Improvements to the SCTP Environment in the INET Framework

Thomas Dreibholz, Martin Becke, Hakim Adhari, Erwin P. Rathgeb
University of Duisburg-Essen, Institute for Experimental Mathematics
Ellernstraße 29, 45326 Essen, Germany
{dreibh, martin.becke, hakim.adhari, rathgeb}@iem.uni-due.de

Irene Rüngeler, Robin Seggelmann, Michael Tüxen
Münster University of Applied Sciences, Department of Electrical Engineering and Computer Science
Bismarckstraße 11, 48565 Steinfurt, Germany
{i.ruengeler, seggelmann, tuexen}@fh-muenster.de

***Abstract*—This code contribution paper provides an overview of the enhancements to the Stream Control Transmission Protocol (SCTP) environment of the INET FRAMEWORK. Besides a highly improved SCTP core protocol, a couple of SCTP extensions – being either already standardized or currently under discussion in the IETF – have been realized. Furthermore, extended test applications and an auto-routing module with multi-homing support have been added. These enhancements make state-of-the-art SCTP simulation scenarios possible.**[1]

Keywords: SCTP, SCTP Extensions, Simulation, Multi-Homing, Auto-Routing

I. INTRODUCTION

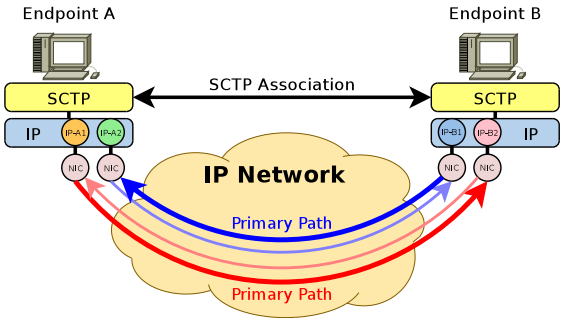

Figure 1. Multi-Homing with SCTP

The Stream Control Transmission Protocol (SCTP), defined in [1], is a general-purpose, connection-oriented, unicast Transport Layer protocol which provides the reliable transport of user messages. Unlike TCP, each SCTP endpoint can use multiple IPv4 and/or IPv6 addresses to transmit to its peer. This feature – which is illustrated in Figure 1 – is denoted as *multi-homing*. Each peer address defines a unidirectional *path*. In each direction, one of these paths is selected as so-called *primary path*. This selected path is used to transfer the user data. All other paths remain as backup and are only used for retransmissions. The redundancy features of SCTP make this protocol particularly useful for Reliable Server Pooling (RSerPool) [2] as well as Signalling Transport (SIGTRAN) applications.

A model of SCTP is already included in the INET FRAMEWORK since 2008 [3]. However, within the last three years, many improvements have been made and SCTP extensions have been added. This code contribution advances the SCTP environment of the INET FRAMEWORK to the state-of-the-art in SCTP development.

[1]Parts of this work have been funded by the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG).

II. SCTP MODEL IMPROVEMENTS

Besides various bugfixes and performance improvements of the SCTP code, a couple of SCTP extensions [4] which are already standardized or currently under discussion by the IETF have been incorporated into the SCTP model.

A. Partial Reliability

SCTP was designed as a reliable transport protocol. However, there are applications where the retransmission of data might be undesirable. An example is video streaming that needs the data immediately or never. The partial reliability extension (PR-SCTP) was specified in [5]. It gives the user the opportunity to control the level of reliability by limiting the time in which the data is worth being retransmitted or by limiting the number of retransmissions.

B. Dynamic Address Reconfiguration

As associations sometimes last very long, there might be the need to reconfigure the addresses, for example in a mobile environment. The Dynamic Address Reconfiguration extension [6] allows to change the set of addresses. The API realized in the INET FRAMEWORK enables the user to set a time for the deletion or addition of an address. Dynamic Address Reconfiguration requires another extension, called Chunk Authentication [7]. The realization of chunk authentication in the INET FRAMEWORK was motivated by the possibility to connect the framework to real networks [8]. Thus, this feature could be tested under real conditions. In addition to the chunks that announce an address reconfiguration, the user can determine which other chunks have to be authenticated.

C. Stream Reset

The ability to reset a stream [9], i.e. restart the numbering of the stream sequence number from 0, has often been requested and therefore implemented in the INET FRAMEWORK, too.

D. Non-Renegable Selective Acknowledgement

SCTP provides a selective acknowledgement (SACK) protocol mechanism, similar to state-of-the-art TCP implementations. However, SACKs are renegable, i.e. a receiver instance may – at any time – revoke such acknowledgements and request a retransmission of the previously SACK'ed data [1]. The sender instance must be prepared to handle such reneging. In order to reduce send buffer space requirements, the non-renegable selective acknowledgement (NR-SACK) extension, as defined

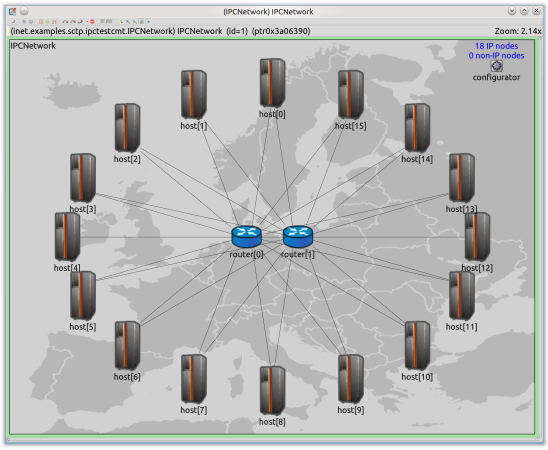

Figure 2. A Dual-Homed Example Network

in [10], adds non-renegable selective acknowledgements. That is, NR-SACK'ed data can be removed from the send buffer immediately.

E. Stream Scheduling

SCTP supports multiple streams, which are unidirectional channels within an association. They can be used to separate logically independent data. Since the message sequence is only retained within a single stream, less messages have to be delayed after a loss to restore the order. The messages of an arbitrary number of streams have to be multiplexed into packets in order to be sent. This is the task of a stream scheduler, which can pursue different strategies. The default in SCTP-capable operating systems is either First-Come, First-Serve (e.g. Linux, Solaris) or Round-Robin (e.g. FreeBSD). There are, however several other possible approaches to scheduling, such as Fair Bandwidth or Priorities as described in [11]. The scheduler of the SCTP implementation in the INET FRAMEWORK is now configurable.

F. Other SCTP Extensions

In addition to the extensions just described, the INET FRAMEWORK encompasses some more new features. Packet Drop Reporting [12] handles packets that were rejected by the Data Link Layer. This increases the throughput on lossy links considerably. The time needed to receive a SACK can be controlled by the sender applying the SACK Immediately extension [13]. UDP encapsulation [14] can guide SCTP packets through the network when SCTP is not known to middle boxes (e.g. NAT routers).

III. MULTI-HOMED AUTO-ROUTING

Typical SCTP application scenarios use multi-homed endpoints, in order to make use of the SCTP path redundancy features by SCTP. An example, consisting of two networks (i.e. dual-homing) – each as star topology with a router in its centre – is shown in Figure 2. However, the existing auto-routing modules provided by the INET FRAMEWORK did not support multi-homed networks. Therefore, the new model MultihomedFlatNetworkConfigurator has been added. Links are assigned a network ID (netID), which is used to differentiate between networks. The special setting of netID=0 configures links shared by ALL networks, which may be used for backbones, etc..

IV. THE SOURCE PACKAGE

The SCTP source package can be downloaded from the project page [15]. It is also provided as a patch against the current GITHUB INET FRAMEWORK tree. The package consists of:

- the updated and extended SCTP module sources in `src/transport/`,
- the adapted TCPDUMP and EXTINTERFACE sources in `src/util/`,
- small extensions to PPP and IP to allow for bit error handling by Packet Drop Reporting,
- queue bugfix and improvements in `src/networklayer/queue`,
- updated main `Makefile` and
- an example in `examples/sctp/newfeatures` (the included `README` file provides further details).

Note, that Chunk Authentication makes use of OPENSSL libraries and headers (Ubuntu/Debian package: `libssl-dev`). The main `Makefile` has been updated accordingly and a new run of `make makefiles` is required!

V. CONCLUSIONS

SCTP is a powerful Transport Layer protocol. This code contribution updates the SCTP model in the INET FRAMEWORK to the state of the art in SCTP development. Furthermore, it adds a couple of important SCTP extensions which are also provided by current SCTP implementations.

REFERENCES

[1] R. R. Stewart, "Stream Control Transmission Protocol," IETF, Standards Track RFC 4960, Sept. 2007, ISSN 2070-1721.

[2] T. Dreibholz and E. P. Rathgeb, "Overview and Evaluation of the Server Redundancy and Session Failover Mechanisms in the Reliable Server Pooling Framework," *International Journal on Advances in Internet Technology (IJAIT)*, vol. 2, no. 1, pp. 1–14, June 2009, ISSN 1942-2652.

[3] I. Rüngeler, M. Tüxen, and E. P. Rathgeb, "Integration of SCTP in the OMNeT++ Simulation Environment," in *Proceedings of the 1st ACM/ICST International Workshop on OMNeT++*, Marseille/France, Mar. 2008, ISBN 978-963-9799-20-2.

[4] T. Dreibholz, I. Rüngeler, R. Seggelmann, M. Tüxen, E. P. Rathgeb, and R. R. Stewart, "Stream Control Transmission Protocol: Past, Current, and Future Standardization Activities," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 82–88, Apr. 2011, ISSN 0163-6804.

[5] R. R. Stewart, M. Ramalho, Q. Xie, M. Tüxen, and P. T. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension," IETF, Standards Track RFC 3758, May 2004, ISSN 2070-1721.

[6] R. R. Stewart, Q. Xie, M. Tüxen, S. Maruyama, and M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration," IETF, Standards Track RFC 5061, Sept. 2007, ISSN 2070-1721.

[7] M. Tüxen, R. R. Stewart, P. Lei, and E. Rescorla, "Authenticated Chunks for the Stream Control Transmission Protocol (SCTP)," IETF, Standards Track RFC 4895, Aug. 2007, ISSN 2070-1721.

[8] M. Tüxen, I. Rüngeler, and E. P. Rathgeb, "Interface Connecting the INET Simulation Framework with the Real World," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools)*, Marseille/France, Mar. 2008, pp. 1–6, ISBN 978-963-9799-20-2.

[9] R. R. Stewart, P. Lei, and M. Tüxen, "Stream Control Transmission Protocol (SCTP) Stream Reconfiguration," IETF, Internet Draft Version 12, Aug. 2011, draft-ietf-tsvwg-sctp-strrst-12.txt, work in progress.

[10] N. Ekiz, P. D. Amer, P. Natarajan, R. R. Stewart, and J. R. Iyengar, "Non-Renegable Selective Acknowledgements (NR-SACKs) for SCTP," IETF, Network Working Group, Internet Draft Version 08, Aug. 2011, draft-natarajan-tsvwg-sctp-nrsack-08, work in progress.

[11] R. Seggelmann, M. Tüxen, and E. P. Rathgeb, "Stream Scheduling Considerations for SCTP," in *Proceedings of the 18th IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sept. 2010, ISBN 978-953-290-004-0.

[12] R. R. Stewart, P. Lei, and M. Tüxen, "Stream Control Transmission Protocol (SCTP) Packet Drop Reporting," IETF, Individual Submission, Internet Draft Version 12, June 2011, draft-stewart-sctp-pktdrprep-12.txt, work in progress.

[13] M. Tüxen, I. Rüngeler, and R. R. Stewart, "SACK-IMMEDIATELY Extension for the Stream Control Transmission Protocol," IETF, Individual Submission, Internet Draft Version 08, Oct. 2011, draft-tuexen-tsvwg-sctp-sack-immediately-08.txt, work in progress.

[14] M. Tüxen and R. R. Stewart, "UDP Encapsulation of SCTP Packets," IETF, Transport Area Working Group, Internet Draft Version 01, Oct. 2011, draft-ietf-tsvwg-sctp-udp-encaps-01.txt, work in progress.

[15] T. Dreibholz, "Thomas Dreibholz's SCTP Page," 2012. [Online]. Available: http://tdrwww.iem.uni-due.de/dreibholz/sctp/