This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

Performance Comparison between MIPv6 and SEMO6

Md Sazzadur Rahman,
Mohammed Atiquzzaman
School of Computer Science,
University of Oklahoma,
Norman, OK 73019-6151.
{sazzad,atiq}@ou.edu

Wesley Eddy,
William Ivancic
NASA Glenn Research Center,
21000 Brookpark Rd. MS 54-8,
Cleveland, OH 44135.
{weddy,wivancic}@grc.nasa.gov

***Abstract*—Internet mobility protocols are designed to support handover between different wireless networks. Many of them, such as Mobile IPv6, suffer problems such as high handover latency, high packet overhead, high packet loss during handoff, requirement for infrastructure change, etc. To solve these problems, SEamless MObility using SHIM6 (SEMO6), a multihoming based mobility protocol framework for host mobility, has been proposed. The goal of this paper is to experimentally validate that SEMO6 can perform better than MIPv6. We show that SEMO6 can improve the performance of applications in IP-based mobile networks.**

***Index Terms*—Mobility, Handoff, Multihoming, MIPv6, SEMO6**

I. Introduction

With the rapid growth in the number of mobile nodes, mobility management in wireless networks has received significant attention. A mobility management protocol enables mobile nodes to maintain continuity of ongoing connections during a change of the point of attachment. With the availability of various wireless access technologies, such as 802.11, 3G and GPRS, a mobile node is likely to have multiple network interfaces using multiple wireless technologies. The ability of a mobile node to communicate concurrently with different network interfaces is sometimes called multihoming[1]. Multihoming can provide several advantages over single-homing, such as fault tolerance and traffic engineering. However, current mobility management protocols, such as Mobile IPv6 [1], do not provide the benefits offered by multihoming.

Mobile IPv6, a network layer based mobility solution, is the most widely known IPv6 mobility management protocol. But it suffers from a number of drawbacks, such as: (1) high handover latency arising from multiple levels of indirection during handover, and (2) packet loss during handover. Moreover, the high handover latency does not allow Mobile IPv6 to provide the quality of service (QoS) guarantees required by real-time applications. Since Mobile IPv6 uses single interface, it has to break the old connection to make a new connection ('Break-Before-Make') during handoff, resulting in high handover latency.

To resolve the above mentioned problems, we have proposed a multihoming based host mobility management protocol, SEamless MObility using SHIM6 (SEMO6) [2] that can achieve low handover latency along with the added advantages of multihoming.

The research reported in this paper was funded by NASA Grant NNX06AE44G.

[1]Other definitions of multihoming have been used, specifying either multiple providers, multiple prefixes, etc.; the content in this paper addresses only the multiple interface setting.

A number of schemes have been proposed in the literature to incorporate multihoming with mobility protocols such as MIPv6. Wakikawa et al. [3] propose extensions to MIPv6 to incorporate multihoming features using multiple care of address registration (MCoA). Although it describes how a multihomed MIPv6 node can register multiple addresses with its home agent, it does not address how the node can utilize multiple addresses to reduce handover latency. Therefore, in terms of handover latency during handoff period, performance of MIPv6 and MCoA based MIPv6 is similar. In contrast to MCoA, our proposed scheme, SEMO6, yields a low handover latency with millisecond resolution. Moreover, SEMO6 and MCoA based MIPv6 are different in approach. MCoA tries to add multihoming feature in mobility protocol MIPv6 where SEMO6 tries to add mobility feature in multihoming protocol SHIM6.

Bagnulo et al. [4] propose a mobile host multihoming solution for IPv6 that uses both SHIM6 and MIPv6 architectures. Others [5], [6] also propose host mobility support in an IPv6 environment by incorporating MIPv6 and SHIM6. All these proposals try to add multihoming features to MIPv6 by incorporating SHIM6 with MIPv6 for host mobility. On the contrary, our proposed mobility scheme, SEMO6, uses only SHIM6 for multihoming features.

Also, a number of schemes have been proposed in the literature that extend multihoming protocols, such as, Host Identity Protocol (HIP), Stream Control Transmission Protocol (SCTP), etc. to support host mobility. However, except for SHIM6, other multihoming protocols suffer from different problems such as infeasible deployment and lack of application transparency. Therefore, SHIM6 is the most promising solution for IPv6 multihoming [7]. Ronan et al. [8] propose a mechanism to support host mobility through multihoming SHIM6 layer for IP Multimedia Subsystem (IMS) networks, but do not quantify the performance gain using an experimental testbed. Dhraief et al. [9] investigate the applicability of the SHIM6 protocol in a mobile environment and empirically show the handover latency is more than two seconds. On the contrary, SEMO6 uses SHIM6 as its underlying protocol and yields handover latency of 22 msec., which is significantly smaller than two seconds as proposed by Amine et al. [9].

The *objective* of this paper is to compare SEMO6 and MIPv6 in terms of throughput, Round Trip Time (RTT), handoff latency and effect on upper layer protocols - four of the major performance criteria for any mobility management scheme. Since performance of MIPv6 and MCoA based MIPv6 is similar during handover period, we compared the performance of SEMO6 with MIPv6 instead of MCoA based MIPv6. Our *contributions* in this paper are comparing the performance of MIPv6 and SEMO6 using experimental results from MIPv6 and SEMO6 testbeds; in-depth experimental studies are important to ensure that all real-world phenomenon have been taken into consideration when comparing performance.

Based on the experimental results (see Sec. V), we demonstrated that SEMO6 has a negligible handoff latency (22 msec) and can achieve seamless handoff, while MIPv6 suffers long discontinuity in transmission during handoff (5.69 sec).

The rest of the paper is organized as follows. Sec. II briefly discusses MIPv6 architecture. Sec. III describes the architecture of SEMO6. Sec. IV describes the details of the experimental setup of SEMO6 and MIPv6. Comparison of performance between MIPv6 and SEMO6, based on experimental results, are given in Sec. V. Finally, concluding remarks are included in Sec. VI.

II. MIPv6 Architecture

The MIPv6 protocol [1] extends IPv6 to support basic host mobility. In MIPv6, when a Mobile Host (MH) moves from its home network to a visited network, it obtains a Care-of-Address (CoA) from the Access Router (AR) in the visited network. Then it sends a Binding Update (BU) message to the Home Agent (HA) in its home network informing the CoA and receives a Binding Acknowledgement (BA) message from the HA. Afterwards, the HA intercepts any packets destined to the MH and forwards the packet to the MH in its visited network using an IPv6-over-IPv6 tunnel. After receiving an encapsulated packet from the HA, the MH decapsulates the packet and delivers it to the upper layer.

III. SEMO6 Architecture

Our proposed SEMO6 (SEamless MObility using shim6) is a network layer based host mobility scheme that uses SHIM6 [10], an IPv6 based, host-centric multihoming protocol. SHIM6 introduces an intermediate layer located above the IP routing sub-layer, but below the IP endpoint sub-layer. It interprets each IPv6 address as having two possible semantics: either as a *locator* for IP routing to a specific interface in the network topology or as an *identifier* or Upper layer ID (ULID) for identification of a particular stack or node without regard to topological location or specific interface. For any failure of the existing communication, SHIM6 uses new locators to continue the communication without changing the ULIDs and thus the change of locators are transparent to the Upper Layer Protocols.

For failure detection and recovery, SHIM6 uses a defined REAchability Protocol (REAP) [11]. Utilizing multihoming framework of SHIM6, SEMO6 uses the ‘Make-Before-Break’ technique and yields a handover latency of 22 msec, which is significantly lower than Mobile IPv6 (5.69 sec). The handover process of this architecture can be described by the following four steps:

1) STEP 1: Obtain new IP address: The handover process begins when the MH moves into the overlapping radio coverage area of two adjacent subnets. In this region, MH should receive an unsolicited Router Advertisement (RA) from the new Access Router (optionally, the MH can send a Router Solicitation (RS) message to the network to receive an immediate RA) and forms a global IPv6 address typically using the IPv6 stateless auto-configuration.

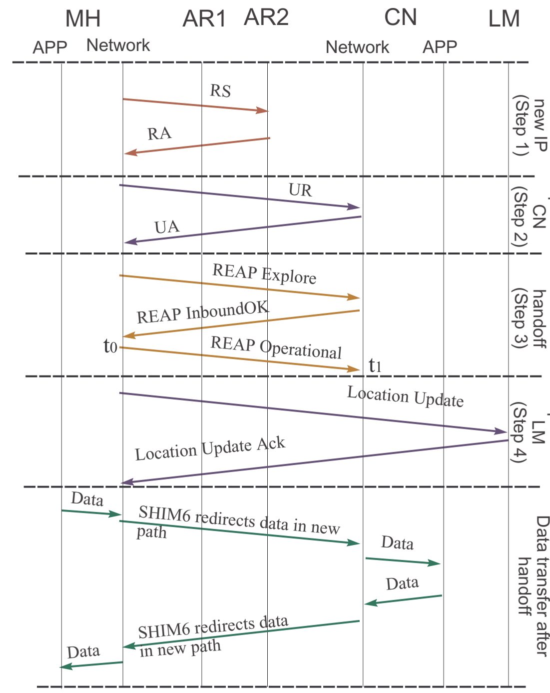

Fig. 1. Message flow of L3 handover of SEMO6.

2) STEP 2: Update Correspondent Node (CN) with the new IP address: Now, MH should notify CN about the availability of the new locator using SHIM6 Update Request (UR) message. After receiving it, CN updates its peer locator list with the new locator of MH and replies back with Update Acknowledge (UA) message.

3) STEP 3: Redirect communication to new IP address: When MH moves further into the coverage area of the new Access Router, the CN starts redirecting data traffic to the new IP address from the old IP address of the MH. In order to do that in SHIM6, MH initiates REAP signaling with message type explore using its new IP address as source. Receiving a REAP explore message from MH, the CN sends a REAP inbound ok message to the MH's old and new IP address. After receiving this message, the MH replies with REAP operational message to CN and starts redirecting packet using the new path. When the CN receives a REAP operational message from the MH, it also redirects data traffic to the new path.

4) STEP 4: Update location manager: Location management can be supported by deploying a location manager (LM) which maintains a database recording the correspondence between the MH's identity and the MH's current primary IP address.

Fig. 1 shows the signaling sequence during SEMO6 handover. Here, t_0 and t_1 are defined as the MH's and CN's handoff completion time respectively. Only the old path is used for communication before t_0, only the new path is used after t_1. Both old and new paths are used between t_1 and t_0. Thus SEMO6 ensures a soft handover and zero necessary packet loss during handover.

IV. Experimental Setup

In this section, we describe testbeds for SEMO6 and MIPv6 that have been used to collect results presented in Sec. V.

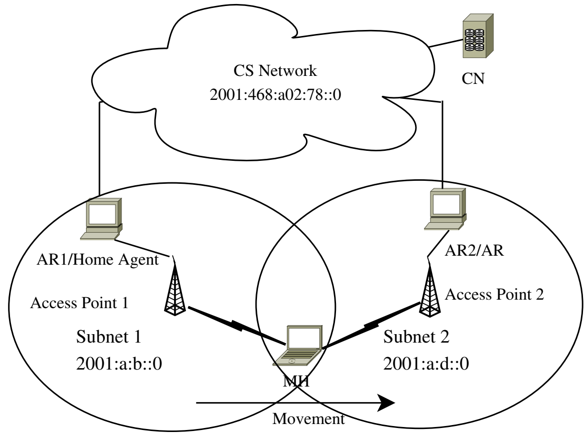

Fig. 2. SEMO6 Testbed architecture.

A. SEMO6 testbed setup

Fig. 2 shows the experimental testbed of SEMO6. It consists of Access Routers (AR1 and AR2), MH and CN. AR1 is the gateway of subnet1 and interconnects the University of Oklahoma CS Network to the Mobile Network (MN), and advertises network prefix 2001:a:b:0::/64 to MHs. AR2 is the gateway of subnet2 and interconnects the University of Oklahoma CS network to the MN and advertises network prefix 2001:a:d:0::/64 to MHs inside this subnet. For the SHIM6 implementation, we used LinShim6, an open source implementation of SHIM6.

B. MIPv6 testbed setup

Fig. 2 also shows the MIPv6 testbed that we built to compare the performance between SEMO6 and MIPv6. It is very similar to the SEMO6 testbed except that it introduces Home Agent instead of Access Router in the Home Network. The Home Network advertises prefix 2001:a:b:0::/64 to MHs. We used the same Access Router in the visited network in both SEMO6 and MIPv6 testbeds which advertises network prefix 2001:a:d:0::/64 to MHs. For MIPv6 daemon, we used NEPL, an open source implementation of MIPv6.

The CN, Home Agent, AR1 and AR2 are connected to the Computer Science network of the University of Oklahoma, an operational network carrying production traffic.

V. Results

In this section, we compare the performance SEMO6 and MIPv6 in terms of throughput, handoff latency, RTT and their effect on the upper layer protocols.

In the experiments, the CN was used as a data source and the MH was used as a data sink. We used the FTP server in the CN and FTP Client in the MH. The MH moved from subnet1 into subnet2.

A. Handover latency

We define handover latency as the time interval between the last application data segment received through the old path and the first data segment received through the new path from the CN to the MH.

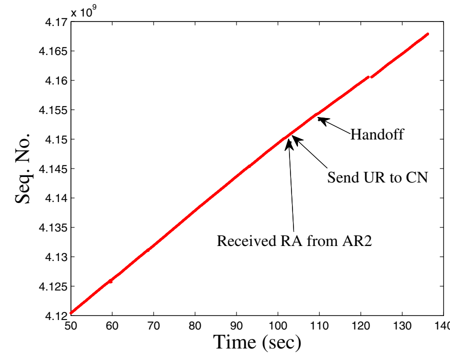

Fig. 3. Data received in MH with the TCP sequence number in SEMO6.

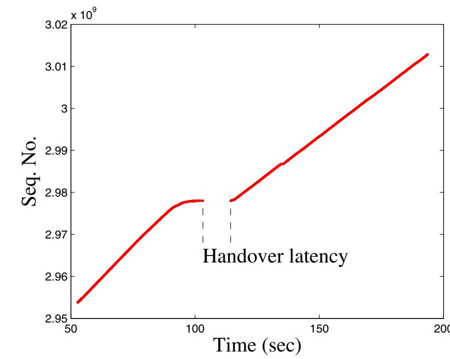

Fig. 4. Data received in MH with the TCP sequence number in MIPv6.

1) Measure of handover latency: From the packet trace (Fig. 3) at the MH, we found that the last data segment received through the old path and the first data segment received through the new path from the CN to the MH are $t = 109.343$ sec. and $t = 109.345$ sec., respectively. Therefore, the handover latency is $t = 2$ msec. $(109.345 - 109.343)$. Moreover, in Fig. 3, we can see that there is no gap between sequence numbers during the SEMO6 handoff. This means that there was no disruption in receiving data by the MH during the handoff period which essentially shows the seamless handoff capability of SEMO6.

On the contrary, we found a significant gap of $t = 11.127$ seconds between two sequence numbers across time during the handoff using MIPv6 as shown in Fig. 4. We measured this gap from the time difference between the first packet through the new path ($t = 114.283$ sec.) and the last packet through the old path ($t = 103.156$ sec.) received by the MIPv6-enabled MH. According to our definition of handover latency, this gap of $t = 11.127$ sec. represents the handover period of MIPv6.

We investigated the reason behind this large handover latency of MIPv6. We found that, although the MH received a Binding Acknowledgement at $t = 108.033$ sec. after sending the Binding Update at $t = 106.933$ sec., it received the first data packet from the CN through the new prefix at $t = 114.283$ sec. The TCP sender in the CN increases the timeout value with exponential backoff due to the initial connection disruption, from $t = 103.156$ sec. to $t = 108.033$ sec. resulting in this large delay of $t = 6.25$ sec. $(114.283 - 108.033$ secs.) for the CN to send data packets to the MH after handover.

2) Hypothesis testing on handover latency: The handoff experiment was performed multiple times to measure the handover

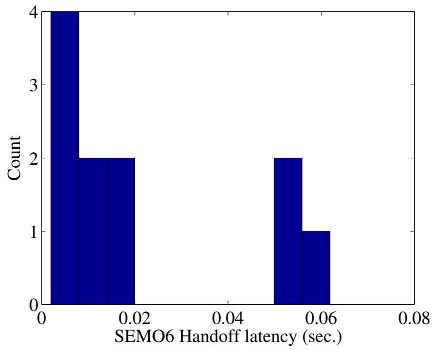

Fig. 5. Handoff latency histogram of SEMO6.

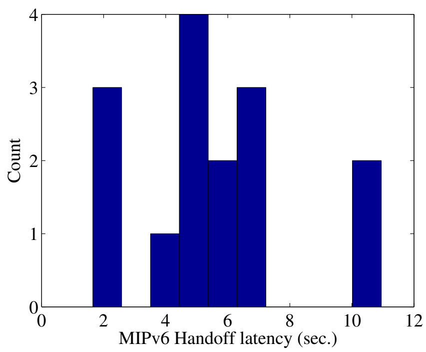

Fig. 6. Handoff latency histogram of MIPv6.

latency of SEMO6 and MIPv6. For measuring MIPv6 handover latency, we consider the time difference between the last data segment through the old path and Binding Acknowledgement received through the new path. Thus, we eliminate the TCP timeout delay on measuring handover latency as discussed in Sec. V-A1.

Figs. 5 and 6 show the frequency distribution of the handoff latency of SEMO6 and MIPv6, respectively. We found that, the mean of MIPv6 handover latency was around 5.69 sec. On the contrary, SEMO6 yielded a mean handover latency of 22 msec.

We ran a two-sample right-tail t-test to verify the null hypothesis. We found that the two-sample t-test rejects the null hypothesis (h=1) with 5% significance level. The 95% confidence interval on the mean of the difference of the handover latency of MIPv6 and SEMO6 will be at least 4.254 s. This hypothesis testing establishes that the mean handoff latency of MIPv6 is greater than that of SEMO6.

B. Throughput

Throughput is measured by the rate at which payload data are received at a node. Fig. 7 shows the average throughput (from seven experiments) received at the MH (the CN sends data to the MH in our experiment) during the handoff between subnet1 and subnet2. The variations in the throughput within a network are due to network congestion arising from cross traffic in the production CS network. In Fig. 7, we see that the throughput after the handover remains the same as before the handover, and never falls to zero during the session between the MH and the CN. On the contrary, we found that the average throughput (from

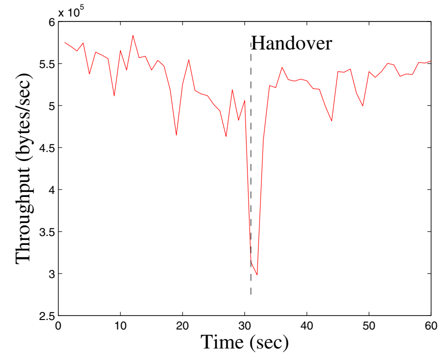

Fig. 7. Average throughput received in the MH in SEMO6.

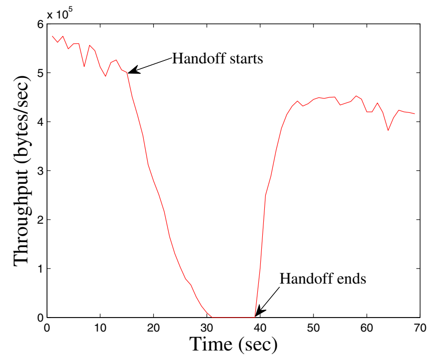

Fig. 8. Average throughput received in the MH in MIPv6.

ten experiments) was zero during the handover using MIPv6 as shown in Fig. 8.

C. Round Trip Time (RTT)

The RTT is the time required for a data packet to travel from the source to the destination have an acknowledgement come back. As the CN sends data to the MH in our testbeds, we measured the RTT at the CN. Fig. 9 shows the RTT for SEMO6. We can see that there is no large spike of RTT for a significant amount of time during the handoff, which implies a seamless handoff. The small spikes are due to the varying levels of cross traffic in the production CS network. On the contrary, the RTT observed using MIPv6 went as high as 16 secs. (from t = 208 sec. to t = 224 sec.) as shown in Fig. 10. During this period, the connection between the MH and the CN is interrupted, resulting in the CN's retransmission timer to fire due to loss

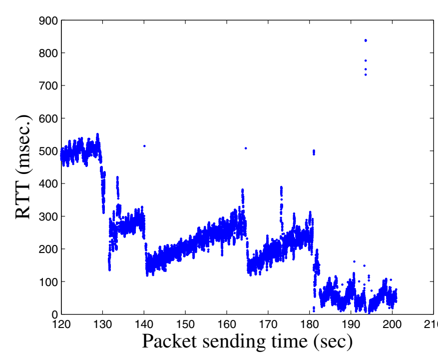

Fig. 9. RTT observed in CN in SEMO6.

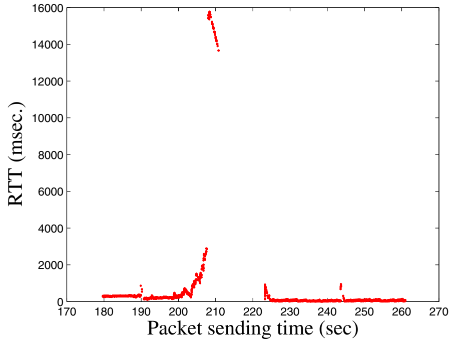

Fig. 10. RTT observed in CN in MIPv6.

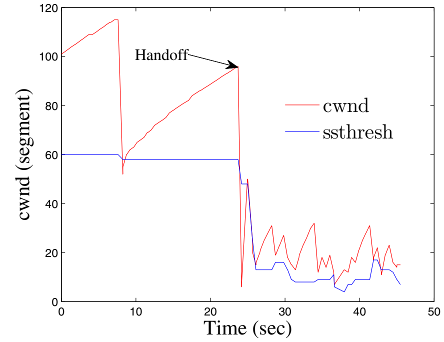

Fig. 11. cwnd and ssthresh in CN of SEMO6.

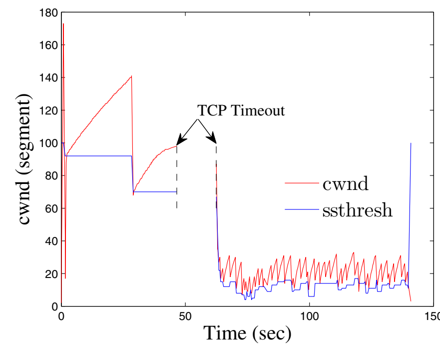

Fig. 12. cwnd and ssthresh in CN of MIPv6.

or delay of data or acknowledgement packets. The RTT was calculated from the difference in time between the the CN sending a packet and receiving the corresponding acknowledgement. For example, CN sent a packet with sequence number of 2978025061 at $t = 208.036$ and retransmitted at $t = 223.294$ sec. and then received an acknowledgement at $t = 223.587$ sec. Therefore, the RTT for this packet is 15.5 sec as shown in Fig. 10.

D. Effect of handover latency on upper layer protocols

Upper layer protocols should not suffer during the handover process. To investigate the effect on the upper layer protocols, like TCP, during the handover of SEMO6 and MIPv6, we decided to focus on the congestion window. The congestion control mechanism is an integral part of most end-to-end transport protocols, such as TCP, and generally limits the rate and amount of data sent into the network. TCP's algorithm maintains several variables including the congestion window (cwnd) and slow-start threshold (ssthresh). We extended the FTP Server code for adding socket APIs for gathering congestion window data for the cwnd and ssthresh variables.

Fig. 11 shows the congestion window during the FTP session in the CN when using SEMO6. Here, the CN is able to quickly recover from the handoff due to the make-before-break nature of the handoff, allowing immediate use of the new path to rebuild the congestion window. On the contrary, due to the slower handoff in MIPv6, the sender's suffers multiple time outs (with exponential backoff), causing additional latency in recovery even after the handoff is completed, as shown in Fig. 12. In this figure, we found that, from $t = 46.528$ sec. to $t = 62.427$ sec., the FTP server in the CN did not make progress and parameters could not be read using the $tcp\_info()$ API. During this period, the MH hands over from subnet1 to the subnet2, the CN experiences timeout for data sending, and the application progress stalls as discussed in Sec. V-A1.

VI. Conclusion

In this paper, we compared the performance of SEMO6, our proposed seamless host mobility scheme, with MIPv6 using Linux based experimental testbeds. Results show the handoff latency of MIPv6 and SEMO6 to be 5.69 sec and 22 msec respectively. Moreover, in contrast to SEMO6 where the throughput remains unchanged after handoff, we found a drop in MIPv6 throughput after handoff due to bidirectional tunneling. We thus conclude that SEMO6 outperforms MIPv6 in terms of throughput and handoff latency.

References

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

[2] M. Rahman and M. Atiquzzama, "SEMO6 - A multihoming-based seamless mobility management framework," in *IEEE MILCOM 2008*, San Diego, CA, Nov 17-19, 2008, pp. 1–7.

[3] R. Wakikawa, V. Devarapalli, T. Ernst, and K. Nagami, "draft-ietf-monami6-multiplecoa-12.txt," Internet Draft, March 2009.

[4] M. Bagnulo, A. Garcia-Martinez, and A. Azcorras, "IPv6 multihoming support in the mobile internet," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 92–98, Oct 2007.

[5] T. Chen and L. Wenyu, "A novel IPv6 communication framework: mobile SHIM6 (M-SHIM6)," in *2nd International Conference on Wireless Communications, Networking and Mobile Computing*, Wuhan, China, Sep 22-24, 2006, pp. 1–3.

[6] A. Kiani, S. Khan, and Y. Wenbing, "A novel mechanism to support session survivability in heterogeneous MIPv6 environment," in *International Conference on Emerging Technologies*, Peshawar, Pakistan, Nov 13-14, 2006, pp. 38–43.

[7] C. Launois and M. Bagnulo, "The paths toward IPv6 multihoming," *IEEE Communications Surveys &Tutorials*, vol. 8, no. 2, pp. 38–51, Second Quarter 2006.

[8] J. Ronan, S. Balasubramaniam, A. Kiani, and W. Yao, "On the use of SHIM6 for mobility support in IMS networks," in *4th International Conference on Testbeds and research infrastructures for the development of networks and communities*, Innsbruck, Austria, March 18-20, 2008, pp. 1–6.

[9] A. Dhraief and N. Montavont, "Toward mobility and multihoming unification the SHIM6 protocol: a case study," in *2008 IEEE Wireless Communications and Networking Conference*, Las Vegas, NV, March 31 - April 3, 2008, pp. 2840–2845.

[10] E. Nordmark and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6," IETF RFC 5533, June 2009.

[11] J. Arkko and I. van Beijnum, "Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming," IETF RFC 5534, June 2009.

978-1-4244-5638-3/10/$26.00 ©2010 IEEE