

SCTP Sendbuffer Advertising

CS4099 Project
Midterm Evaluation

Arpan Kapoor, Deepak Sirone J, K Prasad Krishnan

Guided By:

Dr. Vinod Pathari

Mr. V Anil Kumar, Principal Scientist, CSIR, Bengaluru

March 3, 2016

Outline

Introduction

Problem Statement

Work Done in the previous semester

Design

Work Done in the current semester

Future Work

References

Introduction

- ▶ Stream Control Transmission Protocol (SCTP):
 - ▶ Supports multiple logical channels called streams
 - ▶ Multi-homing
- ▶ Sendbuffer Advertising:
 - ▶ specialized chunks will carry the amount of backlogged data present in the sender's buffer.

Problem Statement

- ▶ To propose a scheme to
 - ▶ advertise sendbuffer occupancy information in SCTP
 - ▶ implement it in the Linux kernel and
 - ▶ study the performance and security implications of the same.

Work Done in the previous semester

- ▶ Modified kernel module `sctp_probe` to measure sendbuffer.
- ▶ Explored Linux kernel SCTP implementation.
- ▶ Identified parameter to be advertised.

Design

- ▶ New chunk type with Chunk Type value between 128 to 190.
- ▶ Highest order 2 bits determine action to be taken if Chunk Type is unknown.
- ▶ This ensures that unmodified hosts won't send a Unrecognized Chunk Type Error chunk upon reception.

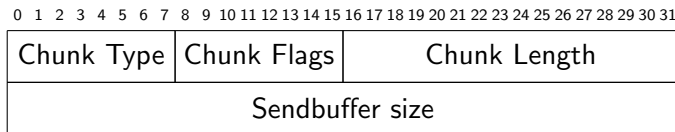


Figure: Proposed Chunk for sendbuffer advertisement

Work Done in the current semester

- ▶ A working prototype of sendbuffer advertisement was implemented in the Linux kernel.
 - ▶ Added a timer with appropriate modifications to the state machine function table.
- ▶ The interval for sending the sendbuffer advertisement chunk can be modified at runtime.
 - ▶ Added a procfs entry to change the sendbuffer advertisement interval

Future Work

- ▶ There are several scheduling algorithms which prioritizes packets based on some criteria. One of these priority based scheduling algorithms can be modified to consider the sendbuffer information and to improve QoS for high volume flows.
- ▶ Future Directions
 1. Use the testbed available at CSIR, Bengaluru.
 2. Create a testbed at NITC.
 3. Use a simulated network environment.

References I

- [1] A. Agache and C. Raiciu. *TCP Sendbuffer Advertising*. Internet-Draft draft-agache-tcpm-sndbufadv-00.txt. IETF Secretariat, July 2015.
- [2] Alexandru Agache and Costin Raiciu. “Oh Flow, Are Thou Happy? TCP Sendbuffer Advertising for Make Benefit of Clouds and Tenants”. In: *7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15)*. Santa Clara, CA: USENIX Association, July 2015. URL: <https://www.usenix.org/conference/hotcloud15/workshop-program/presentation/agache>.
- [3] Ian F. Akyildiz et al. “A roadmap for traffic engineering in SDN-OpenFlow networks”. In: *Computer Networks* 71 (2014), pp. 1–30. ISSN: 1389-1286. DOI: <http://dx.doi.org/10.1016/j.comnet.2014.06.002>. URL: <http://www.sciencedirect.com/science/article/pii/S1389128614002254>.

References II

- [4] Karthik Budigere. “Linux Implementation Study of Stream Control Transmission Protocol”. In: *Proceedings of Seminar on Network Protocols in Operating Systems*, p. 22.
- [5] M. Tim Jones. *Better networking with SCTP*. Feb. 28, 2006. URL: <http://www.ibm.com/developerworks/library/1-sctp/>.
- [6] L. Ong and J. Yoakum. *An Introduction to the Stream Control Transmission Protocol (SCTP)*. RFC 3286. RFC Editor, May 2002, pp. 1–10. URL: <http://www.rfc-editor.org/rfc/rfc3286.txt>.
- [7] Jon Postel. *Transmission Control Protocol*. RFC 793. RFC Editor, Sept. 1981, pp. 1–85. URL: <http://www.rfc-editor.org/rfc/rfc793.txt>.
- [8] R. Stewart. *Stream Control Transmission Protocol*. RFC 4960. RFC Editor, Sept. 2007, pp. 1–152. URL: <http://www.rfc-editor.org/rfc/rfc4960.txt>.

References III

- [9] R. Stewart et al. *Sockets API Extensions for the Stream Control Transmission Protocol (SCTP)*. RFC 6458. RFC Editor, Dec. 2011, pp. 1–115. URL:
<http://www.rfc-editor.org/rfc/rfc4960.txt>.