RESEARCH ARTICLE OPEN ACCESS

Simulation of SCTP Multi-Streaming using Protocol Independent Socket API

Sangram Keshari Nayak [1], Sarojananda Mishra [2]
[1],[2] Department of Computer Science and Engineering
Indira Gandhi Institute of Technology, Sarang - India.

ABSTRACT
With introduction of new features such as multihoming and multi-streaming in Stream Control Transmission Protocol (SCTP) [1], it has become a promising protocol like a general purpose transport protocol. SCTP has gone considerable number of improvements and it has become a reliable protocol. Research works have been done to investigate capabilities and performance of SCTP. Multi-streaming is an interesting property that makes higher throughput as it reduces head-of-line (HOL) blocking. In this paper we tested SCTP performance in protocol independent way.
Keywords: -SCTP, TCP, multi-streaming, multi-homing, HOL

I. INTRODUCTION

In the past years SCTP has undergone a number of developments among the available transport layer protocols such as TCP [2] and UDP [3]. SCTP is a reliable transport protocol. It provides more robustness than TCP and provides message boundaries preservation like UDP. It provides more advantageous features like multi-homing and multi-streaming. Despite many benefits and features, it not more common in the sense of using the protocol by the developers. This is due to the problem of taking the chance of challenge. The challenge is of taking the risk of new protocol. So there must be a introduction of benefit to the end user of using a new protocol. This becomes a problem as the OS developers and applications developers must implement the new protocol. So there happens a restriction of using new protocol. It is a difficult task as all the users at all endpoints and developers would modify the functionalities and operating systems. Second alternative is to use a protocol translator as a layer. so it requires the introduction of services and requirements in stead of specific protocols. This is the basis of this paper using a translator with requirement and services. The way of using the underlying layer as translator makes using the new transport capabilities in a protocol independent way.

II. RELATED WORK

Main problem of new protocol is the deployment issues. Many scientists [4] have worked in the area of simple change of protocols at transport level to get a benefit of performance or new features. One of the translation software is the withsctp. It was developed in the intension of porting the TCP applications to SCTP in a simple way and to benefit from advanced features. Since TCP is the most frequently used transport protocol in the Internet, its replacement will be of most interest. SCTP has at least two distinct features which lead to an improved functionality compared to TCP are multi-homing and multi-streaming. Some researchers [5] have tried to access the protocol in a protocol independent way. Some important properties hav been investigated ignoring some unwanted services. Some basic services has been presented and the performance has been evaluated. Socket API for SCTP [6] has been chosen among available APIs. Socket API is the simplest that provides abstractions to be used easily. As per the design specifications of jorer [4], protocol features for TCP, UDP and DCCP [7] have been studied and design of the proposed concept has been discussed. The approach has been started with investigation of services that the protocols provide. Some of the features have been included and also excluded in the proposed implementation. In this paper we tried to implement the multi-streaming features with simulation by extending the proposed work done by Jorer [4].

III. GENERAL PROPERTIES OF TCP

This section describes the general properties of TCP that are relevant in this study.

A. Connection oriented

TCP is a connection oriented protocol meaning that a connection must be established before any data can be sent.

B. Bidirectional

A TCP connection is bidirectional, data can be sent by both peers using a single connection.

C. Congestion Control

TCP has a congestion control functionality which monitors the transmission and tries to adapt to the networks capacity.

The retransmission of lost packets is tightly coupled with the congestion control.

IV. GENERAL PROPERTIES OF SCTP

This section presents some properties of SCTP that have been found relevant when comparing with TCP.

A. Message Oriented

SCTP's data transfer is message oriented, as opposed to the byte stream oriented TCP.

B. Congestion Control Capability

SCTP has as TCP a congestion control functionality which monitors the transmission and tries to adapt to the networks capacity. SCTP's congestion control is made as similar as possible to TCP but there are differences.

C. Multi-stream

Multi-streaming feature is only supported by SCTP.

V. SCTP PERFORMANCE

A. Simulation Scenario

In this Section we provide a detailed analysis of the experimental results performed with the NS2 [8] simulator in order to show how networks benefit from the multistreaming functionality implemented in SCTP. To achieve such purpose,we present a comparison of SCTP behavior versus TCP. To conducting the experiments, a simple simulation has been designed using NS-2 Network simulator. NS-2 is a free software simulator that is very suitable for the protocol simulation. NS-2 is a discrete-time simulator whose development began in 1989 with the development of REAL Network Simulator. Probably one of the main reasons for its success is the fact that the distribution has General Public License (GPL) condition that drives the free development of the same. For this work , the latest version a network simulation was NS 2.35 installed on Ubuntu 14.04 [9] machine with core i3 (2.2 GHz) of processor. A set of experiments was carried out using five constant bit rate CBR, SCTP and TCP traffic sources. In this setup, the node started transmission with predefined delays to simulate the real environment. A simulation scenario has been considered to measure throughput for both of SCTP and TCP separately. The network consists of ten source node, ten destination node along with two routers. The five source SCTP node would generating FTP over SCTP traffic destined for the SCTP node on the other portion of the network, while the CBR over UDP generated by the rest of the nodes would be used to simulate the actual networking environment. Network is setup to measure a packet loss rate of SCTP is chosen for the current scenario. Similar Scenario is applied to the simulation topology as in Figure 1. The Programs run with corresponding nam files and trace files obtained from simulation. We used some awk [10] scripts to obtain desired data. Performances are measured and corresponding graphs obtained using xgraph utility.

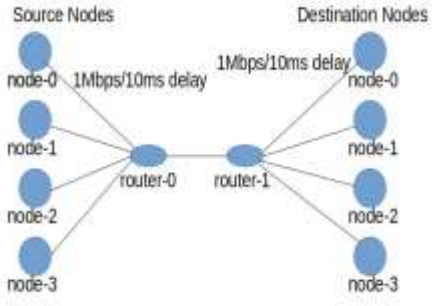

Fig. 1. Simulation Topology

VI. RESULTS AND DISCUSSION

Graph obtained from the simulation is shown in figure 2. It shows throughput (mbits/s) for corresponding time (msec). The graph yields major observation on throughput differences. It is clear from the graph that higher throughput obtained for SCTP as compared to TCP. SCTP shows steady graph over time, where as TCP graph is not a steady and ranges within 1Mbps only.

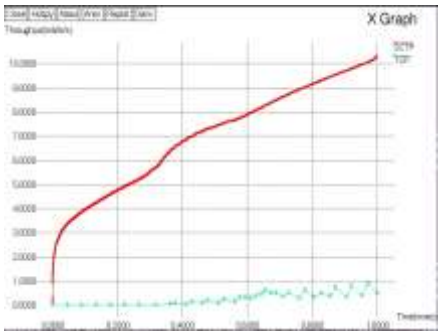

Fig. 2. Throughput SCTP vs TCP

VII. CONCLUSION

From the discussion of result, it is known that we tried to show the protocol performance using socket API. Multi-streaming feature has been included in the program and performance has a good sign of benefit for SCTP over TCP. It is also clear that the protocol independent program works

well for multi-streams. It has a indication that the socket API is a best choice for testing protocols under our discussion.

REFERENCES

[1] 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. sRFC 2960(Proposed Standard), October 2000. URL http://www.ietf.org/rfc/rfc2960.txt. Obsoleted by RFC 4960, updated by RFC 3309.

[2] J. Postel. Transmission Control Protocol. RFC 793 (Standard), September 1981. Updated by RFCs 1122,3168.

[3] J. Postel. User Datagram Protocol. RFC 768 (Standard), August 1980.

[4] S. Jorer, A Protocol-Independent Internet Transport API, 2010. URL http://hfi.uio.no

[5] Florian Niederbacher. Beneficial gradual deployment of SCTP. Master thesis at University of Innsbruck, February 2010. http://www.flori.bz.it/sctp/download/thesis.pdf.

[6] R. Stewart, K. Poon, M. Tuexen, V. Yasevich, and P. Lei. Sockets API Extensions for Stream Control Transmission Protocol (SCTP).

[7] E. Kohler, M. Handley, and S. Floyd. Datagram Congestion Control

Protocol (DCCP). RFC 4340 (Proposed Standard), March 2006 [8] Network Simulator ns-2 available at http://www.isi.edu/nsnam/ns.

[9] Ubuntu Linux. URL http://www.ubuntu.com.

[10] awk/perl script to calculate throughput/delay/jitter from trace files http://mailman.isi.edu/pipermail/ns-users/2004-September/044655.html