

Cellular SCTP: A Transport-Layer Approach to Internet Mobility

Ilknur Aydin
Computer and Information Sciences
University of Delaware
aydin@cis.udel.edu

Woojin Seok
KISTI Supercomputing Center
Korea
wjseok25@kisti.re.kr

Chien-Chung Shen
Computer and Information Sciences
University of Delaware
cshen@cis.udel.edu

Abstract—In this paper, we describe a transport layer mobility scheme termed Cellular SCTP, or cSCTP for short, based on the Stream Control Transmission Protocol (SCTP), for seamless soft handoff. We compare Mobile IPv6 and two of its variants (Mobile IPv6 with Fast-Handover and Hierarchical Mobile IPv6) with cSCTP on various aspects (especially the handoff process of micro-mobility). We also analyze the Mobile IPv6 handover mechanism in detail to reveal the operations that constitutes extra delays or packet losses caused by handoffs. Furthermore, we describe the interworking of cSCTP with SIP for mobility management.

I. INTRODUCTION

Wide spread use of wireless and mobile computing and communication devices has signified the need of host mobility support in the Internet. Mobile IP supports host mobility at the network layer by deploying specially functioning routers (Home and/or Foreign Agents) into the network to keep track of current location of the mobile host, and hence be able to route the packets destined for the mobile host to its current location (usually by means of tunnelling). In contrast, Mobile SCTP [1], or mSCTP for short, has proposed a transport layer approach to host mobility based on the Stream Control Transmission Protocol (SCTP) [2].

SCTP is a new, general-purpose transport layer protocol originally designed to transport telephony signaling messages over IP networks. Like TCP, SCTP provides a connection oriented, reliable service. Moreover, SCTP provides two core features that benefit not only telephony signaling applications but also other Internet and wireless networking applications: *multi-homing* and *multi-streaming*. SCTP multi-homing allows a transport layer connection (an *association* in SCTP terminology) to be defined between a set of local IP addresses and a set of remote IP addresses. If connectivity is lost on the primary IP address being used for the association, the association seamlessly fails over to an alternate IP address. SCTP multi-streaming allows data to be partitioned into multiple streams, and each stream to be sequentially delivered to the destination end point independently of the other streams. Hence, a packet loss in one stream does not incur head-of-line blocking to other streams. In particular, mSCTP extends the base SCTP to facilitate mobility in the Internet at the transport layer.

Basically, mSCTP states that both Mobile Node (MN) and Correspondent Node (CN) need to support the *Dynamic Address Reconfiguration* extension [3] to SCTP for seamless handover¹. The base SCTP protocol allows a set of IP addresses at both source and destination end points to be decided in the association establishment phase. In contrast, dynamic address reconfiguration allows two SCTP end-points to add new IP addresses into and delete IP addresses from an active association as well as to set the primary IP address of the association with the help of newly defined ASCONF (Address Configuration Change) and ASCONF-ACK (Address Configuration Acknowledgment) chunks. Furthermore, [4] elaborates mSCTP and discusses generic procedures for seamless handover and some implementation issues for connections initiated by the MN, but is not concerned with the performance improvement.

In this paper, we propose an SCTP-based mechanism, termed Cellular SCTP or cSCTP for short, to support better handoff performance than mSCTP. In addition, we describe the interworking of cSCTP with SIP (Session Initiation Protocol) [5] to facilitate mobility management of Mobile Nodes (from Corresponding Nodes). We also review Mobile IPv6 [6], and compare Mobile IPv6 and two of its variants (Mobile IPv6 with Fast Handover [7] and Hierarchical Mobile IPv6 [8]) with cSCTP. We also analyze the handover process in Mobile IPv6 in detail to reveal the factors that constitute extra delays or packet losses caused by handoffs.

The organization of this paper is as follows. Section II details Cellular SCTP on handoff and mobility management. Section III review the operations of Mobile IPv6 along with its fast handover and hierarchical variants, and compares various aspects (especially the handover procedure of micro-mobility) of standard Mobile IPv6, Mobile IPv6 with Fast-Handover, and Hierarchical Mobile IPv6 with cSCTP. Section III also presents a simple analysis of standard Mobile IPv6 handover procedure. Related works are reviewed in Section IV. Section V concludes the paper with future research effort.

¹During handover (or handoff) process MN changes its point of attachment to the Internet. But it should still be possible for the MN to transmit and receive packets with minimum service disruption.

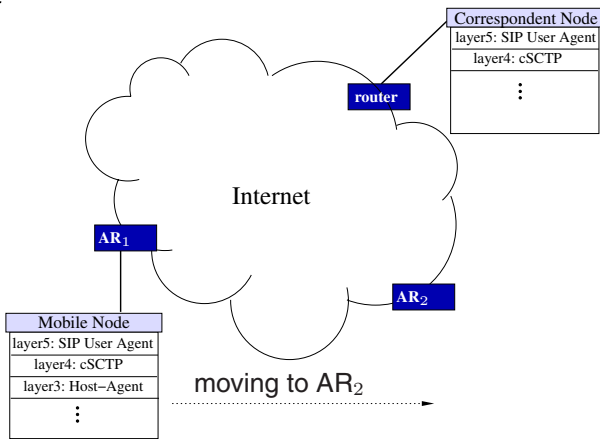


Fig. 1. Cellular SCTP operation: A cSCTP-enabled Mobile Host moving from one access router to another one. The Correspondent Node is also shown.

II. CELLULAR SCTP (cSCTP)

In cSCTP (Figure 1), a Mobile Node (MN) has three main components. (1) The Host-Agent component communicates with the Access Router(s) (ARs) mainly to help the cSCTP component learn about reaching a new AR and/or leaving the previous AR². (2) The cSCTP component is basically an SCTP protocol entity with the dynamic address reconfiguration extension [3] plus the handover procedure proposed in this paper. The Correspondent Node (CN) also needs to have a cSCTP component. (3) To facilitate mobility management, there is a SIP User Agent running at the application layer of both MN and CN. Moreover, each AR will need to support a neighbor discovery protocol such as [9]. The Cellular SCTP handover works as follows.

- *Detecting and obtaining a new IP address:* The Host-Agent and ARs communicate via a neighbor discovery protocol. The Host-Agent sends ROUTER SOLICITATION messages and ARs send ROUTER ADVERTISEMENT messages. With the help of DHCP or Stateless Address Auto configuration [10], the Host-Agent obtains a new IP address in the new point of attachment.
- *Adding the new IP address into the association:* After obtaining a new IP address in the new point of attachment, the Host-Agent informs the cSCTP component of MN about the new IP address.

cSCTP introduces a new boolean variable per SCTP association, termed *handoff_mode*. After cSCTP of the MN learns the existence of the new IP address, handover starts, and the cSCTP of MN perform the following. (1) MN sets its *handoff_mode* to *true*. (2) MN sends an ADD-IP ASCONF chunk to the CN to inform the CN about the start of the handoff mode, and to let the CN add the new IP address into the association. To do so, both ASCONF and ASCONF-ACK chunks need to be modified to notify the CN about the start of the handoff mode at the MN. We use one bit (H) of the Chunk Flags to indicate the start of handoff mode (Figure 2).

²The Host-Agent can also help cSCTP to obtain physical layer information such as the strength of the wireless signal, etc.

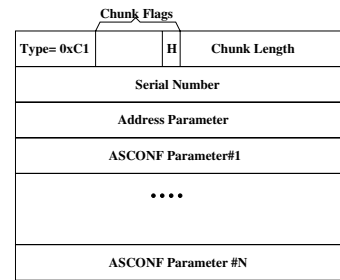


Fig. 2. Modified ASCONF Chunk. New flag H is set to signal start of handoff mode. ASCONF-ACK Chunk will be modified in the same way by adding a H flag for handover.

- (3) MN adds the new IP address into the association³.

Upon receiving the ADD-IP chunk with *handoff_mode* flag set, CN does the following. (1) CN sets its *handoff_mode* to *true*. (2) CN adds the new IP address into the association. (3) Both the old IP address of MN and the newly added IP address are considered as primary addresses to MN. Congestion window value (*cwnd*)⁴ for each of these primary addresses is set to be *half* of the *cwnd* value of the old primary IP address to the MN.

- *Data Transfer During Handover:* (Direction of data transfer is assumed to be from CN to MN to simplify explanation.) CN duplicates and sends packets to *both* of the primary addresses for MN in the rate of newly calculated, reduced *cwnd* value. Therefore, the same data will be transferred to the MN via two different paths, to reduce the probability that MN would miss the data packets sent by the CN. Actually when the old and the new attachment points of MN to the Internet are close to each other (i.e., both access routers are within the same ISP, for instance), mostly only the last hop (i.e., the wireless hop) to the MN will change.

In contrast, for mSCTP, before MN sets the new IP address to be the primary IP address of the association, data packets are sent to the old IP address. However, if MN is not reachable by the old IP address anymore, *cwnd* of the old IP address will be set to one MTU (due to timeout, and hence back to slow start at the old IP address), and then the retransmissions will be sent to the newly added IP address. The retransmissions will result in delays and decrease the performance of the SCTP association. Furthermore, in the case where MN moves to access point B from access point A, stays in access point B for a little time, moves back to access point A again, and repeats this movement pattern, each handoff will degrade the transmission rate of the CN due to timeouts and re-transmissions. Whereas, by duplicating the data transfer during the handoff mode in cSCTP, we aim to mitigate these negative effects.

- *Deleting old IP address from the association:* When the

³Unless an error is reported by the responding endpoint (CN for our case), ASCONF requests to add/delete IP addresses are considered successful. For the sake of describing cSCTP's handover mechanism, we don't consider error cases.

⁴Notice that SCTP maintains a separate set of congestion control parameters for each path if host is multihomed.

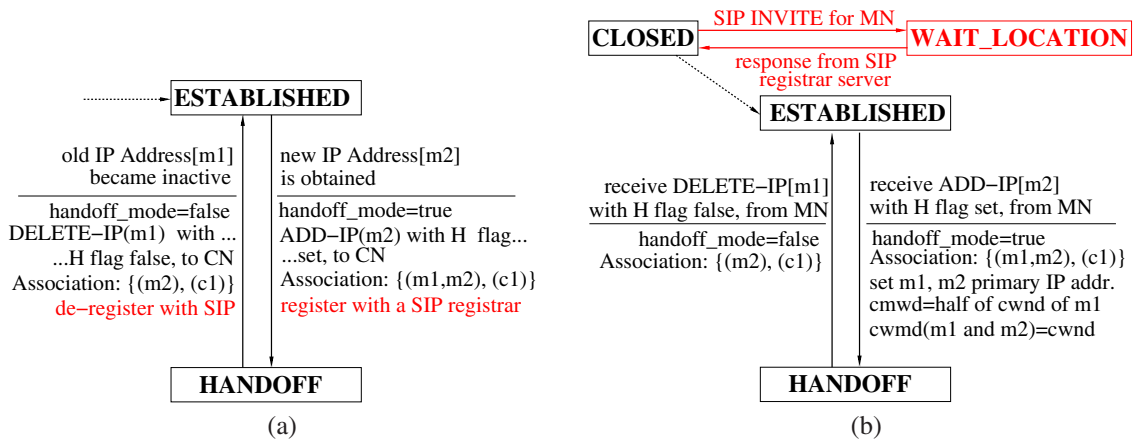


Fig. 3. The Finite State Machine diagrams for (a) Mobile Node and (b) Correspondent Node. Dotted arrows show some other paths from one state to another. Gray texts show the mechanisms added to locate MN with SIP for associations initiated by CN while black texts describe handover. Before any handoff occurs (i.e., ESTABLISHED mode entered by following the dotted arrows), the association is defined by $\{(m1), (c1)\}$ (i.e., both MN and CN are single-homed and (primary) IP address for the MN is $m1$ and (primary) IP address for the CN is $c1$.)

cSCTP at the MN decides that the old IP address is inactive⁵, the MN exits the handoff mode by setting *handoff_mode* to *false*, removes the old IP address from the association, and sends DELETE-IP ASCONF chunk (with *handoff_mode* flag set to false) to the CN.

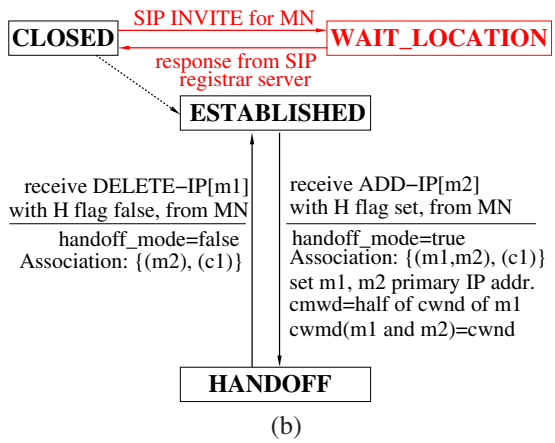
Upon receiving the DELETE-IP ASCONF chunk, CN also exists the handoff mode, deletes the old IP address from the association, and uses the new IP address as the primary address to the MN. Figure 3 depicts the finite-state-machine diagrams that summarize the steps during the cSCTP handover process.

A. Mobility Management Using SIP

One key design objective of cSCTP is have a transport layer mobility solution that is independent of network layer support such as Mobile IP. To facilitate mobility management when CNs initiate the associations and need to locate MNs, we propose to use the mobility management function of SIP to locate the (current) location(s)/address(es) of the callee (MN in our case).

The main components of SIP include *User Agents* (which initiates the requests and produces corresponding responses on behalf of the users), *Proxy Servers*, *Redirect Servers*, and *Registrar Servers*. The ways Proxy and Redirect servers process the (INVITE) requests by callers differ. For our purpose of locating the callee (i.e., MN), the redirect server would be sufficient. Registrars are the servers that users register their current location(s) with. The interworking of cSCTP and SIP to locate a MN is as follows.

Each MN runs a SIP User Agent at its application layer. Whenever the MN obtains a new IP address, the SIP User Agent registers the new IP address with the local Registrar server(s)⁶ by using SIP REGISTER request. The two important



fields of the REGISTER's request-header for the purpose of location registration are *To* and *Contact*⁷. 'Objects' in SIP are addressed as 'users at hosts' similar to an email address, identified by SIP URLs, such as *user@host*. Hence, the *To* and *Contact* fields of the REGISTER's request-header needs to be filled with such a proper SIP syntax. The *To* field will contain the "name" of the MN and the *Contact* field will contain the new IP address of the MN. For instance, the MN in Figure 1 can be given a name $\langle \text{sip:MN1@adhocNetwork.org} \rangle$ in the *To* field⁸. If MN moves to the network of AR_2 , and receives a new IP address, let's say 10.1.2.3, then the *Contact* field of the REGISTER request could contain address $\langle \text{sip:MN1@10.1.2.3} \rangle$ ⁹.

CNs will also run SIP User Agents at the application layer. A CN, who wants to initiate an association with a MN, will send a SIP INVITE request to the local redirect server¹⁰ for MN. CN will need to give the "name" of MN in the *Request-URI* (and the *To* field) of the SIP INVITE request. Continuing with the same example, CN will write $\langle \text{sip:MN1@adhocNetwork.org} \rangle$ into the *Request-URI* of the INVITE request. Then, finally the redirect server, contacting with the location service, will return the current location(s) of the MN to the CN (i.e., only $\langle \text{sip:MN1@10.1.2.3} \rangle$ in this case). SIP User Agent at the CN receiving the response from redirect server, will send the response to the cSCTP layer, where the response is processed further and finally the current location(s) of the MN is decided. Then, the cSCTP layer of the CN will use the current location(s) of MN to initiate an association with the MN.

The gray texts in the finite-state-machine diagrams of Figure

⁷For the syntax of SIP REGISTER request, please refer to [5].

⁸Note that the value of user and the host parts of the name of the MN is not important for our purpose. We just want to "name" the MN in a SIP-suitable way.

⁹Again, the value of the user part of the address is not important for our purpose.

¹⁰For the details of how a caller (CN in our case) locates a SIP server, please refer to Section 1.4.2 of [5].

⁵An MN can decide if the old IP address is inactive when MN is not receiving any data from the CN via old IP address or Host-Agent could inform the cSCTP at the MN about not receiving any Router Advertisements from the old Access Router, etc.

⁶For the details of how the local registrar server(s) are located by User Agent, please refer to Section 4.2.6 of [5].

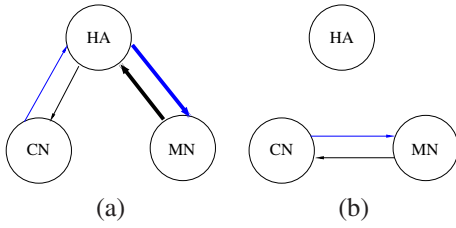


Fig. 4. Modes of operation of Mobile IPv6. Thick arrowed lines show the existence of encapsulation and tunnelling (a) Bidirectional Tunelling Mode (b) Route Optimization Mode.

3 describe the mechanisms to support location management as explained in this section.

III. ANALYSIS OF MOBILE IPV6

In this section we give an overview of the Mobile IPv6 [6], and compare the handoff efficiency of Mobile IPv6 with cSCTP from various aspects.

Three main entities of Mobile IPv6 are Home Agent (HA), Correspondent Node (CN), and Mobile Node (MN)¹¹. Unlike Mobile IPv4, there are no special local routers (i.e., Foreign Agents) required. MN updates its HA (and CN), about its (primary) care-of address by sending Binding Updates. There are two modes of operation in Mobile IPv6: *bidirectional tunnelling mode* and *route optimization mode*. In bidirectional tunnelling mode, MN does not send Binding Updates to CN. Hence, while transmitting packets¹² to the MN, CN sends packets to the home address of the MN. The packets are intercepted by the HA at the home address and then tunelled to the current (primary) care-of address of MN. In the same way, packets from MN to CN are first (reverse) tunelled from MN to the home agent and then routed normally by the HA to the CN (Figure 4(a)). In contrast, route optimization mode requires MN to update both HA and CN about its current binding¹³. Therefore, CN will have the entry in its Binding Cache about the current binding of the MN. Hence, CN directly sends data to the (primary) care-of address of MN. In the same way, (if MN is sure that current binding of MN is in the Binding Cache of CN) MN sends packets to the CN directly. (Figure 4(b)).

Obviously, route optimization helps the use of shortest path between MN and CN, while the overhead at the HA and the possible congestion at the home link of MN being reduced in comparison to bidirectional tunnelling mode. However, even if CN and MN are communicating in route optimization mode, there could be a certain time period where CN and MN are only able to communicate in bidirectional tunnelling mode, or not able to communicate at all. Hence, during this time period, packets from CN to MN (and vice versa) will be delayed or lost. We name this resulting increase in delay (due to losses or packets following a longer path) during handover *delay spike*. As a result of delay spike, there will be a decrease in the overall throughput of the connection. Later in the following

¹¹For the formal definitions of these entities, see [6].

¹²Packet in this context is an IP header plus payload.

¹³Formally defined, a binding is an association between the home address and a care-of address of MN.

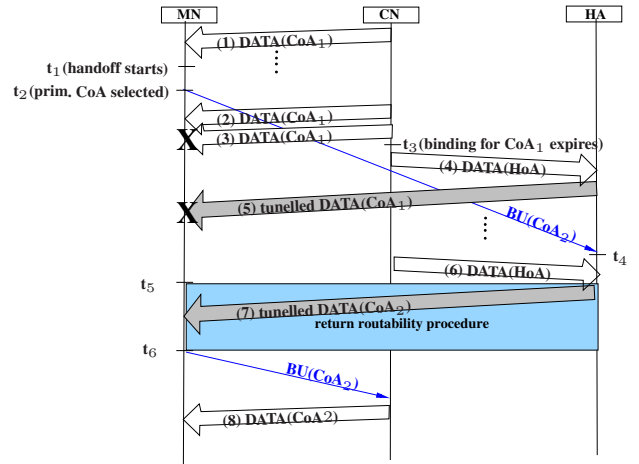


Fig. 5. Timing diagram showing the messages, delays, and possible losses during a handover where MN moves from CoA₁ to CoA₂.

subsection we will discuss delay spike further and make a simple analysis to show what contribute to the delay spike.

Let's assume a scenario where MN is to download a file from CN while moving from one access point into another (as in Figure 1). MN has a home address HoA at its home link, a care-of address CoA₁ at AR₁, and will obtain CoA₂ at AR₂. Figure 5 depicts a timing diagram¹⁴ to show the possible points of packet losses and increase in the delay. In this scenario, MN and CN operates in route optimization mode. Initially, MN establishes a connection with CN and requests the file download (not shown in the diagram). CN sends data packets to the CoA₁ of MN (Data packets labeled as (1)). At time t₁, MN detects the movement and handoff mode starts. At time t₂, MN decides that CoA₂ will be the primary CoA¹⁵ and sends a BU to its HA. In the meantime, CN possibly still having (HoA, CoA₁) binding in its Binding Cache sends the data packets to the CoA₁ (packets labeled (2) and (3)). However, if MN is not reachable via its CoA₁ anymore, some of these packets can be lost (like packets labeled as (3). Possibility of packet loss is shown with a X at the end of the arrows in the figure). At time t₃, the (HoA, CoA₁) binding in CN's Binding Cache expires; hence, CN will need to send the new data packets or retransmissions (packets labeled (4)) to HoA of MN, which is then intercepted by the HA, and tunelled to the MN. In the meantime, MN will send the ACKs back to CN via HA too¹⁶. Hence, no more route optimization at this point. Due to ACKs' (and data packets') suddenly following a longer path than before, there will be an increase in round-trip time (RTT) which could cause ACKs sent by MN to reach to the CN after current RTO expires, in which case CN goes back to slow start and hence slows down its transmission rate. Moreover,

¹⁴The timing diagram shows data packets and Binding Update (BU) messages. ACKs for data packets and BU-ACKs are not shown to keep the diagram simple and easy to follow.

¹⁵See Section 11.5 of [6] about how MN detects movement and decides on its primary CoA.

¹⁶A MN can send packets to CN directly only if MN is sure that CN has a binding in the Binding Cache, due to security related issues [6].

while HA is tunneling data packets, HA will use the existing Binding Cache entry for the MN, which would include CoA₁ of MN unless BU for CoA₂ reaches to the HA. Therefore, packets tunelled by HA to the CoA₁ of MN could also be lost if MN is not reachable by the CoA₁ anymore (Data packets shown with label (5)). At time t₄, BU for CoA₂ reaches to HA, and HA can start tunnelling packets to CoA₂. After updating its HA, MN is ready to update its CN, but first a return routability procedure¹⁷ needs to be completed. This procedure is shown as a grey box, starting at time t₅ and ending in time t₆ in the timing diagram. Finally, MN can send a BU to CN (after time t₆). After the binding procedure between CN and MN is completed, CN can send data packets directly to CoA₂ of MN. Hence, MN and CN go back to communicate in route optimization mode again.

As it is seen from the scenario above, one of the drawbacks of the standard Mobile IPv6 handover procedure is that, there will be some time period, where MN is not able to send/receive packets at all (when both MN is not reachable from its previous CoA and the Binding procedures with the HA (and CA) is not completed). Fast handover mechanism [7] intends to solve this problem. Fast handover is based on the idea of establishing a bidirectional tunnel between old and new access routers. Later, the traffic from CN to MN is tunneled from the old access router to the new access router until MN establishes itself totally at the new access point (i.e., completion of binding updates, etc.). In the same way, traffic from the MN to CN has to be tunneled from the new access router to the old access router and routed to CN. Therefore, though packet losses during the handover can be reduced with fast handover, extra processing overhead and delay are introduced at the access routers due to tunnel establishment and encapsulation/decapsulation while tunneling traffic in both directions. Whereas, cSCTP does not suffer from any of these problems. Once a new IP address is added into the association (which can be initiated right after MN obtains the new IP address), CN can receive packets at least from one of its addresses with traditional routing.

Another drawback of standard Mobile IPv6 is its inefficiency in supporting *micro mobility*. Each time MN changes its location, MN needs to send binding updates to its HA and CN. If MN is migrating frequently, the overhead due to binding update messages and delays during the binding update procedures¹⁸ will be high. Therefore, Hierarchical Mobile IP (H-MIP) [8] is introduced on top of Mobile IPv6, to provide better micro mobility handling for frequently migrating mobile nodes. H-MIP introduces the concepts of Mobility Anchor Point (MAP) and MAP domain, which can contain certain number of access routers. In H-MIP, a MN has two types of care-of addresses: local care-of address (LCoA) and a regional care-of address (RCoA). While MN is moving within

a MAP domain from one access router to another one, the LCoA of MN changes, whereas while moving between MAP domains, the RCoA of MN changes. Whenever, the LCoA of MN changes, MN updates its MAP(s)¹⁹ with local binding updates. Whenever the RCoA of MN changes, MN updates its HA and CNs with (global) binding updates. Therefore, the movement of the MN within a MAP domain becomes invisible to CN and HA. MAP behaves like a local home agent for the MN and needs to tunnel all the packets, sent by the CN to the RCoA of the MN, to the current LCoA of the MN. Hence, we could think that Mobile IPv6 is providing a mobility management scheme for macro mobility while H-MIP is providing a mobility management specifically for micro mobility. Being network-layer mobility management schemes, both Mobile IPv6 and H-MIP requires assistance from the network (routers basically) unlike cSCTP. Another drawback of H-MIP in comparison to cSCTP is that as the number of levels in the hierarchy increases, extra complexity is introduced due to MAPs' tunnelling the packets sent by CNs to the current LCoA of MN.

A. Analyzing Delay Spike in Mobile IPv6

In the scenario above (the timing diagram in Figure 5), MN downloads a file from a CN with a TCP connection over Mobile IPv6 while moving. We could define *latency* of this TCP connection as the time MN starts the TCP connection, requests the file from the CN, until it obtains the entire file. A model to derive the latency of this file downloading application for a very simplified scenario, where there are neither losses nor node movements, is presented in [11]. In addition, there are other TCP short-flow or steady-state latency and throughput models developed for the stationary networks [12], [13]. For the mobile network case, obviously, due to the possible retransmissions and timeouts during handovers, the file will take longer to download to the MN, and hence the latency of the connection (as defined above) will increase. We call this extra delay introduced to the TCP connection *delay spike*. How much the delay spike increases the latency of the connection depends on several factors.

The first of these factors is the duration of handover process. The handover starts when MN detects the movement and obtains a new care-of address. Then MN sends BUs to HA and receives BU ACKs from the HA, which we could say it would take one RTT, if successful. Then a return routability procedure, which would take approximately 1.5 RTT [8], if successful, will take place, followed by a binding update procedure between CN and MN which would again take one RTT, if no loss. Then we can formulate the duration of the handover process, termed $Time_{DH}$, as:

$$Time_{DH} = (\text{time for MN decide on its primary CoA}) + (k_1 * RTT_{MN-HA}) + (\text{HA Binding Cache update time}) + (k_2 * 1.5 * RTT_{MN-CN-HA}) + (\text{CN Binding Cache update time}) + (k_3 * RTT_{MN-CN}),$$

¹⁷Return Routability procedure is used for security purposes to help the CN to make sure that MN is really addressable in its claimed care-of address and home address.

¹⁸It takes at least one RTT to update the HA and the return routability procedure takes approximately 1.5 RTT in the best (no losses) case [8].

¹⁹H-MIP allows a MN to register with more than one MAP at the same time.

where k_1 , k_2 , and k_3 are factors added if there are losses during binding updates or return routability procedures.

Also, due to the retransmission and timeouts congestion window of the CN gets reduced and the rate at which CN sends the segments of the file to MN slows down. Yet another factor is the *handover rate* to reflect the number of handovers occurring during the transfer of the file.

In this paper, we only elaborate some of the factors that can effect the overall TCP latency in a Mobile IP environment. Research is in progress to develop models and derive closed-form TCP delay/throughput equations like [11–13] for Mobile IPv6 with handoffs.

IV. RELATED WORK

There are other network-layer mobility management schemes besides Mobile IP. Cellular IP [14] supports micro-mobility management for frequently migrating mobile hosts. In [15], an approach based on hierarchical Mobile IP is proposed to support fast handover while still providing route optimization. This scheme uses Gateway Edge Nodes (G-ENs) and Temporary Home Agents (THAs) together to act like a MAP. In addition to LCoA, RCoA, and home addresses, a new care-of address called Temporary Care-of address (TCoA) per node is introduced. While THAs keep (TCoA, LCoA) bindings, G-ENs have (RCoA, LCoA) bindings, and HA and CNs have (TCoA, HoA) bindings. However, all these newly introduced protocol entities and address make the protocol even more complicated in comparison to Cellular SCTP. Moreover, a change to Mobile IP is proposed in [16] which represents a mobile host with multiple (changing) IP addresses, instead of a single (fixed or home) IP address, by changing the semantics of the API between the IP layer and the upper layers. This approach can be considered as a patch to Mobile IP to support multi-homed hosts at the network layer (instead of transport layer).

In addition to Mobile SCTP, there are other transport layer mobility schemes, such as Indirect-TCP (I-TCP) [17] and MSOCK [18]. I-TCP is based on the idea of *indirection* at the transport layer, where a transport layer connection between MN and CN is divided into two separate connections, one between MN and Mobile Support Router (MSR) at the wireless link, and another between the current MSR and CN. MSOCK architecture is based on a similar technique called *TCP Splice*, where a logical TCP connection between MN and a CN is divided into two separate TCP connections between MN and a proxy, and a proxy and the CN. Both of these approaches require some changes to the TCP/IP protocol stack, while our Cellular SCTP supports transport-layer mobility as an ‘application’ of SCTP.

V. CONCLUSIONS

In this paper, we described a transport-layer mobility solution for the Internet, termed Cellular SCTP, based on the

emerging SCTP protocol. Cellular SCTP handles handovers seamlessly and without delay spikes in comparison to network-layer based mobility schemes, such as Mobile IP. We also investigated Mobile IPv6 along with its fast handover mechanism and hierarchical schemes, and compared them with Cellular SCTP on various aspects.

We suggested to use two primary addresses in parallel to duplicate the packet transmissions (while halving the transmission rate) during handoff to provide ‘soft’ handover. However, more effective load balancing techniques (combined with SCTP’s multi-streaming feature) for softer and more efficient handover support require further investigation. Moreover, though our mechanism takes care of associations initiated by either MN (to CN) or CN (to MN), as a future work we will investigate associations initiated by MN to another MN to offer a complete solution.

REFERENCES

- [1] M. Riegel and M. Tuxen, “Mobile SCTP,” February 2003, draft-riegel-tuxen-mobile-sctp-02.txt.
- [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,” February 2003, draft-ietf-tsvwg-addip-sctp-07.txt.
- [4] S. J. Koh, H. Y. Jung, S. H. Kim, and J. S. Lee, “SCTP with Mobile IP for IP Mobility Support,” February 2003, draft-sjkoh-mobile-sctp-mobileip-00.txt.
- [5] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, “SIP: Session initiation protocol,” March 1999, rFC 2543.
- [6] D. Johnson, C. Perkins, and J. Arkko, “Mobility Support in IPv6,” February 2003, draft-ietf-mobileip-ipv6-21.txt.
- [7] R. Koodli, “Fast Handovers for Mobile IPv6,” March 2003, draft-ietf-mobileip-fast-mipv6-06.txt.
- [8] H. Soliman, C. Castelluccia, K. El-Malki, and L. Bellier, “Hierarchical Mobile IPv6 Mobility Management (HMIPv6),” October 2002, draft-ietf-mobileip-hmipv6-07.txt.
- [9] T. Narten, E. Nordmark, and W. Simpson, “Neighbor Discovery for IP Version 6 (IPv6),” December 1998, RFC 2461.
- [10] S. Thomson and T. Narten, “IPv6 Stateless Address Autoconfiguration,” December 1998, RFC 2462.
- [11] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 2nd ed. Addison Wesley, 2003.
- [12] N. Cardwell, S. Savage, and T. Anderson, “Modelling TCP Latency,” in *IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.
- [13] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modelling TCP Throughput: A Simple Model and its Empirical Validation,” in *ACM SIGCOMM 1998*, Vancouver, CA, 1998, pp. 303–314.
- [14] A. G. Valko, “Cellular IP - A New Approach to Internet Host Mobility,” *ACM Computer Communication Review*, January 1999.
- [15] T. Kato, R. Takechi, and H. Ono, “A Study on Mobile IPv6 Based Mobility Management Architecture,” *Fujitsu Scientific & Technical Journal*, pp. 65–71, June 2001.
- [16] P. Nikander, “TCP and UDP in the Mobile World, or What is Wrong with Mobile IP version 6, and How to Fix it,” in *in Proceedings of NordU’2001*, Stockholm, Sweden, February 14–16 2001, invited Talk.
- [17] A. Bakre and B. R. Badrinath, “I-TCP: Indirect TCP for Mobile Hosts,” *15th International Conference on Distributed Computing Systems*, 1995. [Online]. Available: citeseer.nj.nec.com/bakre94iTCP.html
- [18] D. A. Maltz and P. Bhagwat, “MSOCKS: An Architecture for Transport Layer Mobility,” in *IEEE INFOCOM 1998*, 1998, pp. 1037–1045. [Online]. Available: citeseer.nj.nec.com/60438.html