World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:12, 2009

ECA-SCTP: Enhanced Cooperative ACK for SCTP Path Recovery in Concurrent Multiple Transfer

GangHeok Kim, *Non-Member, IEEE,* SungHoon Seo, *Member, IEEE,* JooSeok Song, *Member, IEEE*

Abstract—Stream Control Transmission Protocol (SCTP) has been proposed to provide reliable transport of real-time communications. Due to its attractive features, such as multi-streaming and multi-homing, the SCTP is often expected to be an alternative protocol for TCP and UDP. In the original SCTP standard, the secondary path is mainly regarded as a redundancy. Recently, most of researches have focused on extending the SCTP to enable a host to send its packets to a destination over multiple paths simultaneously. In order to transfer packets concurrently over the multiple paths, the SCTP should be well designed to avoid unnecessary fast retransmission and the mis-estimation of congestion window size through the paths. Therefore, we propose an Enhanced Cooperative ACK SCTP (ECA-SCTP) to improve the path recovery efficiency of multi-homed host which is under concurrent multiple transfer mode. We evaluated the performance of our proposed scheme using ns-2 simulation in terms of cwnd variation, path recovery time, and goodput. Our scheme provides better performance in lossy and path asymmetric networks.

Keywords—SCTP, Concurrent Multiple Transfer, Cooperative Sack, Dynamic ack policy

I. INTRODUCTION

DURING the last decade, Stream Control Transmission Protocol (SCTP) has been proposed and standardized for the signalling of PSTN networks [1]. The most attractive features of SCTP like multi-homing and multi-streaming are often expected to alternate other transport layer protocols such as TCP and UDP. The SCTP is capable transmitting data by using more than one additional path (*i.e.*, a secondary path) as an alternative to a primary path but it may increase path redundancy. That is, when the transmission via the primary path fails by a specific reason (*e.g.*, link or route failure), the secondary path is used for the packet delivery instead of the primary path. Recently, many efforts has been investigated to extend the SCTP's multi-homing feature. The multi-homing allows each host to maintain multiple IP addresses at the same time within an association in a transport layer protocol. Especially, authors of [2], [3], [4] provide the extended multi-homing solutions for concurrent multiple transfer (CMT). Using the CMT extensions, a multi-homed host is able to send multiple packets simultaneously via its available paths so that it increases the efficiency of data transmission within the entire networks. However, the CMT has several limitations when out of order packet is delivered from a source to a destination host due to the asymmetric characteristics of multiple paths (*i.e.*, loss rate, round trip time, bandwidth). This may introduce unnecessary fast retransmission procedures at the source host. In addition, when the source host misestimates the missing chunks report from its available path, congestion window size (cwnd) is not determined properly. Therefore, in this paper, we proposed an Enhanced Cooperative ACK based SCTP (ECA-SCTP) to transfer packets over multiple paths concurrently and rapidly recover the transmission rate from the congested path using alternative path. Furthermore, the proposed scheme improves the overall performance of throughput by using cooperative selective ack which contains information of other available paths even when the ack packets are lost.

The rest of the paper is organized as follows: Section II presents CMT extensions provided in the previous literatures. In Section III, we introduce the details of our proposed ECA-SCTP and show how the enhanced cooperative ack improves the performance of the SCTP in CMT. The analysis and the performance evaluation of the proposed mechanism are provided in Section IV and Section V concludes this paper.

II. RELATED WORKS

A number of researches on the concurrent multiple transfer and bandwidth aggregation at the transport layer are currently in progress [2]- [5]. Most of related works focus on reducing or detecting out-of-order packets induced by different path characteristics such as round trip time, loss rate, bandwidth[6]. In Load-Sharing SCTP (LS-SCTP)[3], authors suggest a mechanism to support concurrent multiple transfer, which adds available paths and/or remove disable paths dynamically by monitoring the status of paths. Through available paths, the LS-SCTP introduces a per-association and per-path data-unit sequence numbering so that LS-SCTP separates flow control from congestion control. That is, the LS-SCTP manages each paths independently and send gap report to the path only when packets from each path are delivered out-of-order. So the mechanism can avoids spurious fast retransmission and does not increase Selective Acknowledgments (SACKs) traffic but it can not notify that packets may be lost quickly.

In [2],[7], the original SCTP proposer et al suggest to change the source host's operation not but data or SACK chunk format to solve the problems introduced by out-of-order packets in CMT. The mechanism called Concurrent Multiple Transfer (CMT) uses the association sequence number (ASN) which is unique in an association to track packets sent over multiple paths. The sender host maintain per-path virtual queues and send the packets over available paths within the limit of the congestion window. However, just by association sequence number, the destination host can not determine

---

G. Kim and J. Song is with the Department of Computer Science, Yonsei University, Seoul, Korea, e-mail: drface@emerald.yonsei.ac.kr

S. Seo is with the Department of Computer Science at Columbia University, New York, USA e-mail: hoon@cs.columbia.edu

whether the packet is lost or just out-of-order by relative path characteristics such as bandwidth, delay, etc. In latter case, it can not be concluded that packet loss occurred because the packets might be delivered by slow path and receive after a while. So retransmissions are triggered only when several consecutive sacks report missing chunks within the path.

In [4], the authors propose Concurrent Multi-Path SCTP (cmpSCTP), similar to LS-SCTP[3], which is also based on the idea of separating the association flow control from congestion control. That is, the cmpSCTP use both per-path sequence number and per-association sequence number. For the flow control, a association have a single, shared send buffer and each connections have its own send buffer. When the receiver host send a sack which includes information of other paths such as advertised receiver window credit, the shared send buffer and send buffer of connections are refreshed together. On the other hand, congestion control is performed on per path basis, thus the sender host has a separate congestion control for each path and maintain congestion variables such as round trip time(RTT), retransmission timeout(RTO), advertised receiver window credit(RWND). In contrast with LS-SCTP, the mechanism can infer missing chunks of the specific paths because sacks which reports missing chunks are based on association sequence number. But cmpSCTP may send sacks false gap report introduced owing to difference of asymmetric path characteristics.

To support concurrent multiple transfer in wireless networks, the authors in [8] suggest Wireless networks using Multi-Path transmission(WiMP-SCTP) which enables a multi-homed mobile host to transmit data through available paths concurrently. WiMP-SCTP use mode selection scheme which can be chosen according to the network status by using 1 bit flag of chunk header. By using the mode selection scheme, a mobile host can stripe its data across available paths for goodput when the network status is good. Also a multi-homed host can transmit data through available paths for reachability when the network status is bad.

To support concurrent multiple transfer at transport layer, reordering problem of packets should be solved. Most works modifies sender operation so that the sender can detect false gap report of missing chunks, which be induced by the differences of path characteristics and may not be out-of-order in fact. But unnecessary false gap reports increase ack traffic and wastes the bandwidth of paths. To reduce reordering of packets, other approaches use packet scheduling algorithms so that the receiver can get packets in order. However, none of the previous proposals fully addresses the case in which the paths included in the SCTP association exhibit various bandwidths and round trip times(RTTs). In such scenario, the packets reaches the receiver out of order. Not to degrade performance, it is important to report true missing packets during out-of-order packets as soon as possible. In next section, we will refer to the key features of Enhanced Cooperative SACK SCTP (ECA-SCTP) and present the detailed operations of the ECA-SCTP protocol.

III. PROPOSED SCHEME

Our scheme uses two ways to solve the reordering problem and support concurrent multiple transfer effectively in SCTP. One of that is to use cooperative sack which includes information of available paths. the other is to provide dynamic sack policy which is determined by the sender host according to the status of paths.

A. *Using Cooperative Sack*

TABLE I: Notation of ECA-SCTP

Term	Notation
ASN	association sequence umber to distinguish order during chunks in one association
PID	path identifier to specify path between specific sender host and receiver host
PSN	path sequence number to distinguish order during chunks in one path
CumPSN	cumulative path sequence number in specific path
Prwnd	advertised receiver window credit for specific path
PS	path specifier to indicate dynamic ack policy

We modified the data and sack chunk format so that the host can perform per-path congestion control and send the cooperative sack. Table I presents notation of term used in modified data or sack chunk format. Association Sequence Number (ASN) is the unique sequence number used to order packets transmitted over available paths in one association. On the contrary, Path Sequence Number (PSN) means the unique sequence number increasing monotonically within one path. Path Identifier (PID) is used to indicate a specific path to transmit packets. Therefore, there can be some chunks having same PSN but different PID in an association. The receiver host tracks PSN to order packets in one path. So, the host can perform the congestion control by PSN and PID independently with other paths. On receiving packets out-of-order in same path, the receiver sends gap report of missing chunks on the basis of PSN. Cumulative Path Sequence Number (CumPSN) means cumulative acknowledgement, which records the highest sequential PSN received. Path advertised Receiver WiNDow credit (Prwnd) presents buffer space left in the path of the receiver. Finally Path specifier (PS) flag is used to indicate dynamic ack policy to the receiver host. According to path status(*e.g.*, loss rate, round trip time), the sender host decides ack policy and inform the receiver host by using PS flag ,which is 3 bits of flag field in the data chunk format. detailed mechanism about dynamic ack policy will be described in next section.

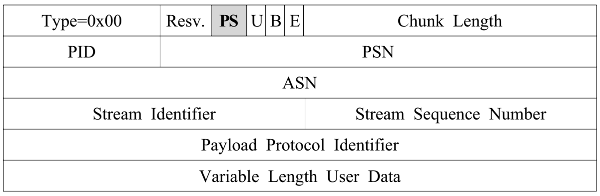


Fig. 1: Modified Data Chunk Format

As shown in figure 1, PID and PSN are used to check in-sequence delivery of packets within one path. Also to prevent false fast retransmission which is occurred when path characteristics are asymmetric relatively, we perform congestion control on the basis of Path Sequence Number(PSN). the sender host maintains highest outstanding path sequence number and cumulative path sequence number of each path. if outstanding packets are acknowledged by the receiver, the sender host update congestion variables of the path. So even if packets are out-of-order on the basis of association sequence number, packets may not be out-of-order on the basis of path sequence number. therefore by using modified data chunk, our scheme can send the gap report only when the packets are delivered out-of-order within one path. Also figure 2 describes sack chunk format that modified to use cooperative ack over multiple paths. Cooperative sack includes cumulative path sequence number of each paths to acknowledge outstanding packets. and by using cumulative association sequence number, the sender host update association sender buffer on the basis of association sequence number. Also time stamp is used to order sacks received out-of-order from the different paths. and sack may include gap ack block or duplicate ack block of each path. So path identifier is written together with gap ack blocks. By using cooperative sack, we can inform missing chunk quickly through the best path to the sender. Consequently, the sender host can determine the status of the path that the packets may be lost and retransmits the packet earlier.

<table>
<tr><td>Type=3</td><td>Flag=0</td><td colspan="2">Length=variable</td></tr>
<tr><td colspan="4">Timestamp</td></tr>
<tr><td colspan="4">Cum_ASN</td></tr>
<tr><td>PID #1</td><td colspan="3">Cum_PSN</td></tr>
<tr><td colspan="4">Advertised Receiver window credit(rwnd1)</td></tr>
<tr><td>PID #2</td><td colspan="3">Cum_PSN</td></tr>
<tr><td colspan="4">Advertised Receiver window credit(rwnd2)</td></tr>
<tr><td colspan="2">Num of Fragment = N</td><td colspan="2">Num of Duplicate = M</td></tr>
<tr><td>PID #1</td><td colspan="2">Gap ack start #1</td><td>Gap ack end #1</td></tr>
<tr><td>PID #2</td><td colspan="2">Gap ack start #N</td><td>Gap ack end #N</td></tr>
<tr><td>PID #1</td><td colspan="3">Duplicate_PSN #1</td></tr>
<tr><td>PID #2</td><td colspan="3">Duplicate_PSN #M</td></tr>
</table>

Fig. 2: Modified Cooperative Sack Format

B. Dynamic Sack Policy

For fast path recovery, our proposed scheme provides dynamic ack policies so that the sender host can select the policy according to network path status. when the network status is bad and most of packets are lost, the sender may selects the duplicate mode for reachability of the packets. then the receiver sends the ack packet over all of paths. On the contrary, if the network status is good and the packet loss is occurred often, the sender host may choose the specific mode for efficiency. then in the normal case, the receiver returns sack to the path that the last data packet is received. But if missing chunk is occurred on the basis of path sequence number, the receiver return sack on the most appropriate path indicated by the sender host for fast path recovery. To notify the sender's ack policy to the receiver, our proposed scheme uses 3 bits of flag field in DATA chunk, of which 3 bits are used just to indicate data fragmentation and unordered delivery originally in RFC4960[1] and other bits is being unused.

TABLE II: Dynamic Ack policy of ECA-SCTP

PS flag	Notation
000	Normal
001	Duplicated mode
100	Specific mode
101-111	1st/2nd/3rd path specifier

The receiver sends gap report to one of the paths which PS flag is set from 101 to 111 when missing chunks are discovered. When the first bit of PS flag is 1, 2 bits of the flag indicate the priority in sending gap report. If packets of the first path, that flag is set to 101, are lost, the receiver send gap report to second path that flag is 110. Consequently, we can gain the effect of fast path recovery through different ack policies according to network path status.

IV. PERFORMACE EVALUATION

In this section, we explore performance of ECA-SCTP. To evaluate our scheme, we have experimented three factors such as cwnd variation, goodput, fast recovery time. We present the effectiveness of our proposed scheme with compared to CMT-SCTP in some simulation scenarioes. For this, we evaluated our scheme by modifying ns-2 SCTP module which is recently released to support concurrent multiple transfer.

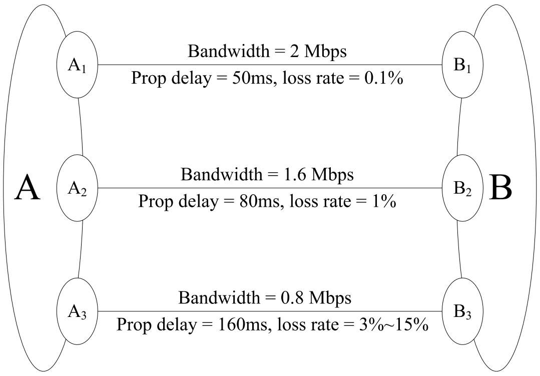


Fig. 3: Simulation Topology

A. Simulation Setup

1) Assumption: For simulation, we assume that an SCTP association was already initiated between the two nodes and each nodes are to send the data packet to the other. In addition, we use only one stream number within the association. So the advantages caused by multi-streaming is omitted. Also we assume some values and environments for simulation. First, the bottleneck link is not shared. it means that paths is independent with each other. So when one path is congested, other paths are unrelated. In addition, chunk bundling was excepted to simplify the simulation. Some variables for simulation was set as follows.

- Initial ssthresh is 65536 for each path.

- Initial cwnd is 2*MAX_CHUNK_SIZE.
- Initial Receive buffer size is 65536 bytes for each path.
- Maximum transmission unit is 1500 bytes.
- MAX_CHUNK_SIZE is 1468 bytes.
- SCTP common header size is 12 bytes.

B. *Simulation Scenario*

Also we used the network topology shown in Figure 3 to investigate the effect of different loss rate about throughput and end-to-end delay. We investigated communication between two hosts with 3 multi-interfaces. The bandwidths of each path are 2.0Mbps for path1, 1.6Mbps for path2 and 0.8Mbps for path3. Also propagation delays are respectively 50ms, 80ms, 160ms. The loss rate of path 1 and path 2 is fixed as 0.1% and 1%. The loss rate of path 3 varies from 3% to 15%. we investigated the effect of loss rate about performance of our scheme in changing loss rate of path 3 from 1% to 15%. Also path means route binding the one interface of sender node and that of receiver node. In next section, we analyze the performance of our scheme in comparison with the simulation result of the CMT-SCTP.

C. *Simulation Result*

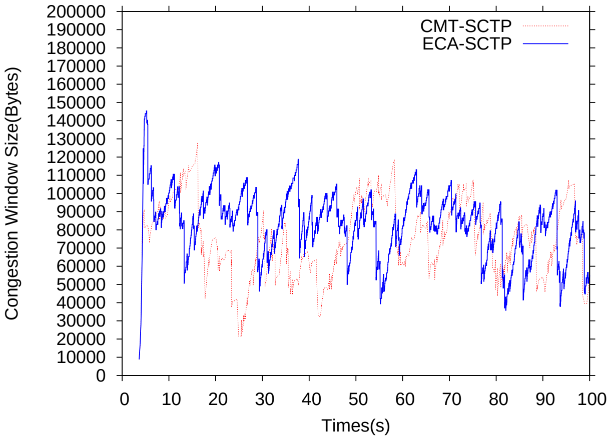
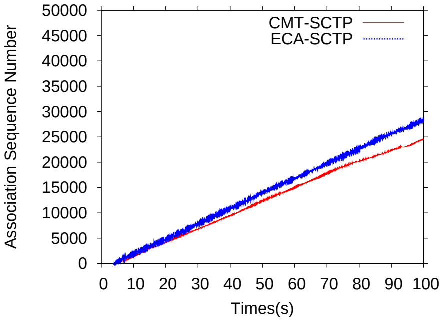
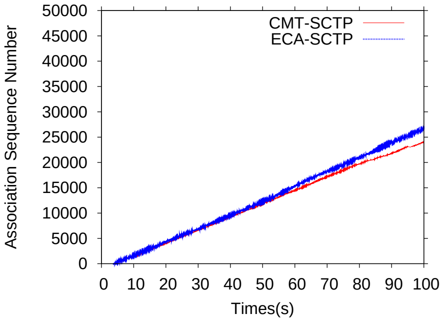


Fig. 4: Cwnd Variation at lossrate path1=0.1% path2=1% path3=15%

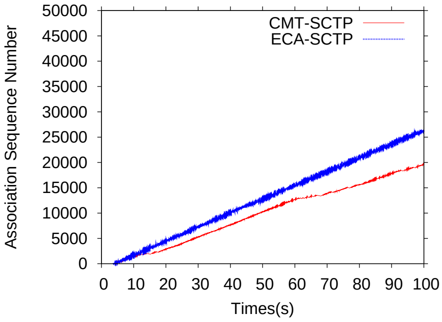
1) Cwnd Variation: Firstly, we explored the effect of different loss rates on cwnd variation. Figure 4 show overall cwnd variation of the association under different loss rates - 3%,5%,10%,15% in ECA-SCTP and CMT-SCTP. In CMT-SCTP, Receive Buffer is maintained commonly in an association. So cwnd may have bigger value than maximum receiver buffer size of one path. but because decision of missing chunk production is based on association sequence number, there may be many cwnd degradation of other path. It shows that smaller cwnd deviation relatively in ECA-SCTP rather than CMT-SCTP. it is the reason why receiver node send the cooperative sack over fastest path when missing chunk is detected in ECA-SCTP. So, path recovery time is smaller than the other scheme and it makes deviation of cwnd variation smaller. Because our scheme send sack including missing chunk over faster path, sender node can decide fast retransmission and adjust cwnd size quickly than other schemes.



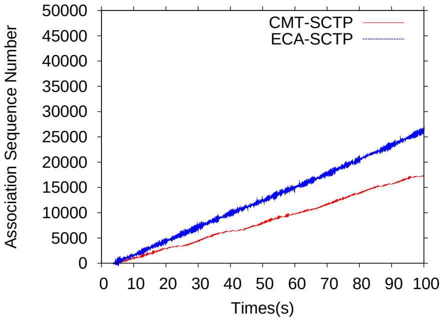
(a) path1=0.1% path2=1% path3=3%



(b) path1=0.1% path2=1% path3=5%



(c) path1=0.1% path2=1% path3=10%



(d) path1=0.1% path2=1% path3=15%

Fig. 5: goodput at different loss rates

2) goodput: We examined total transmitted packet number to know overall goodput. Figure 5(a) Figure 5(d) show how many packets were delivered during the simulation time. As shown in the figures, we can know that ECA-SCTP provide little better goodput than CMT-SCTP in environments that path loss rate is bigger. Also the bigger difference of path loss rate is, the bigger difference of performance between our scheme and CMT-SCTP is. it is the reason why our scheme select the best path to send gap report.

3) Fast Retransmission Time: Finally, we examined the fast recovery time of lost packets. Fast recovery time means the time between the time that packet is entered in queue to transport and the time that packet is dropped and retransmitted and received by receiver node. Table III shows that average time for fast retransmission and the number of lost packets with regard to different path3 loss rate. the left side of figures is the result of CMT-SCTP and the right side is the result of ECA-SCTP. Figures shows that fast retransmission time varies largely in CMT-SCTP but the time is steady in ECA-SCTP. Also as the path 3 loss rate is bigger, the fast retransmission average time is bigger and the number of lost packet is more and more. it is the reason why our scheme use the best path to send sack including missing chunk quickly. but CMT-STCP use last acknowledged path to send sack.

TABLE III: Fast retransmission time

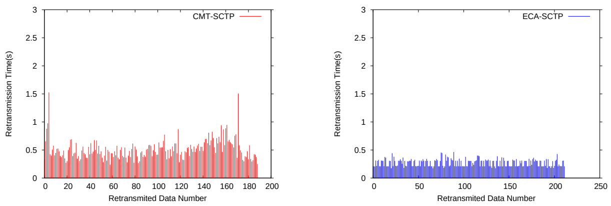
Experiment	CMT-SCTP		ECA-SCTP	
	Average Time(s)	Count	Average Time(s)	Count
Loss 3%	0.51192	189	0.272309	213
Loss 5%	0.506046	203	0.280514	261
Loss 10%	0.6035	215	0.306223	357
Loss 15%	0.644724	202	0.322321	459

V. CONCLUSION

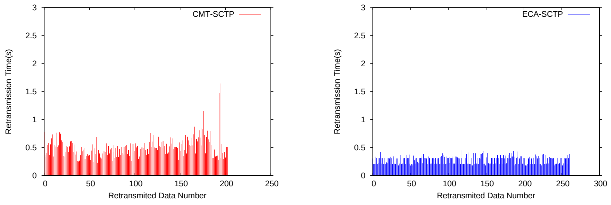
In this paper, we proposed ECA-SCTP mechanism which extend SCTP to support concurrent multiple transfer effectively by some modification. By using cooperative selective acks and selecting sack policy according to network status and controlling congestion in concurrent multiple transfer, we improved overall performance. our scheme send the cooperative sack including information of all paths and receiver node can update all of paths status. Also our scheme notify to receiver node the best path to acknowledge by using dynamic ack policy that is written in data chunk flag field by sender node. So sender node receive the acknowledge including missing chunk information quickly. So fast recovery time decreased and throughput is increased. As showed in simulation result, ECA-SCTP show the better performance in environment that characteristics of network paths are different and lossy network than other scheme. So our scheme will be appropriate in wireless network that variation of path loss rate is severe and characteristics of path are asymmetric. Finally ECA-SCTP show better performance than other schemes in lossy and path-asymmetric networks.

ACKNOWLEDGMENT

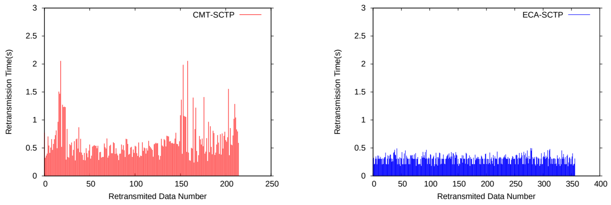
This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST)(2009-0076476).



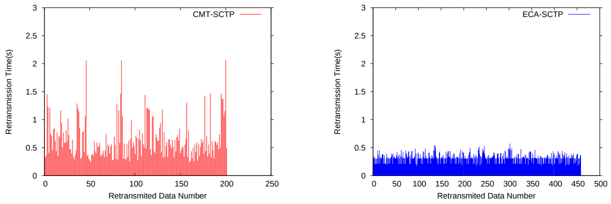
(a) lossrate path1=0.1% path2=1% path3=3%



(b) lossrate path1=0.1% path2=1% path3=5%



(c) lossrate path1=0.1% path2=1% path3=10%



(d) lossrate path1=0.1% path2=1% path3=15%

Fig. 6: Fast Retransmission Time at different loss rates

REFERENCES

[1] E. R. Stewart, “RFC4960: Stream Control Transmission Protocol,” Sep. 2007.

[2] J. R. Iyengar, P. D. Amer, and R. Stewart, “Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths,” *IEEE/ACM Trans. Netw.*, vol. 14, no. 5, pp. 951–964, Oct. 2006.

[3] A. Abd El Al, T. Saadawi, and M. Lee, “LS-SCTP: a bandwidth aggregation technique for stream control transmission protocol,” *Computer Communications*, vol. 27, no. 10, pp. 1012–1024, Jun. 2004.

[4] J. Liao, J. Wang, and X. Zhu, “cmpSCTP: An Extension of SCTP to Support Concurrent Multi-Path Transfer,” in *Proc. IEEE International Conference on Communications ICC ’08*, 19–23 May 2008, pp. 5762–5766.

[5] H. Y. Hsieh and R. Sivakumar, “A Transport Layer Approach for Achieving Aggregate Bandwidths on Multi-Homed Mobile Hosts,” *Wireless Networks*, vol. 11, no. 1, pp. 99–114, Jan. 2005.

[6] *Dynamic load balancing without packet reordering*, vol. 37, no. 2. New York, NY, USA: ACM, 2007.

[7] J. R. Iyengar, P. D. Amer, and R. Stewart, “Receive buffer blocking in concurrent multipath transfer,” in *Proc. IEEE Global Telecommunications Conference GLOBECOM ’05*, vol. 1, 28 Nov.–2 Dec. 2005, p. 6pp.

[8] C. M. Huang and C. H. Tsai, “WiMP-SCTP: Multi-Path Transmission Using Stream Control Transmission Protocol (SCTP) in Wireless Networks,” in *Proc. 21st International Conference on Advanced Information Networking and Applications Workshops AINAW ’07*, vol. 1, 21–23 May 2007, pp. 209–214.

GangHeok Kim GangHeok Kim received the B.S degree in Physics from Yonsei University, Seoul, Korea, in 2003. He is currently working on M.S degree at Yonsei University, Seoul, Korea. His research interests include concurrent multiple transfer in wireless and mobile networks.

SungHoon Seo SungHoon Seo received M.S. degree in computer science from Yonsei University, Seoul, Korea, in 2002. From 2002 to 2004, He had worked at LG Electronics CDMA Handset Laboratory as a research engineer. In 2009, he received the Ph.D. degree in computer science from Yonsei University, Seoul, Korea. He is currently a postdoctoral research fellow in Department of Computer Science at Columbia University, New York, USA His research interests include mobility management in mobile and wireless communications, and interworking in heterogeneous networks.

JooSeok Song JooSeok Song received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1976, and the M.S. degree in electrical engineering from KAIST, Korea, in 1979. In 1988, he received the Ph.D. degree in computer science from University of California at Berkeley. He had been an assistant professor of Naval Postgraduate School, California, USA, from 1988 to 1989. He is currently a professor of computer science at Yonsei University, Seoul, Korea. His research interests include cryptography, network security, and data communications.