A Comprehensive Study on Handover Performance of Hierarchical Mobile IPv6

Youn-Hee Han[1] and Dongwon Jeong[2]

[1] School of Internet-Media, Korea University of Technology and Education, Cheonan, Korea, E-mail: yhhan@kut.ac.kr
[2] Department of Informatics and Statics, Kunsan National University, Gunsan, Korea, E-mail: djeong@kunsan.ac.kr

Abstract. Recently, IETF has standardized Mobile IPv6 (MIPv6) and Hierarchical Mobile IPv6 (HMIPv6) for supporting IPv6 mobility. Even though existing literatures have asserted that HMIPv6 generally improves MIPv6 in terms of handover speed, they do not carefully consider the details of the whole handover procedures. In this paper, based on the current IETF standards of both MIPv6 and HMIPv6, we conduct a comprehensive study of all IP-level handover procedures. We also provide a mathematical analysis on MIPv6 and HMIPv6 performance in terms of handover speed, and reveal that the average HMIPv6 handover latency is not always lower than the average MIPv6 handover latency. A vital finding of our analysis is that some optimization techniques for movement detection and duplicate address detection are essential to increasing the benefit of HMIPv6.

1 Introduction

To support Internet mobility, a lot of research has been studied so far. IETF Mobile IPv6 (MIPv6) [1, 2] is one of the most important protocols to accommodate the increasing demand of end-to-end mobility in IPv6 Internet. In MIPv6, a mobile node (MN) is always addressable by its home address (HoA), which is the IPv6 address assigned within its home network. When an MN is away from its home network, packets can still be routed to it using the MN's HoA. Whenever an MN moves to a new subnet, it typically acquires a temporary address, called care-of address (CoA), through the address auto-configuration according to the methods of IPv6 neighbor discovery [3, 4]. Through the binding update procedure, an MN registers its temporal location to its home agent (HA) in its home network and all correspondent nodes (CNs) every time it moves. So, support for the route optimization is built in as a fundamental part of the protocol.

Even though MIPv6 provides the optimal packet routing, it requires high signaling cost and long handoff latency so that MIPv6 is not appropriate for environments in which MNs frequently change their point of attachment to the network. IETF Hierarchical Mobile IPv6 (HMIPv6) [5, 6] is an enhanced MIPv6 to minimize the signaling cost and the latency of updating the location of MN by using a local anchor point called Mobility Anchor Point (MAP). The MAP is intended to limit the amount of MIPv6 registration signaling outside the local domain. Domain can be an ISP network, a campus network, a company network, a set of LANs, or even a single LAN.

In this paper, we focus to qualitatively and quantitatively study the performance of MIPv6 and HMIPv6 in terms of handover speed. Recent IETF RFC 4140 [5] where the protocol of HMIPv6 is minutely described, firmly assures that MIPv6's handover latency is reduced by HMIPv6 since HMIPv6's registration procedure is mostly conducted within a domain. But, we have been wondering whether or not the average of HMIPv6's handover speed is really faster than the one of MIPv6's handover speed. A mathematical analysis is described in this paper to gratify the curiosity based on the recent IETF RFCs [1, 3–5]. The analysis points out that the average HMIPv6 handover latency is not always lower than the average MIPv6 handover latency. Furthermore, even the intra-domain handover latency of HMIPv6 is not much small compared with MIPv6 handover latency. We reveal that some optimization techniques for movement detection and duplicate address detection are required to shorten HMIPv6 handover latency and increase the benefit of HMIPv6.

This paper is organized as follows. In Section 2, we represent our system architecture and provide the parameters and metrics for our performance analysis. In Section 3, we analyze MIPv6 and HMIPv6 handover procedures. Section 4 conducts the performance evaluation about handover latency comparison between MIPv6 and HMIPv6. Finally, concluding remark is given in Section 5.

2 System Architecture and Performance Metrics

We consider an IPv6-based mobile system described in Fig. 1. The home network is where an MN gets its permanent IP address (home address). A foreign domain network (or a domain) can be a campus network, a company network, or a public access network. A domain is comprised of a border gateway (BG), access routers (ARs), and wireless point of attachments to which an MN can make a connection. Between a BG and ARs, there may be some intermediate routers. We assume that each AR has an interface with connection to a distinct set of wireless point of attachment. It also assigns only one network prefix per such an interface. The same network prefix cannot be assigned to the different AR's interface.

When MN moves to new subnet while preserving communication with a CN, MIPv6 and HMIPv6 usually assume that the network-layer handover procedure is initiated only after the link-layer handover procedure comes to end. We thus define *handover latency* as the elapsed time after the link-layer handover comes to end (MN re-establishes its connection to a new point of attachment) until MN receives the first data packet of ongoing session through the new AR of the new network.

To describe an analytical model, let us define the following parameters:

a - average number of hops between AR and MAP;
b - average number of hops between MAP and HA;
c - average number of hops between MAP and CN;

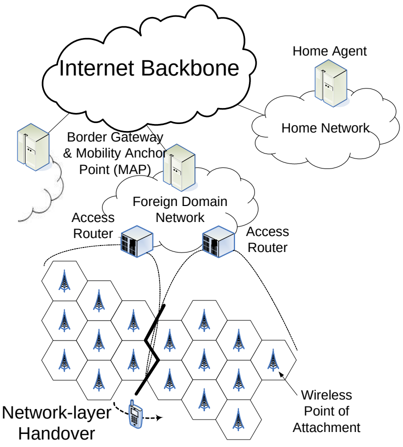


Fig. 1. Reference system architecture

d - average number of hops between HA and CN;
t_α - latency of an IP packet delivery between MN and AR (over wireless and wired medium);
t_β - latency of an IP packet's one hop delivery only through wired medium;
t_H - latency of a packet delivery between MN and HA ($t_H = t_\alpha + (a+b)t_\beta$);
t_N - latency of a packet delivery between MN and CN ($t_N = t_\alpha + (a+c)t_\beta$);
t_L - latency of a packet delivery between MN and CN via HA ($t_L = t_\alpha + (a+b+d)t_\beta$);
t_M - latency of a packet delivery between MN and MAP ($t_M = t_\alpha + at_\beta$).

The beginning procedure of network-layer handover is to determine whether or not it moves to a new IP subnet. This is called the *movement detection (MD)* procedure. The changes on the underlying link-layer status of MN might be locally relayed to network-layer of MN in the form of *link-up event notification*. Although such link-layer event trigger can facilitate MD procedure, it is not always available at all types of MNs. In this paper, the latency of MD procedure will be denoted T_{MD}. If it is decided to move to a new subnet, MN should generate a new CoA by using IPv6 stateless or stateful address auto-configuration. To verify the uniqueness of this CoA, MN should run *duplicate address detection (DAD)* procedure, before assigning the address to its interface.

The two procedures, MD and DAD, are the common parts in both MIPv6 and HMIPv6 handover procedures. After the two procedures come to an end, the location registration procedure starts. In fact, it is the location registration that differentiates HMIPv6 from MIPv6. We will use T_R to denote the latency of MIPv6 registration procedure. On the other hand, T_{HR} will denote the latency of HMIPv6 registration procedure. Using the defined symbols, the average handover latencies of MIPv6 and HMIPv6 are defined as follows:

$$\overline{T}_M = T_{MD} + T_{DAD} + T_R. \tag{1}$$

$$\overline{T}_H = T_{MD} + T_{DAD} + T_{HR}. \tag{2}$$

In the following sections, each latency parameters, T_{MD}, T_{DAD}, T_R, and T_{HR}, will be analyzed and concretized.

3 Analysis of MIPv6 and HMIPv6 Handover Procedures

3.1 Movement Detection Analysis

A complete MD analysis and its results are presented by [7], which is a outcome of our preceding research. In this paper, therefore, we give a concise description of them. For further details, refer to [7]. In IPv6-based mobile systems, MD usually relies on the reception of the router advertisement (RA) message [4] from a new AR. An RA message includes the AR's network prefix information from which an MN can determine if it moves to a new network. An AR sends (pseudo) periodic unsolicited RA messages to its all nodes. Whenever such an RA is sent from an AR, a timer is reset to a uniformly-distributed random value between the router's configured *MinRtrAdvInterval* and *MaxRtrAdvInterval*, which are respectively symbolized by R_m and R_M in this paper. When the timer is expired, new unsolicited RA is again sent. Movement is confirmed by the receipt of a new RA from a new AR. Let us assume T denotes the time interval between link-layer re-establishment and MN's first reception of a new RA. Using the equations introduced by [7], we can get the expectation of T as follows:

$$E_T = \frac{R_M^2 + R_M R_m + R_m^2}{3(R_M + R_m)} + t_\alpha. \tag{3}$$

There is another method to conform the movement with network-layer movement hint. MIPv6 specification [1] defines a new *RA interval option*, which is used in RA messages to advertise the interval at which the AR sends unsolicited RAs. An MN can keep monitoring for periodic RAs and interpret the absence of the exiting RAs as a movement hint. The indicated interval is equal to the router's configured value, *MaxRtrAdvInterval*. To differentiate new AR's *MaxRtrAdvInterval*, the old AR's *MaxRtrAdvInterval* will be symbolized by R_L. An MN may implement its own policy to determine the number of missing RAs needed to interpret that as a movement hint. In this paper, just one missing is assumed a movement hint. On receiving such a movement hint, MN can send the router solicitation (RS) message [4] to the all-routers multicast address. A new AR receiving such RS responses it by sending its RA message to the all-nodes multicast address. Movement is finally confirmed by the reception of such solicited RA from the new AR. Let us assume S denotes the time interval between the link-layer re-establishment and the receipt of the new solicited RAs. We can get the expectation of S as follows:

$$E_S = \frac{R_L}{2} + 2t_\alpha. \tag{4}$$

In order to get the average latency T_{MD}, we calculate the probability $P\{S < T\}$ that an MN first gets a solicited RA by noticing the existing RA's absence prior to get a new unsolicited RA. It is as follows:

$$\begin{aligned} P\{S<T\} &= \int_{t_\alpha}^{R_M+t_\alpha} P\{S < T|T = t\} f_T(t)dt = \int_{t_\alpha}^{R_M+t_\alpha} F_S(t) f_T(t)dt \\ &= \int_{t_\alpha}^{2t_\alpha} F_S(t) f_T(t)dt + \int_{2t_\alpha}^{R_M+t_\alpha} F_S(t) f_T(t)dt = \int_{2t_\alpha}^{R_M+t_\alpha} F_S(t) f_T(t)dt \end{aligned} \tag{5}$$

Then, we can finally acquire T_{MD} as follows:

$$T_{MD} = P\{S < T\} \cdot E_S + P\{S \geq T\} \cdot E_T. \tag{6}$$

Some MN can accelerate MD procedure by using the link-layer trigger [8, 9]. That is, the changes on the underlying link-layer status can be relayed to IP layer in a form of link-layer event notification, such as *link-up event*. On receiving such a link-up notification, the network-layer of MN can expedite (or optimize) MD procedure by sending RS message to get a solicited RA. A new AR receiving such RS responses it by sending RA message to the MN. Movement is finally confirmed by the reception of such solicited RA from the new AR. So, if MD procedure is executed with the link-layer notification, the optimized MD latency will be $T_{MD} = 2t_\alpha$.

3.2 Address Configuration Analysis

The current and simplest form of DAD was laid out as part of RFC 2462 "IPv6 Stateless Address Autoconfiguration" [3]. When a node wishes to create a new address on an interface, it combines the network prefix with a suffix generated from its interface identifier (IID). The IID can be either obtained from the interface hardware or generated randomly. This address is referred to as the *tentative address*. The node sends a neighbor solicitation (NS) message from the unspecified address to the tentative address. If the address is already in use by another node, that node will reply with a neighbor advertisement (NA) message defending the address.

Once a node has sent such an NS, it should waits for *RetransTimer* (=*RT*) milliseconds to see if a defending NA is forthcoming, and this solicit-and-wait process is repeated *DupAddrDetectTransmits* (=$DTimes$) times. During this process, the node cannot communicate with a node. As a result, with the assumption that there is no colliding node in the same subnet, DAD latency T_{DAD} is simply as follows:

$$T_{DAD} = RT \cdot DTimes. \tag{7}$$

According to [3], the default value of *RetransTimer* is 1000 ms, and by default the process is only done once, resulting in $T_{DAD} = 1000$ ms. Thus, the DAD procedure of RFC 2462 [3] is obviously a big factor to deteriorate the performance of MIPv6 and HMIPv6, particularly when MN in need of seamless handover runs it. So, some DAD optimization schemes have been suggested to reduce this problem [10, 11]. Optimistic DAD (oDAD) [10] is proposed based on the premise that DAD is far more likely to succeed than fail. An optimistic MN starts communication without RFC 2462 DAD procedure. oDAD modifies its IPv6 neighbor discovery protocol [4] while keeping backward interoperability. On the other hand, advance DAD (aDAD) [11] is a solution of pessimistic approach for IPv6 address conflict. In aDAD, each AR reserves a bunch of unique CoAs in advance and allocates one of them into a newly connected MN. Both schemes, oDAD and aDAD, do not cause any latency. So, if these DAD optimization schemes are applied to MIPv6 and HMIPv6, DAD latency will be written by $T_{DAD} = 0$.

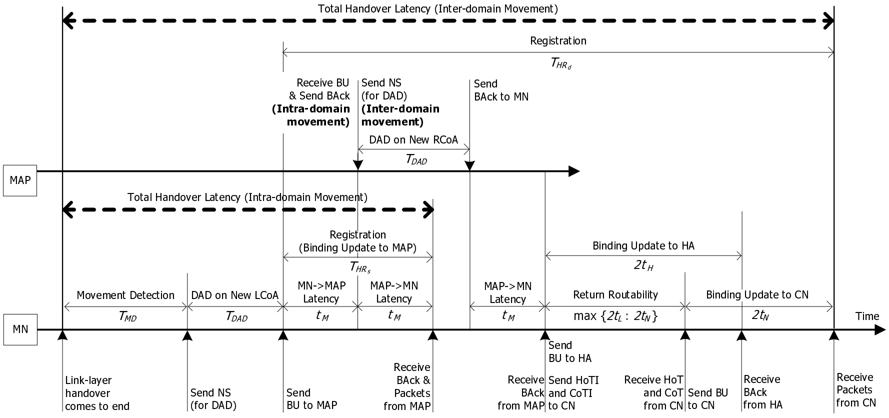


Fig. 2. Handover Procedure and Timing Diagram in HMIPv6

3.3 Registration Procedures

MIPv6 Registration Procedures: In MIPv6, an MN registers its temporal location to its HA in its home network and all CNs every time it moves. Each CN can have its own binding cache where HoA plus CoA pairs are stored. So, CN is able to send packets directly to an MN when it has a recent binding entry for the MN in its binding cache. The achieved route optimization improves data transmission rates between the MN and the CN. The time to complete MIPv6 registration procedure actually depends on how to fast update CN's binding cache with new CoA.

To prevent attackers from sending false BU messages, the BU should be authenticated using a cryptographic binding management key. The process to create the key value is referred to as the *return routability* procedure. For the details of return routability process, refer to [1]. As a result, MIPv6 handover latency is further increased by the time needed for the return routability as well as the BU procedures. Assuming $c < b + d$ (refer the parameter list in Section II), MIPv6 registration latency T_R is given by:

$$\begin{aligned} T_R &= max\{2t_L,\ 2t_N\} + 2t_N \\ &= 2t_L + 2t_N = 4t_\alpha + 2(2a + b + c + d)t_\beta. \end{aligned} \tag{8}$$

HMIPv6 Registration Procedures: In HMIPv6, an MN gets two CoAs: an on-link CoA (LCoA) and a regional CoA (RCoA) when it enters into a new domain. The former is the on-link CoA configured on an MN's interface based on the prefix advertised by its default AR, while the latter is an address on the MAP's subnet. The prefix used to form RCoA is usually advertised by MAP to all routers and MNs in the same domain through the dynamic MAP discovery scheme using MAP option [5].

For the location registration, an MN uses *local binding update (LBU)* as well as BU. LBU is sent only to the domain MAP and specifies the binding between MN's RCoA and LCoA. On the other hand, normal BU specifies the binding between MN's HoA and RCoA. It is sent to the HA and each of CNs outside

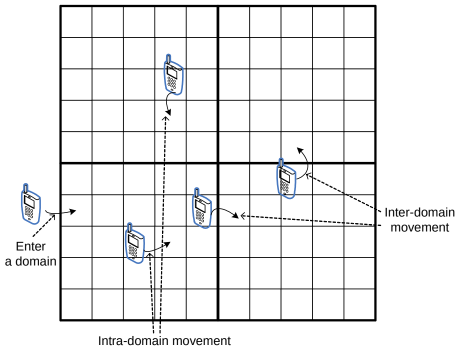


Fig. 3. Our subnet and domain overlay structure

the domain. When an MN moves within the domain and attaches to an AR, it configures new LCoA by using the on-link prefix advertised by the AR. It should be noted that RCoA remains constant in this intra-domain movement. MN then sends LBU to MAP. No BU message is sent outside domain. After MAP receives LBU from MN, all packets destined to the MN will be rightly transferred. Let us denote by T_{HR_s} the registration latency of intra-domain movement. It is simply given by:

$$T_{HR_s} = 2t_M = 2t_\alpha + 2at_\beta. \tag{9}$$

When MN moves into a new MAP domain and attaches to an AR, it configures new RCoA as well as new LCoA. RCoA is usually auto-configured by MN using the advertised MAP option. Then, it sends LBU to MAP. Only if the uniqueness of RCoA is confirmed from RFC 2462 DAD procedure, MAP records this binding in its binding cache, and sends a binding acknowledgement (BAck) message to MN. After receiving BAck from MAP, MN also sends normal BUs to its HA and CNs. Like MIPv6, the return routability procedure is still required to authenticate the BU message. Let us denote by T_{HR_d} the registration latency of inter-domain movement. Assuming $c < b + d$, it is as follows:

$$\begin{aligned} T_{HR_d} &= 2t_M + t_{DAD} + max\{2t_L,\ 2t_N\} + 2t_N \\ &= 6t_\alpha + (6a + 2b + 2c + 2d)t_\beta + RT \cdot DTimes. \end{aligned} \tag{10}$$

If a DAD optimization scheme, such as oDAD or aDAD, is used for RCoA configuration, it will be written by:

$$T_{HR_d} = 6t_\alpha + (6a + 2b + 2c + 2d)t_\beta. \tag{11}$$

4 Performance Evaluation

4.1 System and Mobility Models

The system architecture described in Section II is modeled as follows. Our model is similar to the one described in [12]. But, we slightly change it and include a new concept of subnet and domain overlay structure by which the service area is assumed to be covered (see Fig. 3). In the overlay architecture, the lower layer is comprised of subnets and the higher layer represents domains. For simplicity, we assume all domains overlay the same number of subnets. We also assume the homogeneous network of which all subnets in a domain have the same shape and size of mesh plane.

Let us assume that MN resides in a subnet for a period and moves to one of its four neighbors with the same probability, i.e., with probability 1/4. A domain is referred to as an *n-layer domain* if it overlays $4n^2 - 4n + 1$ subnets. We define a random variable M so that each MN moves out of a domain network at M's subnet movement. Then $E[M] = 1$ when $n = 1$. For $n \geq 2$, the expected number of M in a n-layer domain is computed as:

$$E[M] = \sum_{k=1}^{\infty} \sum_{y=0}^{2n-3} \sum_{j=0}^{2n-3} q_{(n-1,y)} p_{k,(n-1,y)(n,j)} k. \tag{12}$$

The detailed description of (12) can be found in [13], which is other outcome of our preceding research. By using (12), the latency of HMIPv6 registration procedure is more completely written as follows:

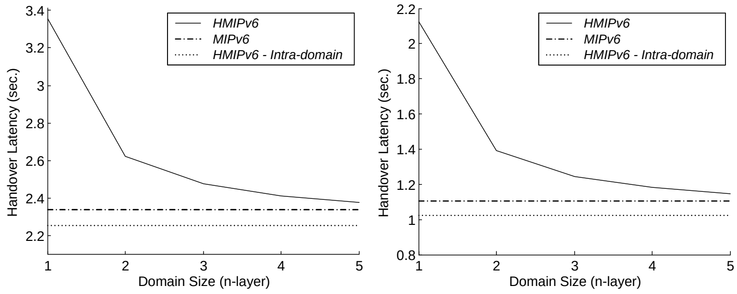
$$T_{HR} = \frac{(E[M] - 1)T_{HR_s} + T_{HR_d}}{E[M]}. \tag{13}$$

4.2 Analytical Results

We demonstrate the performance comparison between MIPv6 and HMIPv6. For our analysis, the following default parameters are used: $a = 3$, $b = 7$, $c = 6$, $d = 3$, $t_\alpha = 3ms$, $t_\beta = 2ms$, $R_m = 1.0s$, $R_M = 3.0s$, $R_L = 3.0s$, $RT = 1.0s$, and $DTimes = 1$. Fig. 4 shows the variation in the handover latencies of MIPv6 and HMIPv6 as the domain size is changed.

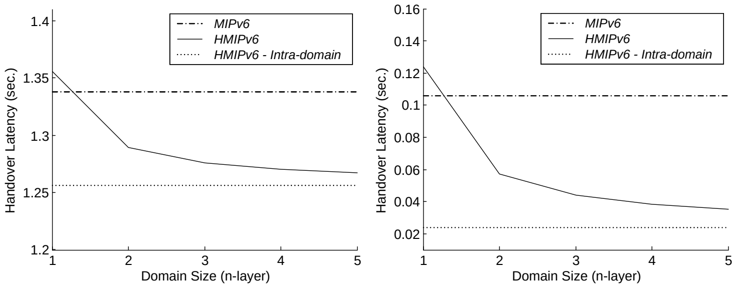
The results indicate that the more subnets a domain contains, the lower HMIPv6 handover latency is. On the other hand, MIPv6 handover latency and the intra-domain HMIPv6 handover latency are fixed regardless of the domain size. In addition, we can find that the intra-domain handover latency of HMIPv6 is always lower than MIPv6 handover latency. That is to say, if MN moves only within a domain, HMIPv6 always outperforms MIPv6 in terms of handover speed. However, it is noted that the intra-domain handover latency of HMIPv6 is not much low compared with MIPv6 handover latency unless both MD and DAD optimizations are supported. The intra-domain HMIPv6 handover reduces MIPv6 handover latency just by about 3% at the case (see Fig. 4 (a)). Only with MD optimization and only with DAD optimization, it reduces MIPv6 handover latency just by about 8% and 6.5%, respectively (see Fig. 4 (b) and (c)).

Another surprising results are seen from Fig. 4 (a) and (b) which report that the average HMIPv6 handover latency is always higher than the average MIPv6 handover latency unless DAD optimization is supported. This is justified



(a) No MD and DAD optimizations

(b) Only MD optimization



(c) Only DAD optimization

(d) Both MD and DAD optimization

Fig. 4. Handover latency comparison

by noting that HMIPv6 requires DAD procedure for RCoA as well as DAD procedure for LCoA when MN moves to a new domain. Although inter-domain handover does not frequently occur, its latency becomes so long due to the DAD procedures for both LCoA and RCoA. On the average, therefore, HMIPv6 handover latency becomes longer than MIPv6 handover latency.

Fig. 4 (c) and (d) show that HMIPv6 can outperform MIPv6 in terms of handover latency if DAD optimization is used for CoA configuration in MIPv6 and HMIPv6. Only with DAD optimization, however, we see from Fig. 4 (c) that HMIPv6 handover latency is not much reduced. Compared with MIPv6 handover latency, there is only about 6% decrease when the domain layer is 3, 4, and 5 (each number of subnets in a domain is respectively 25, 49, and 81). It explains that, although DAD latency is diminished, MD latency is still a dominating factor of the whole handover latency. Of more practical and remarkable interest shown by Fig. 4 (d) is the case that both MD and DAD optimizations are supported. In this case, on the average, HMIPv6 handover latency can theoretically becomes below 50 *ms* when the domain layer is above three. We can also know that, compared with MIPv6 handover latency, there is about 50 $\sim$ 60% decrease. In particular, the intra-domain handover latency of HMIPv6 is very low (reduction by about 79% compared with MIPv6 handover latency). It means HMIPv6 can nicely support a real-time application like VoIP in terms of handover speed, if both MD and DAD optimizations are supported.

5 Conclusions

In this paper, we studied the details of MIPv6 and HMIPv6 handover procedures based on the recent IETF RFCs. While existing literatures have asserted that HMIPv6 generally improves MIPv6 in terms of handover speed, we revealed that the average HMIPv6 handover latency is not always lower than the average MIPv6 handover latency. Furthermore, even the intra-domain handover latency of HMIPv6 is not much small compared with MIPv6 handover latency unless both MD and DAD optimizations are supported. In order to increase the benefit of HMIPv6 and make HMIPv6 appropriate to a real-time application requesting considerably short handover latency, both MD and DAD optimizations are essential requirements.

References

1. Johnson, D.B., Perkins, C.E., Arkko, J.: Mobility Support in IPv6. IETF RFC 3775 (2004)
2. Vaughan-Nichols, S.J.: Mobile IPv6 and the Future of Wireless Internet Access. IEEE Computer **36**(2) (2003) 18–22
3. Thomson, S., Narten, T.: IPv6 Stateless Address Autoconfiguration. IETF RFC 2462 (1998)
4. Narten, T., Nordmark, E., Simpson, W.: Neighbour Discovery for IP version 6. IETF RFC 2461 (1998)
5. Soliman, H., Castelluccia, C., Malki, K.E., Bellier, L.: Hierarchical Mobile IPv6 Mobility Management (hmipv6). IETF RFC 4140 (2005)
6. Castelluccia, C.: HMIPv6: A Hierarchical Mobile IPv6 Proposal. ACM Mobile Computing and Communications Review **4**(1) (2000) 48–59
7. Han, Y.H., Hwang, S.H.: Movement Detection Analysis in Mobile ipv6. IEEE Communication Letters **10**(1) (2006) 59–61
8. Yegin, A.: Link-layer Event Notifications for Detecting Network Attachments. IETF Internet draft, draft-ietf-dna-link-information-01.txt (work in progress) (2005)
9. Choi, J., Daley, G.: Goals of Detecting Network Attachment in IPv6. IETF RFC 4135 (2005)
10. Moore, N.: Optimistic Duplicate Address Detection. IETF Internet draft, draft-ietf-ipv6-optimistic-dad-05.txt (work in progress) (2005)
11. Han, Y.H., Hwang, S.H., Jang, H.: Design and Evaluation of an Address Configuration and Confirmation Scheme for IPv6 Mobility Support. In: Proc. of IEEE Wireless Communications and Networking Conference (WCNC). (2004)
12. Akyildiz, I.F., Lin, Y.B., Lai, W.R., Chen, R.J.: A New Random Walk Model for PCS Networks. IEEE Journal on Selected Areas in Communications **18**(7) (2000) 1254–1260
13. Hwang, S.H., Han, Y.H., Min, S.G.: Performance Analysis of IPv6 Mobility Protocols over IEEE 802.11 Network. IEICE Transaction on Communication **E87-B**(9) (2004) 2613–2625