Performance Comparison between NEMO BSP and SINEMO

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

William Ivancic
NASA Glenn Research Center,
21000 Brookpark Rd. MS 54-8,
Cleveland, OH 44135
Email: `wivancic@grc.nasa.gov`

***Abstract*—IETF has proposed Mobile IPv6-based Network Mobility (NEMO) basic support protocol (BSP) to support network mobility. NEMO BSP inherits all the drawbacks of Mobile IPv6, such as inefficient routing path, single point of failure, high handover latency and high packet overhead. To address these drawbacks, a new network mobility scheme, called Seamless IP-diversity based Network Mobility (SINEMO), has been proposed. The goal of this paper is to validate with experimental data that SINEMO performs better than NEMO BSP. We show that SINEMO can improve the performance of IP based mobile networks.**

***Index Terms*—Network Mobility, Mobility Management, Multihoming, Performance Evaluation.**

I. INTRODUCTION

IETF has proposed NEtwork MObility (NEMO) Basic Support Protocol (BSP) [1] to support the mobility of a network. NEMO BSP is an extension of Mobile IPv6 and allows the hosts in the Mobile Network (MN) to continue ongoing connection while the handoff is taking place. In the NEMO BSP architecture, a Mobile Router (MR) takes care of the hosts within MN and thus, hosts connected to the MR, are not aware of mobility.

NEMO BSP is based on Mobile IPv6, and therefore, it inherits the following drawbacks: (1) all packets are routed through the Home Agent (HA) of the MR, giving rise to inefficient routing path, (2) increased packet overhead due to encapsulation, (3) increased handover latency during handover, and (4) packet loss during handover. Petander et al. [2] proposed a make-before-break handover scheme to improve the handover and routing performance of NEMO BSP. Cho et al. [3] proposed route optimization techniques to improve the routing of NEMO BSP. Although, these papers individually address various drawbacks of NEMO BSP, none of them provides a complete solution to all the drawbacks of NEMO BSP.

To address the aforementioned drawbacks of NEMO BSP, we proposed Seamless IP diversity based NEtwork MObility (SINEMO) [4] to handle mobility of a network. It is an extension of SIGMA (Seamless IP diversity based Generalized Mobility Architecture) [5]. Unlike NEMO BSP, which is a network layer based solution, SINEMO is an end-to-end network mobility solution having the following advantages over NEMO BSP. SINEMO : (1) exploits IP diversity to achieve a seamless handover between adjacent access points, (2) is easier to deploy because no changes are required in the Internet infrastructure, (3) cooperates with internet security protocols, (4) is efficient in utilization of network bandwidth due to the absence of tunneling, (5) utilizes wireless links efficiently (i.e. low signalling), and (6) has low latency and packet loss during handover.

Previous analytical model-based performance comparison between NEMO BSP and end-to-end mobility management scheme, such as SINEMO has demonstrated better performance of SINEMO [4]; performance comparison between the two schemes using experimental testbed is not available. Simplifying assumptions are usually required to render an analytical model tractable. It is thus important to carry out in-depth experimental studies to ensure that all real-world phenomenon have been taken into consideration when comparing performance.

The *objective* of this paper is to compare NEMO BSP and SINEMO in terms of throughput, Round Trip Time (RTT) and handoff latency - the three major performance criteria for any mobility management scheme. Our *contributions* in this paper are illustrating and analyzing the experimental results gathered from NEMO BSP and SINEMO testbeds and comparing the performance between the two schemes.

Based on the experimental results (see Sec. V), we demonstrated that SINEMO has a negligible handoff latency (48 msecs.) and can achieve seamless handoff, while NEMO BSP suffers discontinuity in transmission during handoff (14 secs.). Moreover, we showed that, unlike SINEMO, with the increased path length between Home and Foreign Networks, throughput becomes worse after handoff in NEMO BSP due to bidirectional tunneling.

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

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

978-1-4244-2324-8/08/$25.00

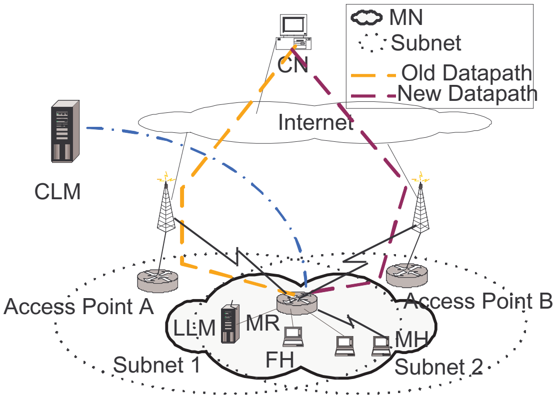

Fig. 1. Architecture of SINEMO.

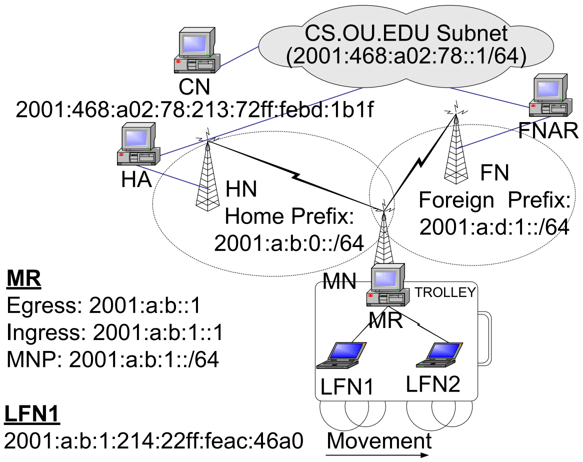

Fig. 2. NEMO experimental testbed.

II. ARCHITECTURE OF NEMO BSP

In NEMO BSP [1], MR ensures connectivity of all hosts when the MR changes its point of attachment to the Internet when moving from a Home Network (HN) to a Foreign Network (FN). MR has a unique IP address and one or more Mobile Netowk Prefixes (MNP) that it advertises to the Mobile Hosts attached to it. MR establishes a bidirectional tunnel with its HA to pass all the traffic between the Mobile Hosts and the correspondent nodes.

When MR changes its point of attachment, it acquires a new care-of-address from the Foreign Network. It sends a Binding Update to its HA which creates a cache entry, binding MR's home address with its care-of-address, and creates a bidirectional tunnel between HA and MR. When a Correspondent Node (CN) sends a packet to a host, the packet is routed to the HA of the corresponding MR. HA looks at its cache entry and forwards the packet to the MR using the bidirectional tunnel. Finally, MR receives the packet, decapsulates it, and forwards it to the host in the MN.

III. ARCHITECTURE OF SINEMO

In this section, we provide a brief description of SINEMO architecture [6]. As shown in Fig. 1, MN consists of a multihomed MR which connects to wireless networks through either Access Point A (manages Subnet 1) or Access Point B (manages Subnet 2). Hosts inside the MN can be Fixed Hosts (FH) or Mobile Hosts (MH).

MR acts as a gateway between the hosts and the Access Points for Internet access. When MN moves into Subnet 1, MR obtains its own public IP address and one or more public IP address prefixes for delegation to its hosts from Access Point A. MR provides each host with a private IP address and reserves a public IP address for the host. The hosts are not aware of their public IP addresses; only the private IP addresses are used for communication by hosts. MR maintains one-to-one mapping of the public and private IP addresses. When MN moves in Subnet 2 and makes handover, MR gets a new public IP address and prefixes from the new Access Point B and injects control packets to CN on behalf of LFN to initiate handover. After handover, only the public addresses are changed in the address mapping at MR, the private IP addresses of the hosts inside MN remain unchanged. MR thus hides mobility from the hosts, although as a transport layer mobility management scheme, CN needs to have support of multihoming protocol such as SCTP in SINEMO.

A Central Location Manager (CLM) maintains the host names of the hosts inside the MN. A Local Location Manager (LLM), collocated with the MR, maintains the public IP addresses of the hosts. When CN wants to send packets to a host, it queries the CLM which forwards the query to the LLM. LLM responds directly to the CN with the public IP address of the host. Packets sent by CN to the host are intercepted by the MR and forwarded to the host after address translation.

IV. EXPERIMENTAL SETUP

In this section, we describe experimental testbeds of NEMO BSP and SINEMO that have been used to collect results presented in Sec. V.

A. NEMO BSP testbed setup

There are several open source implementations available for NEMO BSP, like NEPL [7] and SHISA. [8]. We chose NEPL for testing NEMO BSP in our testbed due to the following reasons:

1) Unlike NEPL which is based on Linux platform, SHISA uses BSD platform. As we are using Linux platform for SINEMO, we used Linux based NEMO BSP for a fair comparison between NEMO BSP and SINEMO.
2) NEPL has been tested thoroughly which includes interoperability tests with SHISA [8].

Fig. 2 shows the experimental testbed of NEMO BSP with single level of nested mobility. It consists of HN, FN, HA, MR, Local Fixed Nodes (LFN1 and LFN2), Foreign Network Access Router (FNAR) and CN. Table I summarizes the hardware and software configuration of devices used in NEMO BSP testbed.

As SINEMO uses SCTP as its underlying transport protocol to achieve multihoming, hence, to make a fair comparison between

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

TABLE I
CONFIGURATION OF DEVICES FOR NEMO BSP TESTBED

No.	Device Type	Software Configuration	Hardware Information
1	HA	Debian 2.6.22 kernel + NEPL	CPU: Intel P4, 1500 MHz Mem: 256 MB
2	CN	Fedora Core 5 + lksctp-tools 1.0.6	CPU: Intel Celeron, 2.8 GHz Mem: 512 MB
3	MR	Debian 2.6.22 kernel + NEPL	CPU: Intel P4, 2.20 GHz Mem: 256 MB
4	LFN	Fedora Core 5 + lksctp-tools 1.0.6	CPU: Intel P4, 1.73 GHz Mem: 1 GB
5	APs	Channel 6 and 11	DLink WBR-1319

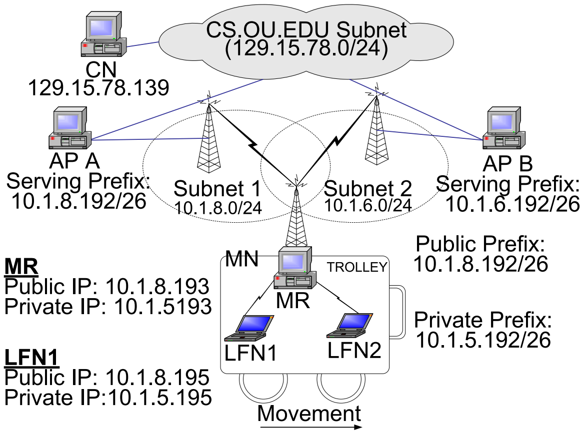

Fig. 3. SINEMO experimental testbed.

NEMO BSP and SINEMO, we run a SCTP based client server program in LFN (client) and CN (server) for data communication in NEMO BSP testbed.

To make the performance comparison more real world oriented, we connected the CN, HA and FNAR to the operational network of University of Oklahoma which carries production traffic.

B. SINEMO testbed setup

Fig. 3 shows the experimental testbed of SINEMO with single level of nested mobility. A trolley with a MR and Local Fixed Nodes (LFN1 and LFN2) works as the MN. Table II summarizes the hardware and software configuration of devices used in SINEMO testbed.

TABLE II
CONFIGURATION OF DEVICES FOR SINEMO TESTBED

No.	Device Type	Software Configuration	Hardware Information
1	CN	Red Hat 2.6.6 + lksctp-tools 1.0.6	CPU: Intel P4, 2.2 GHz Mem: 256 MB
2	MR	Fedora Core 5 + iptables	CPU: Intel Celeron, 2.20 GHz Mem: 256 MB
3	LFN	Fedora Core 5 + lksctp-tools 1.0.6	CPU: Intel P4, 2.20 GHz Mem: 512 MB
4	APs	Channel 6 and 11	DLink WBR-1319

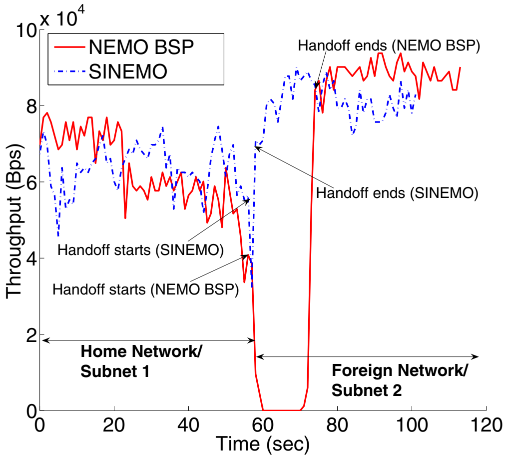

Fig. 4. Throughput of NEMO BSP and SINEMO.

V. RESULTS

In this section, we compare the performance of NEMO BSP and SINEMO in terms of throughput, RTT and handoff latency. From our NEMO BSP and SINEMO testbeds, we captured packet flows at LFN, CN and MR using Wireshark network protocol analyzer, and used the data for measuring throughput, RTT and handoff latency.

A. Throughput

Throughput is measured by the rate at which payload data are received at a node. Fig. 4 shows the throughput at LFN (CN sends data to LFN in our experiment) during NEMO BSP handoff between HN and FN. The variations in throughput within a network are due to network congestion arising from cross traffic in the production CS network. We see that the throughput falls to zero between $t = 58$ to $t = 72$ secs. (14 secs. duration) during handoff when MR is disconnected from CN and hence LFN does not receive data. This disruption will be discussed in detail in Sec. V-C.

Fig. 4 also shows the throughput of SINEMO where, unlike NEMO BSP, the throughput does not fall to zero during handoff between $t = 56$ to $t = 56$ secs. Comparing the throughput of NEMO BSP and SINEMO, it is obvious that SINEMO throughput is not affected by handoff.

Ideally, NEMO BSP should exhibit lower throughput after handoff due to the tunneling as discussed in Sec. V-D. But Fig. 4 does not reflect this event due to small propagation delay and single level of nesting used in our testbed. Although it looks like that NEMO BSP has higher throughput than SINEMO after handoff in Fig. 4, it is quite random due to the varying nature of the wireless medium (i.e. interference) between MR and FNAR.

B. RTT

RTT was calculated from the difference in time between the CN sending a packet and receiving the corresponding acknowledgement. Fig. 5 shows the RTT between CN and LFN, measured

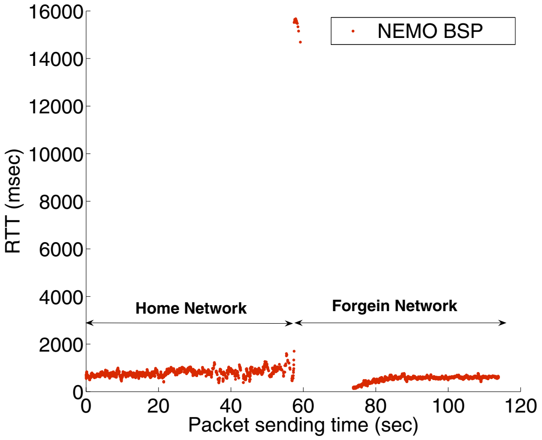

Fig. 5. RTT of NEMO.

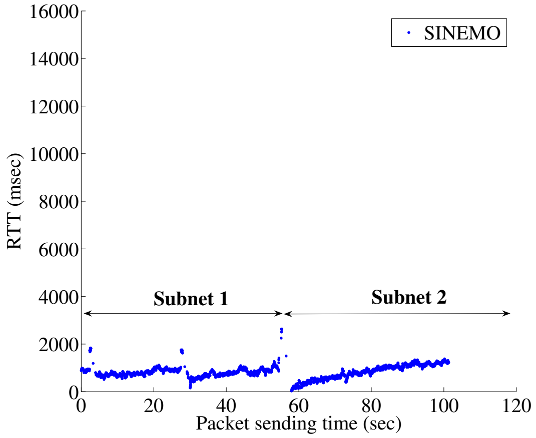

Fig. 6. RTT of SINEMO.

at CN for NEMO BSP. This period $t = 58$ sec. to $t = 72$ sec. represents the handoff when the connection between MR and CN is interrupted. Because of this connection interruption, retransmission timer of CN fires for some packets due to packet loss or delayed acknowledgements resulting in CN retransmitting those packets. For example, CN sent a packet with sequence number of 2999 at $t = 57$ and subsequently retransmitted the packet at $t = 61, 63, 66$ and 73 secs. and received acknowledgement at $t = 73$ sec. Hence, the RTT for this packet is 16 secs.

Fig. 6 shows the RTT for SINEMO where the RTT is fairly constant during handoff ($t = 56$ sec. to $t = 56$ sec.), implying a seamless handoff. The small spikes are due to the varying levels of cross traffic in the production CS network.

C. *Handoff Latency*

We define handoff latency as the time interval between the MR receiving the last packet from HN and the first packet from FN. Fig. 7 shows the packet trace during NEMO BSP and SINEMO handovers. In NEMO BSP, we found that MR sent Binding

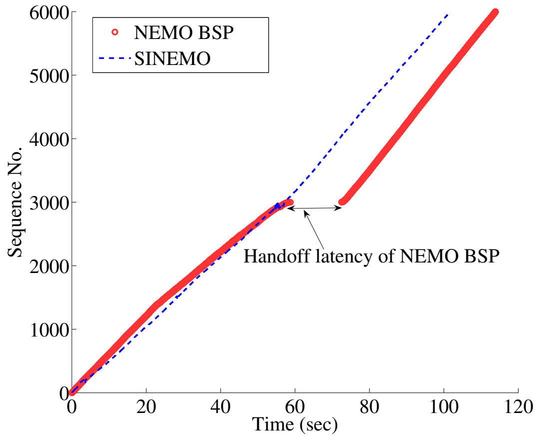

Fig. 7. Handoff latency of NEMO BSP and SINEMO.

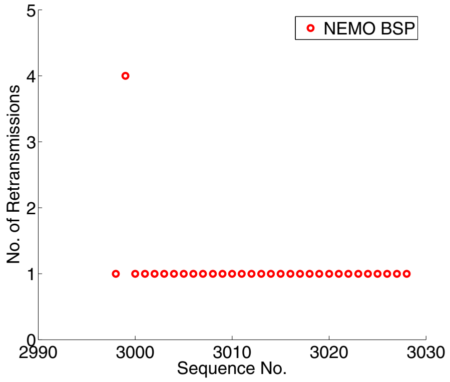

Fig. 8. No. of retransmission in NEMO BSP during handoff.

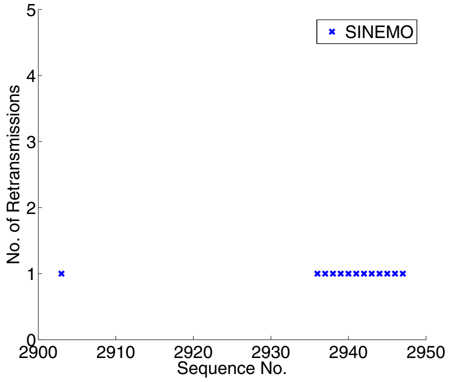

Fig. 9. No. of retransmission in SINEMO during handoff.

Update at $t = 65.367$ sec. and it received Binding Acknowledgement at $t = 66.710$ sec. The last packet MR received from CN before sending Binding Update was at $t = 58.750$ sec., and the first packet MR received from CN after receiving Binding Acknowledgement was at $t = 72.394$ sec. Hence, the *handoff latency for NEMO BSP was 13.645 secs.* Further, we found that CN retransmitted a large number of packets during handoff as shown in Fig. 8 for NEMO BSP. For example, packet no. 2999 was retransmitted four times with the following packets retransmitted only once. This large number of retransmission causes CN to backoff, resulting in poor throughput for a long duration (See Fig. 4).

For SINEMO, as shown in Figs. 4 and 7, we found that the last and first packets from subnets 1 and 2 (equivalent to HN and FN respectively for NEMO BSP) arrived at MR (from CN) at $t = 56.382$ and $t = 56.430$ secs., respectively, resulting in *SINEMO handoff latency of about 48 milliseconds*. Fig. 9 shows the packet trace during SINEMO handover where CN retransmitted packets with sequence numbers 2936 to 2947. This number of retransmitted packets (12) is much lower than that of NEMO BSP (30).

D. *Effect of delay in AR*

To investigate the effect of inefficient routing path on the performance of NEMO BSP after handoff, Fig. 10 shows the measured throughput of NEMO BSP with increasing level of delays (using Netem network emulator) between HN and FN, representing increasing path length between Home network and Foreign network. To find the bottleneck of NEMO BSP bidirectional tunneling, we considered the case where LFN was communicating with CN under the same FNAR as shown in Fig. 11. According to the architecture of NEMO BSP, all communications to and from Mobile Network Nodes after handoff must go through the bidirectional tunnel between the MR and HA. Hence, although CN and MR are now under same FNAR (Fig. 11), every packet traverses twice the path between FNAR and HA due to the bidirectional tunneling when CN communicates with LFN. This results in poor throughput in the LFN (as throughput decreases with the increasing delay for NEMO BSP as shown Fig 10). On the contrary, as the two hosts are under the same FNAR, and there is no concept of HA in SINEMO, throughput of SINEMO does not suffer after handoff due to this added delay.

VI. CONCLUSION

In this paper, we compared the performance of SINEMO, our proposed IP-diversity based seamless network mobility scheme, with NEMO BSP using Linux based experimental testbeds. Results show the handoff latency of NEMO BSP and SINEMO to be 14 secs. and 48 msecs. respectively. Moreover, in contrast to SINEMO where the throughput remains unchanged after handoff, we found a drop in NEMO BSP throughput after handoff due to bidirectional tunneling. We thus conclude that SINEMO outperforms NEMO BSP in terms of throughput and handoff latency.

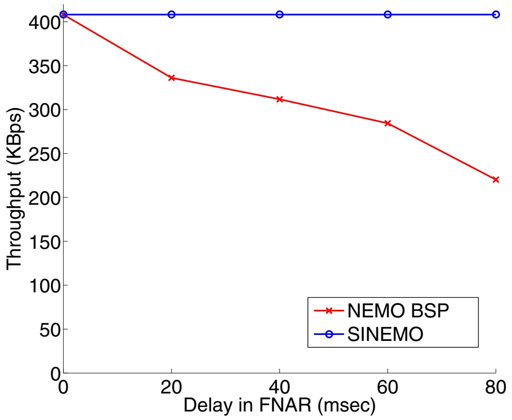

Fig. 10. Throughput of NEMO BSP and SINEMO after handoff with added delay.

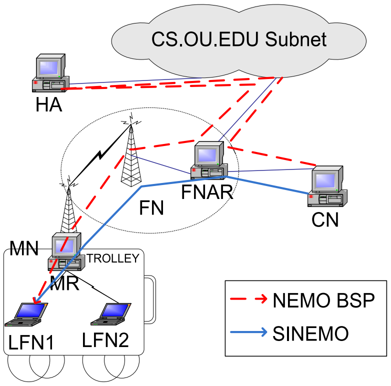

Fig. 11. NEMO BSP vs SINEMO Traveled Path after handoff.

REFERENCES

[1] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, “Network mobility (NEMO) basic support protocol,” RFC 3963, January 2005.

[2] H. Petander, E. Perera, K. Lan, and A. Seneviratne, “Measuring and improving the performance of network mobility management in IPv6 networks,” *IEEE Journal on selected areas in communications*, vol. 24, no. 9, pp. 1671–1681, September 2006.

[3] H. Cho, T. Kwon, and Y. Choi, “Route optimization using tree information option for nested mobile networks,” *IEEE Journal on selected areas in communications*, vol. 24, no. 9, pp. 1717–1724, September 2006.

[4] P. K. Chowdhury, A. S. Reaz, M. Atiquzzaman, and W. Ivancic, “Performance analysis of SINEMO: Seamless IP-diversity based network mobility,” in *IEEE International Conference on Communications, ICC’07*, Glasgow, Scotland, Jun 24-28, 2007, pp. 6032–6037.

[5] S. Fu, L. Ma, M. Atiquzzaman, and Y. Lee, “Architecture and performance of SIGMA: A seamless handover scheme for data networks,” in *IEEE ICC*, Seoul, South Korea, May 16-20, 2005, pp. 3249–3253.

[6] P. K. Chowdhury, M. Atiquzzaman, and W. Ivancic, “SINEMO: An IP-diversity based approach for network mobility in space,” in *Second International Conference on Space Mission Challenges for Information Technology*, Pasadena, CA, July 17-21, 2006.

[7] “Nepl (nemo platform for linux) howto,” www.nautilus6.org/doc/nepl-howto/.

[8] R. Kuntz, “NEMO Basic Support implementation tests at the 6th IPv6 TAHI interoperability test event,” www.nautilus6.org/doc/tc-nemo-tahi-interop-20050207-KuntzR.txt, February 2005.