# Performance Comparison of SCTP and TCP over Linux Platform

Jong-Shik Ha, Sang-Tae Kim, and Seok J. Koh

Department of Computer Science,
Kyungpook National University, Korea
{mugal1, saintpaul1978, sjkoh}@cs.knu.ac.kr

**Abstract.** Stream Control Transmission Protocol (SCTP) is the third transport layer protocol next to TCP and UDP. The SCTP provides some distinctive features over the TCP. This paper is purposed to compare SCTP and TCP in the performance perspective. We compare the throughput of SCTP and TCP for the three different test scenarios: the performance comparison of SCTP and TCP for the different size of the user input data for the socket system call, the analysis of the fairness under competition of SCTP and TCP traffic, and the performance comparison of the SCTP multi-homing and single-homing cases. From the results, it is shown that the SCTP provides better throughput over TCP for a larger user input data. We also see that the SCTP traffic tends to compete fairly with TCP and that the multi-homing SCTP provides better performance than the single-homing case.

## 1 Introduction

Stream Control Transmission Protocol (SCTP) is a new transport protocol next to TCP and UDP, which was standardized in the IETF [1]. Similarly to the TCP, the SCTP is a connection-oriented reliable transport protocol. Differently from the TCP, the SCTP uses the four-way handshake procedure for association establishment and the three-way handshake scheme for association termination. In particular, the SCTP provides the 'multi-streaming' and 'multi-homing' features.

Some previous studies [2, 3] include the performance analysis of the SCTP itself. In this paper, we focus on the comparison of the performance of SCTP and TCP in the viewpoint of the throughput over the Linux platform [4, 5]. The SCTP performance is analyzed for the three kinds of test scenarios: 1) performance comparison of SCTP and TCP for the different size of the user input data in the socket system call, 2) analysis of the fairness under the competition of SCTP and TCP traffics, and 3) performance gain of the SCTP multi-homing.

This paper is organized as follows. Section 2 briefly summarizes the features of the SCTP. In Section 3, we describe the three test scenarios for the performance comparison over the Linux platform. Section 4 shows the experimental results for the performance testing. Section 5 concludes this paper.

D.S. Huang, X.-P. Zhang, G.-B. Huang (Eds.): ICIC 2005, Part II, LNCS 3645, pp. 396 – 404, 2005.

## 2 SCTP Features

In this section we describe the distinctive features of SCTP, which include the SCTP association setup, association termination, multi-streaming and multi-homing features.

### 2.1 Four-Way Association Establishment

Differently from the 3-way handshake mechanism of TCP, the SCTP uses the 4-way handshake scheme for establishment of an SCTP association. Let us consider the two SCTP endpoints, A and B.

First, the endpoint A sends an SCTP INIT chunk to the endpoint B for initiation of an SCTP association. The endpoint B will respond with the INIT-ACK chunk to A, which contains the 'cookie' information for the security purpose. The endpoint A will then send the COOKIE-ECHO chunk to the B. The endpoint B completes the association establishment by sending the COOKIE-ACK chunk to the A.

It is noted in Figure 1 that this 4-way handshake scheme of SCTP is employed for preventing the so-called TCP SYN flooding. That is, the SCTP endpoint B will allocate the relevant kernel memory for the connection from the endpoint A, only after receiving the third COOKIE-ECHO chunk (after confirmation that the peer endpoint is a secure host).

### 2.2 Three-Way Association Terminations

The SCTP also uses the 3-way handshake mechanism for termination of an SCTP association for the purpose of the graceful close (shutdown). It is noted that the TCP provides the 4-way connection termination scheme, as shown in Figure 2.

To terminate an association, the endpoint A may send a SHUTDOWN chunk to the endpoint B. If there is no data to send, the endpoint B will respond with the SHUTDOWN-ACK chunk to the A. Finally, the endpoint A completes the association termination by sending the SHUTDOWN-COMPLETE chunk to the endpoint B.

Differently from the TCP, the SCTP does not support the so-called 'half-open state', wherein one side may continue sending data while the other end is closed.

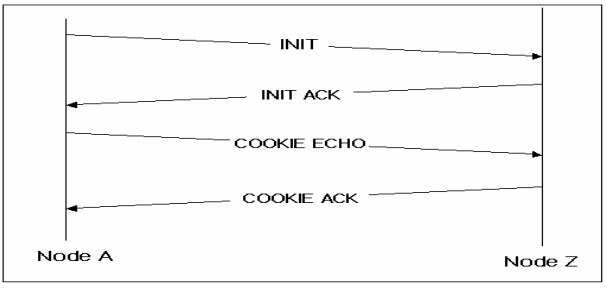



**Fig. 1.** SCTP four-way association establishment

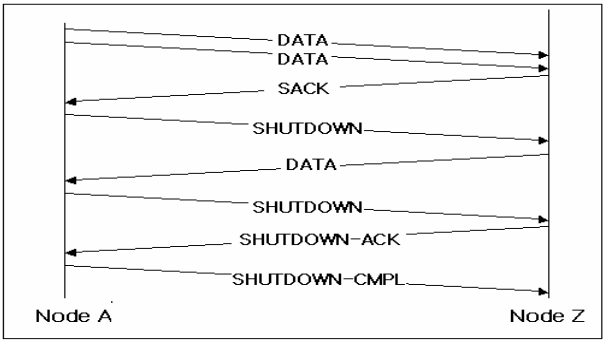



**Fig. 2.** SCTP three-way association terminations

### 2.3 SCTP Multi-streaming

The multi-streaming is a distinctive feature of SCTP. The SCTP user may assign each datagram to one of multiple streams within an association. In the association establishment phase, the two SCTP endpoints will exchange the number of available streams in the association each other. For each stream in the association, the SCTP increases the Stream Sequence Number (SSN) for the data chunk generated by the application user, as shown in Figure 3.

These SSN numbers are used by the receiver to determine the sequence of delivery. The SCTP performs in-sequence delivery per stream. This mechanism helps to avoid the head-of-line (HoL) blocking of TCP, since each stream data can be independently delivered to the peer endpoint within one association.

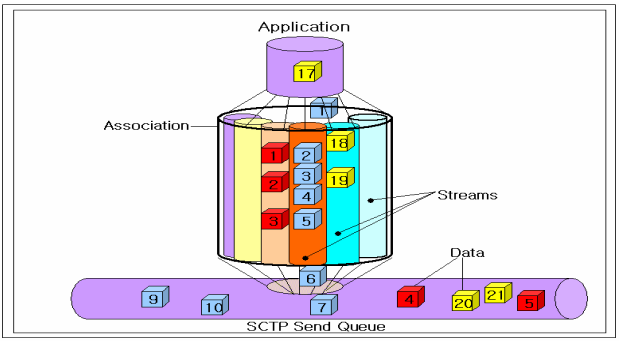



**Fig. 3.** SCTP multi-streaming

### 2.4 SCTP Multi-homing

From the multi-homing feature, the SCTP endpoint can use one or more IP addresses for data transport in the association, as shown in Figure 4.

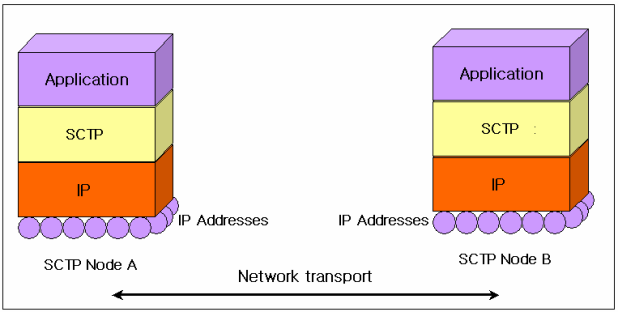



**Fig. 4.** SCTP multi-homing

The SCTP multi-homing feature can be used to protect an association from potential network failures by steering traffic to alternate IP addresses. During the initiation of an association, SCTP endpoints exchange the lists of IP addresses used at the remote endpoint. One of the listed IP addresses will be designed as the primary address. If the primary address repeatedly drops chunks, however, all chunks will be transmitted to an alternate address.

## 3 Test Scenarios

In this section, we describe the test scenarios employed in the experimentations for comparison of the SCTP and TCP performance.

### 3.1 Scenario 1: Different Size of User Input Data

The first test scenario is employed to compare the throughput of the SCTP and TCP for the different size of the use input data in the socket system call.

For the test purpose, a test network is configured as shown in Figure 5.

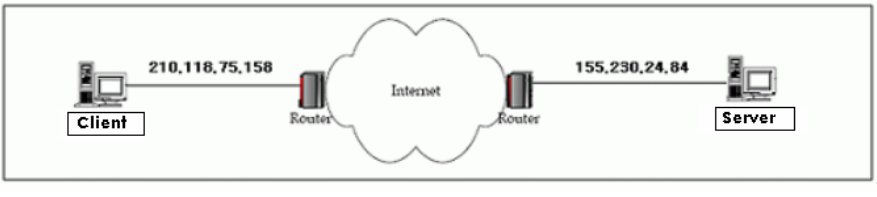



**Fig. 5.** Network configuration for Scenario 1

In the figure the client and server hosts are equipped with the Linux-Kernel 2.6.10 and LK-SCTP toolkit [4]. After establishing an SCTP association with the server, the client begins to download a file of 100 Mbytes from the server. As a performance metric, we measured the throughput of data transmission (i.e., the totally transmitted data bytes during the association period).

### 3.2 Scenario 2: Competition of SCTP and TCP Traffic

This scenario is tested to see how fairly the SCTP and TCP traffics compete in the network. Both the SCTP and TCP connections are established at the same time between the client and the server, as shown in Figure 6.

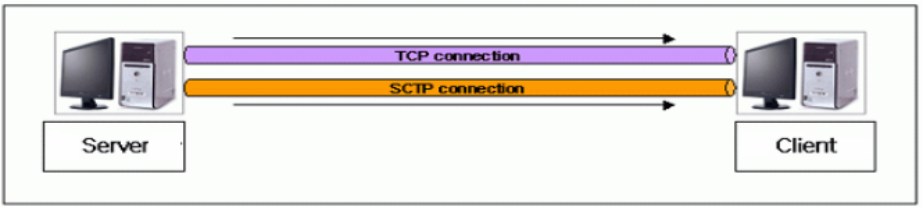



**Fig. 6.** Competition of SCTP and TCP traffic

For the two connections, we measured the traffic between the client and the server.

### 3.3 Scenario 3: Performance of SCTP Multi-homing

For the SCTP multi-homing, the test network is configured, as shown in Figure 7.


**Fig. 7.** Network configuration for SCTP multi-homing

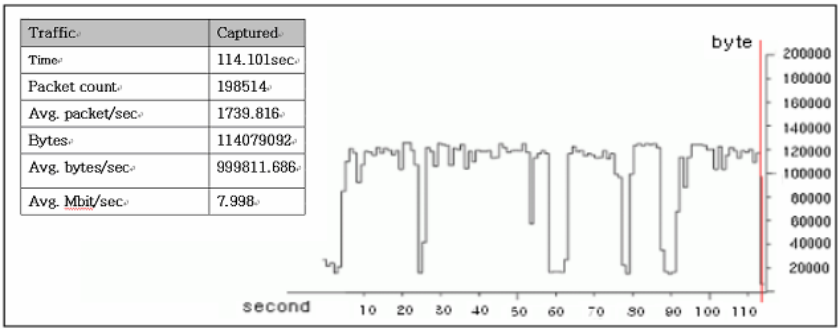
In the figure, the client is in the dual-homing state and uses the two different IP addresses for data packets and SACK packets, respectively.

## 4 Experimental Results

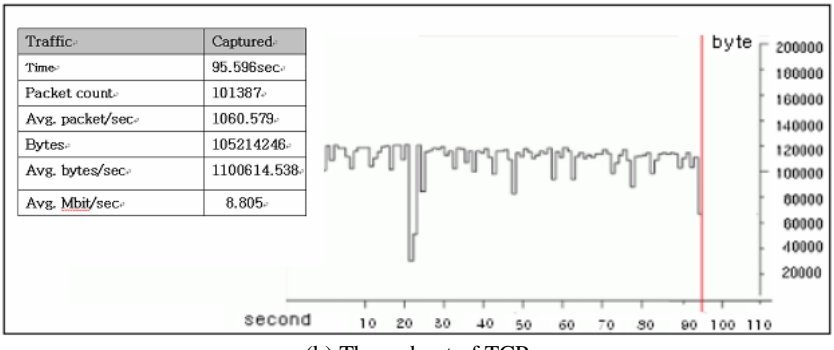
In this section, we discuss the results for the test experimentations of SCTP and TCP.

### 4.1 Results for Scenario 1

Figure 8 and 9 show the test results for the different sizes of user input data for each send() socket system call by using the 'ethereal' tool [6].




(a) Throughput of SCTP




(b) Throughput of TCP

**Fig. 8.** Results for the user input data size of 2048 bytes

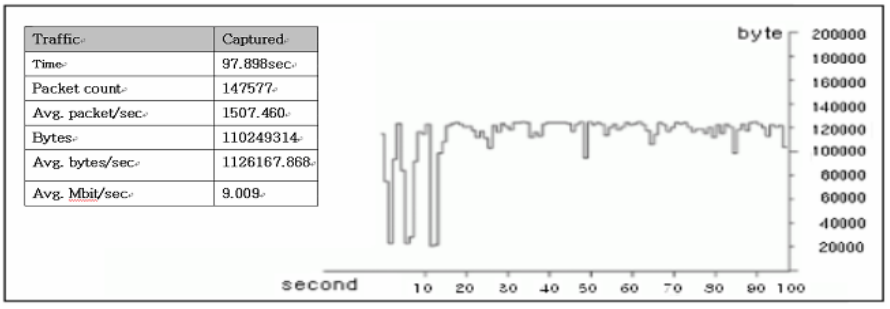
In Figure 8, we show the data packets (in byte) transmitted over the association period, for SCTP (Fig. 8(a)) and for TCP (Fig. 8(b)), in which the user input data of 2048 bytes are sent by the socket *send()* call.

In Fig. 8(a), we see that the SCTP transmits the total 198,514 packets and 114,079,092 bytes (including the data and control packets) over the association period of 114 seconds, which corresponds to the average throughput of 999,811 bytes per second.

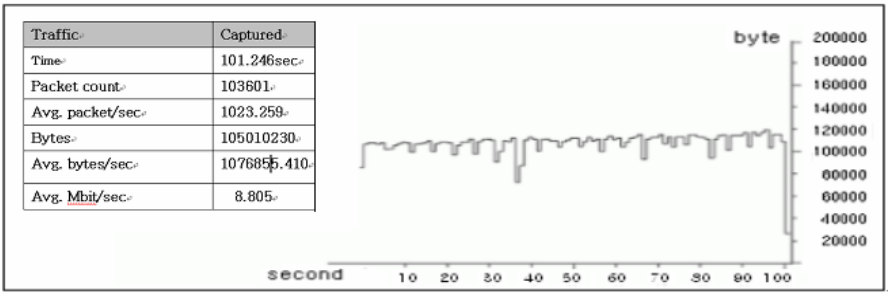
On the other hand, we see in Fig. 8(b) that the TCP sends 101,387 packets over the connection period of 95 second, with the average throughput of 1,100,614 bytes per second. In summary, from the figure we see that the TCP provides better throughput than the SCTP for the user input data of 2,048 bytes.

In Figure 9, we show the results of the throughput for SCTP (Fig. 9(a)) and for TCP (Fig. 9(b)) with the user input data of 8,192 bytes.

It is noted that the results in Figure 9 are different from those in Figure 8. Fig. 9(a) shows that the SCTP gives the average throughput of 1,126,167 bytes per second, whereas Fig. 9(b) shows that the TCP provides the throughput of 1,076,685 bytes per second, for the user input data of 8,192 bytes.




(a) Throughput of SCTP




(b) Throughput of TCP

**Fig. 9.** Results for the user input data size of 8192 bytes

From the results of Figure 8 and 9, it is interesting to note that the SCTP tends to provide better throughput performance over the TCP, when the size of the user input data for each socket system call gets larger. That is, the SCTP performance will benefit from the transport of the large bulk data, compared to TCP.

On the other hand, this performance gain of SCTP over TCP seems to come from the congestion control schemes associated [7, 8]. That is, the TCP uses the initial *Congestion Window* (*CWND*) as 1*MTU, whereas the SCTP starts from the *CWND* of 2*MTU. Overall, the SCTP tends to provide better throughput than TCP for the large-scale bulk data transport.

### 4.2 Results for Scenario 2

Figure 10 shows the results of the traffic traces for SCTP and TCP, in which the two SCTP and TCP connections are activated at the same time in the single computer.

From the figure, we see that the SCTP competes with the TCP for the data transmission under the same condition, in which the traffic generated by SCTP and TCP is almost equally distributed. The TCP connection completes the data transmission earlier than the SCTP, since the SCTP generates more data and control chunks.

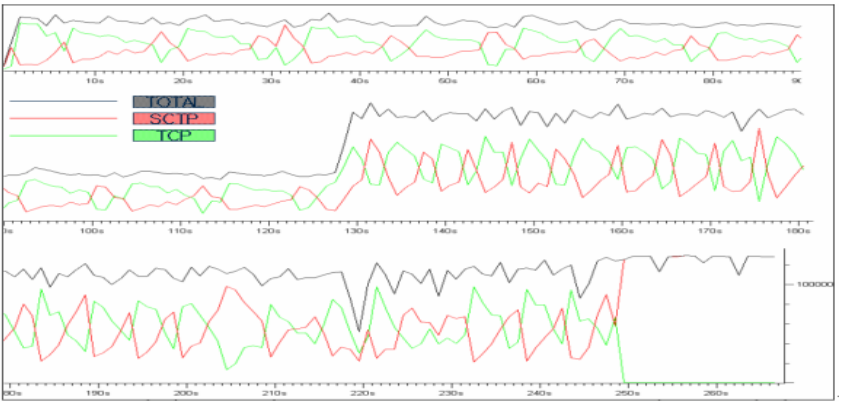



**Fig. 10.** Results for competition of TCP and SCTP traffic

### 4.3 Results for Scenario 3

Figure 11 shows the results of the SCTP single-homing and multi-homing association, as shown in Figure 7. It is noted in the multi-homing SCTP that the data and control SACK chunks are delivered over the different IP addresses [9].

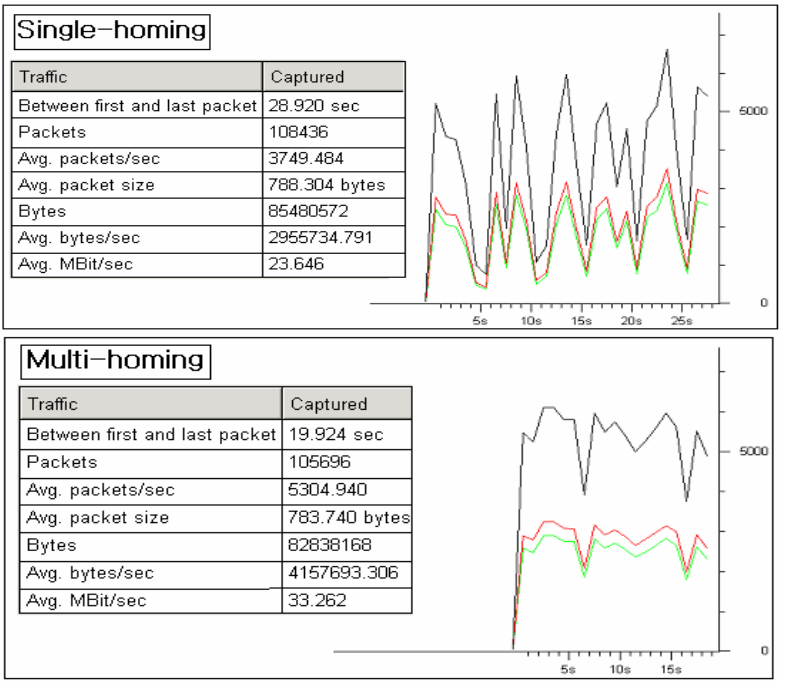



**Fig. 11.** Effects of SCTP single-homing and multi-homing

From the figure, we see that the multi-homing SCTP completes the data transmission earlier, with the better throughput of 4,157,693 bytes per second, than the single-homing SCTP (2,955,734 bytes per second).

It is clear from the results that the SCTP multi-homing feature can be used to improve the throughput of the data transmission. In this experiment, the SCTP control chunks are delivered using the different IP address from the SCTP data chunks.

## 5 Conclusion

In this paper, we have described the comparison of SCTP and TCP in the viewpoint of the throughput performance over the Linux platform. We compare the throughput of SCTP and TCP for the three different test scenarios: the performance comparison of SCTP and TCP for the different size of the user input data for the socket system call, the analysis of the fairness under competition of SCTP and TCP traffic, and the performance comparison of the SCTP multi-homing and single-homing cases.

From the results, it is shown that the SCTP provides better throughput over TCP for a larger user input data. We also see that the SCTP traffic tends to compete fairly with TCP, and that the multi-homing SCTP provides better performance than the single-homing case.

## References

1. Stewart, R., et al.: Stream Control Transmission Protocol. RFC 2960, October 2000
2. Jungmajer, M schopp and M. Tuxen.: Performance Evaluation for the Stream Control Transmission Protocol. IEEE ATM Workshop 2000, June 2000
3. Ravier, T., et al.: Experimental studies of SCTP multi-homing. First Joint IEI/IEE Symposium on Telecommunications Systems Research, 2001
4. Linux Kernel SCTP Project. Available from http://lksctp.sourceforge.net/
5. Stewart, R., et al.: Sockets API Extensions for Stream Control Transmission Protocol. IETF Internet Draft, draft-ietf-tsvwg-sctpsocket-10.txt, Feb. 2005
6. Ethereal, available from http://www.ethereal.com
7. Allman, M., et al.: TCP Congestion Control. RFC 2581, April 1999
8. J. Hoe.: Improving the Startup Behavior of a Congestion Control Scheme for TCP. ACM SIGCOMM, August 1996
9. Koh, S., et al.: mSCTP for Soft Handover in Transport Layer. IEEE Communications Letters, Vol. 8, No.3, pp.189-191, March 2004