Chunk Checksum of SCTP for Throughput Enhancement

Lin Cui and Seok Joo Koh

***Abstract*— Stream Control Transmission Protocol (SCTP) uses a checksum in the common header for each packet, by which a corrupted SCTP packet will be regarded as a loss. This Letter proposes a new "chunk checksum" scheme for SCTP, in which each data chunk contains its own checksum field to determine a corruption of data chunk. Simulation results show that the proposed scheme could improve the throughput of SCTP transmission in the networks with a high bit error rate.**

***Index Terms*— SCTP, checksum, corruption, throughput.**

I. Introduction

In Stream Control Transmission Protocol (SCTP) [1], each packet includes a 32-bit checksum in the common header. On reception of an SCTP packet, if the checksum field indicates a corruption, the receiver will discard an entire packet containing one or more data chunks. Moreover, this corruption will be regarded as a packet loss, and the sender will reduce the congestion window size, which tends to degrade the SCTP throughput, in particular, in the wireless networks with a high bit error (corruption) rate.

This Letter proposes a new chunk checksum scheme for SCTP, named CC-SCTP, in which an SCTP data chunk contains its own checksum field. The chunk checksum will be used to determine a corruption of data chunk at the receiver side. The purpose of this chunk checksum is to handle the chunk corruption differently from the loss, so that the congestion window size should not be reduced at the sender side in case of a chunk corruption.

II. SCTP Chunk Checksum

A. Chunk-based Checksum

In the CC-SCTP scheme, it is assumed that both the sender and receiver should agree to use the proposed chunk-based checksum in the association, which might be done by exchanging the associated information using SCTP INIT and INIT-ACK chunks in the association establishment. The details of such a signaling scheme are outside the scope of this Letter.

In the CC-SCTP scheme, a sender will construct a data chunk with an additional 32-bit checksum in the chunk header, as shown in Fig. 1.

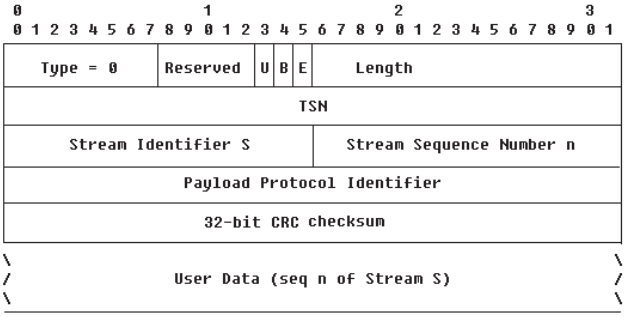

Fig. 1. Data chunk format with the 32-bit chunk checksum.

For each data chunk, a sender will calculate the chunk header checksum that covers the first 20-byte header fields of the data chunk. It is noted that this checksum is used to check the validity of the header fields of the data chunk, such as Transmission Sequence Number (TSN), Stream Identifier (SI), and Stream Sequence Number (SSN). For ensuring the integrity of the packet's common header which contains port numbers and verification tag information, the chunk checksum of the first data chunk should cover the 12-byte common header of the SCTP packet in addition to its own 20-byte chunk header.

B. Generation of SACK Chunks by Receiver

On reception of a SCTP packet, the receiver will first verify the whole SCTP packet by checking the overall checksum in the common header. If the packet is corrupted, the receiver will in turn verify the integrity of each individual data chunk contained in the packet by investigating the proposed chunk checksum.

Once the chunk checksum is proved to be invalid and it is the first chunk, the receiver will discard this packet, since the common header may contain some wrong information. Otherwise, if it is not the first one, the receiver discards the data chunk only and then continues to check the next one. On the contrary, passing the verification of the integrity of chunk header means that the corruption only occurs in the chunk data portion. Thus the receiver will record the corresponding TSN number of a "corrupted data chunk". This information will be delivered to the sender by using the subsequent SACK chunk. For this purpose, the SACK chunk needs to include additional fields to represent such TSNs. The detailed procedures are illustrated in Fig. 2.

Manuscript received February 6, 2006. The associate editor coordinating the review of this letter and approving it for publication was Prof. Iakovos Venieris.

L. Cui is with the School of Electronic Engineering and Computer Science, Kyungpook National University, 1370 Sankyuk-dong, Buk-gu, Daegu, 702-701, Korea (email: cuilin@cs.knu.ac.kr).

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 (email: sjkoh@knu.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2006.060184.

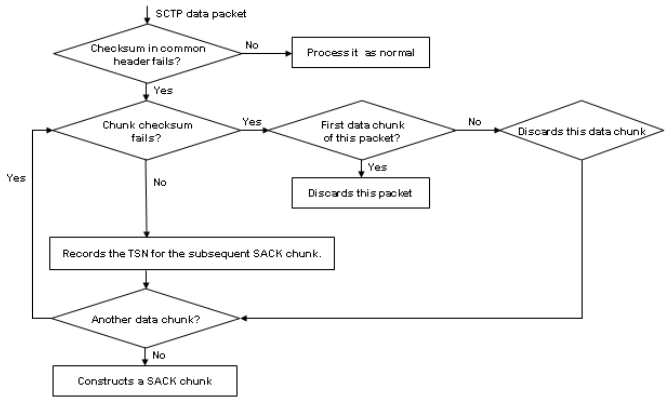

Fig. 2. The procedure of the chunk chechsum investigation.

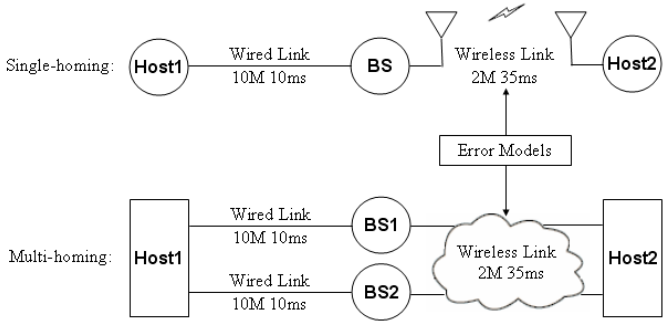

Fig. 3. Simulation topology.

C. Window Adjustment by Sender

In the conventional SCTP, the sender uses the two different retransmission strategies for a lost packet: timer-based retransmission and duplicate ACK-based fast retransmission, in which the congestion window size will be reduced in the retransmission phase. In the proposed CC-SCTP scheme, however, the "corruption" event (that will be reported by SACKs) is handled differently from the loss event, since a corruption itself indicates an explicit retransmission request.

When a corruption event is reported from the receiver, together with its associated TSN, the SCTP sender will retransmit the corresponding data chunk immediately to the receiver. It is noted in the corruption case that the sender may not decrease the congestion window size. With this immediate retransmission and congestion control strategy, the proposed CC-SCTP can improve the SCTP throughput performance.

III. Experimental Results

To evaluate the performance of the proposed CC-SCTP scheme, the ns-2 network simulator [2] is used with a simple test network, as shown in Fig. 3, in which two SCTP endpoints communicate over either a single path (single-homing) or two non-overlapping paths (multi-homing).

In the figure, the end-to-end paths are configured in the wired-cum-wireless manner. Each wired link has the link bandwidth of 10 Mbps and the transmission delay of 10 ms, whereas the wireless link is of bandwidth of 2 Mbps and transmission delay of 35 ms. In our simulations, every packet has a size of 1500 bytes and the queue size of each base station (BS) is set to 15 packets.

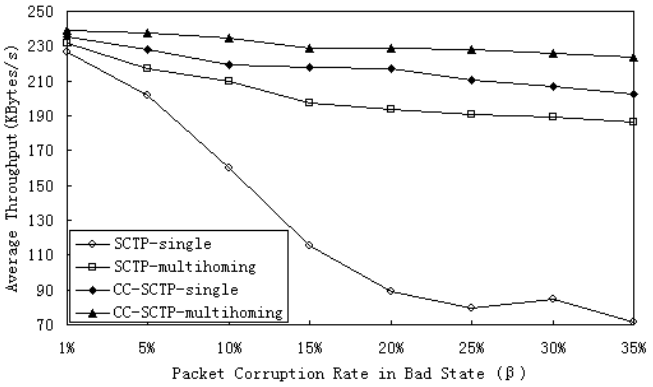

Fig. 4. SCTP throughputs for different β values.

To generate the chunk corruption events in the wireless environments, we employ the Gilbert model [3], and marked each corrupted packet with a corruption flag in the simulations. In the error model, the two states of "good" and "bad" are expressed in terms of transition probabilities p = 0.1 (from "good" to "bad") and q = 0.3 (from "bad" to "good"), with the average sojourn periods (2.5 seconds for "good" state and 0.5 seconds for "bad" state).

In the simulation, the Gilbert model is extended by imposing a certain corruption rate on "good" state as well as "bad" state. It is noted that in the original Gilbert model a corruption rate is given to "bad" state only.

It is assumed in simulation that an SCTP packet contains a single data chunk. In the "good" state, an SCTP packet will experience a corruption with the probability α (fixed to 0.0001), whereas in the "bad" state a SCTP packet will experience a corruption with the probability β (variable), where $\beta \gg \alpha$.

Furthermore, at the receiver side, in order to emulate the scenario where corruption occurs in packet's header portion, we apply the proportion of the header size (IP header + SCTP common header + SCTP chunk header) over the total packet size to the corruption events generated, and those corrupted chunks will be dropped without recovering their TSNs by the receiver of CC-SCTP scheme.

For each ns-2 experiment, we performed a file transfer application for 100 seconds. We measured the average throughputs (Kbytes/s) of the existing SCTP and CC-SCTP schemes for the single-homing and multi-homing cases.

A. Throughput for Different Corruption Rates

Figure 4 shows the average throughputs of CC-SCTP and SCTP for different corruption rates (β), ranged from 0.01 to 0.35, in the single-homing and multi-homing networks.

From the figure, we can see that the CC-SCTP schemes provide better throughput over the SCTP schemes for both the single-homing and multi-homing cases. It is noted in the SCTP single-homing case that the SCTP throughput gets worse drastically, as the corruption rate (β) increases. On the contrary, the CC-SCTP scheme gives better throughput, with the help of the proposed chunk checksum scheme. This is because the CC-SCTP can successfully avoid the adjustment of the congestion window size due to corruption. It is also

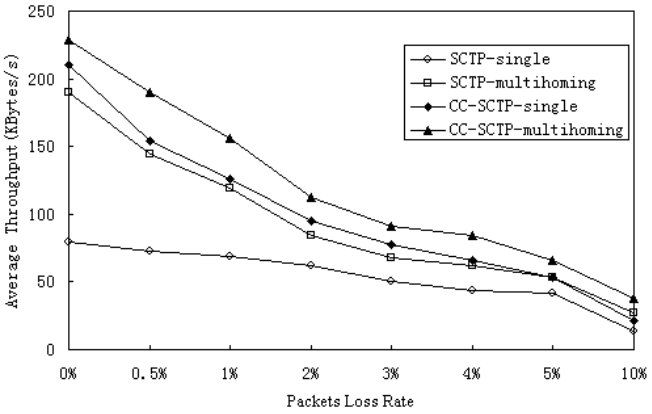

Fig. 5. SCTP throughputs for networks with packet losses.

noted that the performance of CC-SCTP is not sensitive to the corruption rate, compared to those of the SCTP.

By comparison of single-homing and multi-homing SCTP, the multi-homing SCTP provides better performance than the single-homing case, since the multi-homing SCTP can use the secondary path for retransmission of corrupted data chunks. Overall, the multi-homed CC-SCTP gives the best throughput over all the test cases.

B. Throughput for Different Loss Rates

We also compare the throughput of CC-SCTP in the networks with packet "loss" as well as packet corruption since wireless link is generally featured by both high packet loss rate and high packet corruption rate, compared to wired link. We apply a uniform random error (loss) model to generate packet losses for the primary wireless link, with a packet loss rate ranged from 0 to 0.1.

Figure 5 shows the throughputs of the SCTP in the networks with the different packet losses and the fixed packet corruption rate of 25% in "bad" state based on the modified Gilbert model.

In the figure, we can see that the proposed CC-SCTP scheme provides the better throughput than the standard SCTP for both the single-homing and multi-homing cases. In particular, when the packet loss rate is less than 2%, the proposed CC-SCTP scheme gives the significant performance gains, compared to the standard SCTP.

IV. Conclusion

In this Letter, we propose the data chunk-based checksum scheme, CC-SCTP, which is designed to enhance the SCTP throughput even in the network environment with a high bit error rate. From the simulation results, we can see that the proposed CC-SCTP scheme provides better throughput than the standard SCTP for both the single-homing and multi-homing cases in the networks with a high bit error or corruption rate. In particular, the proposed CC-SCTP schemes give the better performance gain in the multi-homing case. It is noted that the performance gain of the CC-SCTP scheme comes from the following features: 1) the proposed scheme can distinguish the chunk corruption from the chunk loss by using additional chunk checksum; 2) hence, the CC-SCTP scheme can avoid unnecessary halving the congestion window in case of packet corruptions.

Acknowledgment

This research was supported by the MIC, Korea, under the ITRC support program supervised by the IITA (IITA-2006-C1090-0603-0026).

References

[1] R. Stewart, *et al.*, "Stream control transmission protocol," IETF RFC2960, Oct. 2000.
[2] The Network Simulator (ns-2). Available: http://www.isi.edu/nsnam/ns/.
[3] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell System Technical Journal*, vol. 39, pp. 1253-1265, 1960.