

Performance Analysis of SCTP for Multiple Parallel Streams

Sangram Keshari Nayak, Sarojananda Mishra, Manoranjan Dash

Abstract: SCTP is a connection oriented general purpose transport protocol and applications presented in the transport layer for concept of streams. Different streams provide multiple end to end connections that enables independent send and receiver buffer at two end points. In this paper we performed performance measurement for simultaneous parallel streams. Performance of SCTP is closely examined for two parallel streams and considerable improvement of bandwidth utilization observed as compared to single stream.

Keywords : SCTP, TCP, UDP, DCCP, lksctp, NS2 tcl, awk, RTT.

I. INTRODUCTION

Following are two well known protocols TCP and UDP and also two emerging protocols SCTP and DCCP. Later two protocols are given less attention in deployment. TCP enables connection oriented reliable transmission, operates in layer 3. TCP has been described in RFC 793 [1] and a lot of changes has been proposed. First major significant change was made by Van Jacobson [2] incorporating congestion control. Moreover TCP supports flow control, error detection and in-order delivery. The UDP protocol enables transmission in an unreliable and unordered manner and it provides connection-less service. UDP has not been undergone significant changes and described in RFC 768 [3]. The Stream Control Transmission Protocol (SCTP) is a connection oriented reliable transport layer protocol on the top of the Internet Protocol (IP). SCTP has been described in RFC 2960 [4] and later updated in RFC 4960 [5]. Some applications like Internet telephony, online game and streaming audio requires timely delivery over reliability. So from this point of view Datagram Congestion Control Protocol (DCCP) has been introduced and described in RFC 4340 [6]. The recent past has experienced unexpectedly growing number in the Internet traffic and increasing use of Hyper Text Transfer Protocol [7]. Also transport layer protocol TCP/IP experiences extensive usage in communication at client and server ends. SCTP is another protocol which finds no difficulty in succeeding TCP, but the problem is that reprogramming is possible to

support TCP. This is somewhat easier to use a general protocol translation software without rewriting the existing applications. SCTP is a reliable transport layer protocol on top of the unreliable protocol IP. As compared to TCP, one association of SCTP may have multiple streams which are partially or completely ordered streams. This property of SCTP makes advantageous over TCP which exhibits head-of-line blocking problems. Also using SCTP provides many advantages like multi-homing and multi-streaming as discussed below.

II. SCTP OVERVIEW

SCTP is a connection oriented transport layer protocol which contains features of both TCP and UDP. It uses minimal message-oriented transport feature as in UDP and uses in-sequence transport and congestion control mechanism of TCP. Some important features of SCTP are discussed in the following section.

A.State

State is maintained in SCTP where as TCP is stateless. That means one side of connection can not send data while other end has closed its socket.

B.Reliable

Both TCP and SCTP have reliable transmission. It uses checksum and SACK to detect damaged or duplicate data and reorder the data allowing for faster transmission.

C.Message boundary

Each message is framed and message boundary is preserved. This is not available in TCP since it uses byte stream.

D.SYN flood attack

This occurs at the server end with target to server resources. It causes overload to the server. SCTP uses Security cookies against SYN flood attack making robust protocol from any kind of attacks.

E. Multi-homing

SCTP being a reliable protocol, Multi-homing in SCTP allows end points with multiple associations. It enables SCTP connection across more than one paths. Two hosts are connected with multiple sets of IP addresses. One being primary IP address and others as remote IP address. Connectivity is monitored seamlessly as when connection is lost with primary IP address, the connection is not aborted. This is because the SCTP connection migrates from one IP address to other.

Revised Manuscript Received on November 19, 2019

* Correspondence Author

Sangram Keshari Nayak, Department of Computer Science and Engineering, Indira Gandhi Institute of Technology, Sarang, India ,

Sarojananda Mishra , Department of Computer Science and Engineering, Indira Gandhi Institute of Technology, Sarang, India.

Manoranjan Dash, Faculty of Management Sciences, Siksha O Anusandhan (Deemed to be University), Bhubaneswar, India.

F. Multi-streaming

Multi-streaming in SCTP can be advantageous over data transmission. Various data types can be transmitted within the same association. This new approach has benefits of multiple connections of TCP. It also exhibits some benefits of multiplexing and demultiplexing applications of UDP.

III. RELATED WORK

Deployment of new protocol is a difficult matter from application point of view, since it requires application to support the new protocol at both ends client and server. Since TCP is the widespread and dominant protocol in the Internet world, it discourages for development of new protocols. Accordingly application developers also do not take any risk of using new protocols. Experimental work has been initiated in small change in transport layer protocol. The solution may lead to development of new protocol. Authors in [8] present that use of SCTP multi-streaming can be beneficial from many reasons. Some steps were initiated by Florian Niederbacher [8] to incorporate automatic multi-streaming feature, which also requires no extra effort for the network application developer. Here investigations were done to discuss advantages and disadvantages of replacing TCP connection by SCTP association. Also steps has been taken to study benefits of SCTP multi-streaming feature and portability of TCP based application to SCTP. One of the work initiated for protocol replacement is use of appropriate software “shim-layer” [9]. In this approach TCP to SCTP translation is done by the kernel space implementation. This is a standard module in FreeBSD [10] operating system. Being a kernel module, it requires some programming knowledge and is somewhat more complex. Existing protocol can also be replaced using appropriate software. This is available as “withsctp” [11] tools in user terminal available in linux operating system as LKSCTP-tool package [11]. The Linux Kernel Stream Control Transmission Protocol (lksctp) project is a software with intension to translate TCP to SCTP in the Linux kernel. This enables protocol support for SCTP. It introduces security at startup, multi-homing and message based communication. It uses one to one mapping of translation that corresponds to every TCP connection. Using this software multi-homing feature is enabled. *Withsctp* is flexible to use as it runs in operating system address space, hence requires no kernel programming knowledge. Hence it can also be implemented to different Linux systems easily. So *withsctp* tool is attached to higher operating system level as shown in the figure 1. *Withsctp* starts as a terminal command with the application which is less complex to implement. Preservation of former protocols like TCP is required during connection setup. So deployment of new protocol is a big challenge from the compatibility reason. In this regard authors in [12] describe transition of HTTP services from TCP to SCTP. Here required algorithms are described and investigated. Still it requires close examination of the properties and parameters. Another different approach was proposed by authors in [13]. They described a connection establishment method for SCTP.

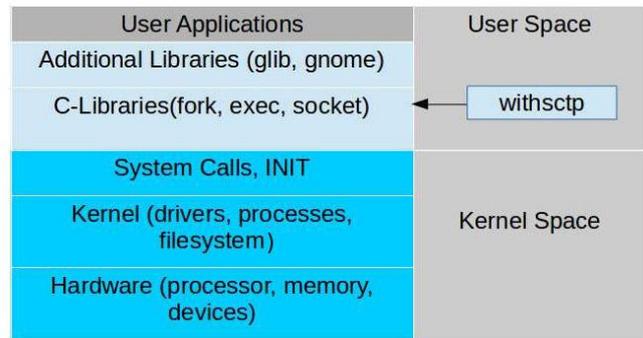


Figure 1. Position of withsctp in Linux kernel

IV. SCTP MULTI-STREAMING

End users are now experiencing a growing number of multimedia traffic usage. However these are limited to available bandwidth and comfortable throughout. So improvement of performance is dependent on transport layer services. One of the available innovative services is SCTP multi-streaming. A stream in a SCTP association is the logical channel established between two endpoints. The association is unidirectional and all ordered messages are delivered in sequence. Multi-streaming in SCTP association separates logical flow of data into different independent streams. It separates the data packets into logically different channels. Different data packets are received at receiver end within same SCTP association. This property allows applications to identify different flows of data. The responsibility of flow of data is managed by transport layer since application layer can no longer open more than one end to end connections to the same host. Each stream is delivered independently and has its own Stream Sequence Number (SSN) used for determining sequence of delivery. But sometimes it happens that one stream may be blocked for waiting next in-sequence message delivery. In the mean time delivery from other stream may proceed. Normally number of streams are decided during SCTP association set up. This number is negotiated at the receiver end. SCTP normally performs ordered delivery, but sometimes it may bypass the sequenced delivery and deliver the received message as soon as received. Thus head-of-line blocking problem does not happen in case of using multiple streams. From various discussions described above another proposal for protocol independent Internet transport API has been proposed by Stefan Jorer [14], that offers services that the application programmers need to see. It also describes a much simpler alternative to be used.

V. EXPERIMENTAL SETUP

This section presents discussion of performed measurements, which were made to obtain various results and evaluate the SCTP network under Linux. As the idea of network performance intended to Internet, some problems might occur due to SCTP, as all the middle box softwares do not support SCTP. Hence setup was made without Internet as it is not appropriate for practical measurements since it inherits some side effects.

That means background noise is to be minimized and no additional processes were assumed to be running. Consequently practical measurements were carried out taking two desktop computers as client and server. These computers were having same hardware and software configurations. Another additional computer was configured as Linux router.

A.Experimental setup details

Experiment was been carried out taking the network testbed with hardware configuration as given in table 1. The experiment has been made on CPU of Intel® Core™ i3-4005U CPU with 1.70GHz × 4 having four channel and 4GB memory. Both the hosts run under Linux operating system Ubuntu 14.04 [15] with basic configuration. The detailed softwares used for the basic configuration are described in table 2.

Table 1. Hardware configuration

Switch	Netgear Prosafe 5 port fast Ethernet switch
Network interface A	Qualcomm Atheros Lite-On Communications Inc Device
Network interface B	Qualcomm Atheros Lite-On Communications Inc Device
Ethernet controller	Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller

Table 2. Software specification

Operating system	Ubuntu 14.04
Kernel version	3.4.10
Network emulator	Netem kernel component(already enabled)
LKSCTP versions	Lksctp-tools 1.0.9
Network measurement tool	Iperf [16] 2.0.5

B.Simulation

This section presents experiments conducted using network simulator NS-2 [17] and University of Delaware’s SCTP module [18] for NS-2. This module is in wide use and provides a reliable baseline for SCTP functionality. We used NS-2 2.35 for simulation purpose. Our transport is SCTP and we measure quality through delay, packet loss and throughput. We found 1 tcl script and run over NS. With NS-2, traffic is with constant bit rate, simulation was carried out to obtain trace file. We use some awk [19] scripts to run on trace files to generate the expected results. A message is tracked when it is enqueued from application at one host and delivered to application of another host. To effectively investigate the number of streams that affects throughput, delay and bandwidth, we simulate it 100 times to get the result.

C.Performance evaluation

In this experiment we create a link between two hosts, a sender and receiver. We establish SCTP association between two streams in both direction. In order to simulate in real network, the test bed is designed with three computers installed Ubuntu Linux [15]. Both end hosts must be enabled with lksctp [11] and middle one with network emulator netem that generates constant path delay. Traffic sent through this network are affected by network delay.

1) Performance measurement with iperf

Iperf [16] is a terminal command can be used for evaluation of various traffic. Performance estimation was made using iperf. Since this software is used to measure TCP and UDP performance, SCTP support is not implemented. However protocol translation can be carried out with iperf in combination of withsctp [11] software. Experiments were carried out using standard test case, default window size being 1024K and port 8000 for 10 secs. From various analysis the graph is obtained by using Xgraph utility for Linux systems. This is a plotting program which can be used to create graphic representations of simulation results.

VI. RESULT AND DISCUSSION

In this section we discuss the simulation results. Figure2 shows the results of bandwidth values for single and two number of streams. with different RTT (Round Trip Time), obtained from ping command. We note bandwidth values is 5.09Mbps being constant for single stream but it is 9.90Mbps for two streams. Since multiple streams are independent streams, monitoring of data transmission revealed that SCTP does not use full bandwidth in case of multiple parallel streams and bandwidth utilization is done efficiently.

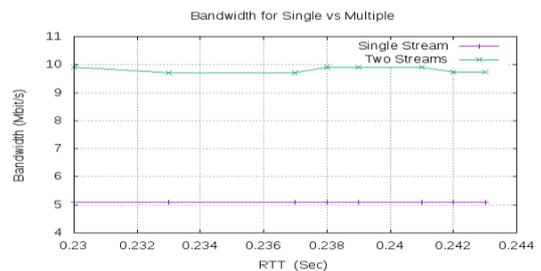


Figure 2. Bandwidth comparison

Throughput measurement results of this experiment is shown in figure 3 . Multiple streams data are shown in dotted lines. Maximum throughput is observed as 0.867Mbps in case of single stream and the corresponding value is 5.7Mbps for multiple streams. Comparison result observed as huge gain in SCTP performance for multiple streams.

Performance Analysis of SCTP for Multiple Parallel Streams

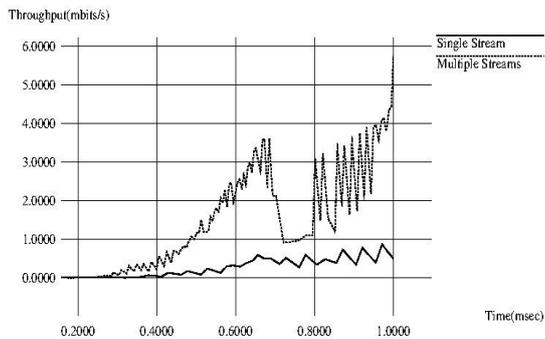


Figure 3. Throughput comparison of streams

Latency in the network is the difference in time received from packets. Figure 4 shows the latency vs time graph. The delay is 12.0ms for two streams and 5.0ms for single stream. Delay is shorter for single stream than multiple streams. For stream oriented protocols transmission overhead is desired to be reduced. Hence possible overhead might cost more for higher delay. Not only flow control is affected by path delay but also congestion control affects it. Thus delay may occur due to congestion control algorithm. This occurs also due to error detection functionality. This may be a disadvantageous, but it is overcome by higher throughput.

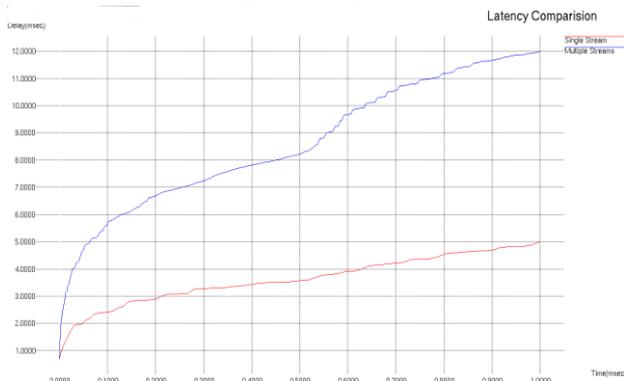


Figure 4. Latency for single vs multiple streams

VII. CONCLUSION

Our experimental result confirm that more number of parallel streams are beneficial in terms of bandwidth utilization. It enhances higher throughput and shares same end to end path, thereby reducing number of connections. Thus SCTP with two parallel streams performs better than two independent streams. The major gains of multistreaming may be more predominant when transferring multiple files. The consequence would be a fairer network bandwidth sharing for parallel transmission in a faster way.

REFERENCES

1. J. Postel. Transmission Control Protocol. RFC 793 (Standard), September 1981. Updated by RFCs 1122, 3168.
2. V. Jacobson. Congestion avoidance and control. In SIGCOMM '88: Symposium proceedings on Communications architectures and protocols, pages 314-329, New York, NY, USA, 1988. ACM.
3. J. Postel. User Datagram Protocol. RFC 768 (Standard), August 1980.
4. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream Control Transmission Protocol. RFC 2960 (Proposed Standard), October 2000. Obsoleted by RFC 4960, updated by RFC 3309
5. R. Stewart. Stream Control Transmission Protocol. RFC 4960 (Proposed Standard), September 2007.

6. E. Kohler, M. Handley, and S. Floyd. Datagram Congestion Control Protocol (DCCP). RFC 4340 (Proposed Standard), March 2006.
7. T. Berners-Lee, R. Fielding, H. Frystyk, Hypertext Transfer Protocol - HTTP/1.0. RFC 1945, IETF, May 1996.
8. Florian Niederbacher. Beneficial gradual deployment of SCTP. Master thesis at University of Innsbruck, February 2010. <http://www.flori.bz.it/sctp/download/thesis.pdf>.
9. Ryan W. Bickhart. Transparent tcp-to-sctp translation shim layer. Master's thesis, University of Delaware, 2005. URL <http://www.cis.udel.edu/~amer/PEL/poc/pdf/BickhartMStthesis.pdf>.
10. FreeBSD. URL <http://www.freebsd.org/>.
11. The Linux Kernel Stream Control Transmission Protocol Project - LKSCTP. URL <http://lksctp.sourceforge.net/>.
12. D. Wing, A. Yourtchenko, and P. Natarajan. Happy Eyeballs: Successful Introduction of New Technology to HTTP. INTERNET-DRAFT Expires on January 3, 2010, July 2009. URL <http://tools.ietf.org/html/draft-wing-http-new-tech-00>.
13. B. Ford and J. Iyengar. Breaking Up the Transport Logjam. Seventh ACM Workshop on Hot Topics in Networks, August 2007. URL www.bford.info/pub/net/logjam.pdf.
14. S. Jorer, "A Protocol-Independent Internet Transport API", 2010. http://home.ifi.uio.no/michawe/teaching/dipls/stefan_joerer.pdf
15. Ubuntu Linux. URL <http://www.ubuntu.com>.
16. IPERF. URL <http://iperf.sourceforge.net/>
17. Network Simulator ns-2 available at <http://www.isi.edu/nsnam/ns>
18. A Caro and J. Iyengar. ns-2 SCTP module. <http://pel.cis.udel.edu>.
19. [ns] awk/perl script to calculate throughput/delay/jitter from trace files <http://mailman.isi.edu/pipermail/ns-users/2004-September/044655.html>