Fast Selective ACK Scheme for Throughput Enhancement of Multi-Homed SCTP Hosts

Lin Cui, Seok Joo Koh, and Woo Jin Lee

Abstract—**This Letter proposes a fast selective ACK scheme for Stream Control Transmission Protocol (SCTP) to enhance transmission throughput in multi-homing scenarios. In the proposed scheme, a multi-homed receiver sends SACK chunks to the sender over the fastest reverse path, which facilitates to inflate the congestion window and to retransmit the lost data packets as quickly as possible. Simulation results show that the proposed scheme could improve the throughput performance of multi-homed SCTP hosts under both normal SCTP and CMT-SCTP cases if only the reverse paths have various one-way delays.**

Index Terms—**SACK, Fast SACK, multi-homing, CMT, SCTP.**

I. Introduction

FOR end-to-end transport protocols, one of the critical issues is throughput degradation that may be induced by asymmetric transmission delays. The works in [1], [2] discussed TCP performances in the networks with asymmetric delays for forward and reverse paths, but did not consider the multi-homed hosts. Based on the multi-homing scenarios, the authors in [3] tend to use a symmetric round-trip routing with the minimum round-trip time (RTT) to improve the throughput. But the authors in [4] argue that it is better for the sender to use the asymmetric data and Selective ACK (SACK) paths with the minimum RTT. Both works [3], [4] ignore the Concurrent Multi-path Transport (CMT-SCTP) cases [5], [6]. In CMT cases, on one hand, the problem might get worse since the increased out-of-order data and SACK segments will result in the more unnecessary fast retransmissions, the much reduction of congestion window and the higher traffic of SACKs; on the other hand, data segments will be delivered along all available forward paths, thus only the minimum RTT does not matter for the CMT schemes [5], [6].

Our study is based on the notion that the throughput of multi-homed SCTP hosts can be enhanced, if a receiver transmits SACK chunks to a sender, as fast as possible, which will be helpful to inflate the congestion window and to retransmit the lost data segments quickly, especially in the following cases: 1) static CMT case [5]; 2) case with frequent slow start and/or frequent fast retransmission/recovery phases; 3) handover with CMT in overlap area between heterogeneous networks, as described in [6].

Manuscript received November 30, 2009. The associate editor coordinating the review of this letter and approving it for publication was F.-N. Pavlidou.

L. Cui is with the School of Information Technology Engineering, Tianjin University of Technology and Education, 1310 South Dagu Road (East of Liulin), Hexi District, Tianjin City, 300222, China (e-mail: cuilin.academic@gmail.com).

S. J. Koh is with the School of Electronic Engineering and Computer Science, Kyungpook National University, 1370 Sankyuk-dong, Buk-gu, Daegu, 702-701, Korea (e-mail: sjkoh@knu.ac.kr).

W. J. Lee is with the School of Electronic Engineering and Computer Science, Kyungpook National University, 1370 Sankyuk-dong, Buk-gu, Daegu, 702-701, Korea (e-mail: woojin@knu.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2010.06.092335

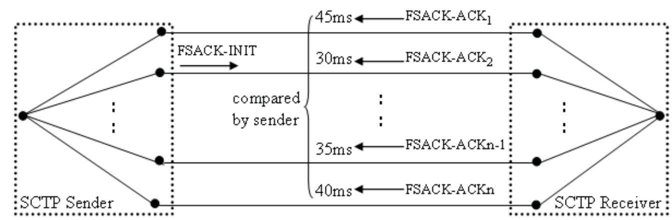

Fig. 1. Step 1 and 2 for configuration of the FSACK path.

In this Letter, we propose a Fast SACK (FSACK) scheme of SCTP, in which a receiver sends SACK chunks, as many as possible, to the sender over the fastest path among the available return paths. The proposed scheme can be applied to both SCTP [7] and CMT-SCTP cases [5], [6]. Different from [3] and [4], we aim at selection of the optimal return path only in this Letter because we argue that the optimal forward path strongly depends on its bandwidth in addition to its delay, and does not make sense for the CMT cases [5], [6] yet.

II. Proposed Fast SACK Scheme

In the proposed scheme, the sender determines the FSACK path with the minimum one-way delay from the receiver, and informs the receiver about its decision. Since then the receiver will transmit all the subsequent SACK chunks to the sender over the designated FSACK path. The FSACK path will be dynamically updated, depending on the network conditions.

A. Configuration of FSACK Path

In data transmissions, a sender transmits the data chunks to a receiver over the primary path (in SCTP) or multiple paths (in CMT-SCTP). In the proposed scheme, FSACK path can be determined in the course of data transmission as follows.

Step 1: Initially, the sender transmits a FSACK-INIT chunk over the primary path, as shown in Fig. 1, which contains a 4-byte timestamp (current time).

Step 2: On reception of the FSACK-INIT chunk, the receiver responds with the FSACK-ACK chunks over all the available return paths, as shown in Fig. 1. Once any FSACK-ACK chunk reaches, the sender can immediately designate the FSACK path as that one from which the fastest FSACK-ACK chunk arrives since all FSACK-ACK chunks have the same timestamp value of the FSACK-INIT chunk. The sender also sets/resets "FSACK-RTT-threshold" to the arrival time

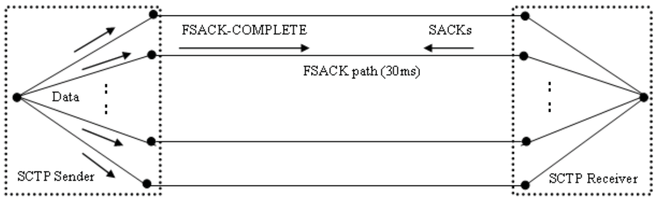

Fig. 2. Step 3 for configuration of the FSACK path.

Type = X or X+1	Chunk Flags	Length=8
Timestamp (4 bytes)		

Fig. 3. FSACK-INIT and FSACK-ACK chunks.

Type = X2	Chunk Flags	Length=8
IP Address of the FSACK path (In case of IPv4)		

Fig. 4. FSACK-COMPLETE chunk.

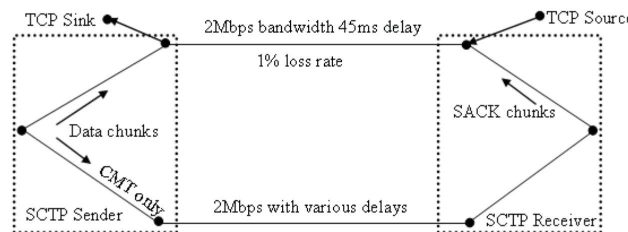

Fig. 5. Simulation topology for heterogeneous networks.

minus the timestamp value for asymmetric FASCK-INIT and FSACK-ACK paths (sometimes they might be symmetric too).

Step 3: After that, the sender informs the receiver about the FSACK path by sending the FSACK-COMPLETE chunk, as shown in Fig. 2 and 4. Once the FSACK path is determined, the receiver transmits all the subsequent SACK chunks to the sender over the FSACK path until the FSACK path is changed.

During data transmission, these three-step procedures can be performed repeatedly, as long as the sender realizes that the current asymmetric FSACK RTT is greater than the 'FSACK-RTT-threshold' value. This ensures that the dynamic network conditions are reflected on the FSACK path configuration. In addition, we will apply the *maximum time interval* for FSACK path update (e.g., 1 second), so as to guarantee that the FSACK path can be confirmed or updated at least once during a certain time period.

B. Extensions of SCTP

To apply the proposed scheme, we define the following new three types of SCTP chunks: FSACK-INIT, FSACK-ACK and FSACK-COMPLETE. The formats of FSACK-INIT and FSACK-ACK chunks are similar, as shown in Fig. 3, in which "X" represents the new chunk type number. The FSACK-COMPLETE chunk is used to inform the IP address of the determined FSACK path, as shown in Fig. 4.

III. Numerical Results

We implemented the proposed FSACK scheme on top of both SCTP and CMT-SCTP for performance evaluation over heterogeneous and homogeneous networks, using the ns-2 network simulator [8]. The simulation topology for heterogeneous networks, where all available paths may experience different propagation delays, is given in Fig. 5.

In figure 5, all links have the bandwidth of 2Mbps and the upper link has the end-to-end propagation delay of 45ms, while the end-to-end propagation delay of the lower link is variable. The other simulation parameters are configured based on the *sctp-cmt-2paths-64K* function for CMT-SCTP and the *sctp-multihome2-2Rtx1* function for normal SCTP. Both are already given in the *test-suite-sctp.tcl* of ns-2.30 [8] where the initial slow start threshold is set to 16000 bytes and each data chunk has a fixed size of 1468 bytes, without background traffic.

In the simulation suit, we lasted a file transfer application with 1% loss rate over the forward path for 100 seconds, and set the *maximum time interval* to 1 second for FSACK path update. The simulation result is shown in Fig. 6.

From the results, fist of all, we can see in the comparison of SCTP and SCTP with FSACK that when the end-to-end propagation delay of the alternative network (i.e., the lower path of Fig. 5) is smaller, the proposed scheme (SCTP with FSACK) outperforms the normal SCTP. However, with increase of the propagation delay, the proposed scheme tends to give nearly similar performance with the normal SCTP. This is resulted in due to dynamic SACK routing capacity of the proposed scheme. In particular, when the delay of the alternative network gets much smaller (e.g., less than 45 ms in Fig. 5), the proposed scheme will switch (i.e., reconfigure) the FSACK path to the alternative network in order to inflate its congestion window and/or retransmitting the lost segments more quickly than the existing scheme, while the normal SCTP uses the original primary path all along (i.e., the upper path of Fig. 5).

On the other hand, in the comparison of the CMT-SCTP and CMT-SCTP with FSACK, as the propagation delay of the alternative network is larger, the proposed scheme (CMT-SCTP with FSACK) outperforms the CMT-SCTP scheme

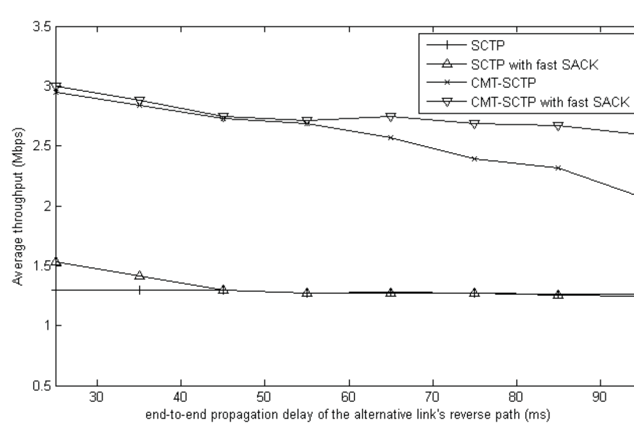

Fig. 6. Comparison of throughputs over heterogeneous networks.

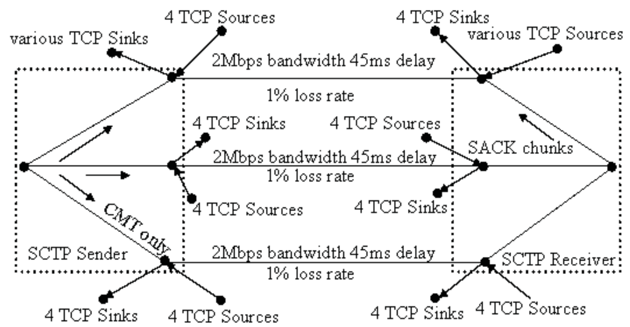

Fig. 7. Simulation topology for homogeneous networks.

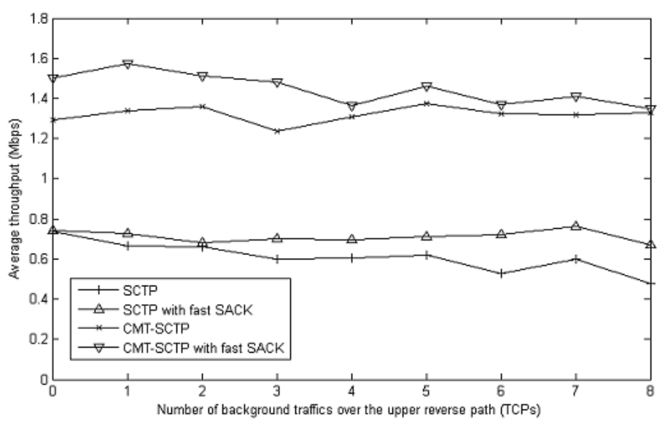

Fig. 8. Comparison of throughputs for homogeneous networks.

much more. This also benefits from the dynamic SACK routing capacity. That is, in the proposed scheme all SACK chunks are delivered to the FSACK path with a shorter delay, while the existing CMT-SCTP scheme sends SACK chunks through both heterogeneous networks without consideration on their delay difference.

We then compare the schemes' throughput performances over homogeneous networks. The related simulation topology is given in Fig. 7. In the figure, we employ three independent links between sender and receiver. Each link has the capacity of 2 Mbps, 1% packet loss rate and 45ms propagation delay. Each node has a queue buffer of 256 packets. To simulate dynamic network conditions, a various number of TCP traffics (ranged from 0 to 8) are generated over the upper reverse path. For each of the other forward and reverse paths, 4 TCP background traffics are randomly generated.

Notice that all background traffics are randomly generated by using FTP applications with 1500 bytes per packet, and thus the transmission delays and error rates of the forward and reverse paths will vary dynamically. The other parameters are set based on the *sctp-cmt-3paths-64K* function for CMT-SCTP and the *sctp-multihome3-3timeout* function for SCTP, which are also given in the *test-suite-sctp.tcl* of ns-2.30 [8].

For SCTP, we perform an FTP application for 100 seconds, and the *maximum time interval* is also set to 1 second for FSACK path update. Fig. 8 shows the simulation results with different number of TCP background traffics over the upper reverse path. From Fig. 8, specifically, we saw that for a single data flow the normal SCTP tends to experience severe performance degradation as background traffics increase along the primary reverse path (i.e., the upper link of Fig. 7), compared to the SCTP with FSACK scheme. This is because when the background traffics are lighter, both SCTP and SCTP with FSACK schemes will deliver most of SACK chunks to the same primary reverse path. However, once the background traffics get heavier, the proposed scheme changes its FSACK path to the alternative path with a shorter transmission delay, whereas SCTP uses the existing primary path all the time.

On the contrary, in case that the TCP background traffics are relatively low along the upper reverse path, performance of the existing CMT-SCTP scheme will get worse than that of the proposed CMT-SCTP with FSACK scheme because its receiver always sends SACK chunks to all of reverse paths in response to the arrival of data chunks, whereas the proposed scheme only delivers SACK chunks to the FSACK path with the shortest delay.

IV. Conclusion

This Letter proposed a fast SACK scheme for SCTP to enhance the throughputs of SCTP and CMT-SCTP schemes. From the simulation results, we saw that the proposed scheme can improve the transmission throughputs of SCTP schemes over both heterogeneous and homogeneous networks by effectively exploiting the fast SACK path. This implies that the proposed scheme has deep potential to enhance the transmission throughputs of SCTP schemes and warrants a further exhaustive study.

Acknowledgment

This research was partially supported by the MKE, Korea, under the ITRC support program supervised by the NIPA: (NIPA-2009-C1090-0902-0009).

References

[1] C. Barakat, *et al.*, "On TCP performance in a heterogeneous network: a survey," *IEEE Commun. Mag.*, vol. 38, pp. 40-46, Jan. 2000
[2] C. P. Fu and S. C. Liew, "A remedy for performance degradation of TCP Vegas in asymmetric networks," *IEEE Commun. Lett.*, vol. 7, pp. 42-44, Jan. 2003.
[3] A. Kelly, G. Muntean, P. Perry, and J. Murphy, "Delay-centric handover in SCTP over WLAN," *Trans. Automat. Contr. and Computer Sci.*, vol. 49, pp. 211-216, 2004.
[4] E. P. Ribeiro and V. C. M. Leung, "Minimum delay path selection in multi-homed systems with path asymmetry," *IEEE Commun. Lett.*, vol. 10, no. 3, pp. 135-137, Mar. 2006.
[5] J. Iyengar, *et al.*, "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths," *IEEE/ACM Trans. Networking*, vol. 14, no. 5, pp. 951-964, Oct. 2006.
[6] C. M. Huang and C. H. Tsai, "The handover control mechanism for multi-path transmission using Stream Control Transmission Protocol (SCTP)," *Computer Commun.*, vol. 30, no. 17, pp. 3239-3256, 2007.
[7] R. Stewart, Stream Control Transmission Protocol, IETF RFC 4960, Sep. 2007.
[8] Network Simulator (NS-2), http://www.isi.edu/nsnam.ns