THE 12th INTERNATIONAL CONFERENCE on ADVANCED COMMUNICATION TECHNOLOGY

ICT for Green Growth and Sustainable Development

Phoenix Park, Korea
Feb. 7–10, 2010

PROCEEDINGS
Volume I

IEEE Catalog Number CFP10561-PRT
ISSN 1738-9445
ISBN 978-89-5519-145-5

Organizers

Electronics and Telecommunications Research Institute(ETRI)
National Information Society Agency(NIA)
Global IT Research Institute(GIRI)

Sponsors

IEEE Communications Society(IEEE ComSoc)
IEEE Region 10 and IEEE Daejeon Section
Korean Institute of Communication Sciences(KICS)
IEEK Communications Society(IEEK ComSoc)
Korean Institute of Information Scientists and Engineers(KIISE)
Open Standards and Internet Association(OSIA)
Korean Communication Commission(KCC)

Session 3A: Wireless Communication Technology (III)

0966

Session 3B: Wireless Sensor Network (III)

Session 3C: IT Service Technology (I)

Analyze and Evaluate the performance of SCTP at transport layer

Tran Cong Hung, Ph.D. (Posts & Telecommunications Institute of Technology, Viet Nam)

E-mail: conghung@ptithcm.edu.vn

Tran Phu Khanh, Eng. (Posts & Telecommunications Institute of Technology, Viet Nam)

E-mail: phukhanh@gmail.com

Abstract - **Stream Control Transmission Protocol (SCTP) is first designed by IETF as a reliable transport protocol to transport SS7 signaling messages over IP networks. But its advanced features which are not provided by TCP or UDP show that SCTP is able to support a wider range of applications than just for signaling transport [4].**

As with TCP, SCTP is a reliable transmission protocol with congestion control and flow control, it is also a connection-oriented protocol with selective acknowledgement. Besides, it is a protocol which improves overall protocol security with four-way handshake association establishment to reduce vulnerability to DOS attacks; it provides framing mechanism and unordered service as UDP to preserve message boundaries; it also provides heartbeat mechanism to keep track of endpoint status and reach ability to detect network error more quickly; especially it is also a multi-homing protocol to improve robustness to failure, its multi-streaming feature or also called partial ordering can provide higher data rate, reduce delay by eliminating the Head-Of-Line blocking problem at the receiver in TCP.

We analysis advanced features and prove the capability of SCTP to show that it can replace TCP/UDP through test-beds [3]

Our paper "Analyze and evaluate the performance of SCTP at transport layer" is divided into the following main parts: The first part present "Introduction". The second part present "Background of SCTP". The third part present "Simulation". The fourth part is "Conclusion".

I. INTRODUCTION

SCTP, a transport protocol providing acknowledged, error-free, non-duplicated transfer of messages, has been proposed to be an alternative to UDP and TCP. The multi-streaming and multi-homing features of SCTP are especially attractive for applications that have stringent performance and high reliability requirements.

In this paper We show key features and enhancements of SCTP over TCP/UDP, explain why SCTP instead of TCP/UDP. Next I defined test-bed architectures and test scenarios to show these features of SCTP. Then, with some performance metrics and network impairment parameters, we measured and showed the advantages of SCTP over TCP/UDP. We also compared the performance of SCTP and TCP as the transport for HTTP

II. BACKGROUND [2],[15]

II.1. SCTP packet format

TCP provides a byte-stream data delivery service, whereas SCTP provides a message-oriented data delivery service. The SCTP packets always begin with an SCTP common header, a minimal structure that provides three basic functions:

- Source and destination ports: together with the IP addresses in the IP header, the port numbers identify the association to which an SCTP packet belongs.
- Verification tags: Vtags ensure that the packet belongs to the current incarnation of an association.
- Checksum: This computed value maintains the entire packet's data integrity.

The remainder of an SCTP packet consists of one or more chunks, concatenated building blocks that contain either control or data information. This format differs from TCP and UDP packets, which include control information in the header and offer only a single optional data field.

Like TCP, SCTP is a connection-oriented protocol, and then an SCTP association has three phases: association establishment, data transfer and association shutdown.

SCTP uses a four-way handshake in which a cookie mechanism establishes an association that prevents blind SYN attacks

SCTP assigns each chunk a *transmission sequence number* (TSN). A TSN is unique within an association and independent of the stream on which the chunk is being sent. TSNs only need to be associated with data chunks. Each data chunk has a different TSN, assigned incrementally, to maintain reliability and proper function of the flow and congestion control algorithms.

Unlike TCP's byte-stream service, SCTP preserves the boundaries of application-layer messages, similar to

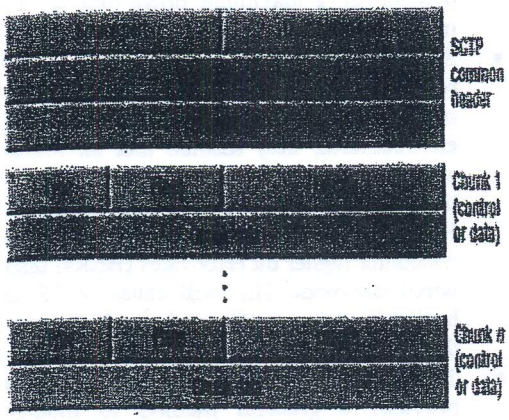

Figure 1: SCTP Packet format

UDP. When an application has a message larger than the destination path MTU, SCTP fragments the message into multiple data chunks, which can be sent in separate packets.

SCTP assigns the same Stream Sequence Number (SSN) to all the data chunks associated with a single application message, it helps the receiver in the reassembly process.

The association shutdown phase is a three-way handshake, as opposed to TCP's four-way handshake. This three-way handshake does not allow half-closed connections, in which one end point shuts down while the other end point continues sending new data. Either one of the end points engaged in the association can initiate the shutdown.

II.2. Enhanced features of SCTP over TCP and the issues of SCTP

- Multihome

  SCTP takes benefit from redundancy at network layer by providing redundancy at transport layer. A multihomed SCTP end point can bind to multiple IP addresses when that end point initializes an association. Currently SCTP uses multi-homing only for redundancy, not for load balancing.

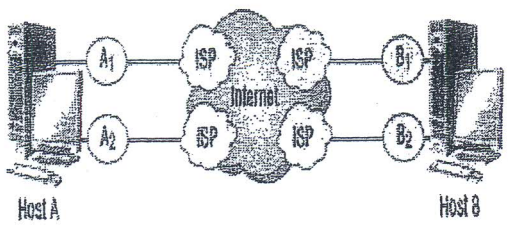

Figure 2: SCTP multihomed hosts

SCTP keeps track of each destination address's reachability through two mechanisms: ACKs of data chunks and *heartbeat* chunks – control chunks that periodically probe a destination's status. According to RFC2960, if six consecutive timeouts occur on either data or heartbeat chunks to the same destination, the sender concludes that destination is unreachable and chooses an alternate destination IP address dynamically.

- Multistreams

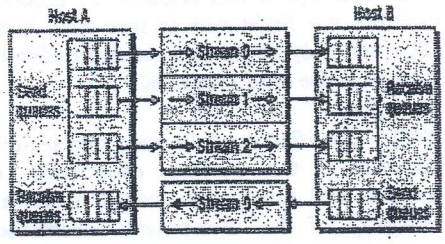

Figure 3: SCTP multi-streams hosts

Multi-streaming is another novel service SCTP includes at the transport layer. This is the logical separation of data within an association. Within streams, SCTP uses *stream sequence number* (SSNs) to preserve the data order and reliability for each data chunk. Between streams, however, no data order is preserved. This approach avoids TCP's head-of-line blocking problem, in which successfully transmitted segments must wait in the receiver's queue until a TCP sending end point retransmits any previously lost segments.

While SCTP manages ordering and packet delivery on a per-stream basis, it manages congestion control on a per-destination basis and flow control on a per-association basis.

- Congestion control

  SCTP congestion control is based on the well proven mechanism used in TCP. However, there are several major differences between TCP and SCTP. SCTP incorporates a fast retransmit algorithm based on SACK gap reports similar to that of TCP SACK. But SCTP does not have an explicit fast recovery phase. SCTP achieves fast recovery automatically with the use of SACK. Compared to TCP, the use of

SACK is mandatory in SCTP, which allows more robust reaction in the case of multiple losses from a single window of data. This avoids a time-consuming slow start stage after multiple segment losses, thus saving bandwidth and increasing throughput. During congestion avoidance of SCTP, *cwnd* can only be increased when the full *cwnd* is utilized; this restriction does not exist in TCP. TCP begins fast retransmission after receipt of three DupACKs; SCTP begins after four DupACKs. SCTP is able to clock out new data on receipt of the first three DupACKs and retransmit a lost segment by ignoring whether the flight size is less than *cwnd*; TCP can only begin data retransmission on receipt of the third DupACK.

- Security

  SCTP identified the following two security objectives: the *service availability* of reliable and timely data transport and the *integrity* of the user-to-user information carried by SCTP.

  Protecting availability of services: One of the most common threats to this objective is blind DoS attacks by flooding the target host with connection setup requests (e.g., SYN attack in TCP). The root of this problem is that the attacked host maintains in its memory useless state information regarding each pending connection, which will eventually exhaust the memory space of the system. SCTP eliminates the risk of DoS attacks by utilizing a *four-way handshake* sequence and a *cookie* mechanism to avoid maintaining state information for incomplete associations.

  Protecting the integrity of user-to-user information: If the objective of the attack is to break the integrity or confidentiality of the user-to-user information transfer, the payload of SCTP will be the target of the attack. In this case, SCTP should be used with IPSec or TLS to protect the confidentiality and integrity of the payload

Although SCTP has some advantages over TCP, it is not widely used at the moment.

One of the reasons is SCTP still has some issues.

- It must meeting the reliability requirements of SS7 that are the time needed to switch to another link when link failure occurs should be less than 800 ms and the availability of communication service between two signaling points should be at least 99,9988 percent, or a maximum down-time of 10 min/yr.
- SCTP congestion control and retransmission schemes are based on those of TCP. They are designed only for wired environments, they assume that all losses are caused by congestion, and round-trip-time (RTT) changes slowly and gradually. However, wireless mobile networks encounter higher bit error rates (BERs) than wired networks. This will cause SCTP to back off unnecessarily and result in poor throughput.
- The dynamic addition/deletion interface provides a graceful method to modify interfaces to an existing association when one of the endpoints in the association wishes to notify its peer that a new IP address will join the association or one of the old IP addresses will be out of service. It is important in mission-critical applications or mobile environments to support service reconfiguration without interrupting ongoing data transfers. This option, however, needs to define new chunk types and parameter types, and is still in the IETF draft stage.

III. SIMULATION

We design two kinds of test-bed. One use NS2 2.34 [10] and other is presented in real network.

With NS2, traffic is with constant bit rate. Our transports are UDP, TCP and SCTP and We measure quality through delay, packet loss and average throughput. Here, packet size is about 1000 bytes and bandwidth link is 200kbps. We have 3 *tcl* scripts for UDP, TCP and SCTP. Each of them run over NS and We found 3 .trace file, then We wrote some *awk* scripts [9] to run on trace files to generate results expected. We also output results to a graph for visual view. A message is tracked when it is en-queued from application at node 0 and delivered to application at node 3

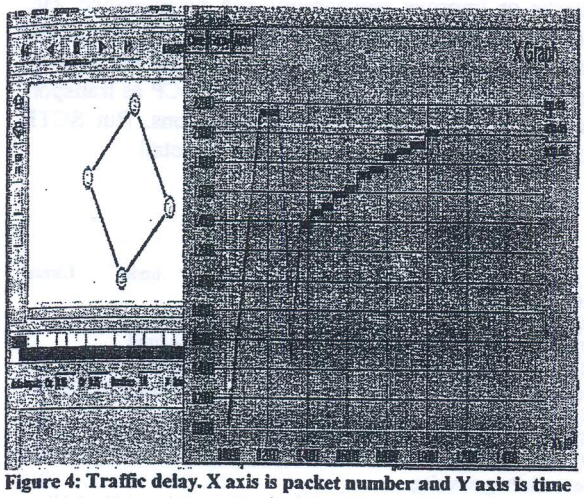

Figure 4: Traffic delay. X axis is packet number and Y axis is time (ms)

Table 1: Compared results among TCP, UDP and SCTP

Protocol	Sent	Received	Packet loss	Average Delay	Average Throughput
TCP	713	694	19	787.624 ms	199.276/200
UDP	1569	723	846	1930.83 ms	199.586/200
SCTP	750	689	61	1600.18 ms	199.267/200

This result is presented in normal network environment, SCTP seem in a better manner. In the case of under impairment network, we'll see the enhanced performance or benefit of SCTP over TCP clearly.

In second test-bed, in order to simulate in real network, test-bed is designed as below with three PCs installed Ubuntu. Lksctp must be enabled on two ends (OS must be integrated SCTP). The middle PC is installed with *netem* [5] package to create network impairment such as network delay, packet loss, packet corruption etc, it's also installed *bridge* [6] package to combine two PC network interfaces as a bridge inside a *netem* box. Traffic transmit through this bridge will be affected by network impairment

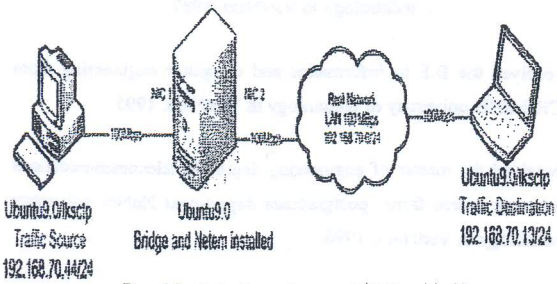

Figure 5: Test-bed to show performance of SCTP in real network

Here, We want to introduce something about lksctp [1] [3]. The Linux Kernel Stream Control Transmission Protocol (lksctp) project is an implementation of the Stream Control Transmission Protocol (SCTP) in the Linux kernel. The primary goal of this project is to provide users with the state of art implementation of SCTP protocol and various extensions. At current

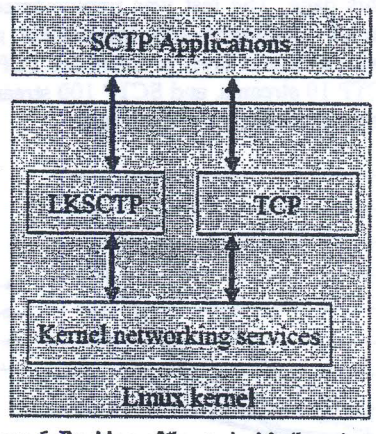

Figure 6: Position of lksctp inside linux kernel

status, lksctp focus to complete SCTP features, not for enhances performance as much as TCP, and at the moment very few applications using lksctp support SCTP as multi-stream transport layer protocol.

We wrote a simple application [7],[15] to compare performance between SCTP and TCP by send messages from source to destination under normal network conditions and impairment network conditions using SCTP and TCP as transport layer protocol, respectively. We varied loss rate from 0% to 5%, We collected results such as bandwidth, loss or time that it took to send data to show on simple graph.

In case of normal network (loss 0%) and high bandwidth, TCP and SCTP are nearly the same, TCP event was better. But under impairment conditions and low bandwidth, SCTP is really outperforming TCP. The test was run four times and the results showed the total time to transmit a message from source to destination for each case

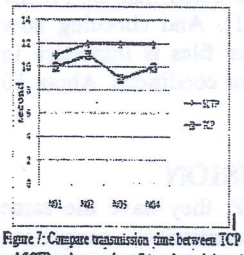

Figure 7: Compare transmission time between TCP and SCTP under normal conditions (lower is better)

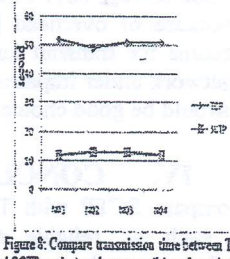

Figure 8: Compare transmission time between TCP and SCTP under impairment conditions (lower is better)

ISBN 978-89-5519-145-5

We also tried testing SCTP as transport layer protocol for HTTP application to transmit files which have different sizes. Here We used *thttpd* [13] as web server application, which is very simple, small and fast web server and web client application is *httperf* [14] .It generates good http traffic so it's used widely for testing. Both programs were modified to support SCTP by Elvis Plutzenreuter [12] in his thesis. We loaded them from his page. The test was implemented under normal network conditions and impairment network conditions with transport layer protocol are TCP, SCTP 1 stream, SCTP 10 streams, and SCTP 100 streams. The result shows the throughput for each case.

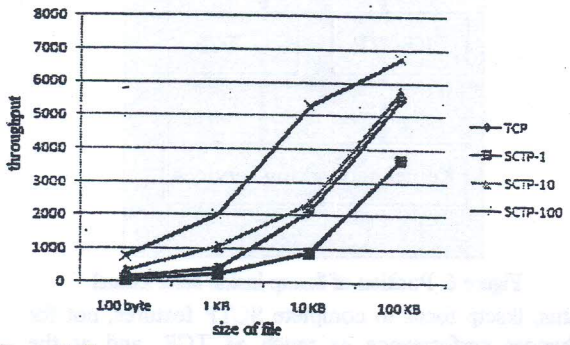

Figure 9: Compare throughput (kbps) between TCP and SCTP as transport protocol for http under normal conditions (higher is better)

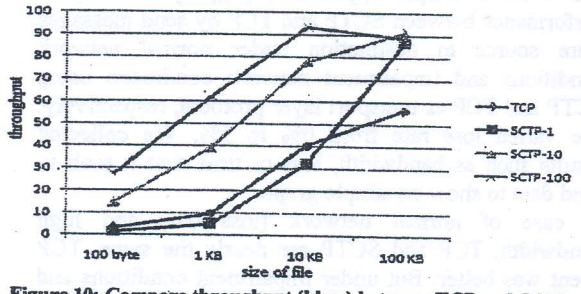

Figure 10: Compare throughput (kbps) between TCP and SCTP as transport protocol for http under normal conditions (higher is better)

The figures indicate that SCTP multi-streams can improve performance significantly on compared to TCP or SCTP one stream. And TCP is better than SCTP one stream because of overhead [11]. And choosing how many streams for transmit larger files is important in case of network under impairment conditions. About 10 streams should be good choice

IV. CONCLUSION

When compare SCTP with TCP, they have the same features such as reliable transmission, congestion control, flow control, connection oriented, SACK. In addition, there are advanced features which are in SCTP but not included in TCP to reduce vulnerability to DoS attacks or improve reachability and performance, etc. Test results present that SCTP completely is suitable for data transmission, especially in case of bad networking conditions. SCTP can replace UDP or TCP as transport layer protocol for networking applications. But SCTP should be research and optimized more detail.

V. References

[1] An implementation of SCTP under Linux, http://lksctp.sourceforge.net/
[2] SCTP – RFC 4960, http://www.ietf.org/rfc/rfc4960.txt
[3] LKSCTP, http://lksctp.sourceforge.net/overview.html
[4] S. Fu and all, SCTP: State of the Art in Research, Products, and Technical Challenges, IEEE Communications Magazine, April 2004
[5] http://www.linuxfoundation.org/en/Net:Netem
[6] http://www.linuxfoundation.org/en/Net:Bridge
[7] W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, UNIX® Network Programming
Volume 1, Third Edition: The Sockets Networking API, Addison Wesley, 2003.
[8] SCTP for Beginners, http://tdrwww.exp-math.uni-essen.de/inhalt/forschung/sctp_fb/index.html
[9] [ns] awk/perl script to calculate throughput/delay/jitter from trace files http://mailman.isi.edu/pipermail/ns-users/2004-September/044655.html
[10] NS-allinone-2.34, http://www.isi.edu/nsnam/ns/
[11] Rajesh Rajamani, Sumit Kumar, Nikhil Gupta, SCTP versus TCP: Comparing the Performance of Transport Protocols for Web Traffic, Computer Sciences Department, University of Wisconsin-Madison, May 2002
[12] Elvis Pfützenreuter, Applicability and performance of the SCTP transport protocol, master thesis, 2003-2004, http://epx.com.br/mestrado/index_en.php
[13] thttpd, http://acme.com/software/thttpd/thttpd_man.html
[14] httperf, http://www.hpl.hp.com/research/linux/httperf/httperf-man-0.9.pdf
[15] Nguyen Anh Dung, Tran Minh Phat, Nguyen Thanh Van: Multi-streaming with SCTP, final report. Page 6-15, 2009

TRAN CONG HUNG was born in VietNam in 1961

He received the B.E in electronic and Telecommunication engineering with first class honors from HOCHIMINH university of technology in VietNam, 1987.

He received the B.E in informatics and computer engineering from HOCHIMINH university of technology in VietNam, 1995.

He received the master of engineering degree in telecommunications engineering course from postgraduate department HaNoi university of technology in VietNam, 1998.

He received Ph.D at HaNoi university of technology in VietNam, 2004.

His main research areas are B – ISDN performance parameters and measuring methods, QoS in high speed networks, MPLS.

Currently, he is a lecturer, Dean of Faculty of Information Technology II in Posts and Telecoms Institute of Technology (PTIT), in HOCHIMINH City, VietNam.

TRẦN PHÚ KHÁNH was born in Vietnam in 1983

Obtained BE in Information Technology from Post and Telecommunication Institute of Technology (PTIT), Vietnam, 2006
Will receive Master in PTIT, 2009, major in Networking and Data Transmission
Networking solution engineer, Network Adminitrator, 2006-2008
IT Leader at Orientsoftware Corp, 2009
Main research fields are NGN, TCP/IP, and Security