

Evaluating Cellular SCTP over One-Hop Wireless Networks

Ilknur Aydin Chien-Chung Shen
Computer and Information Sciences
University of Delaware
Newark, DE 19716, U.S.A.
{aydin,cshen}@cis.udel.edu

Abstract—Support for Internet host mobility is increasingly important with the proliferation of wireless and mobile communication devices. Stream Control Transmission Protocol (SCTP) is a plausible choice in the Internet with its unique features like multi-homing and Dynamic Address Reconfiguration extension. We have proposed Cellular SCTP (cSCTP), which is an SCTP-based transport layer mobility solution that aims better handoff performance. In this paper, our focus is the handoff management functionality of cSCTP. We propose an approach to implementing duplicated data transfers during the handoffs to improve the performance. We also present preliminary simulation results using the QualNet network simulator.

I. INTRODUCTION

The need for “host mobility support” in the Internet is increasing with the proliferation of wireless and mobile communication devices. Network layer solutions (such as Mobile IP) are common to support host mobility in the Internet. Network layer solutions however have their own drawbacks, such as the need of support from the network (Home/Foreign Agents), triangular routing, long handoff latency, etc. Moreover, network layer solutions hide the mobility from the transport protocols, where the flow and congestion control tasks are performed, which hinders performance optimization. Transport layer solutions on the other hand allow the transport protocol to be aware of mobility and hence adjust changes in the path characteristics when required. Therefore, transport layer could be a better place to tackle the host mobility problem in the Internet [1].

SCTP [2] is a transport layer protocol originally designed to transport telephony signaling messages over IP networks, but found useful as a general purpose, reliable transport protocol in the Internet. One of the most prominent features of SCTP is multi-homing where an association can be established between a set of local and remote IP addresses. There is also Dynamic Address Reconfiguration (DAR) extension of SCTP [3] to allow changes in the set of local and remote IP addresses of an association dynamically by introducing two new chunk types named ASCONF and ASCONF-ACK chunks. SCTP with DAR extension is referred as Mobile SCTP (or mSCTP for short) [4].

Furthermore, it is becoming more possible for today’s mobile hosts to have access to different types of communication links with different bandwidth, error, cost, and coverage

characteristics (like WLAN, GPRS, Bluetooth) and to have multiple interfaces to benefit from these multiple links simultaneously [5]. Therefore, SCTP with multihoming and DAR extensions becomes a natural candidate for transport layer host mobility solutions.

A transport layer mobility solution needs to have the following functionalities. *Movement Detection*: A Mobile Node (MN) needs to detect the availability of different types of access networks and obtain addresses in these networks for communication. *Handoff Management*: MN should move from one access network to another with minimum service disruption and handoff latency. *Location Management*: Correspondent Node (CN) needs to locate MN for initiating a communication between them.

We have proposed Cellular SCTP (or cSCTP for short) [6], [7], which is an SCTP-based transport layer mobility solution that aims better handoff performance than mSCTP.

The Movement Detection functionality in cSCTP is provided by a neighbor discovery protocol between the Access Routers (ARs) in the network and the Host-Agent module of a MN (Fig. 1). After obtaining a new IP address in the new access network, Host-Agent in layer 3 of the MN informs the cSCTP module of MN to signal a new handoff. Then, the cSCTP module in MN sends a *modified* ASCONF chunk to the cSCTP module of the CN to add the new IP address into the association and signal the start of the new handoff. After the MN totally establishes itself in the new access network and is not reachable by the old IP address anymore, the MN uses the modified ASCONF chunk this time to remove the old IP address from the association and signal the end of handoff to the CN. Note that when the MN has addresses in more than one access network, both MN and CN assumes that MN experiences a handoff. Once both MN and CN know the state of the MN in terms of the handoff process, MN and CN can better behave (such as applying different congestion control, transmission, and retransmission policies, and taking advantage of multiple paths to the MN) to increase the performance during/after the handoff. As for the Location Management, cSCTP suggests utilizing the mobility management capability of SIP [8]. (Readers can refer to [6] for more detail on cSCTP and SIP).

The main focus of this paper is the handoff management

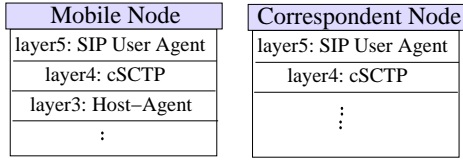


Fig. 1. Protocol stack and functional components of a MN and CN in cSCTP architecture

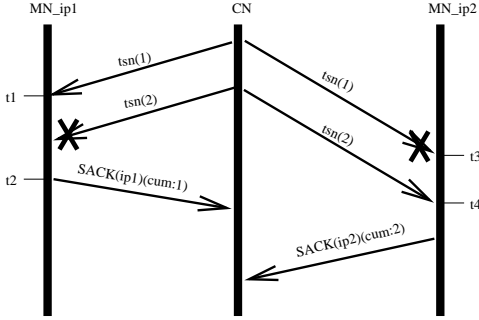


Fig. 2. SACK ambiguity problem for the duplicated data transfer to multiple paths

functionality of cSCTP and the discussion on the modifications and the types of behaviour in the both ends of the association (i.e., MN and CN) to increase the performance. To facilitate the simulation study, we have implemented the cSCTP architecture (except the Location Management functionality) in QualNet network simulator [9].

The paper is organized as follows. Section II describes the cSCTP architecture and implementation in detail. Section III presents initial simulation results, and Section IV concludes the paper with future work.

II. CELLULAR SCTP (cSCTP) ARCHITECTURE

We have implemented the cSCTP architecture across the network and transport layers in the QualNet network simulator [9] using the QualNet SCTP Module [10] implemented by the DEGAS networking group at the University of Delaware.

A. Movement Detection

For the movement detection functionality, a neighbor discovery protocol is implemented in layer 3 at the ARs in the network and the Host-Agent component of the MNs. ARs broadcast periodical beacon messages. Whenever the Host-Agent module of the MN hears a beacon message from a new AR, the Host-Agent communicates with the AR to obtain an IP address in the new access network and informs the cSCTP module of the MN. This triggers transmission of a *modified* ASCONF(add-ip) chunk to the CN for adding the new IP address into the association and signaling the start of a handoff.

In addition, the Host-Agent module stores a “*strength*” value associated with each local IP address of MN to tell how good a particular address is to use. Every time the Host-Agent

receives a beacon message, strength value of the corresponding IP address is updated based on the signal strength of the beacon message¹. The Host-Agent also informs the cSCTP module, as the strength associated with an IP address changes. In turn, the cSCTP module transmits an ASCONF(set-primary) chunk to the CN to make sure that the CN always uses the best path to the MN. Finally, if the Host-Agent can not hear the beacon message from an AR for some threshold value amount of time, the Host-Agent assumes that the connectivity to the access network is lost and informs the cSCTP module. This triggers transmission of a ASCONF(delete-ip) chunk from the cSCTP module of the MN to the CN, to delete the corresponding IP address from the association, and signal the end of the handoff.

The finite state machine diagrams of cSCTP modules at the MN and CN to keep track of handoffs experienced by MN are depicted in Figure 3. The finite state machine diagrams assume that the handoff is between two access networks. But the diagrams can be generalized for handoffs between more number of access networks.

B. Handoff Management

For the handoff management functionality, we have modified the standard SCTP protocol to implement the cSCTP modules at both CN and MN ends.

The idea is that once both of the CN and the MN are synchronized about the handoff state of the MN, cSCTP at both ends can better behave during and/or after a handoff to improve the performance perceived by the MN.

For the following discussion, we assume that the direction of the data transfer is from CN (i.e., the sender) to MN (i.e., the receiver) to simplify the explanations.

Previously [6], we have proposed the idea that during the handoffs, the sender considers the both IP addresses of the MN as primary destinations and duplicates and transmits the data chunks to both primaries. First of all, this will reduce the chance of data chunks missed by the MN and hence result in less timeouts and cwnd reductions at the sender. Another positive effect would be that the cwnd of the new data path is built (by using the duplicated data) *right after* the new path is added into the association.

We aim to improve and implement the idea of duplicating the data transfers during the handoffs. However, we noticed that the main challenge in implementing the idea is the SACK information and processing. Following is the description of the challenge with a sample scenario as in Figure 2.

Assume that the following modifications are already done for creating a cSCTP association with duplicated data transfer: (i) during the handoff periods, the sender duplicates the data chunks to both IP addresses and for each data chunk both destinations are recorded as last destination addresses the data chunk is sent to, (ii) the data receiver has a way to tell to which destination address a data chunk is sent to (the sender could use a bit in the data chunk to tell/differentiate the destination

¹Note that we could also use some other additional measures like bandwidth, etc. in calculating the strength.

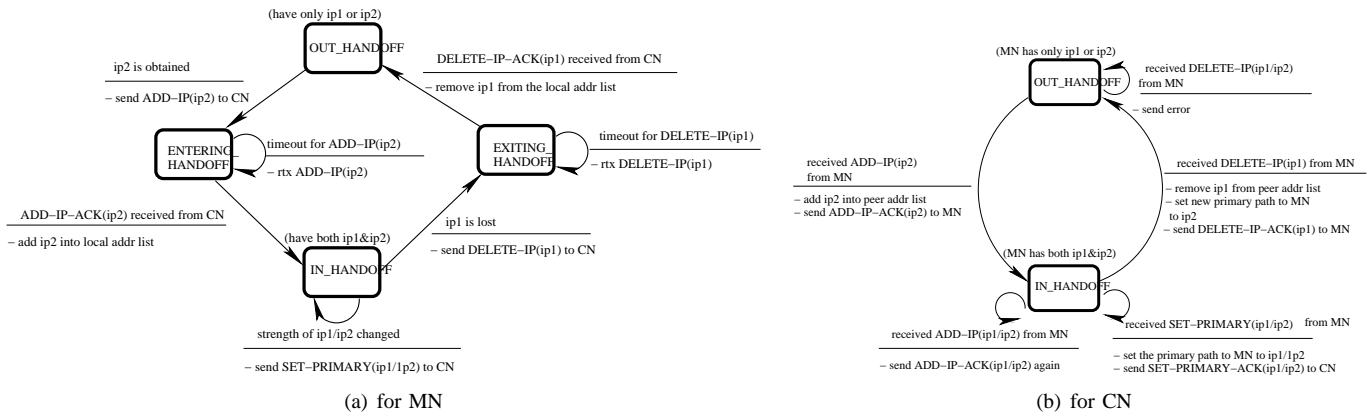


Fig. 3. Finite State Machine diagrams at MN and CN to keep track of handoffs

address of each data chunk), and (iii) the SACK chunks back from the receiver include information about for which path the SACK chunk is sent for.

In Figure 2, MN is in handoff mode and CN sends $tsn(1)$ and $tsn(2)$ onto both paths $ip1$ and $ip2$. At time $t1$, MN receives $tsn(1)$ from path $ip1$ and $tsn(2)$ to path $ip1$ is lost. At time $t2$, MN sends a SACK chunk back for path $ip1$ with cumulative ack of 1. In the meantime, $tsn(1)$ to path $ip2$ is lost but MN receives $tsn(2)$ from path $ip2$ (at time $t3$). At time $t4$, MN sends a SACK chunk for path $ip2$ with cumulative ack of 2. The problem with the information in the SACK sent at time $t4$ for path $ip2$ is that it actually partially acks both paths $ip1$ and $ip2$ (since data chunk with $tsn(1)$ is received via path $ip1$ and data chunk with $tsn(2)$ is received via path $ip2$). CN should not use the total of cumulative ack of 2 to credit the congestion window ($cwnd$) of path $ip2$. Therefore, there will be an ambiguity about what each SACK information actually acknowledges.

This example shows that it is not enough to make the changes (i-iii) above to correctly and properly build the $cwnd$ of the destination paths in the case of duplicated data transfers without ambiguity. Instead, we propose the following approach to solving the ambiguity problem for the duplicated data transfers.

In a multi-homed transport layer protocol like SCTP, data sender keeps a separate set of *congestion related parameters per destination address*. Each set includes parameters like rtt , rto , $flightsize$, etc. (section 7.1 of [2]). In the same way, receiver-side can keep a separate set of parameters per local address. Let's call this set *ack related parameters per local address*. For instance, a set of ack related parameters can include the following parameters per local address: ($lastRcvdTsn$, $rcvdTsnBlocks$) where, $lastRcvdTsn$ is the tsn of the last data chunk received in-order from the particular local address and $rcvdTsnBlocks$ shows which out-of-order tsn 's are received relative to the $lastRcvdTsn$ from the local address. The receiver can then use these parameters to create *cumulative Tsn Ack* and *Gap Ack Blocks* (Section 3.3.4 of [2]) per SACK chunk for a particular destination address.

For the same sample scenario, let's assume that the (shared)

receive buffer of the MN is initially empty. At time $t1$, MN receives $tsn(1)$, saves it in the receive buffer for delivery to the upper layer protocol and updates the set of ack related parameters of $ip1$ as ($lastTsnRcvd$: 1, $rcvdTsnBlocks$: empty). Then, at time $t2$, MN sends the SACK for path $ip1$ to the CN using ack related parameter set of path $ip1$. At time $t3$, MN receives $tsn(2)$, saves in in the receive buffer, and updates the ack related parameter set of $ip2$ as ($lastTsnRcvd$: 0, $rcvdTsnBlocks$: 2). At time $t4$, the SACK is sent to the CN using these parameters for path $ip2$. Therefore, the sender can tell what information each SACK chunk acknowledges without ambiguity.

The gist of our approach is that *the receiver should provide better and more precise acknowledgments per destination path to the sender*. We believe that this type of approach would help a multi-homed transport layer protocol who wishes to take advantage of (duplicated and/or new data) transfers to all of the destination addresses *simultaneously*.

We are currently in the process of improving and implementing the idea. However, in the following section, we present some initial simulation results using our cSCTP architecture implementation in QualNet for the case of using a single primary during handoffs.

III. PERFORMANCE EVALUATION

In our simulation scenario, a MN moves in a straight line from one access network to another (like driving in a road) while downloading information from the CN (Figure 4).

In our simulation set up, ARs and the MN use IEEE802.11 with data rate of 2Mbps and communication range of 300 meters for wireless communication. The length of the intersection (handoff) area between ARs is 100 meters. Each AR sends beacon message in every 1 second. If MN does not hear any beacon message from an AR up to 3 seconds, MN assumes that the connection to the access network is lost and informs the cSCTP layer to delete the corresponding ip address from the association.

The links between the CN (and ARs) and the gateway router have bandwidth of 100Mbps and one-way propagation delay of 10ms. The core link has bandwidth of 10Mbps and one-way

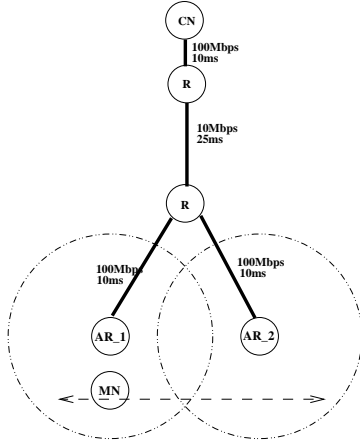
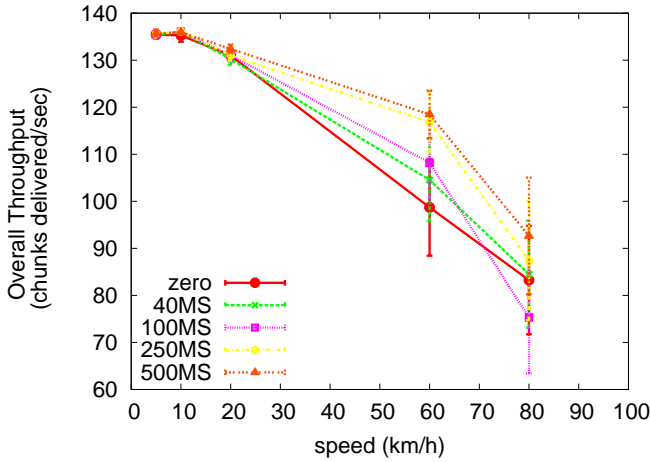
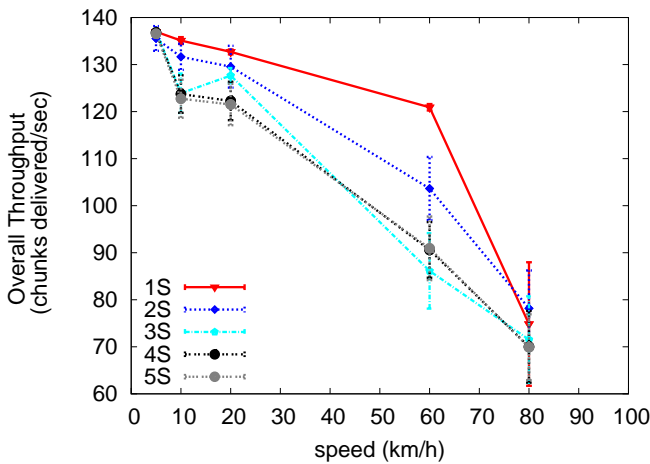


Fig. 4. Experimental Setup



(a)



(b)

Fig. 5. Overall throughput vs. speed (handoff rate) for IP address acquisition times in (a) milliseconds (b) seconds.

propagation delay of 25ms. These delays roughly approximate reasonable Internet delays for distances such as coast-to-coast of the US continental [11].

For these simulations, we have adapted a source-destination pair selection policy to increase the performance. Normally, [2] does not force a policy to select the source-destination address pair for SCTP packets. In most implementations the selection of source address for the SCTP packets destined to a specific address is left to the routing layer. In our scenario, CN is single-homed and MN is (intermittently) multi-homed. Therefore, for the SCTP packets (containing DATA or ASCONF-ACK chunks) going out from CN, the destination address is always (the current) primary path to the MN. For SCTP packets (containing SACK or ASCONF chunks) going out from MN, the source address of is the “strongest” local address of the MN.

Simulation time is 6 minutes and MN continuously downloads data from the CN. The amount of (application layer) data in each SCTP data chunk is 1200 bytes.

We measured the throughput at the MN for speeds of 5, 10, 20, 60, 80 km/h (i.e., 1, 2, 4, 12, 16 handoffs relatively, during the simulation time) with varying IP address acquisition times². Each data point in the graphs is an average of 50 runs with 95% confidence intervals.

Graphs in Fig. 5 show the overall throughput³ at the MN for varying speed (or handoff rate) and IP address acquisition times. Fig. 5(a) and 5(b) show the results for IP address acquisition times in millisecond and second scale, respectively. It is seen in both Fig. 5(b) and 5(a) that as the speed (handoff rate) increases the overall throughput at the MN decreases and confidence intervals get bigger (i.e., both the performance and the stability of the system decreases). Graphs in Fig. 6 and 7 show how the throughput at the MN changes with time for IP address acquisition times of 250ms, 500ms, 1s, and 3s in various speeds. The shaded rectangles on the graphs show when the MN is inside the handoff region. It is clearly seen from these graphs that while MN is experiencing a handoff, its (instantaneous) throughput decreases. One observation is that, when MN enters to the handoff region, CN continues using the current primary address, until the MN tells to the CN to change the primary address. During this time, the data chunks could not be transmitted to the old primary address if old primary address is not strong enough, which causes reduction in cwnd value and retransmissions. Another point is that, the cwnd of the new primary address is built *only after* the CN changes primary destination to the new path. This is another reason affecting the sending rate of the CN. We hope to mitigate these negative effects by implementing our duplicate data transfer approach during the handoffs and present the comparative simulation results.

²In this paper, we define the *IP address acquisition time* as the amount of time elapses for the MN to obtain an IP address at the new access network (by means of DHCP or Stateless Address Autoconfiguration [12]) after the MN detects the new AR with beacon messages.

³Total number of data chunks delivered to the application layer divided by the simulation time.

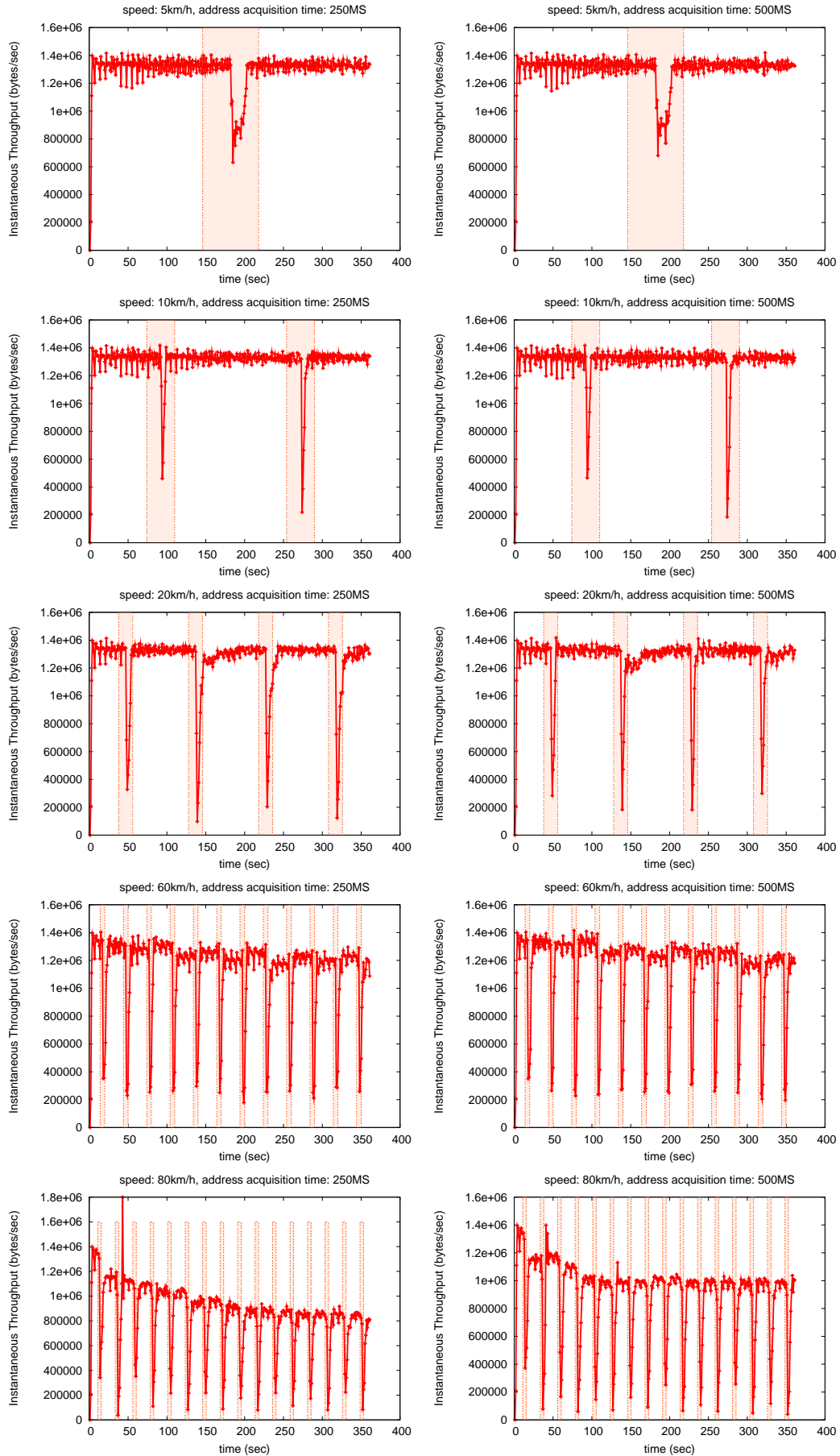


Fig. 6. Throughput at MN with respect to time for IP address acquisition times of 250MS (left-hand side) and 500MS (right-hand side) for various speeds (shaded areas show when MN is inside the handoff region)

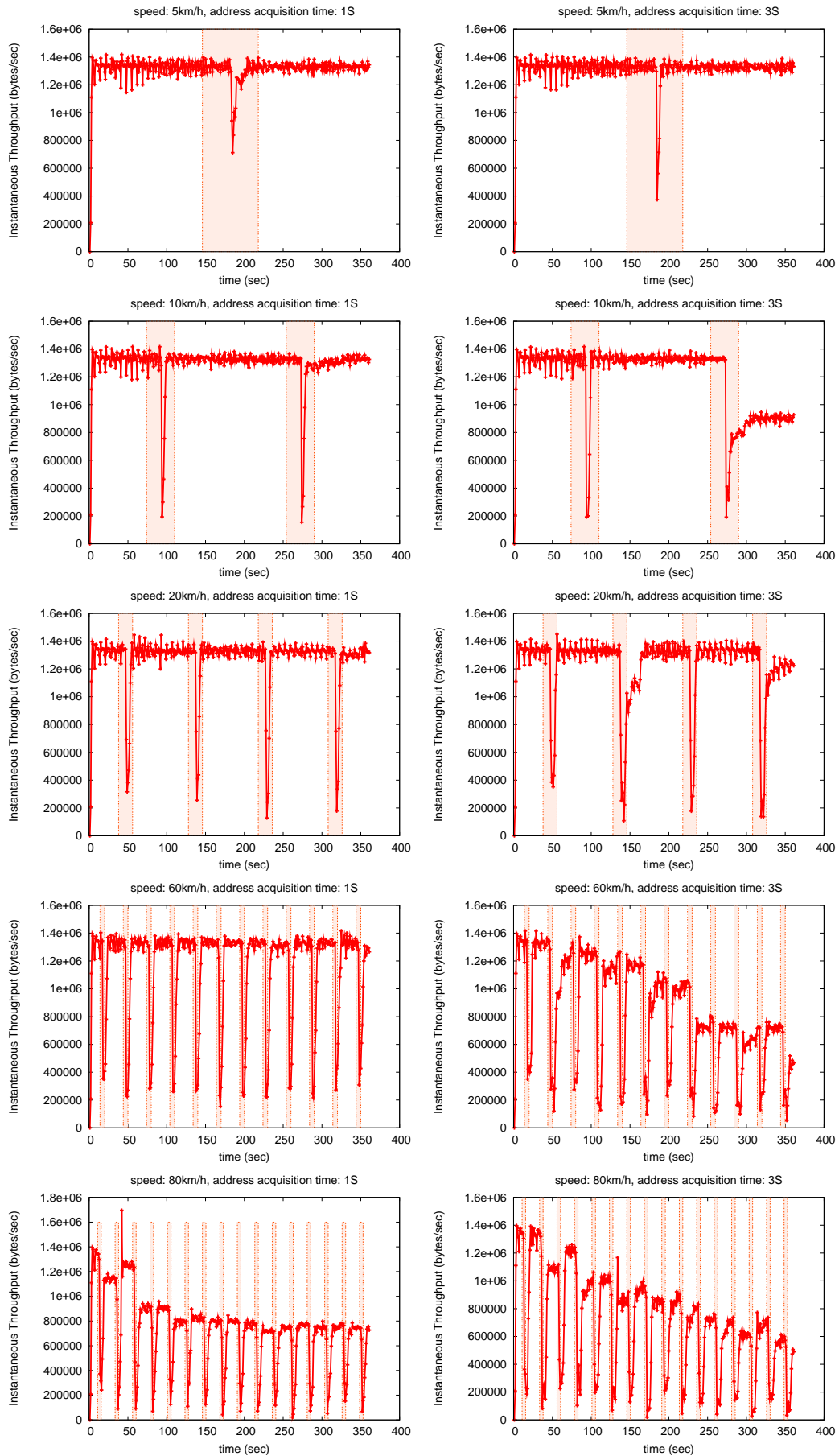


Fig. 7. Throughput at MN with respect to time for IP address acquisition times of 1S (left-hand side) and 3S (right-hand side) for various speeds (shaded areas show when MN is inside the handoff region)

IV. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the Cellular SCTP (cSCTP) architecture as a host mobility solution for the Internet and focused on the handoff management component for better performance. We suggested to use both addresses of MN during handoffs simultaneously by duplicating the data transfer. The main challenge in using the duplicated data chunks is the SACK information and processing. We suggested a preliminary approach to overcoming this challenge. We also evaluated the performance of the cSCTP architecture using QualNet network simulator for the case of using single IP primary during handoff.

We are in the process of improving and implementing the duplicated data transfer idea. As future work, we plan to investigate different congestion control, transmission, and retransmission policies before and after handoffs for horizontal as well as vertical handoffs⁴.

ACKNOWLEDGMENTS

This work is supported in part by National Science Foundation under grant CNS-0335302.

An earlier version of this paper has appeared in the proceedings of IEEE 62nd Semiannual Vehicular Technology Conference (VTC2005-Fall), Sept. 24–28 2005, Dallas, Texas.

REFERENCES

- [1] W. Eddy, "At What Layer Does Mobility Belong?" *IEEE Communications Magazine*, October 2004.
- [2] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream Control Transmission Protocol," October 2000, RFC 2960.
- [3] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, I. Rytina, M. Belinchon, and P. Conrad, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration," June 2005, draft-ietf-tsvwg-addip-sctp-12.txt (work in progress).
- [4] M. Riegel and M. Tuexen, "Mobile SCTP," August 2003, draft-riegel-tuxen-mobile-sctp-03.txt (work in progress).
- [5] P. Bhagwat, D. Maltz, and A. Segall, "MSOCKS+: An Architecture for Transport Layer Mobility," *Computer Networks*, vol. 39, no. 4, pp. 385–403, July 2002.
- [6] I. Aydin and C. Shen, "Cellular SCTP: A Transport-Layer Approach to Internet Mobility," in *12th International Conference on Computer Communications and Networks (ICCCN)*, Dallas, Texas, October 20-22 2003, pp. 32–37.
- [7] —, "Cellular SCTP: A Transport-Layer Approach to Internet Mobility," October 2003, Internet Draft (work in progress).
- [8] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," March 1999, RFC 2543.
- [9] Scalable Network Technologies, Inc., "QualNet Developer's Guide", <http://www.scalable-networks.com>.
- [10] I. Aydin, May 2005, QualNet SCTP Module - release 1.1, <http://chenyen.cs.ucla.edu/projects/whynet/download.php?v=sctp>.
- [11] A. Caro, P. Amer, and R. Stewart, "Retransmission Schemes for End-to-end Failover with Transport Layer Multihoming," in *GLOBECOM'04*, November 2004.
- [12] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration," December 1998, RFC 2462.
- [13] L. Ma, F. Yu, V. Leung, and T. Randhawa, "A New Method To Support UMTS/WLAN Vertical Handover Using SCTP," *IEEE Wireless Communications*, vol. 11, no. 4, pp. 44–51, August 2004.

⁴Horizontal handoff occurs when MN moves between the same type of access networks (for instance, between two WLANs), while vertical handoff occurs when MN moves between different types of access networks (for instance, between a UMTS and WLAN) [13].