Faculty of Economic Sciences, Communication and IT
Computer Science

Per Hurtig

Improving the Timeliness of SCTP Message Transfers

Karlstad University Studies
2008:43

Per Hurtig

Improving the Timeliness of SCTP Message Transfers

Karlstad University Studies
2008:43

Per Hurtig. *Improving the Timeliness of SCTP Message Transfers*

Licentiate thesis

Karlstad University Studies 2008:43
ISSN 1403-8099
ISBN 978-91-7063-202-0

Distribution:
Faculty of Economic Sciences, Communication and IT
Computer Science
SE-651 88 Karlstad
+46 54 700 10 00

www.kau.se

Printed at: Universitetstryckeriet, Karlstad 2008

Improving the Timeliness of SCTP Message Transfers

PER HURTIG
Department of Computer Science, Karlstad University

Abstract

Due to the cheap and flexible framework that the underlying IP-technology of the internet provides, IP-networks are becoming popular in more and more contexts. For instance, telecommunication operators have started to replace the fixed legacy telephony networks with IP-networks. To support a smooth transition towards IP-networks, the Stream Control Transmission Protocol (SCTP) was standardized. SCTP is used to carry telephony signaling traffic, and solves a number of problems that would have followed from using the Transmission Control Protocol (TCP) in this context. However, the design of SCTP is still heavily influenced by TCP. In fact, many protocol mechansisms in SCTP are directly inherited from TCP. Unfortunately, many of these mechanisms are not adapted to the kind of traffic that SCTP is intended to transport: time critical message-based traffic, e.g. telephony signaling.

In this thesis we examine, and adapt some of SCTP's mechanisms to more efficiently transport time critical message-based traffic. More specifically, we adapt SCTP's loss recovery and message bundling for timely message transfers. First, we propose and experimentally evaluate two loss recovery mechanisms: a packet-based Early Retransmit algorithm, and a modified retransmission timeout management algorithm. We show that these enhancements can reduce loss recovery times with at least $30 - 50\%$, in some scenarios. In addition, we adapt the message bundling of SCTP to better support timely message delivery. The proposed bundling algorithm can in some situations reduce the transfer time of a message with up to 70%.

In addition to these proposals we also indentify and report mistakes in some of the most popular SCTP implementations. Furthermore, we have continuously developed the network emulation software KauNet to support our experimental evaluations.

Keywords: SCTP, transport protocols, network emulation, loss recovery, Nagle algorithm.

Acknowledgments

First of all I would like to thank my supervisor Anna Brunström and my co-supervisor Johan Garcia. Without their support, it would have been impossible for me to pull this off. Thank you for the support and the interesting discussions.

Secondly, I would like to thank the people in my research group, the Distributed Systems and Communications Research Group (DISCO). Thanks for the interesting discussions and presentations.

I would also like to thank my colleagues at the department of computer science, for making it such a great place to work at!

Finally, I am also grateful for the financial support given by the Knowledge Foundation of Sweden (KK-stiftelsen) with Ericsson and TietoEnator as industrial partners, and also Nutek. Thank you all!

Karlstad, Novemeber 2008

Per Hurtig

List of Appended Papers

This thesis is based on the work presented in the following four papers. References to the papers will be made using the alphabetic symbols associated with the papers.

Paper A **Per Hurtig**, Johan Garcia, and Anna Brunstrom. Loss Recovery in Short TCP/SCTP Flows. *Technical Report*, 2006:71, Karlstad University Studies, Sweden, December, 2006.

Paper B **Per Hurtig** and Anna Brunstrom. Enhancing SCTP Loss Recovery: An Experimental Evaluation of Early Retransmit. In *Computer Communications*, Elsevier, Vol. 31, Issue 16, October 2008, pp. 3778-3788.

Paper C **Per Hurtig** and Anna Brunstrom. Improved Loss Detection for Signaling Traffic in SCTP. In *Proceedings of the IEEE International Conference on Communications (ICC 2008)*, Beijing, China, May, 2008.

Paper D **Per Hurtig** and Anna Brunstrom. SCTP: Designed for Timely Message Delivery? Submitted for publication. October 2008.

Some of the papers have been subjected to minor editorial changes.

Comments on My Participation

For all the papers, I am responsible for carrying out the experiments, and for most of the written material and ideas.

Other Papers

Apart from the papers included in the thesis, I have authored or co-authored the following papers:

- Johan Garcia, **Per Hurtig**, and Anna Brunstrom. The Effect of Packet Loss on the Response Times of Web Services. In *Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WebIST 2007)*, Barcelona, Spain, March, 2007.

- **Per Hurtig** and Anna Brunstrom. Packet Loss Recovery of Signaling Traffic in SCTP. In *Proceedings of the International Symposium on Performance of Computer and Telecommunication Systems (SPECTS 2007)*, San Diego, USA, July, 2007.

- **Per Hurtig** and Anna Brunstrom. SCTP Retransmission Timer Enhancement for Signaling Traffic. In *Proceedings of the 5th Swedish National Computer Networking Workshop (SNCNW 2008)*, Karlskrona, Sweden, April, 2008.

- Johan Garcia, **Per Hurtig**, and Anna Brunstrom. KauNet: A Versatile and Flexible Emulation System. In *Proceedings of the 5th Swedish National Computer Networking Workshop (SNCNW 2008)*, Karlskrona, Sweden, April, 2008.

- Mark Allman, Konstatin Avrachenkov, Urtzi Ayesta, Josh Blanton, and **Per Hurtig**. Early Retransmit for TCP and SCTP. *Internet-Draft, draft-ietf-tcpm-early-rexmt-00*, work in progress, August, 2008.

Contents

Introductory Summary

1 Introduction

Computer networking is becoming a very large and popular area within computer science. A major reason for the interest in computer networking is the tremendous success of the internet. The internet has evolved, over just a few decades, from being a very small research network with very limited capabilities, to what it is today: a world-wide public network with billions of users, and numerous services. This growth has partly been made possible by the Internet Protocol (IP) [22], which is the underlying technology of the internet.

Not only internet-based applications benefit from the IP-technology. In recent years, technologies like Voice over IP (VoIP) have lead to a revolution in the telecommunications industry. Actually, operators have already replaced large parts of the fixed telephony network core with more capable, and cost-effective, IP-networks.

To allow a smooth transition to IP-based environments, by the means of gradual deployments, the Internet Engineering Task Force (IETF) specified the SIGnaling TRANsport (SIGTRAN) architecture [20]. The SIGTRAN architecture was designed to support telephony signaling in environments where legacy networks and IP-networks co-exist.

As telephony signaling often have stricter requirements than other types of traffic (e.g. bulk traffic), the need for a new transport protocol was identified during the standardization of the SIGTRAN architecture. For example, the required availability of telephony is 99.9998%, which corresponds to a maximum downtime of 10 minutes per year [20]. Furthermore, telephony signaling also has strict requirements of timeliness. The transport protocols at that time, the Transmission Control Protocol (TCP) [23] and the User Datagram Protocol (UDP) [21], were not adequate considering such requirements.

The new protocol was named the Stream Control Transmission Protocol (SCTP) [24], and included a number of features to better support signaling transport. For example, SCTP was designed to be robust to link failures and alleviate timeliness problems present in TCP.

As applications other than signaling applications were found to benefit from SCTP's functionality, IETF decided to standardize SCTP for general internet deployment. To support a general deployment, however, SCTP needed to share network resources with TCP users fairly, i.e. it had to be TCP-friendly. Therefore, IETF decided to incorporate a large number of TCP-like mechanisms into the protocol.

Unfortunately, many of these mechanisms were directly inherited from TCP, a protocol that was designed for applications transmitting bulk traffic and not time critical message-based traffic. As no adaptation of the mechanisms was done, SCTP did not become optimally designed for time critical message-based traffic. The lack of adaptation is, for example, present in SCTP's packet loss recovery and message bundling.

SCTP has two loss recovery mechanisms: fast retransmit and retransmission timeout. Fast retransmit is preferred as it provides the fastest loss recovery. However, the use of fast retransmit is often inhibited when message-based traffic is transported. Hence, message-based traffic has to rely on slower timeout-based loss recovery more often than bulk traffic.

To make SCTP better suited for timely message transport, we propose and evaluate a number of adaptations in this thesis. First, we propose and experimentally evaluate two loss recovery enhancements: the packet-based Early Retransmit algorithm, and a modified retransmission timeout management algorithm. The Early Retransmit algorithm tries to increase the chance of fast retransmit, and the modified management algorithm tries to provide more accurate timeouts.

To inhibit excessive transmission of small packets, which introduces overhead, SCTP can bundle multiple messages into a single packet. However, to bundle messages SCTP typically needs to delay their transmission for a period of time. We propose an adaptation of the SCTP message bundling to better support timely message transfers.

The evaluations of our proposed loss recovery enhancements show promising results. In some scenarios, the time needed for loss recovery is reduced with at least $30-50\%$. The message bundling algorithm also performs well. In some scenarios, the transfer time of messages are reduced with up to 70%.

In addition to our evaluations and proposals, we also identify and report a number of implementation mistakes in the most popular SCTP implementations. Furthermore, we have during this work continuously developed the FreeBSD network emulation software KauNet. This software was used for all evaluations in the appended papers.

2 Research Objective

The objective of this thesis is to enhance the timeliness of SCTP message transfers by better adapting existing protocol mechanisms for time critical message-based traffic.

To accomplish this objective, we focus on two different research problems.

Firstly, we consider the timeliness of SCTP message transfers when packet loss occurs in the underlying network[1]. Thus, we focus on the timeliness of the loss recovery mechanisms, and how to improve it. More specifically:

> *How do we make SCTP's loss recovery mechanisms detect, and recover from, packet loss faster given time critical message-based traffic?*

Secondly, we consider the timeliness of SCTP message transfers during loss-free transmission. For this problem, we restrict ourselves to focus on the timeliness of message bundling and how to improve it. More formally:

> *How do we make SCTP perform message bundling and still support timely message transfers?*

In the next section we briefly cover the above-mentioned research problems. Moreover, we describe the relation between our work and previous work regarding these problems.

3 Research Problems

There is a significant amount of research that relates to our work. This section only describes the most relevant research.

3.1 Lengthy Retransmission Timeouts

When SCTP detects that data has been lost within the network, in enters a recovery phase. SCTP has two methods for invoking loss recovery. We start by describing the most basic method, called retransmission timeout (RTO) [12, 24]: if an acknowledgment for a given message is not received in a certain amount of time, the message is retransmitted by the sender. The RTO process is often very slow which is a serious problem, especially for time critical traffic.

There has been a lot of work on how the current RTO algorithm works and how to optimize it[2]. Basically, two approaches can be used: optimizing the parameterization of the algorithm or defining a new algorithm.

[1]To clarify, one packet can contain several messages. Thus, a lost packet implies at least one lost message.

[2]Please note that the meaning of "optimize" is very contextual. Different RTO proposals may have very different objectives. In this section, we only exemplify the different optimization approaches that can be taken.

In [4] different parameterizations are evaluated. The authors find that the parameter controlling the minimum bound of the RTO timer ($RTO_{\mathtt{min}}$) affects loss recovery performance the most. Thus, by lowering $RTO_{\mathtt{min}}$ retransmissions occur faster. If the bound is set too low, however, spurious retransmissions can occur. Such retransmissions could hurt the timeliness of the protocol even more.

In [10] a new RTO algorithm is presented. The authors try to fix some well-known problems in the algorithm, by specifying a completely new algorithm that relies on more advanced calculations. The evaluation of the new algorithm shows that it is more predictable in its operation, but also requires much more processing capabilities.

We suggest a modification that differs from the above-mentioned approaches. Our modification does not provide a completely new algorithm or a change of parameters. Instead, it fixes a design flaw in the timer management algorithm. Sometimes, the management algorithm does not associate the RTO timer with the right message. This often results in unnecessarily long RTOs. To mitigate this effect, we modify the algorithm to relate the timer to the correct message.

3.2 Loss Recovery with a Small Amount of Outstanding Data

Fast retransmit [12, 24] is a mechanism that usually provides much faster loss detection and recovery than the RTO mechanism. Fast retransmit resends a message when three duplicate acknowledgments arrive at the sender. Duplicate acknowledgments are triggered by out-of-order arrivals at the receiver. However, because out-of-order arrivals at the receiver are triggered by both packet loss and packet reordering in the network path, the sender waits for three duplicate acknowledgments to disambiguate packet loss from packet reordering.

When a small amount of data is outstanding[3] it may not be possible to generate the required number of duplicate acknowledgments to trigger fast retransmit. This is problematic for time sensitive applications as fast retransmit has to be used for timely loss recovery. The problem has been identified and addressed by several researchers (cf. [2, 3, 5–7, 9, 13, 14, 18]).

Many of the proposed solutions try to induce more duplicate acknowledgments, thereby increasing the chance of fast retransmit. Limited Transmit [2] allows a sender to transmit new packets when receiving duplicate acknowledgments. The intention of this proposal is to generate additional duplicate acknowledgments. However, new data is not always available at the sender. This is solved

[3]Outstanding data is data that is currently traveling from the sender to the receiver.

in [6], which allows a sender to transmit zero-byte packets to induce further duplicate acknowledgments.

Another example can be found in [14], where it is suggested that packets are split. The splitting should ensure that at least four packets are outstanding at all times. In this way, one original packet might induce several duplicate acknowledgments.

We measure completion times for flows experiencing packet loss, and verify that flow completion times are much longer when the amount of outstanding data is small. To mitigate this problem we propose the packet-based Early Retransmit algorithm [1], an extension of Early Retransmit (originally proposed in [5]). Early Retransmit differs from the proposals mentioned above as it does not try to induce duplicate acknowledgments. Instead, it dynamically lowers the duplicate acknowledgment threshold to a value that is suitable for the amount of data the sender has outstanding.

3.3 Message Bundling

When transmitting really small messages, e.g. 1 byte, the overhead becomes drastic. For instance, transmitting 1 byte messages would yield 4800% of overhead as each SCTP data message contains a 48 byte header. This problem is known as the small-packet problem and was first discovered in the 1960s [19].

To relieve the network from such overhead J. Nagle specified a both simple and effective algorithm in 1984 [19]. The algorithm can be described as follows: if there is unacknowledged data, then the transport protocol buffers all user data, until the outstanding data has been acknowledged or until the protocol can send a full-sized packet. Thus, the algorithm delays the transmission of small-sized packets to promote more data to be bundled in the packet. The algorithm showed to work well in inhibiting excessive transmission of small packets, and became a requirement for TCP-enabled hosts in 1989 [8].

However, as the algorithm inhibits transmission of packets it also prolongs the average transmission time of messages. Furthermore, the algorithm is known to interact badly with other mechanisms in TCP and SCTP, e.g. the delayed acknowledgment mechanism. Actually, in some situations the algorithm can stall the transmission of a single message with up to 500 ms [17].

To solve such latency problems but still inhibit small packets from being sent, a number of modifications have been proposed (cf. [15, 16]). G. Minshall proposes a simple modification in [15], which probably is the most well-known modification. Instead of always delaying small-sized packets, the Minshall algorithm does only delay small-sized packets if there already is a small-sized packet outstanding. This slight modification of the Nagle algorithm has been proven efficient and is, for example, implemented in the Linux TCP implementation.

To support a similar algorithm in SCTP, we redefine the notion of a full-sized SCTP packet. Using the new definition of a full-sized packet and Minshall's algorithm, SCTP is able to provide message bundling and timely messages transfers.

4 Research Methodology

Two research methodologies are commonly used in computer science: analytical and experimental. Using analytical research methods, problems are often mathematically modeled. When using experimental research methods problems are often modeled by simulations. However, experiments can also be conducted on real entities, or a mix of simulated and real entities. The latter is often referred to as emulation.

For this thesis we mainly used network emulation as research method. We used the network emulator KauNet [11], developed at Karlstad University, and real end-hosts. By using KauNet, we were able to evaluate real protocol implementations with a high level of control over network parameters such as end-to-end delays, bandwidths, queue sizes, and packet loss probabilities. The high level of control made it easy to quickly create network environments and construct repeatable experiments.

5 Main Contributions

The main contributions of this thesis can be summarized as follows:

- We evaluate the loss recovery of SCTP, and suggest some improvements to it. In particular, we:
  - evaluate the loss recovery performance of short data transfers in SCTP, and partly compare the results to experiments with TCP. This contribution is presented in Paper A;
  - evaluate and also extend Early Retransmit, a SCTP loss recovery enhancement. Our extension has recently been included in the IETF standardization process of Early Retransmit. This contribution is presented in Paper B and Paper D;

  - develop, and experimentally evaluate, a modified algorithm for controlling the retransmission timer in SCTP. This contribution is presented in Paper C and Paper D.

- We propose a modified message bundling mechanism in SCTP. This contribution is presented in Paper D.

- We also identify mistakes in some of the most popular implementations of SCTP, and also report them to the corresponding developers. This contribution is presented in Paper A.

- Also, we have continuously developed the network emulation software KauNet, which was used for all the experiments presented in Papers A through D.

6 Summary of Papers

Paper A – Loss Recovery in Short TCP/SCTP Flows

In this paper we experimentally evaluate and compare the loss recovery performance of different TCP and SCTP implementations. The results show, among other things, that packet loss recovery becomes a lengthy process whenever a small amount of data is outstanding. In addition to verifying this loss recovery performance problem, we also identify several implementation mistakes in the protocol implementations. In this paper we also give an introduction to TCP and SCTP, and their almost identical mechanisms for loss recovery and congestion control.

Paper B – Enhancing SCTP Loss Recovery: An Experimental Evaluation of Early Retransmit

To improve the timeliness of the SCTP loss recovery when the amount of outstanding data is small, we evaluate the Early Retransmit algorithm in this paper. To make Early Retransmit better suited for message-based traffic, we also modify the original proposal. The resulting mechanism is better adapted to message-based traffic, and outperforms the standard loss recovery mechanisms of SCTP. In this paper we also show that the retransmission timeout mechanism sometimes unnecessarily extends the loss recovery time.

Paper C – Improved Loss Detection for Signaling Traffic in SCTP

In this paper we set out to solve the above-mentioned problem with the retransmission timeout mechanism. We discover that the management algorithm unnecessarily causes the loss recovery to be significantly delayed. However, it is not a mistake in the implementation. Instead, the problem is located in the specification of the algorithm itself. We modify the specification of the algorithm and evaluate the new algorithm. The results show that significant performance improvements can be achieved using the new algorithm.

Paper D – SCTP: Designed for Timely Message Delivery?

In this paper we question if SCTP is really designed with time critical message-based traffic in mind. We evaluate our earlier loss recovery contributions jointly in this paper. Also, we suggest a modified Nagle-like algorithm. Common for both the loss recovery enhancements and the modified Nagle algorithm is that they are better suited for time critical message-based traffic than the original mechanisms, which were designed for TCP bulk traffic.

7 Future Work

For future work we intend to identify more SCTP mechanisms that can be optimized for time critical message-based transport. A possible way to optimize these mechanisms to their current environment could be to conduct dynamic reconfiguration of the mechanisms during run-time. This would make the deployment of SCTP in different environments easier, as it would result in nearly optimal performance without requiring expertise among the deployers.

Due to modern networking technologies like multi-path routing and node mobility, future network environments are likely to reorder packets more frequently. Packet reordering might cause out-of-order arrivals at a receiving node, which in turn can cause unnecessary congestion control actions that negatively affect the timeliness of message transfers. We therefore plan to investigate how transport protocols can mitigate the negative effects of packet reordering.

References

[1] M. Allman, K. Avrachenkov, U. Ayesta, J. Blanton, and P. Hurtig. Early Retransmit for TCP and SCTP. Internet draft, Internet Engineering Task

Force, August 2008. draft-ietf-tcpm-early-rexmt-00.

[2] M. Allman, H. Balakrishnan, and S. Floyd. Enhancing TCP's loss recovery using limited transmit. RFC 3042, Internet Engineering Task Force, January 2001.

[3] M. Allman, S. Floyd, and C. Partridge. Increasing TCP's initial window. RFC 3390, Internet Engineering Task Force, October 2002.

[4] M. Allman and V. Paxson. On estimating end-to-end network path properties. In *Proceedings of the ACM SIGCOMM'99*, 1999.

[5] U. Ayesta and K. Avrachenkov. The effect of the initial window size and limited transmit algorithm on the transient behavior of TCP transfers. In *Proceedings of the 15th ITC Internet Specialist Seminar*, Wurzburg, Germany, July 2002.

[6] H. Balakrishnan. *Challenges to Reliable Data Transport over Heterogeneous Wireless Networks*. PhD thesis, University of California at Berkely, California, USA, August 1998.

[7] H. Balakrishnan, V. Padmanabhan, S. Seshan, M. Stemm, and R. Katz. TCP behavior of a busy web server: Analysis and improvements. In *Proceedings of the IEEE INFOCOM'98*, San Fransisco, CA, USA, March 1998.

[8] R. Braden. Requirements for Internet Hosts – Communication Layers. RFC 1122, Internet Engineering Task Force, October 1989.

[9] S. Dawkins, G. Montenegro, M. Kojo, and V. Magret. End-to-end performance implications of slow links. RFC 3150, Internet Engineering Task Force, July 2001.

[10] H. Ekström and R. Ludwig. The peak-hopper: A new end-to-end retransmission timer for reliable unicast transport. In *Proceedings of the IEEE INFOCOM'04*, Hong Kong, March 2004.

[11] J. Garcia, E. Conchon, T. Perennou, and A. Brunstrom. KauNet: improving reproducibility for wireless and mobile research. In *Proceedings of the 1st international workshop on System evaluation for mobile platforms*, pages 21–26, San Juan, Puerto Rico, 2007.

[12] V. Jacobson. Congestion avoidance and control. In *Proceedings of the ACM SIGCOMM'88*, 1988.

[13] D. Lin and H.T. Kung. TCP fast recovery strategies: Analysis and improvements. In *Proceedings of the IEEE INFOCOM'98*, San Fransisco, CA, USA, March 1998.

[14] M. Mellia, M. Meo, and C. Casetti. TCP smart framing: a segmentation algorithm to reduce tcp latency. *IEEE/ACM Trans. Netw.*, 13(2):316–329, 2005.

[15] G. Minshall. A suggested modification to nagle's algorithm. Expired internet draft, Internet Engineering Task Force, June 1999. draft-minshall-nagle-01.

[16] G. Minshall, Y. Saito, J.C. Mogul, and B. Verghese. Application performance pitfalls and TCP's Nagle algorithm. *SIGMETRICS Perform. Eval. Rev.*, 27(4):36–44, 2000.

[17] J. C. Mogul and G. Minshall. Rethinking the TCP Nagle algorithm. *SIGCOMM Comput. Commun. Rev.*, 31(1):6–20, 2001.

[18] R. Morris. TCP behavior with many flows. In *Proceedings of the 5th IEEE International Conference on Network Protocols*, October 1997.

[19] J. Nagle. Congestion Control in IP/TCP Internetworks. RFC 896, Internet Engineering Task Force, January 1984.

[20] L. Ong, I. Rytina, M. Garcia, H. Schwarzbauer, L. Coene, H. Lin, I. Juhasz, M. Holdrege, and C. Sharp. Framework architecture for signaling transport. RFC 2719, Internet Engineering Task Force, October 1999.

[21] J. Postel. User Datagram Protocol. RFC 768, Internet Engineering Task Force, August 1980.

[22] J. Postel. Internet Protocol. RFC 791, Internet Engineering Task Force, September 1981.

[23] J. Postel. Transmission Control Protocol. RFC 793, Internet Engineering Task Force, September 1981.

[24] R. Stewart. Stream Control Transmission Protocol. RFC 4960, Internet Engineering Task Force, September 2007.

Improving the Timeliness of SCTP Message Transfers

Due to the cheap and flexible framework that the underlying IP-technology of the internet provides, IP-networks are becoming popular in more and more contexts. For instance, telecommunication operators have started to replace the fixed legacy telephony networks with IP-networks. To support a smooth transition towards IP-networks, the Stream Control Transmission Protocol (SCTP) was standardized. SCTP is used to carry telephony signaling traffic, and solves a number of problems that would have followed from using the Transmission Control Protocol (TCP) in this context. However, the design of SCTP is still heavily influenced by TCP. In fact, many protocol mechanisms in SCTP are directly inherited from TCP. Unfortunately, many of these mechanisms are not adapted to the kind of traffic that SCTP is intended to transport: time critical message-based traffic, e.g. telephony signaling.

In this thesis we examine, and adapt some of SCTP's mechanisms to more efficiently transport time critical message-based traffic. More specifically, we adapt SCTP's loss recovery and message bundling for timely message transfers. First, we propose and experimentally evaluate two loss recovery mechanisms: a packet-based Early Retransmit algorithm, and a modified retransmission timeout management algorithm. We show that these enhancements can reduce loss recovery times with at least 30-50%, in some scenarios. In addition, we adapt the message bundling of SCTP to better support timely message delivery. The proposed bundling algorithm can in some situations reduce the transfer time of a message with up to 70%.

In addition to these proposals we also indentify and report mistakes in some of the most popular SCTP implementations. Furthermore, we have continuously developed the network emulation software KauNet to support our experimental evaluations.

Karlstad University Studies

ISSN 1403-8099
ISBN 978-91-7063-202-0