normark, W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", *RFC 2461*, December 1998.