

Improving Performance of SCTP over Broadband High Latency Networks

ISHTIAQ Ahmed
(ishtiaq@ais.sys.i.kyoto-u.ac.jp)

OKABE Yasuo
(okabe@i.kyoto-u.ac.jp)

KANAZAWA Masanori
(bwv147@i.kyoto-u.ac.jp)

Graduate School of Informatics
Kyoto University, Japan

Abstract

Stream Control Transmission Protocol (SCTP) is newly emerged protocol, which combined good qualities of TCP and UDP. It is a reliable message oriented protocol providing Multistreaming and Multihoming as well. The congestion control scheme of SCTP is more or less similar to that of TCP with the exception of the Fast Recovery algorithm. The performance of SCTP over the Internet and satellite links is improved as compared with TCP. However, the congestion control scheme of SCTP over high latency broadband networks needs further refinement in case of multiple packet losses on a link. We propose a new congestion control for SCTP and proved that performance of SCTP has significantly improved.

1. Introduction

The Stream Control Transmission Protocol (SCTP) is newly emerged protocol for the Internet. It was originally developed by the IETF [1] to carry out telephony signaling messages over IP networks. It is evolved as reliable, message oriented data protocol that provides Multistreaming and Multihoming. The good features of TCP and UDP are adopted to facilitate their respective multimedia applications.

SCTP has an advanced congestion control scheme that is used to recover from segment losses effectively and efficiently. The congestion control scheme of SCTP is also based on AIMD (Additive Increase Multiplicative Decrease) principle. The transport algorithms based on AIMD principle perform poorly over broadband high latency networks in case of multiple packet losses [2]. In this paper, we will focus on congestion control of SCTP and examine its Goodput performance over a broadband high latency network. The rest of the paper is organized as follows: Section 2 describe the congestion control mechanism of SCTP, Section 3 lays out the network topology, the modified congestion control algorithm is explained in Section 4, results are discussed in Section 5 and finally Section 6 is the conclusion this paper.

2. Congestion Control of SCTP

The advanced congestion mechanism of SCTP consists of three basic algorithms: 1) slow-start, 2) congestion avoidance and 3) Fast Retransmit. SCTP end points maintain three variables receiver advertised window (*rwnd*, in bytes), congestion window (*cwnd*, in bytes) and slow start threshold (*ssthresh*, in bytes) to regulates data transmission rate. SCTP requires an additional control variable *partial_bytes_acked* (*pba*) that is used during congestion avoidance phase.

Slow-start is used for data transmission into a network with unknown conditions or after a sufficiently long idle period and SCTP will try to determine the available network capacity. Slow-start is also used after repairing a segment loss detected by the retransmission timer. The data sender rate of the SCTP sender during slow-start is not greater than $2 * MTU$ of destination path. The SCTP receiver will generate an acknowledgement (called SACK in SCTP) for at least every second packet (not every second DATA chunk) received, and this SACK should be generated within 200 ms of the arrival of any unacknowledged data chunk. The value of *cwnd* is less than or equal to *ssthresh* during slow-start algorithm. When a SACK chunk arrives, the *cwnd* will be increased by the minimum value of these two conditions: 1) the total size of the outstanding data chunks acknowledged, and 2) the destination's path MTU. Thus, the *cwnd* is not exponentially increased as that of the case of classical TCP algorithms like TCP Reno. However, the packets injected into the network are exponentially increased during slow-start algorithm as that of TCP. The congestion avoidance algorithm will start in which if *cwnd* is greater than *ssthresh*, then *cwnd* will be incremented by $1 * MTU$ per RTT.

If a packet is lost in the network, four duplicate SACK chunks are required at the SCTP sender. Congestion window will be reduced to half of its current value. The Fast Retransmit will retransmit the missing data chunks bundled into same packet (multiple data chunks will be bundled only if their size is quite small as compared with MTU size of the association) to avoid retransmission timeout.

3. Network Topology

The network topology is shown in Figure 1. Two end nodes (labeled as S and D) are connected with each other through Drop Tail queue. The link bandwidth and delay of the link are indicated in the figure. We have used the network simulator ns-2 [3] to perform our experiments and SCTP module has been ported from Protocol Engineering Lab at university of Delaware [4].

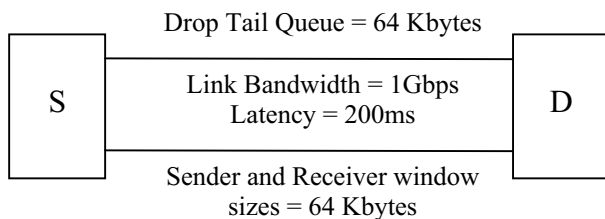


Figure 1. Simulation Topology

4. Modified Congestion Control of SCTP

It is a known fact that TCP performs very poorly over network with very high congestion window and latency due to multiple packet losses [2,5]. The congestion window is reduced to half of its current value; therefore a congestion window with a very high value will take long time to reach to its present position. A Scalable TCP algorithm is proposed to alleviate this problem [5] and similar approach is adopted in [2] as well. Scalable TCP proposed two modifications. During congestion avoidance algorithm for each acknowledgement received in a round trip time the congestion window is increased by

$$cwnd = cwnd + [0.01 * cwnd] \text{ ----- (1)}$$

And on the first detection of the congestion, the congestion window is reduced by the equation

$$cwnd = cwnd - [0.125 * cwnd] \text{ ----- (2)}$$

The selection of congestion window increase and decrease in equations (1, 2) is based on analytic design analysis of the networks possessing large congestion windows in [5]. The congestion control of SCTP follows the same congestion window reduction mechanism as that of TCP. Therefore, SCTP might behave similarly in the event of multiple packet losses. The new packets are injected into the network depending upon the size of the congestion window. The larger the congestion window is, the higher data input into the network will be. We thus modified the congestion control of SCTP as given by equation (2). The results are presented in the next section.

5. Results and Discussion

The Goodput (Number of Bytes received at the receiver) performance of original and modified SCTP is shown in Figure 2. The Goodput of modified SCTP is 1.7

Mbps whereas that of SCTP is 0.9Mbps. We conducted another experiment to show the effectiveness of modified SCTP over LAN and WAN. The error rate seen on the link is 0.001 and simulation time is 100 sec. The modified SCTP does not improve Goodput over LAN, however, the improvement is significant over networks having latency in the range of 20ms to 90 ms.

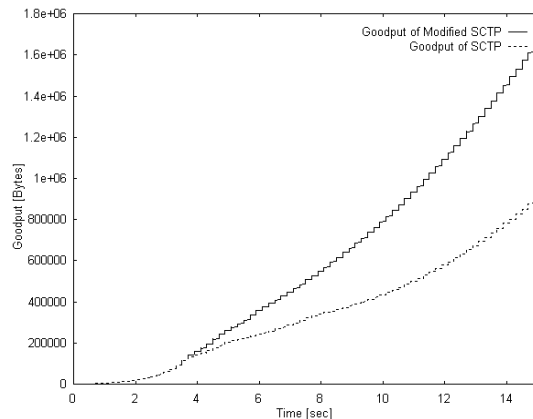


Figure 2. Goodput comparison

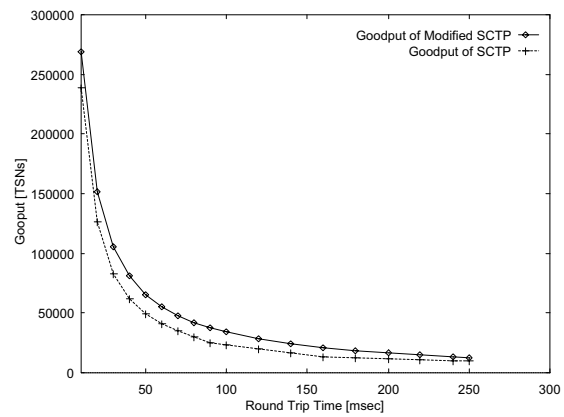


Figure 3. Goodput versus RTT

6. Conclusions

We have modified the congestion control mechanism of SCTP by adjusting the congestion window size and have proved that the performance of SCTP has improved significantly as compared with the original SCTP.

Reference

- [1] R. Stewart, Q. Xie, K. Morneault, and C. Sharp et. Al., "Stream Control Transmission Protocol" RFC 2960, Oct 2000.
- [2] S. Floyd, "HighSpeed TCP for Large Congestion Windows", IETF Internet Draft, Feb 2003.
- [3] The Network Simulator ns-2 at <http://www.isi.edu/nsnam/ns/>
- [4] SCTP Module for ns-2 available at <http://pe1.cis.udel.edu/>
- [5] T. Kelly, "Scalable TCP: Improving Performance in HighSpeed Wide Area Networks", Dec 2002, <http://www-lce.eng.cam.ac.uk/~ctk21/scalable/>.