Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 49(63), Fascicola 1, 2004

Performance Analysis of Stream Control Transmission Protocol

Flavius Copaciu[1], Virgil Dobrota[1], Tudor Blaga[1], Bogdan Moraru[1]

Abstract – SCTP is a new Transport Layer protocol recently standardized by IETF and a possible replacement of TCP. In this paper, we describe SCTP and one of its implementation. Then we evaluate the performances of the protocol in an emulated network test-bed. The scenarios included bandwidth limitation, variation of the user message size, activation and deactivation of Nagle's algorithm and a throughput comparison between a TCP connection and a SCTP association. Based on our results, we make several recommendations regarding SCTP usage and its ability to function as a TCP replacement.
Keywords: SCTP, TCP, Transport Layer, UDP

I. INTRODUCTION

In the last years, we have seen strong efforts aimed at the integration of telephony and data networks. Although the services offered by these two networks were initially different, they are currently evolving towards a convergent digital network, based on NGN (Next Generation Network) concept [1]. The main features are related to packet-switching transfer and broadband capabilities with end-to-end QoS and transparency. The fixed and mobile networks based on IP will provide convergent services, independent to the lower layers technologies. Other features are related to the interworking with legacy networks via open interfaces, by decoupling the service provision from network.

Unfortunately, the choice of transporting the voice over an IP-based network is not enough to obtain the desired convergence of data and telephone networks. The signaling system, used to establish, monitor, terminate and supervise all telephone calls, is playing a key role. All digital telecommunications networks nowadays are governed by SS7 (Signaling System No. 7). It runs over a logical network that is separated from that one used for user data, although the two networks may share the same physical links. SS7 requirements regarding delay, data loss and out of order reception are much more stringent than the voice requirements. Therefore, the transport of SS7 signaling over IP has encountered serious problems, as it was detailed in [2].

In order to solve these problems a new working group, namely SIGTRAN (Signaling Transport) has been formed in IETF and a new Transport Layer protocol has been proposed: SCTP (Stream Control Transmission Protocol). SCTP has been design to transport signaling messages; it provides end-to-end flow and error control, sequenced delivery of messages within multiple data streams and takes advantage of multi-homing in order to provide better fault tolerance.

This paper studies the performances of a SCTP implementation, provided by Open SS7, within different test scenarios. They included bandwidth limitation, variation of the message size and activation and deactivation of Nagle's algorithm. A throughput comparison between a TCP connection and a SCTP association is also discussed. In section 2, we present SCTP and its advantages over UDP and TCP. Section 3 describes the tools, network configuration, methodology and test scenarios of our experiments. Finally, section 4 and 5 present the results of the experiments, our conclusions and further work.

II. THE STREAM CONTROL TRANSMISSION PROTOCOL

SCTP is a Transport Layer protocol, like TCP and UDP, proposed by IEFT [3]. The main task of the new protocol is to provide reliable data transmission between two end-points, over an IP-based network. For a long time TCP and UDP were the only transport protocols endorsed by IETF. The new coming protocol has been adopted because it provides solutions to problems that plagued the other two competitors

The first attempts to transport telephony signaling over IP have used TCP. Unfortunately the limitations have become obvious very soon:

- TCP provides reliable data transmission, based on a 16-bit checksum. However, this mechanism is considered obsolete by the modern standards, as it

[1] Technical University of Cluj-Napoca, Communications Department, 26-28 George Baritiu Street, 400027 Cluj-Napoca, Romania, Tel +40-264-401816, Fax: +40-264-597083, E-mail: {Flavius.Copaciu, Virgil.Dobrota, Tudor.Blaga, Bogdan.Moraru}@com.utcluj.ro

does not fulfill the requirements of the SS7-based digital network.

- It provides ordered data transmission, which is not always required for telephony signaling. The last one is sending signaling messages that must be received in a strict order only if they belong to the same call. When TCP detects that a data segment was lost, all segments following the lost one will be buffered until the lost segment is retransmitted from the source. This situation is known as HOL (Head-Of-Line) blocking and may lead to delays greater than the ones acceptable by a telephone signaling system.

- TCP is byte oriented and it does not preserve message boundaries. Unfortunately, this preservation is very important for a signaling system.

- It is vulnerable to the series of attacks. One of the most common types of attack is known as denial of service and protection against it is very important in order to ensure the reliability.

Another approach to signaling transport over IP has taken into consideration UDP. This protocol has major limitations that make it useless for this task: it does not provide reliable data transmission, does not guarantee orderly data reception and makes difficult any implementation of congestion control mechanisms. However, there is a major advantage when considering UDP: it is already implemented in every operating system. Therefore, an application running on top of UDP and taking care of all its limitations may be deployed faster than the other one included in the kernel space and running on top of IP.

To solve the requirements imposed by the transport of telephony signaling IETF has proposed SCTP. This protocol was initially design for PSTN (Public Switched Telephone Network) signaling over IP networks. Fortunately, its new features and capabilities to overcome the limitations of TCP and UDP have recommended it as a general use Transport Layer protocol. The most important improvements and characteristics are the following [4]:

- *Support for multi-homed hosts*: a host is multi-homed when it can be addressed by multiple IP addresses. This could be done either by having a host with multiple network interface cards (each with its own IP address), either by having a single network interface card with multiple IP addresses assigned to it. SCTP uses multi-homing in order to provide path redundancy and to improve reliability. In practice, only the first case may generate advantages for SCTP comparing to TCP/UDP.

- *Logical streams of data inside an association*: ordered data reception is only provided inside a stream and data loss only delays delivery of messages belonging to the same stream. By using streams, the HOL blocking problem was solved.

- *Highly reliable data transmission*: the first specification required an ADLER-32 checksum. However, it has been proved that ADLER-32 does not work well in the case of small packets and ADLER-32 was replaced with CRC-32c [5]. All new implementations of SCTP use CRC-32c and some of them provide ADLER-32 for compatibility with older implementations.

- *4-way handshake algorithm*: it includes cryptographic algorithms to protect against denial-of-service attacks. User data can be bundled during the third and fourth message exchange sequence.

- *Latest congestion control algorithms*: in addition, selective acknowledgements are mandatory for SCTP. This provides better performances in cases of congested networks. However, due to the multi-homing nature of SCTP further research is currently carried out.

- *The SCTP association is equivalent to TCP connection.* A TCP connection could be seen as a relation between the source port and IP address and the corresponding destination port and IP address. In a similar way, a SCTP association is in fact a relation between a source port and a list of source IP addresses and the corresponding destination port and a list of destination IP addresses.

III. METHODOLOGY

A. SCTP implementation

OpenSS7 provided the SCTP implementation used by our experiments [6]. OpenSS7 offers an open source implementation of SS7 stack for Linux, where SCTP is a part of this stack (two types of implementations). The first SCTP implementation is based on Linux STREAMS. This leads to a better integration into the SS7 stack (also based on Linux STREAMS) but it involves an extra overhead. Because of the performance concerns, this solution was not chosen during the trials.

The second implementation is a kernel space implementation and provides better performances than user space solution. It is based on a patch for a fresh Linux 2.4.18 kernel. The patched kernel was customized to the hardware configuration of the machine and then compiled. For SCTP we have used the default setting that came from OpenSS7. Detailed socket usage information is available in

`/proc/net/sockstat`. The implementation does not support IPv6, so all our experiments were limited to IPv4.

B. Test bed topology

All experiments have been done using a single machine, running a Linux 2.4.18 SCTP patched and enabled kernel, the NIST Net emulator, the server and client test programs. The logical topology is depicted in Fig. 1. There were three entities: the SCTP server, the NIST Net router and the SCTP client.

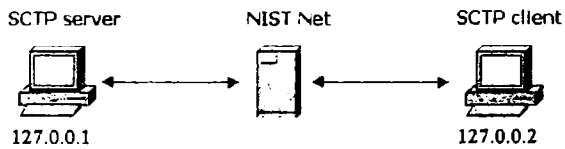

Fig. 1. Network topology

NIST Net is a WAN emulator designed for Linux machines and developed at National Institute of Standards and Technology, USA [7]. It is a software tool that can be used to emulate real IP based network behavior. A router running NIST Net software has the ability to generate adjustable packet delays, losses and duplication according to a specified probability and bandwidth limitation. The software tool has the ability to simulate different network characteristics of the complex topologies.

C. Test scenario

First, the NIST Net emulator is configured according to the performed experiment. After proper configuration, the NIST Net module is loaded and the emulator is started. From this moment, all traffic will be influenced according to the configuration file.

The trials involved two pairs of client-server programs included within the OpenSS7 SCTP patch: *test-sctps* and *test-sctpc* versus *test-tcps* and *test-tcpc*. *test-sctps* runs a SCTP server and waits for a client to connect. *test-sctpc* is a client that connects to the SCTP server. As soon as the SCTP association is establishes the server and client application exchange data according to the option specifies at the start. The output of the server and client application is logged to a file, used for data analysis. The SCTP traffic generated has been captured using *tcpdump* and later analyzed using *Ethereal*. For all experiments, only one stream has been defined in the SCTP association.

The experiments were the following:

- The delay introduced by the emulator was varied, taking the values: 0, 50, 100, 500 and 1000 ms;
- Several sizes for user messages were taken: 100, 200, 400, 600, 800 and 1000 bytes;
- Trials were carried out with and without enabling Nagle's algorithm within the server and the client applications.

In order to make a throughput comparison of SCTP and TCP similar experiments were performed using the other pair of programs *test-tcps* and *test-tcpc*. These programs behave similar to *test-sctps* and *test-sctpc*, but make use of a TCP connection instead of a SCTP association. They are almost identical, because the differences between the TCP's API and the one-to-one SCTP's API are minor [8].

IV. DATA ANALYSIS AND RESULTS

Four types of experiments were performed, all of them on the same machine, over the local loop interface. NIST Net was employed in order to limit the link to 100 kbps.

The first experiment analyzes the behavior of SCTP when used in conjunction with different user messages sizes. The higher levels send data to SCTP in the form of user messages. The size of the user message may vary and SCTP will split larger messages in order to conform to maximum MTU (Maximum Transmission Unit) and may bundle together several smaller messages and protocol data. This experiment evaluates the performance of SCTP when the user message size varies from 100 to 1000 bytes (100, 200, 400, 600, 800 and 1000 bytes). An overview of the results is provided in Fig. 2.

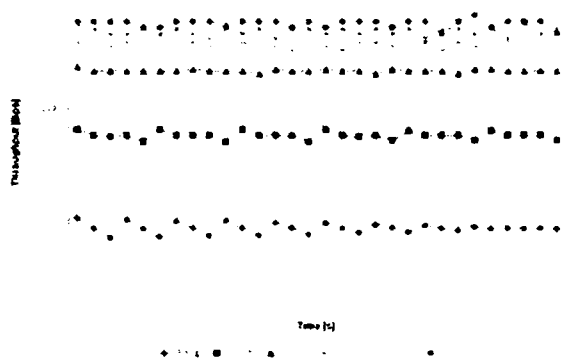

Fig. 2. SCTP throughput for different sizes of user messages

The throughput of the SCTP association increases with the size of the user messages. This behavior is explained by the decrease of the overhead introduced by the protocol per unit of useful user data.

The second experiment investigates the behavior of the SCTP association over links with different delays. Using NIST Net we were able to set the delay of the link to 0, 50, 100, 500 and 1000 ms. The existing delay over the local loop was ignored.

We can see in Fig. 3 that for packages with 100 bytes of user data the throughput decreases as the delay

increases. A similar behavior can be seen when using larger packages with 1000 bytes, as in Fig. 4. This throughput decrease is greater for small size user messages and the overall performances are increased when using larger user messages sizes.

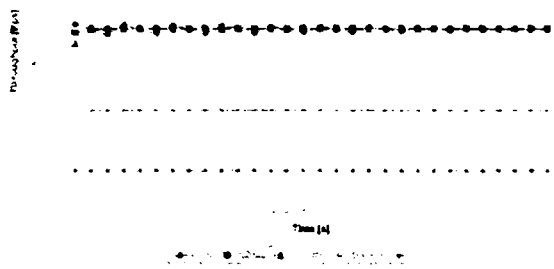

Fig. 3. SCTP throughput for various link delays and 100 bytes of user messages

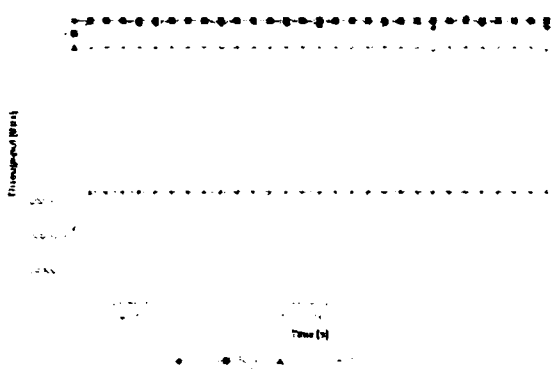

Fig. 4. SCTP throughput for various link delays and 1000 bytes of user messages

The third experiment looks at the influence of the Nagle's algorithm over SCTP performance. According to Fig. 5, the activation of the Nagle's algorithms results in a decrease of performances. The size of the packet does not affect the decrease of performance.

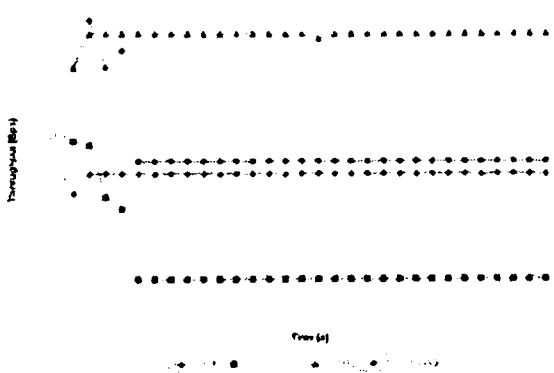

Fig. 5. The influence of Nagle's algorithm on SCTP throughput

The last experiment compares the throughput of TCP and SCTP. The results show comparable performances for a packet size of 1000 bytes (see Fig. 6) and better TCP performances for a packet size of 100 bytes (see Fig. 7).

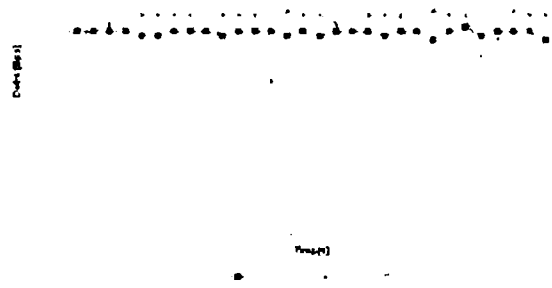

Fig. 6 SCTP and TCP throughput for 1000 bytes of user messages

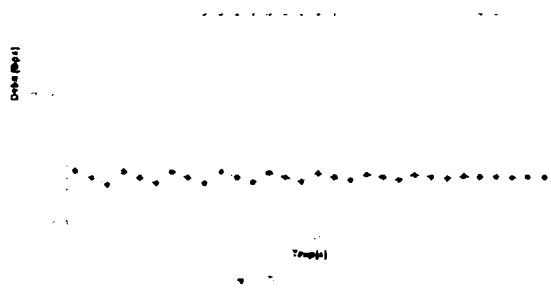

Fig. 7 SCTP and TCP throughput for 100 bytes of user messages

V. CONCLUSIONS AND FURTHER WORK

The first experiment showed that SCTP provides better performances when working with large user messages. The best results are achieved when the size of message is close to or greater than MTU. SCTP is a good choice for applications involving a large amount of data (*ftp* or *http*). Obviously, it is not recommended for applications that transfer small amount of data, such as *telnet*.

During the second experiment, SCTP performances were better for low delay links, but similar to those of other Transport Layer protocols, like TCP.

Nagle's algorithm, designed to reduce the number of (small) segments sent, was known for its behaviour to introduce delays and to decrease the performances of TCP. In the case of SCTP, the same behavior was observed. The recommendation is to disable Nagle's algorithm in order to get a better performance.

Following the trials presented in this paper, we concluded that SCTP has the potential to become the

successor of TCP, involving the best congestion control algorithms available and offering solutions to TCP problems. However, in terms of throughput, the SCTP association has, in average, 30% less than a TCP connection, for small packet size (100 bytes). Performance is similar for packet sizes close to usual MTU (greater than 1000 bytes). As the work is under progress, we are confident that future SCTP implementations will surpass the overall TCP performances.

Further work will be carried out in order to study the new SCTP implementation from the 2.6 series of Linux kernels [9] and based on IPv6.

REFERENCES

[1] http://www.NGN2004.com

[2] L. Ong, I. Rytina, M. Garcia, H. Schwarzbauer, L. Coene, H. Lin, I. Juhasz, M. Holdrege, C. Sharp, "Framework Architecture for Signaling Transport", *RFC 2719*, 1999

[3] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson, "Stream Control Transmission Protocol", *RFC 2960*, 2000

[4] S. Fu, M. Atiquzzaman "SCTP: State of the Art in Reserch, Products, and Technical Challenges", *IEEE Communications Magazine, April 2004, Vol. 42 No. 4*, 2004, pp. 64.

[5] J. Stone, R. Stewart, D. Otis, "Stream Control Transmission Protocol (SCTP) Checksum Change", *RFC 3309*, 2002

[6] http://www.openss7.org/

[7] http://snad.ncsl.nist.gov/itg/nistnet

[8] W. R. Stevens, B. Fenner, A. M. Rudoff, *UNIX Network Programming. Volume 1: The Sockets Networking AP*, Third Edition, Addison Wesley, 2003.

[9] http://lksctp.sourceforge.net/