

SEMO6 - A MULTIHOMING-BASED SEAMLESS MOBILITY MANAGEMENT FRAMEWORK

Md. Sazzadur Rahman, Mohammed Atiquzzaman
Telecommunications and Networks Research Lab
School of Computer Science,
University of Oklahoma, Norman, OK 73019.
{sazzad, atiq}@ou.edu

Abstract—*Mobility protocols are designed to support handover between different wireless networks. Many of them suffer problems such as high handover latency, high packet overhead, high packet loss during handoff, lack of application transparency, etc. To remove these problems, we proposed a network layer based mobility protocol framework, SEamless MObility using shim6 (SEMO6), for host mobility. It is based on using the SHIM6 protocol and provides multihoming, mobility and application transparency. The goal of this paper is to present the system framework, analyze signalling cost, and evaluate the performance of this mobility management protocol.*

Index Terms—**Multihoming, Host Mobility, Handover latency, Performance, SHIM6**

I. INTRODUCTION

With the rapid growth in the number of mobile nodes, mobility management in wireless networks has received significant attention. A mobility management protocols enables mobile nodes to maintain continuity of ongoing connections during change of point of attachment. The recent development 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 called multihoming. Multihoming provides several advantages, such as fault tolerance and traffic engineering. Fault tolerance ensures session survivability across failure event and thus increases the reliability and availability of a node, and traffic engineering provides load-sharing, policy routing, etc. However, current mobility management protocols do not provide such benefits which are offered by multihoming. The focus of this paper is to develop a multihoming-based mobility management protocol.

MIPv6, a network layer based solution, is the most widely known IPv6 mobility management protocol to date. Although Mobile IPv6 provides application transparency,

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 do not allow Mobile IPv6 to provide the quality of service (QoS) guarantees required by real-time applications. As 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 address the aforementioned drawbacks of Mobile IPv6, Seamless IP diversity based Generalized Mobility Architecture (SIGMA) has been proposed to handle host mobility. Analytical model [1] and experimental data [2] have been used to demonstrate that SIGMA performs better than Mobile IPv6 and does not suffer from the drawbacks of Mobile IPv6. SIGMA uses the benefit of multihoming with multiple interfaces and thus can make a new connection before breaking the old connection ("make-before-break") resulting in very low handoff latency. However, SIGMA suffers from the following drawbacks: (1) supports only SCTP transport layer protocol and does not support TCP/UDP, (2) does not support backward compatibility, i.e. TCP/UDP applications must be recompiled using SCTP to work in SIGMA, and (3) each application has to incorporate seamless handoff mechanism for soft handover. Therefore, *our aim* is to design a mobility management protocol that can achieve low handover latency and application transparency along with the added advantages of multihoming. To achieve this aim, *our approach* implements the proposed protocol stack in the network layer to ensure application transparency, and utilizes the "make-before-break" concept using multihoming protocols like SHIM6 [3] to ensure low handover latency and fault tolerance.

There are several approaches in the literature for integrating mobility and multihoming. Amine et al. [4] proposes mobility using SHIM6, a Site Multihoming protocol for IPv6, for single interfaced Mobile Host (MH) and has a handover latency of more than 2 seconds. Using SHIM6 as its underlying protocol, our proposed scheme, SEMO6

The research reported in this paper was funded by NASA Grant NNX06AE44G.
978-1-4244-2677-5/08/\$25.00 2008 IEEE

considers MH as a multiple interfaced node and uses "make-before-break" for handover to obtain a handover latency of 25 msec.

Our contributions in this paper include:

- 1) Design of a system framework for seamless mobility management in IPv6.
- 2) Build a lab prototype for performance evaluation of the SEMO6.
- 3) Develop an analytical model for evaluation of signaling cost of SEMO6.

Our *objective* is to design a multihoming-based mobility management protocol which can achieve seamless handover and application transparency. Our experimental results show that the SEMO6 architecture yields a handover latency of 25 msec. which is significantly lower than Mobile IPv6 (handover latency of 4 sec. [5]). We also found that unlike MIPv6 [5], handoff has no significant effect on upper layer protocols, such as TCP.

The rest of the paper is organized as follows. Sec. II summarizes existing works related to mobility and multihoming. In Sec. III, we describe the general architecture of SHIM6 protocol. Sec. IV describes the system framework of SEMO6 mobility management protocol using SHIM6. Sec. V describes the details of the experimental setup of the SEMO6. Performance analysis of SEMO6 based on experimental results are given in Sec. VI followed by analysis of signaling cost in Sec. VII. Finally, concluding remarks are included in Sec. VIII.

II. RELATED WORK

A number of proposals to use multihoming for host mobility have been proposed in the literature. R. Wakikawa et al. [6] proposed to incorporate multihoming feature in MIPv6 using multiple care of address registration (MCoA). Bagnulo et al. [7] proposed mobile host multihoming solution for IPv6 based on the SHIM6 architecture which relies on the cooperation between the MIPv6 and the SHIM6 protocols. Baumann et al. [8] proposed a framework for macro mobility and multihoming notification protocol for wireless mesh network on MIP and SHIM6. Others [9], [10] also proposed host mobility support in IPv6 environment incorporating MIPv6 and SHIM6. But, all proposals tried to add multihoming features to MIPv6 by incorporating SHIM6 with MIPv6 for host mobility. Besides, Amine et al. [4] investigated applicability of SHIM6 protocol in mobile environment and showed the handover latency which is more than 2 sec. On the contrary, our proposed architecture utilizes the characteristics of seamless mobile environment in wireless overlay networks, and able to make soft handover. Our architecture uses SHIM6 protocol and yields handover latency of 25 msec. which is significantly small compared to that of 2 sec.

III. ARCHITECTURE OF SHIM6

SHIM6 is a multihoming solution in IPv6 proposed by IETF multi6 working group and introduces an intermediate layer located above the IP routing sub-layer, but below the IP endpoint sub-layer. It supposes that each host in the network owns multiple global IPv6 address. In SHIM6, each IPv6 address can be interpreted into two different semantics: locator for IP routing and identifier or ULID (Upper layer ID), for upper layer identification. SHIM6 maintains a mapping between locators and ULIDs in all active connections between two hosts. The operation of SHIM6 in a TCP connection can be described using the following steps:

- 1) A TCP connection can be established between two hosts. There is no SHIM6 context available at this point.
- 2) After some time, hosts exchange SHIM6 context. The IPv6 addresses are same for both locators and ULIDs.
- 3) For any failure on the existing communication, SHIM6 uses new locators to continue the communication. At this point, ULIDs remain same to the upper layer protocols (ULP) but the underlying locators change and SHIM6 manages this mapping between locators and ULIDs. Thus the change of locators are transparent to the ULPs.

For failure detection and recovery, SHIM6 uses REAP (REACHability Protocol) [11]. REAP maintains three possible states for operation such as operational, exploring and inbound ok. If the communication is not experiencing any problem, then REAP is in operational state. On the contrary, if the communication is experiencing problem, REAP switches to exploring state and starts exploring the other possible locators by sending REAP Probe messages. If host receives packets (data, probe or keepalive) from its peer but its peer is experiencing receiving packets, it switches to inbound ok state. The details functionalities of REAP can be found in [11].

IV. ARCHITECTURE OF SEMO6

We followed two principles in designing SEMO6.

- 1) Seeking a single platform for mobility and multihoming so that mobility management protocol can enjoy the benefits of multihoming such as fault tolerance, traffic engineering etc.
- 2) Taking the benefit of multihoming (i.e. multipath connectivity) to minimize handover latency.

This section describes the architecture SEMO6.

A. Handover process

The handover process of this architecture can be described by the following four steps:

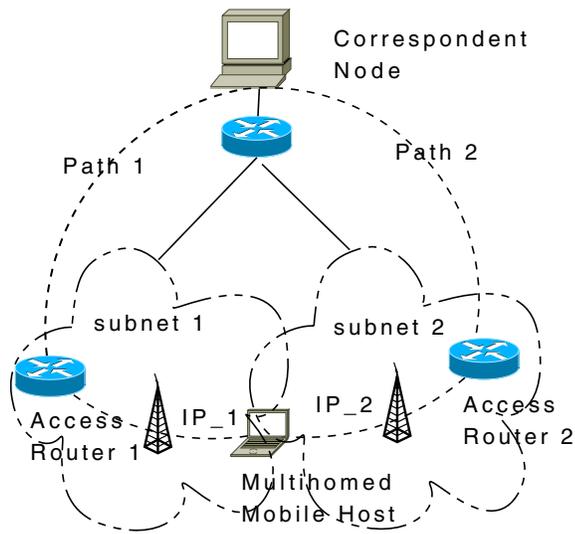


Fig. 1. Mobility scenario for a multihomed host.

1) *STEP 1: Obtain new IP address:* The handover process begins when Mobile Host (MH) moves into the overlapping radio coverage area of two adjacent subnets. In this region, MH should receive an unsolicited Router Advertisement (RA) from Access Router 2 (AR2) (See Fig. 1) as a part of the periodic broadcast of AR2. Optionally, MH can send Router Solicitation (RS) message to the network to receive an immediate RA from AR2. Once MH receives RA from AR2, it forms a global IPv6 address using the prefix in RA and starts duplicate address detection (DAD) by sending Neighbor Solicitation (NS) message. If another node in the network is already using the same address, that node will reply with Neighbor Advertisement (NA) message to the network. If MH does not receive any NA message within a certain time, then obtaining new IP address is completed successfully. This procedure is referred as IPv6 stateless auto-configuration. Another possible way to configure new address is using stateful auto-configuration with DHCPv6.

2) *STEP 2: Update Correspondent Node (CN) the new IP address:* After the MH obtains the new IP address in STEP 1, MH should notify CN about the availability of the new locator (IP_2 in Fig. 1) 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. However, this does not mean that, new locator of MH is verified by CN to be used as destination as the first locator of MH is working fine.

3) *STEP 3: Redirect communication to new IP address:* When MH moves further into the coverage area of AR2, the probability that data can be delivered successfully to the MH increases significantly and CN may take this advantage by redirecting data traffic to the new IP address from the old IP address of MH. In order to do that in SHIM6, MH

initiates REAP signalling with message type explore using its new IP address as source. Receiving REAP explore message from MH, CN sends REAP inbound ok message to MH's old and new IP address. After receiving this message, MH replies with REAP operational message to CN and starts redirecting packet using the new path (Path 2 in Fig. 1). When CN receives REAP operational message from 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 which maintains a database recording the correspondence between MH's identity and MH's current primary IP address as described in SIGMA [1]. The main benefit of this location management is that location management and data traffic forwarding functions are decoupled where they are coupled together in MIP.

B. Timing diagram during handover

Fig. 2 shows the signalling sequences during handover. For configuring a new address, we assume that MH uses IPv6 stateless address auto-configuration. It should be noted that, after sending REAP operational message at t_0 (See Fig. 2, MH starts sending all outgoing packets to CN via new path using new IP address as source. But before time t_1 , all packets from CN will appear through the old path to MH. Thus, between t_1 and t_0 , MH will receive packets in old path where it will send packet using new path. After receiving REAP Operational message at time t_1 , CN starts sending all outgoing packets to MH using new path. To summarize, only old path is used for communication before t_0 , only new path is used after t_1 , and both old and new paths are used between t_1 and t_0 . Therefore, ideally packet loss should not occur during this handoff process in SEMO6.

V. EXPERIMENTAL SETUP

We have prepared the testbed as a proof of concept and conducted experiments with this testbed to collect experimental results presented in Sec. VI. For SHIM6 implementation, we found two open source implementation, LinShim6 [12] and OpenHIP [13]. Between these two implementations, we decided to use LinShim6 in our testbed for the following reasons:

- 1) In terms of number of software release, LinShim6 seemed to be more matured than SHIM6 implementation of OpenHIP. We used LinShim6-0.6.2 release of LinShim6 in our testbed.
- 2) In terms of available feature list, LinShim6 seemed to be more matured than SHIM6 implementation of OpenHIP. For example, CGA (Cryptographically Generated Address) based address validation feature is not available in OpenHIP implementation in Shim6-0.1 release of OpenHIP.

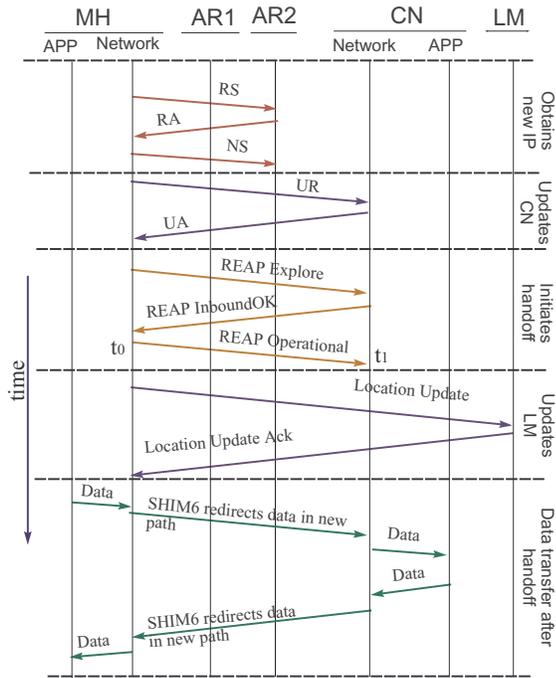


Fig. 2. Message flow of L3 handover of SEMO6.

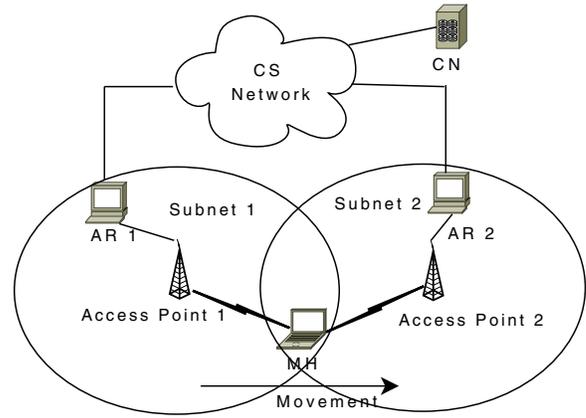


Fig. 3. Testbed architecture.

TABLE I
CONFIGURATION OF DEVICES

No.	Device Type	Software Configuration	Hardware Information
1	MH	Debian 2.6.24.4 kernel + LinShim6 + FTP Client	CPU: Intel P4, 1.73 GHz Mem: 512 MB NIC: 802.11 based two Net-gear MA111
2	AR1	Debian 2.6.22 kernel + radvd-1.0	CPU: Intel P4, 1500 MHz Mem: 256 MB
3	AR2	Fedora Core 2.6.20-1 kernel + radvd-1.0	CPU: Intel P4, 1.73 GHz Mem: 512 MB
4	APs	Channel 6 and Channel 11	DLink WBR-1319
5	CN	Fedora Core 2.6.6 kernel+ LinShim6 + FTP Server	CPU: Intel P4, 1.73 GHz Mem: 512 MB

To implement SEMO6, we have extended the following features of SHIM6 implementation:

- 1) *Implemented Update Request (UR) and Update Acknowledgement (UA) message types in LinShim6* according to [14]. These message types are used to update peer dynamically about the newly configured locator (See Sec. IV-A).
- 2) *Implemented soft handoff mechanism in LinShim6* so that handoff can be initiated when MH moves in the overlapping radio coverage area of two adjacent subnets (See Sec. IV-A).

Fig 3 shows the experimental testbed of SEMO6. It consists of AR1, AR2, MH and CN. AR1 is the gateway of subnet 1 and interconnects the University of Oklahoma CS Network to Mobile Network (MN), and advertises network prefix 2001:a:b:0::/64 to MH's. AR2 is the gateway of subnet 2 and interconnects the University of Oklahoma CS network to MN and advertises network prefix 2001:a:d:0::/64 to MH's inside this subnet. We used radvd-1.0 as IPv6 router advertisement daemon in both AR1 and AR2. The configuration of each device is listed in Table. I.

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

VI. RESULTS

In this section, we present experimental results of SEMO6 and analyze the performance in terms of throughput and handoff latency. From the testbed, we captured packet flows at MH and CN using Wireshark [15] network

protocol analyzer, and used the data for measuring throughput and handoff latency of SEMO6.

In this experiment, CN was used as data source and MH was used as data sink. We run vsftpd FTP server in CN and lftp FTP Client in MH. The MH moved from subnet1 towards subnet2.

A. Handover latency

We define handover latency as the time interval between the last data segment received through the old path and the first data segment received through the new path from CN to MH [2]. In this section, we will first show a packet trace of SHIM6 for mobility and then measure the handover latency from the testbed.

1) *Packet trace of SHIM6:* Fig. 4 shows I/O graph captured in MH. From this graph, it is easy to determine the handoff area of MH as data packets after handoff have SHIM6 header unlike before handoff. From the wireshark capture in MH as shown in Figs. 4 and 5, we found that MH received RA from AR2 at $t = 95.566$ sec. and sent NS at $t = 96.196$ sec. It then configured its IP address

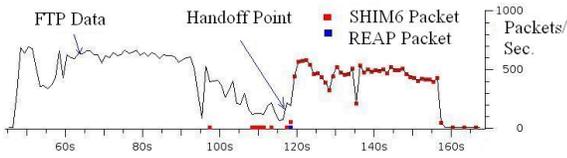


Fig. 4. I/O graph captured in MH.

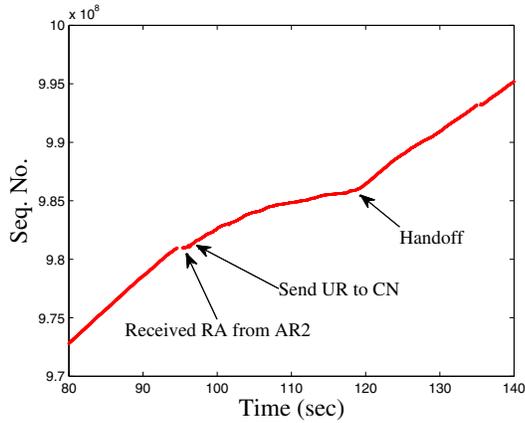


Fig. 5. Sequence number received in MH vs. time.

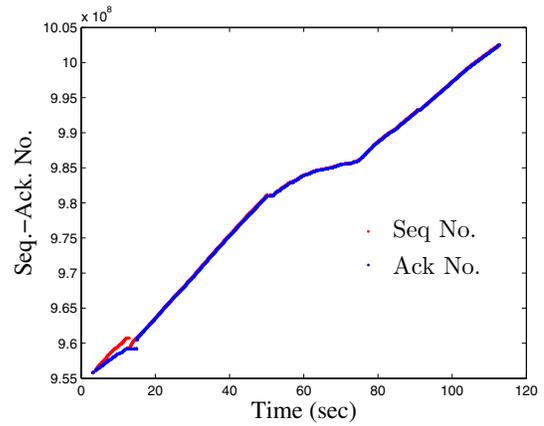


Fig. 6. Segment sequence of SHIM6 during handover captured in CN.

from the network prefix it received from RA and Update the CN using SHIM6 UR message at $t = 97.197$ sec. and received UA message from CN at $t = 97.565$ sec. At $t = 108.748$ sec., MH starts sending REAP Probe message to CN to make a handover. After getting this REAP Probe message, CN replies back with REAP inbound ok message to the old IP address. As MH is not interested to use the old IP address any more, it silently discards this message. After some consecutive attempts, CN sends REAP Probe message to the new IP address of MH at $t = 118.679$ sec. This time MH responds with REAP operational message to CN at $t = 118.682$ sec. and starts sending all outgoing packet via new path with SHIM6 header using the new IP address as source. When CN receives this REAP operational message from MH, it starts sending all outgoing packet via new path with SHIM6 header and using new IP of MH as destination. Therefore, SEMO6 experienced a seamless handover because it could prepare the new path in parallel with data forwarding over the old path. This explains why SEMO6 can achieve a low handover latency and high throughput, as will be shown in Sec. VI-A2 and Sec. VI-B.

2) *Measure of Handover latency:* From the packet trace at MH, we found that the last data segment received through the old path and the first data segment received through the new path from CN to MH are $t = 118.883$ sec. and $t = 118.908$ sec. respectively. Therefore, handover latency (as defined in Sec. VI-A) is $t = 25$ msec. ($118.908 - 118.883$). Moreover, in Fig. 5, we can see that there is no gap between two sequence number across time during handoff. This means that there was no disruption in receiving data by MH

during handoff period which essentially proves seamless handoff. Again, Fig. 6 shows that CN always received immediate ACK from MH after sending data packets and that is why, sequence number and acknowledge number graphs are overlapped. This again proves seamless handoff of SEMO6.

B. Throughput

Throughput is measured by the rate at which payload data are received at a node. Fig. 7 shows the throughput at MH (CN sends data to MH in our experiment) during MH handoff between subnet1 and subnet2. The variations in throughput within a network are due to network congestion arising from cross traffic in the production CS network. In Fig. 7, we see two phenomena: (1) throughput never falls to zero during the session between MH and CN and (2) throughput is reduced during handoff. The first one again proves the seamless handover of MH as there is no disruption during handover of MH. The second one is caused due to the low Radio Frequency (RF) signal of Access Point 1 (AP1) in subnet1 as shown in Fig. 8. According to Fig. 8, RF signal strength of AP1 observed at MH was about 30% during handover from subnet 1 to subnet 2. This low signal strength of AP1 caused packet drops which ultimately affected the throughput. We conclude that the low throughput during handover is not due to the handover scheme itself, but is due to low RF signal strength which can be fixed by appropriate engineering design of the RF coverage areas and placement of the access points.

C. Effect of Handover Latency on Upper Layer Protocols

Upper layer protocols should not suffer during handovers process. To investigate the effect on upper layer protocols, like TCP, during handover of SEMO6, we decided to focus on congestion window, one of the major performance measures of TCP. We extended FTP Server code for adding socket API for gathering congestion window data for `cwnd` and `ssthresh` variables. Fig. 9 shows the congestion window

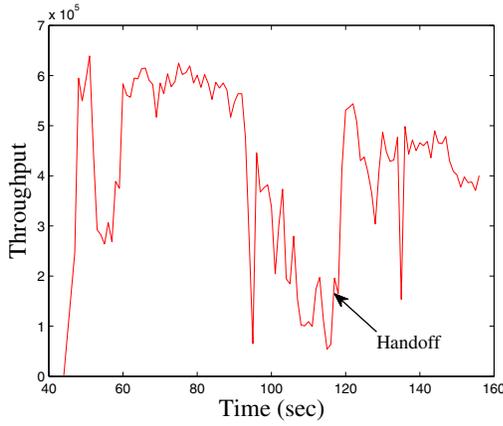


Fig. 7. Throughput received in MH.

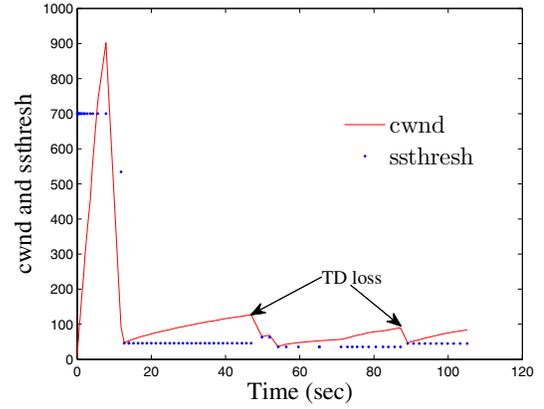


Fig. 9. cwnd and ssthresh in CN.

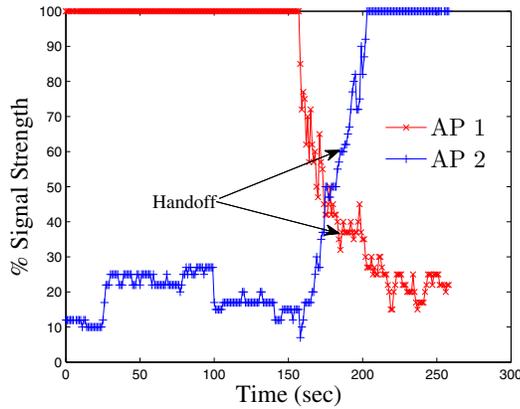


Fig. 8. RF signal strength of AP1 and AP2 observed in MH.

during the FTP session in CN. Here, we see that CN never experienced time out event as time out event will cause cwnd to get 0 which is not present in the graph. Instead CN congestion window experiences some TD losses (due to the duplicate ACK) as shown in Fig. 9 and starts congestion avoidance phase.. This phenomena again can be explained with Fig. 7, where we see the throughput decreases during handoff. As discussed in Sec. VI-B, the low RF signal from AP1 caused some packet losses during handover which is reflected by the TD loss shown in Fig. 9 and throughput decrease in Fig. 7. On the contrary, sender congestion window suffers time out event and congestion window restarts from slow start phase during handoff in MIPv6 mobility protocol as shown in [5].

VII. SIGNALLING COST ANALYSIS

In this section, we develop analytical equations to measure signalling cost of SEMO6 following [1]. We assume that

- 1) No. of hops between MH and CN are same for both downlink and uplink.

- 2) MH and ARs are connected with wireless links and all other nodes are connected with wired links.

A. Notations

The notations to be used in estimating signaling cost are given below:

- N_{mh} total number of MHs
- N_{cn} average number of CNs with which a MH is communicating
- T_r MH residence time in a subnet
- Ψ_{LU} total location update cost
- Ψ_{LUP} total location lookup cost
- Ψ_{BU} total binding update cost
- Ψ_{TOT} total signaling cost per second

B. Binding update cost

Every time MH changes its point of attachment, binding update will be triggered. As we intend to compute the load on network, we will not consider the endpoints, MH and CN, for measuring binding update signaling cost. MH needs to send binding update to each CN it is communicating with. Therefore, estimated binding update cost is:

$$\Psi_{BU} = N_{mh}N_{cn} \frac{BU_{mc}}{T_r} \quad (1)$$

Here, $BU_{mc} = N_{signal}(l_{mc} - 1 + \rho)\delta_B$ and ρ is the average distance between MH and ARs, N_{signal} is the number of signaling required for binding update, l_{mc} is the average distance between MH and CN in hops and δ_B is the per hop binding update transmission cost.

C. Location update and lookup cost

We adopted the same location management of SIGMA [1] and therefore location update and location lookup cost is similar. Location update and lookup cost are estimated as:

$$\Psi_{LU} = N_{mh} \frac{2(l_{ml} - 1 + \rho)\delta_U + \gamma_1}{T_r} \quad (2)$$

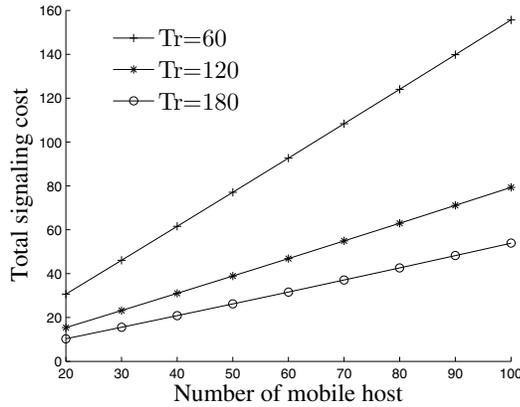


Fig. 10. Signaling cost vs. number of MH.

Here, γ_1 is processing cost at location manager for each location update, l_{ml} is average distance between MH and location manager in hops, and δ_U is per hop location update transmission cost.

$$\Psi_{LUP} = N_{mh}^2 N_{cn} \frac{\Phi \lambda_{sa}}{S} \quad (3)$$

Here, Φ is session-mobility ratio, λ_{sa} is average packet arrival rate and S is number of sessions during an MH-CN communication time.

D. Total signaling cost

Based on the above analysis of binding update cost and location update and lookup cost, the total estimated cost in SEMO6 is:

$$\Psi_{TOT} = \Psi_{BU} + \Psi_{LUP} + \Psi_{LU} \quad (4)$$

E. Numerical results

For computing numerical results of total signaling cost, we assumed the following values (used in [1]) of the variables used in the above equations: $t_r = 0.01$, $l_{mc} = 35$, $\rho = 10$, $\delta_U = \delta_B = 0.2$, $\gamma_1 = 30$, $N_{cn} = 1$, $T_r = 60$, $S = 10$, $\Phi = 0.3$ and $\lambda_{sa} = 0.01$. We used $n_{signal} = 5$ as five signals are required to update binding in CN in SEMO6. Fig. 10 shows the signaling cost with different number of MH varying residence time. From Fig. 10, it is obvious that signaling cost increases with the decrease of residence time and the increase further increases exponentially with the increase of number of MH. Comparison of signaling cost between SEMO6 and other mobility protocols is left as future work.

VIII. CONCLUSION

In this paper, we presented an architecture for managing mobility in wireless networks still achieving low handover latency, application transparency and multihoming simultaneously. Although mobility and multihoming problems

are solved using different protocol architectures, we tried to solve both of the problem using a single solution. We found that SHIM6 can be used for mobility management efficiently with low handover latency (25 msec.) and has very minimal impact on upper layer protocol such as TCP during handover.

REFERENCES

- [1] S. Fu, M. Atiqzaman, L. Ma, and Y. Lee, "Signaling cost and performance of SIGMA: a seamless handover scheme for data networks," *Wireless Communications and Mobile Computing*, vol. 5, no. 7, pp. 825–845, Nov 2005.
- [2] S. Sivagurunathan, J. Jones, M. Atiqzaman, S. Fu, and Y. Lee, "Experimental comparison of handoff performance of SIGMA and Mobile IP," in *2005 Workshop on High Performance Switching and Routing*, Hong Kong, China, May 12-14, 2005, pp. 366–370.
- [3] C. Launois and M. Bagnulo, "The paths toward IPv6 multihoming," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 2, pp. 38–51, Second Quarter 2006.
- [4] 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, USA, 31 March-3 April, 2008, pp. 2840–2845.
- [5] G. Xie, J. Chen, H. Zheng, J. Yang, and Y. Zhang, "Handover latency of MIPv6 implementation in Linux," in *IEEE Global Telecommunications Conference, GLOBECOM 2007*, Washington, DC, USA, Nov 26-30, 2007, pp. 1780–1785.
- [6] R. Wakikawa, V. Devarapalli, T. Ernst, and K. Nagami, "draft-ietf-monami6-multiplecoa-08.txt," Internet Draft, May 2008.
- [7] 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.
- [8] R. Baumann, O. Bondareva, S. Heimlicher, and M. May, "A protocol for macro mobility and multihoming notification in wireless mesh networks," in *21st International Conference on Advanced Information Networking and Applications Workshops/Symposia, AINAW'07*, Niagara Falls, ON, Canada, May 21-23, 2007, pp. 34–37.
- [9] 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.
- [10] 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.
- [11] J. Arkko and I. van Beijnum, "draft-ietf-shim6-failure-detection-13," Internet Draft, June 2008.
- [12] "LinShim6 - implementation of SHIM6," <http://inl.info.ucl.ac.be/linshim6>.
- [13] "OpenHip - implementation of SHIM6," <http://www.openhip.org/>.
- [14] E. Nordmark and M. Bagnulo, "draft-ietf-shim6-proto-10.txt," Internet Draft, February 2008.
- [15] "Wireshark: Network protocol analyzer," www.wireshark.org/.