

Network Friendliness of Mobility Management Protocols

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—In network layer based mobility protocols, such as Mobile IP, the Correspondent Node is not aware of handovers and therefore continues to send data at the same rate as before the handover which may exhibit network unfriendliness, especially when the capacities of the two subnets differ significantly. On the contrary, transport layer based mobility protocols, such as SIGMA, do not show such phenomena. The objective of this paper is to evaluate and compare the network friendliness of network and transport layer based mobility protocols. Experimental results obtained from laboratory prototypes show that network unfriendliness is inevitable in the network layer based mobility protocols during handover when there is a large difference in capacities between the two networks. Transport layer based mobility protocols are, however, more network friendly than network layer based protocols.

Index Terms—Host Mobility, Congestion Control, Handoff, Multihoming, SIGMA, SEMO6

I. INTRODUCTION

A *network-friendly* mobility protocol requires the Correspondent Node (CN) to probe for available bandwidth in the new network after handover [1] especially when there is a significant disparity in network capacities between the two networks involved in the handover. A mobility protocol that does not ensure network friendliness may cause two problems ([2], [3]):

- 1) *Buffer overflow problem* when a handover takes place from a high capacity to a low capacity network. The CN overflows the low capacity network by sending data with at a high data rate, resulting in performance degradation of other users in the new network and exhibiting serious network unfriendliness.
- 2) *Under-utilization problem* when a handover takes place from a low capacity to a high capacity network. The CN continues sending data at a low rate and is unable to adapt and effectively utilize the capacity of the new network.

A number of proposals have been proposed in the literature to address the above mentioned problems. Using experimental testbed, Xie et al. [4] quantify that network layer based mobility protocols cause TCP to suffer from severe performance degradation during handover. To resolve such problems, Goff et al. [5]

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

proposed Freeze-TCP where the Mobile Hosts (MH) advertises a zero window size to force the CN to stop sending data during handover. Gou et al. [2] however, argue that instead of setting zero window size, the MH should advertise the appropriate window size based on BDP (bandwidth-delay product) during handover. Kim et al. [6] further modify Freeze-TCP by starting slow start phase after a handover. All of these solutions suffer from the following problems: a) require MH TCP stack to be modified; b) require precise timing for handover to avoid buffer overflow and under-utilization problems; and c) require explicit handover notification from network layer based mobility protocol to the transport protocol.

Our work *differs* from others in the following ways: a) several authors ([2], [5], [6]) have used network layer-based mobility protocols and proposed to modify TCP stack to overcome such problems. Instead, we used transport layer based mobility to effectively solve these problems; b) unlike previous work where simulation results have been used, we used experimental prototypes and testbeds to compare network friendliness of network and transport layer based mobility protocols. It is important to note that, a fair evaluation requires mobility protocols to be seamless during handovers to avoid the connection restart during handover. Since Mobile IP does not ensure seamless handover [4] (uses ‘Break-Before-Make’) in contrast to SIGMA, we used SEMO6 [7] (uses ‘Make-Before-Break’), a network layer based seamless mobility protocol in our experiments.

The *objective* of this paper is to evaluate and compare the network friendliness of network and transport layer based mobility management protocols. We used SEMO6 and SIGMA as representative network and transport layer based mobility protocols, respectively, and used experimental laboratory prototypes for the evaluation and comparison.

Our contributions in this paper include:

- 1) Evaluation and comparison of network friendliness of network and transport layer based mobility protocols.
- 2) Building laboratory testbeds, designing experiments to carry out the comparison and evaluation of network friendliness.

Our results (Sec. IV) show that network unfriendliness is inevitable in the network layer based mobility protocols during handover when there is a large difference in capacities between the two networks. Transport layer based mobility protocols are, however more network friendly than network layer based protocols.

The rest of the paper is organized as follows. Sec. II describes the architectures of SIGMA and SEMO6 as representative transport and network layer based mobility protocols, respectively. In Sec. III, we describe the detail of the experimental setup of SIGMA and SEMO6. Experimental evaluation of network friendliness of transport and network layer based mobility protocols are given in Sec. IV. Finally, concluding remarks are included in Sec. V.

II. ARCHITECTURE OF MOBILITY PROTOCOLS

In this section, we review the architecture of two mobility protocols, SIGMA [1] and SEMO6 [7] as representative transport and network layer based schemes, respectively and define network friendliness.

A. Architecture of SIGMA

SIGMA (Seamless IP diversity-based Generalized Mobility Architecture) is a transport layer based host mobility scheme that utilizes multihoming and IP diversity to achieve seamless handoff between networks. The main idea of SIGMA is trying to keep the old data path alive until the new data path is ready to take over the data transfer by exploiting the IP diversity at MH, thus achieve a low latency, low loss handover between adjacent subnets. To achieve multihoming, SIGMA uses SCTP as its underlying protocol.

B. Architecture of SEMO6

SEMO6 (SEamless MObility using shim6) is a network layer based host mobility scheme that uses SHIM6 [8], a 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 assumes that each host in the network owns multiple global IPv6 address and each IPv6 address can be interpreted in two different semantics: locator for IP routing and identifier or ULID (Upper layer ID) for upper layer identification. For failure detection and recovery, SHIM6 uses REAP (REAchability Protocol) [9]. Utilizing multihoming framework of SHIM6, SEMO6 uses the ‘Make-Before-Break’ technique and yields a handover latency of 25 msec. [7], which is significantly lower than Mobile IPv6 (handover latency of 4 sec. [4]). Fig. 1 shows the signaling sequence during SEMO6 handover. As shown in this figure, 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 . Thus SEMO6 ensures a soft handover and zero packet loss during handover.

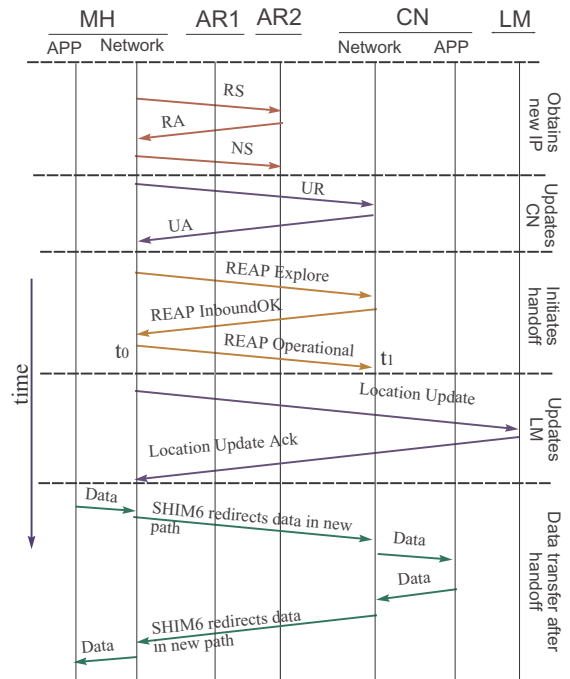


Fig. 1. Message flow of L3 handover of SEMO6.

C. Network friendliness of mobility protocols

In mobile environment, network friendliness means that after MH triggers handover from the old domain to a new domain, CN should not send excessive data flow to MH which may overflow the new domain’s buffer and cause concurrent users suffering from performance degradation. Instead, CN should probe the new domains network capacity and adjust its congestion window accordingly.

In case of SEMO6, CN’s transport protocol is not aware of the handover of MH and thus unable to probe the new domain’s network capacity after handover resulting in excessive data flow to MH if handover occurred from a high to low capacity network. Although we use SEMO6 to demonstrate this phenomena in this paper, the behavior will be similar for all other network layer based mobility schemes. We preferred SEMO6 to other network layer mobility schemes such as Mobile IP since it ensures seamless handover which is important to demonstrate network friendliness.

On the contrary, this phenomena is not observed in transport layer mobility such as SIGMA. SIGMA uses SCTP as its underlying protocol and SCTP maintains different congestion control parameter sets for different path for data transmission. After handover of MH, CN’s SCTP uses a new congestion window to probe the new domains link capacity and adjusts cwnd accordingly. Thus it ensures that CN will not send excessive data flow to MH in case of handover from fast to slow link. Sec. IV demonstrates this phenomena further.

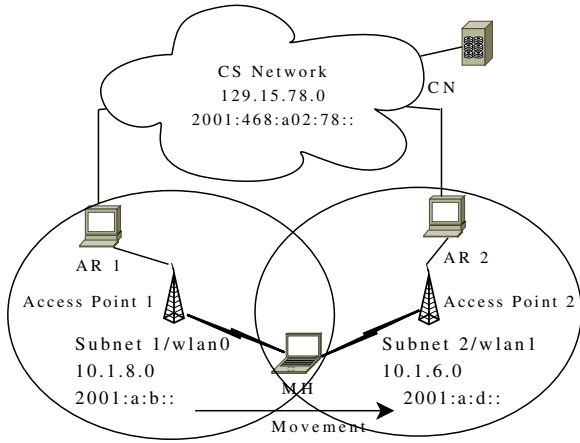


Fig. 2. Testbed architecture.

TABLE I
CONFIGURATION OF DEVICES

No.	Device Type	Software Configuration	Hardware Information
1	MH	Debian 2.6.24.4 kernel + LinShim6	CPU: Intel P4, 1.73 GHz Mem: 512 MB NIC: 802.11 based two Netgear MA111
2	AR1	Debian 2.6.22 kernel + radvd-1.0	CPU: Intel P4, 1500 MHz Mem: 256 MB
3	AR2	Fedora Core 6, 2.6.18 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 SEMO6	Debian 2.6.24.4 kernel+ LinShim6	CPU: Intel P4, 2.2 GHz Mem: 256 MB
6	CN SIGMA	FreeBSD 7.1 kernel	CPU: Intel P D, 3.0 GHz Mem: 1 GB

III. EXPERIMENTAL SETUP

We have prepared SIGMA and SEMO6 testbeds to carry out experiments and collect experimental results presented in Sec. IV. Fig. 2 shows the testbed scenario of SIGMA and SEMO6 in our experiments. Table I summarizes the configuration of different software and hardware used in the testbeds. We setup low capacity network by setting traffic control ('tc' command in Linux) in the egress interfaces of the Access Routers (AR1 and AR2) in Fig. 2. To demonstrate network friendliness, we measured queue length in the egress interface of the low capacity access router using customized 'tc' tool. We collected SCTP congestion window (cwnd and ssthresh) data from FreeBSD7.1 kernel in CN enabling SCTP_LOCAL_TRACE_BUF option. For collecting TCP RENO congestion window data in CN, we used socket API's such as *getsockopt()* and *tcp_info()*. Moreover, we captured packet flows at MH and CN using Wireshark, a network protocol analyzer tool.

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

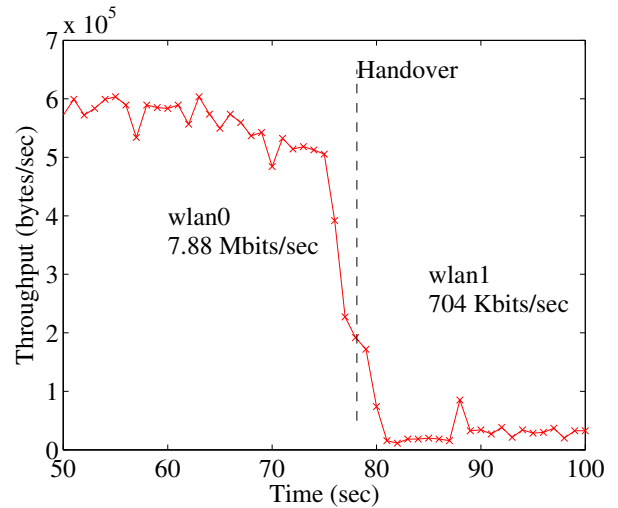


Fig. 3. SEMO6 handover from high to low capacity network in MH.

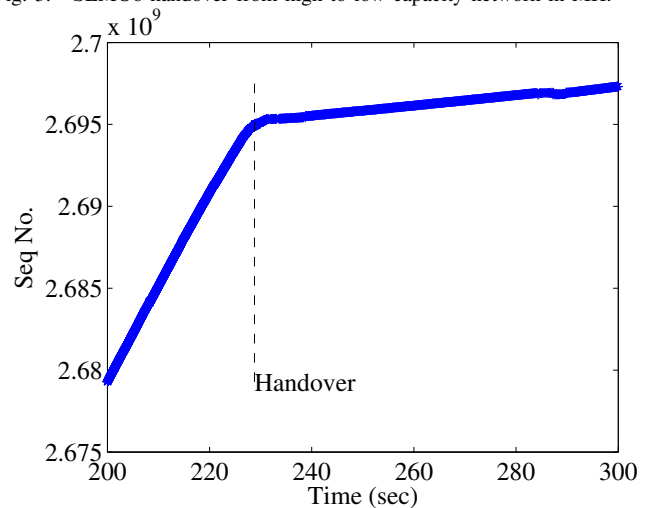


Fig. 4. Data transmission in CN during SEMO6 handover from high to low capacity network.

IV. RESULTS

In this section, we present experimental results collected from testbeds (see Sec. III). Here, we define handover at the time when MH receives first data packet from CN via new domain. In our experiments, CN was used as data source (sends packet with size 1400 bytes) and MH was used as data sink. The MH moved from subnet1 (wlan0) towards subnet2 (wlan1) as shown in Fig. 2.

A. Effect of Buffer overflow problem

In this section, we examine the impact of buffer overflow problem during handover for both network and transport layer based mobility protocols SEMO6 and SIGMA respectively.

1) *Network layer based mobility protocol*: The responsiveness of SEMO6 to a decreased bandwidth environment after handover from wlan0 (7.88 Mbits/sec) to wlan1 (704 Kbits/sec) in terms of throughput (captured in MH) and data transmission (captured in

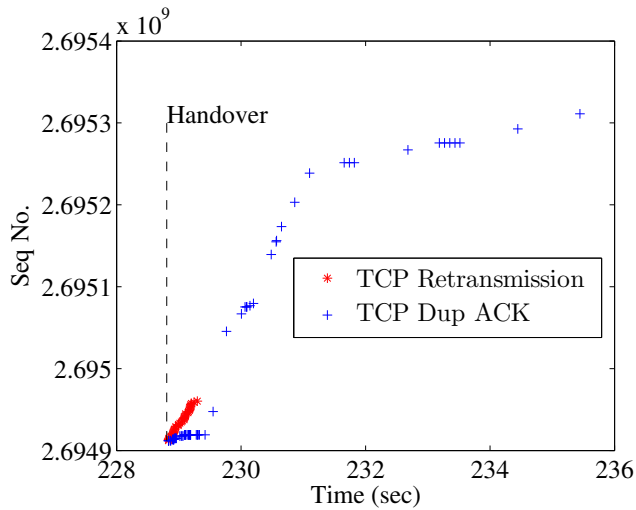


Fig. 5. Packet retransmission by CN after SEMO6 handover.

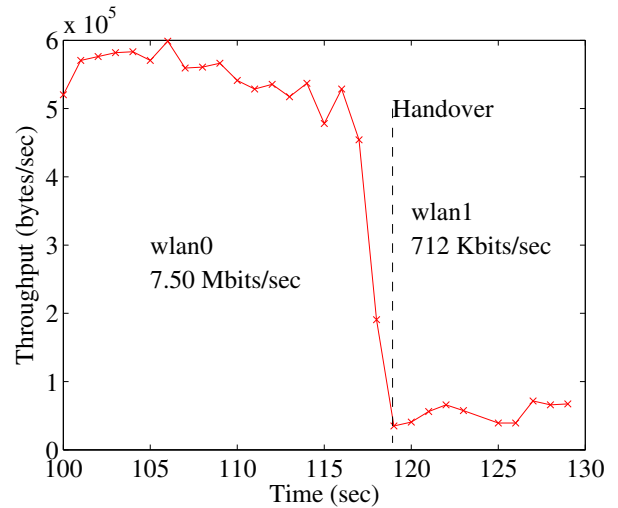


Fig. 8. SIGMA handover from high to low capacity network in MH.

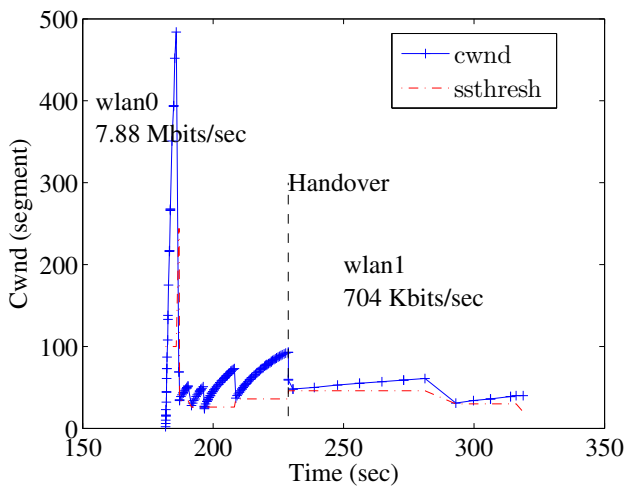


Fig. 6. TCP RENO congestion window in CN during SEMO6 handover from high to low capacity network.

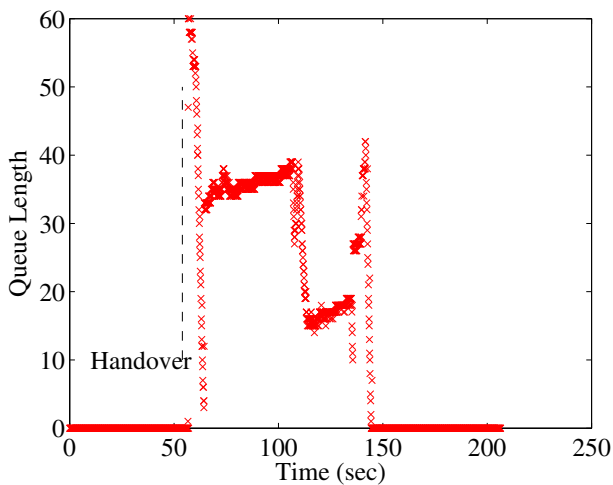


Fig. 7. Queue length in low capacity network's access router (AR2) during SEMO6 handover.

CN) are shown in Fig. 3 and Fig. 4 respectively. Although Fig. 3 and 4 give the impression that TCP adjusts its data sending rate smoothly after handover from fast to slow link, the true effect of the handover is depicted in Fig. 5. In Fig. 5, we found that, CN retransmits a large number of data packet (111 packets) followed by a handover in $t = 228.8$ sec. These retransmissions are triggered by the duplicate ACKs sent from the MH as shown in Fig. 5, possibly due to packet dropping, reordering or duplication in the low capacity network. Based on these duplicate ACKs, TCP sender (CN) invokes fast retransmit algorithm to repair the loss and performs retransmission. Moreover, the fast retransmit algorithm reduces ssthresh and cwnd as shown in Fig. 6. From Fig. 6, we can see that cwnd is in congestion avoidance phase during handover and trying to send data to MH according to the current cwnd value which is not appropriate for the low capacity network and thus results in a sudden increase of the queue length in the access router of the low capacity network (AR2 in our experiment) as shown in Fig. 7. The queue length was increased right after the handover at $t = 54$ sec. and it is likely to cause performance degradation of other network applications in terms of packet loss, increased round trip time, etc. Therefore, it is considered as a serious network unfriendliness caused by the network layer based mobility protocols such as SEMO6.

2) *Transport layer based mobility protocol*: SIGMA network friendliness can be evaluated from Fig. 8 when MH moves from wlan0 (7.50 Mbits/sec) to wlan1 (712 Kbits/sec) in terms of throughput captured in MH. Similar to SEMO6, MH's throughput becomes small right after handover in SIGMA. But opposite to SEMO6, handover causes neither large amount of data retransmission (shown in Fig. 9) in CN nor a sudden large increase of queue length in access router (shown in Fig. 10) in SIGMA. This is because, CN's congestion window adapts new link capacity after handover by probing in the slow start phase (shown in Fig. 11) and starts sending small amount of data appropriate for the low capacity domain. Here, 'bho' and 'aho' stand for

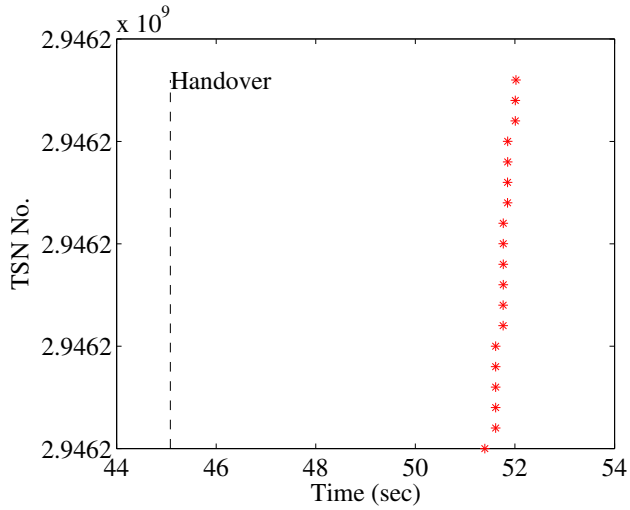


Fig. 9. Packet retransmission by CN during SIGMA handover.

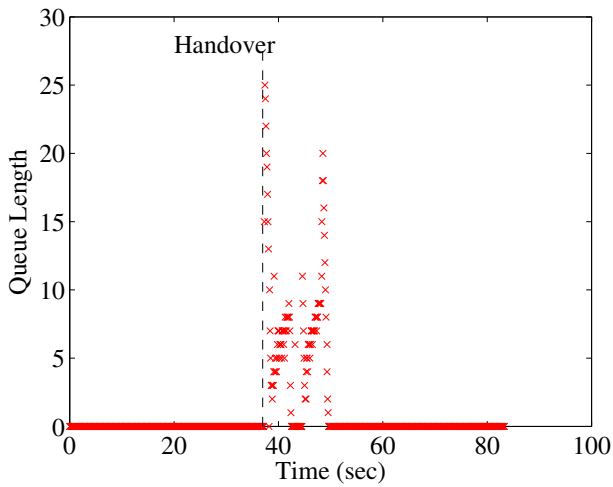


Fig. 10. Queue length in low capacity network's access router (AR2) during SIGMA handover.

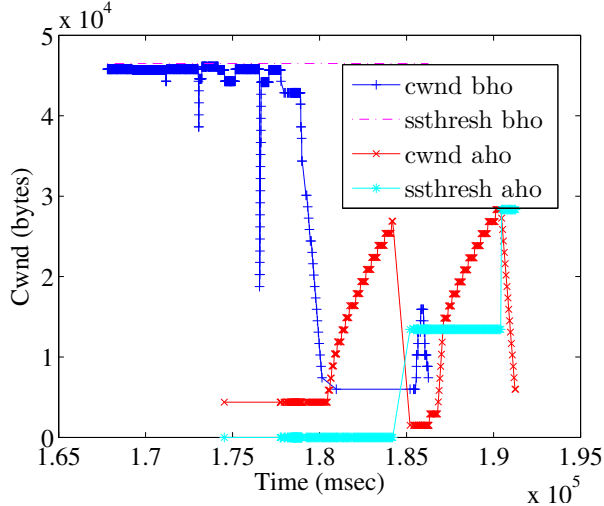


Fig. 11. SCTP congestion window in CN during SIGMA handover from high to low capacity network.

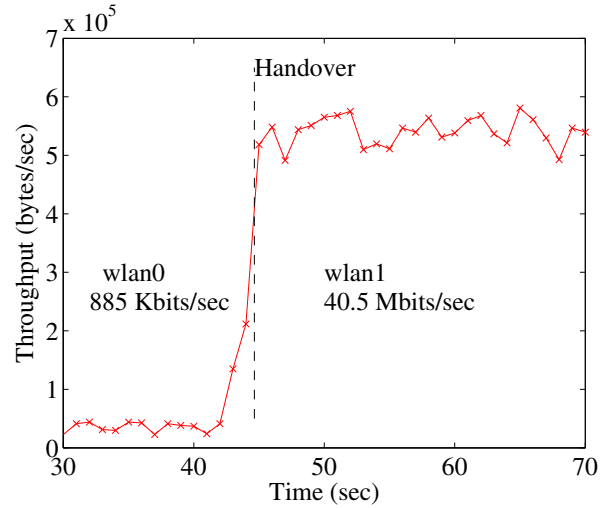


Fig. 12. SEMO6 handover from low to high capacity network in MH.

'before handover' and 'after handover' respectively. In Fig. 9, we found that CN retransmitted small number of data packets (19) at $t = 51$ sec. after six seconds of handover which was triggered in $t = 45$ sec. Due to this significant gap between handover and retransmissions, it is not guaranteed that the retransmission was caused by the handover in contrast to the case of SEMO6. Moreover, as shown in Fig. 10, increase of queue length in access router is smaller compared to the case of SEMO6 and thus should not cause other concurrent network applications suffering from any performance degradation due to this handover. Therefore, transport layer based mobility protocols such as SIGMA exhibit network friendliness in contrast to network layer based mobility protocols such as SEMO6.

B. Effect of Under-utilization problem

In this section, we consider the effect of under-utilization problem during handover for both network and transport layer based mobility protocols SEMO6 and SIGMA respectively.

1) *Network layer based mobility protocol:* In SEMO6 testbed, MH was moving from a low capacity network, wlan0 (885 Kbits/sec) to a high capacity network, wlan1 (40.4 Mbits/sec). According to the under-utilization problem, MH should get higher throughput in the fast link network handing over from the slow link network after some threshold time from handoff due to the fact that CN congestion window is in congestion avoidance phase (as shown in Fig. 13) and thus unable to probe the new domains capacity. But in our experiment, we found that throughput of MH was increased right after handover at $t = 44.639$ sec. from slow link to fast link as shown in Fig. 12. Thus, we could not quantify the under-utilization problem in our experiment. One possible reason is that the low Round Trip Time (RTT) in our experiment made CN possible to find the new domains capacity from the ACKs and helped CN to increase the sending rate utilizing the new domains capacity as shown in Fig. 12.

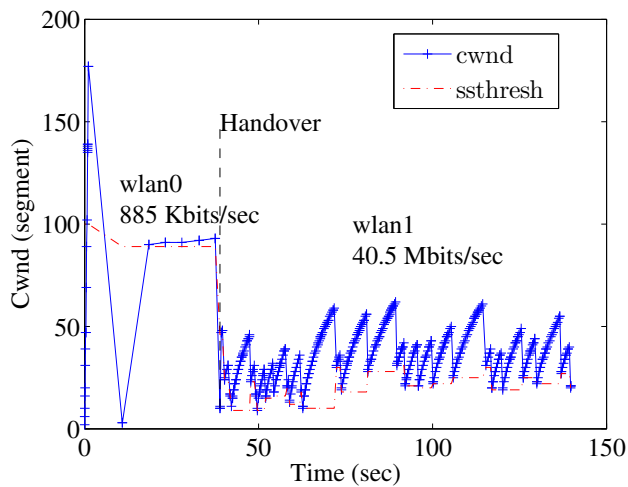


Fig. 13. TCP RENO congestion window in CN during SEMO6 handover from low to high capacity network.

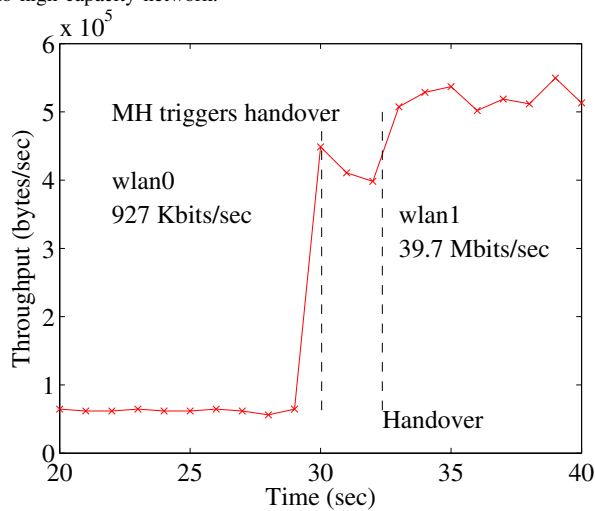


Fig. 14. SIGMA handover from low to high capacity network in MH.

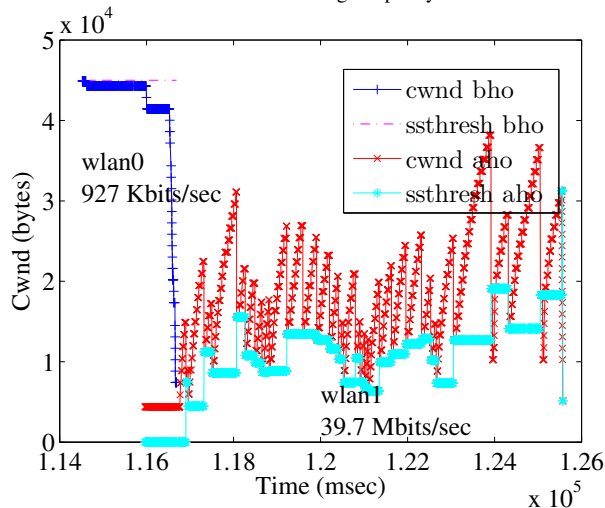


Fig. 15. SCTP congestion window in CN during SIGMA handover from low to high capacity network.

2) *Transport layer based mobility protocol*: Fig. 14 shows the throughput of MH as it moves from a slow link, wlan0 (927 Kbits/sec) to a fast link, wlan1 (39.7 Mbits/sec) in SIGMA testbed. We found that MH triggers handoff from wlan0 to wlan1 in $t = 29.989$ sec. and starts sending acknowledgement via the new domain. Getting these fast acknowledgements, CN increases sending rate and thus MH receives higher throughput at $t = 30$ sec. as shown in Fig. 14, although MH receives first packet from CN via new domain in $t = 32.367$ sec. This utilization was possible due to CN's congestion control mechanism based on SCTP protocol. After handover, CN starts keeping a new congestion window for the new subnet apart from the old one, starts slow start phase and increases the cwnd exponentially as shown in Fig. 15. Due to this reason, CN could utilize new domain's capacity quickly.

V. CONCLUSION

In this paper, we compare and evaluate the network friendliness of network and transport layer based mobility protocols using SEMO6 and SIGMA, respectively, as examples. We focused on scenarios involving networks with significant differences in bandwidths. Our results show that network layer based mobility exhibits network unfriendliness by overflowing network buffers during handover from high to low bandwidth networks. On the contrary, transport layer based mobility protocols exhibit higher network friendliness, and do not suffer from buffer overflow or underutilization problems.

REFERENCES

- [1] S. Fu, M. Atiquzzaman, 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] Y. Gou, D. Pearce, and P. Mitchell, "A Receiver-based Vertical Handover Mechanism for TCP Congestion Control," *IEEE Transactions on Wireless Communications*, vol. 5, no. 10, pp. 2824–2833, Oct 2006.
- [3] A. Gurtov and J. Korhonen, "Effect of Vertical Handovers on Performance of TCP-Friendly Rate Control," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 73–87, July 2004.
- [4] G. Xie, J. Chen, H. Zheng, J. Yang, and Y. Zhang, "Handover latency of MIPv6 implementation in Linux," in *IEEE Globecom 2007*, Washington, DC, Nov 26-30, 2007, pp. 1780–1785.
- [5] T. Goff, J. Moronski, D. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments," in *IEEE INFOCOM*, Tel Aviv, Israel, March 26-30, 2000, pp. 1537–1545.
- [6] S. Kim and J. Copeland, "TCP for seamless vertical handoff in hybrid mobile data networks," in *IEEE Globecom 2003*, San Francisco, CA, Nov 1 - 5, 2003, pp. 661–665.
- [7] S. 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.
- [8] E. Nordmark and M. Bagnulo, "draft-ietf-shim6-proto-12.txt," Internet Draft, February 2009.
- [9] J. Arkko and I. van Beijnum, "draft-ietf-shim6-failure-detection-13," Internet Draft, June 2008.