Considerations on Handling Link Errors in SCTP

Rüngeler, Irene; Tüxen, Michael; Rathgeb, Erwin P.

In: ICB Research Reports - Forschungsberichte des ICB / 2009

DOI: https://doi.org/10.17185/duepublico/47092

URN: urn:nbn:de:hbz:464-20180918-135524-4

Link: https://duepublico.uni-duisburg-essen.de/servlets/DocumentServlet?id=47092

Source: ICB-Research Report No. 35, August 2009

ICB
Institut für Informatik und
Wirtschaftsinformatik

Irene Rüngeler
Michael Tüxen
Erwin P. Rathgeb

Considerations on Handling Link Errors in SCTP

35

ICB-RESEARCH REPORT

UNIVERSITÄT
DUISBURG
ESSEN

ICB-Research Report No. 35
August 2009

Authors' Address:

Dipl.-Inform. Irene Rüngeler
Prof. Dr. Michael Tüxen

Fachbereich Elektrotechnik und Informatik
Fachhochschule Münster
Stegerwaldstraße 39
D-48565 Steinfurt

i.ruengeler@fh-muenster.de
tuexen@fh-muenster.de

Prof. Dr. Erwin P. Rathgeb

Institut für Experimentelle Mathematik und Institut für Informatik und Wirtschaftsinformatik
Universität Duisburg-Essen
Ellernstr. 29
D-45326 Essen

erwin.rathgeb@uni-due.de

Contact:

Institut für Informatik und Wirtschaftsinformatik (ICB)
Universität Duisburg-Essen
Universitätsstr. 9
45141 Essen

Tel.: 0201-183-4041
Fax: 0201-183-4011
Email: icb@uni-duisburg-essen.de

ISSN 1860-2770 (Print)
ISSN 1866-5101 (Online)

Abstract

Today wireless network connections gain more and more importance. Yet a drawback of this technology is the higher packet error rate compared to wired networks. This has a negative impact on SCTP associations as packet loss is considered a congestion indication. The SCTP extension PKTDROP reacts to the notification of corrupted frames and circumvents the congestion control mechanism by retransmitting lost packets without changing the congestion window. In this paper we will point out the difference between TCP and SCTP due to message-orientation of SCTP and provide a formula to predict the throughput of an error-prone link, which can be beneficially applied for network planning. The validation of the PKTDROP extension will show that the effects of lossy links on the throughput can be fully compensated by this feature.

Keywords: SCTP, packet loss, PKTDROP reporting, theoretical throughput

Contents

List of Figures

1 Introduction

The Stream Control Transmission Protocol (SCTP) is a multi-purpose reliable message-oriented transport protocol, which was originally designed to deliver telephony signaling data over IP based networks. SCTP inherited some features from TCP, one of them is the congestion control mechanism. Although the principles used are the same, some issues arise from the fact that SCTP operates message-oriented whereas TCP operates byte-stream oriented. SCTP also supports bundling of multiple small user messages into one SCTP packet. As a result, the overall overhead of an SCTP packet depends on the user message size and the number of user messages that are bundled into the packet. Therefore, the corresponding assumptions inherited from TCP, are only true for SCTP, when packets with a large payload size are sent. For smaller message sizes, which are typical in the telephony signaling environment, the amount of overhead in the packet has to be considered.

As network planning can be performed more efficiently, if the upper limit for the achievable throughput of a link, whether it is error-prone or not, can be predicted, formulae have to be developed which reflect the message-orientation and the bundling properties of SCTP. Therefore, after an introduction to SCTP and its features relevant for this paper in Section 2, we will develop a formula for the maximum throughput that can be achieved by an SCTP association in the presence of errors and delay in Section 3. We will consider the different options of taking the headers into account when calculating the congestion window.

The need to deal with lossy links is growing as the number of wireless networks increases. In TCP and SCTP, a packet loss is considered a congestion indication, which triggers a reduction of the number of transmitted packets and, hence, the throughput. The packet drop reporting (PKTDROP) extension [SLT08] of SCTP is designed to overcome this disadvantage by reporting the loss of packets due to a packet error and thus allowing to retransmit the messages without adjusting the congestion control variables. In Section 4, we will first explain the packet drop reporting mechanism. Then we will show simulation results which have been performed with the OMNeT++ simulation environment [VaHo08] and an extended version of the SCTP simulation model [RTR08] we contributed to the INET framework [INET09]. The results demonstrate that the negative impact of the lossy link on the throughput can be fully compensated. In addition, fairness scenarios are examined. The paper ends with a conclusion and a short outlook on future work.

2 The Stream Control Transmission Protocol

With the emergence of IP-based networks as universal platform for communication services, the need arose to send telephony signaling data across the internet. Signaling data feature relatively small single messages, which can range between 17 and 48 bytes for ISUP traffic (see [SmBe94]), that have to be sent reliably and some of them also in the correct sequence.

To fulfill the strict performance requirements, the new protocol SCTP was designed and finally adopted by the IETF as official standard in RFC 2960 in 2000. After some modifications, RFC 4960 [Ste07] adopted in September 2007, is the current SCTP specification. In contrast to other earlier attempts to introduce new transport protocols, SCTP has gained significant practical relevance and is included in most of the major operating systems.

SCTP is a reliable, connection-oriented transport protocol. Connections are called **associations**. Unlike TCP, SCTP is message-oriented. SCTP packets consist of a 12 byte common header followed by a number of *chunks* containing user and control messages. SCTP uses the same port number concept as TCP or UDP. The source and destination port numbers are contained in the common header. There are control chunks, like the ones used to set up or tear down an association, and `DATA`-chunks.

Each chunk consists of a chunk header that varies with the chunk type. Type specific parameters or values complete the information. For the course of this paper two types of chunks have to be introduced in more detail, `DATA`-chunks carrying user messages and `SACK`-chunks which are used for acknowledgements.

A `DATA`-chunk consists of a chunk header of 16 bytes plus the user data. The protocol requires the chunk payload to be 32-bit aligned. To achieve this, the chunk has to be padded with 0 to 3 bytes at the end. Each `DATA`-chunk is identified by a transmission sequence number (TSN). In order to fully utilize the packet size provided by the maximum transmission unit (MTU) of the link, `DATA`-chunks (and certain control chunks) can be bundled together in one SCTP packet.

As SCTP is a reliable transport protocol, the reception of user data has to be acknowledged. This is handled by `SACK`-chunks. The cumulative TSN ack (cumTsnAck) parameter indicates the highest TSN received in sequence. The acceptance of additional chunks is reflected in gap ack blocks. Here, the beginning and the end of blocks of acknowledged data is announced. Thus, the gaps between these blocks identify the missing TSNs. After the acceptance of three `SACK`-chunks announcing the absence of the same TSN, the corresponding `DATA`-chunk has to be scheduled for fast retransmission. On sending a `DATA`-chunk, the sender starts a retransmission timer. If the TSN has not been acknowledged after the timer expired, a timer-based retransmission has to be initiated.

More information about other distinctive features of SCTP like multi-homing or multi-streaming is provided in RFC 3286 [OnYo02].

3 A Rule of Thumb for the Calculation of the Throughput

The link that connects the communicating hosts is supposed to provide a certain rate. In the case of ideal conditions, when the link is free of errors and the delay is insignificant, the link can almost be fully utilized by one connection, provided the network adapter and the CPU can handle the data. However, the available bandwidth, and thus the throughput, is influenced by the error rate and the delay of the channel. Therefore, it is of interest to find out which throughput can be expected in a specific scenario.

3.1 SCTP Congestion Control

As the throughput is the amount of data delivered from one node to another in a certain time, it is important to look at the mechanisms that influence the data transfer. The amount of data released from the sender depends to a great extent on the congestion control algorithm. The goal of this algorithm is to prevent senders from blocking links by reducing the rate of sending packets. A very important feature to control this rate is the congestion window (cwnd). It determines the amount of data that may be in flight on the link, that is the data, which has not been acknowledged yet, plus the data, that may be injected into the network.

The congestion control mechanism operates in two phases. The first one is called *slow start*. It operates for cwnd values less than or equal to the slow start threshold, which is set to an arbitrary value (mostly the advertised receiver window of the peer during association setup) at the beginning of an association. *Slow start* is characterized by an exponential increase of the congestion window. Every time an incoming `SACK`-chunk announces that the cumulative TSN ack parameter has advanced and the cwnd is fully utilized, i.e. the number of outstanding bytes is greater than cwnd, the minimum of the path MTU and the acknowledged bytes is added to cwnd.

When cwnd exceeds the slow start threshold, *congestion avoidance* makes for a linear increase of cwnd. As the growth of cwnd can lead to an excessive injection of datainto the network, packet loss is the consequence. While fast retransmissions result in halving the congestion window, a timer based retransmission leaves cwnd at the size of the path MTU and in *slow start* again. Thus cwnd follows usually a zigzag curve in the lifetime of a connection.

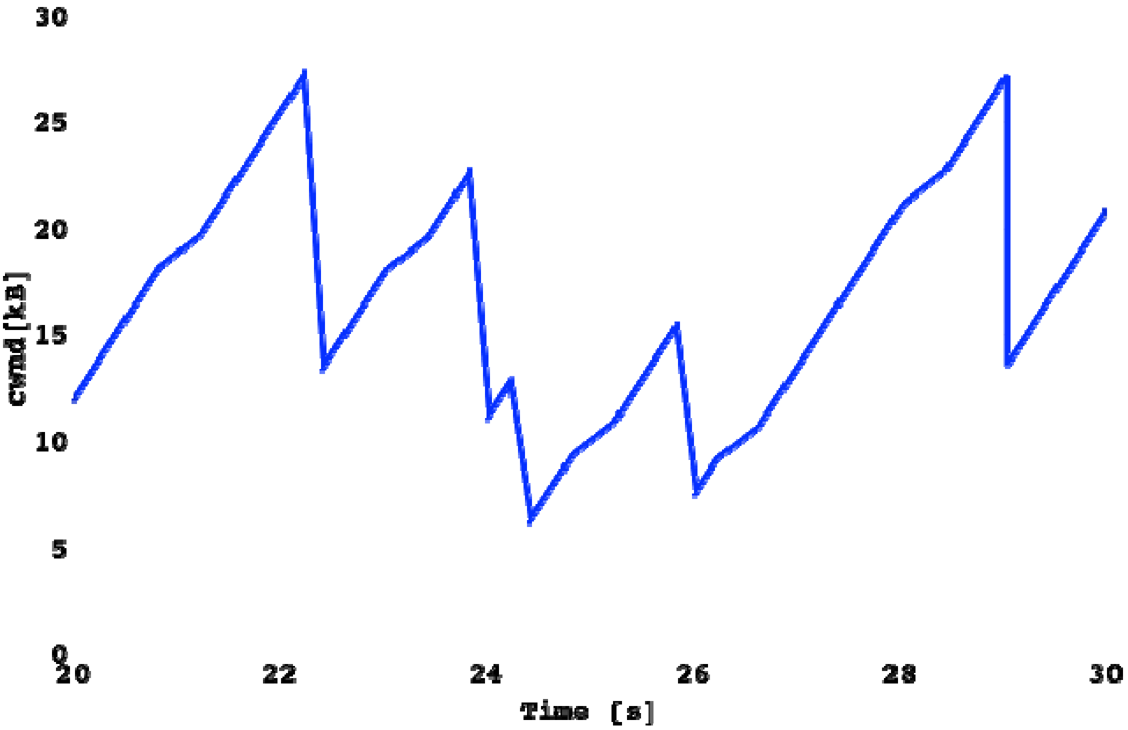

Figure 1: Evolution of the congestion window during a simulation

Figure 1 illustrates the evolution of the congestion window (cwnd) during a simulation over time. First the congestion window rises for a few seconds. Then an event occurs that causes the window to be halved, before it can rise again. This event is a packet loss, that might be caused by a bit error or a full router queue and results in a retransmission.

3.2 Mathis' Formula to Calculate the Throughput for TCP

In [MSM97] Mathis generalized the behavior of the congestion window evolution and introduced a model to predict the throughput of a TCP connection when the packet loss rate P_P and the round trip time RTT are given.

$$Throughput = \frac{MSS \cdot C}{RTT \cdot \sqrt{P_P}} \tag{1}$$

The parameter C is the constant of proportionality, that combines several terms that are typically constant for a given TCP implementation.

As TCP is byte-stream oriented, all data are transmitted in packets of the maximum segment size (*MSS*).

SCTP is message oriented and therefore the packet size depends on the message length. For payload sizes which fill up the packets, Equation (1) can be used for SCTP, too. But for smaller user message sizes, Mathis' model is not applicable. Another important difference to TCP is, that small messages can be bundled in SCTP. Each DATA-chunk consists of its header of 16 Bytes and the user message. Especially for small user message sizes, where the overhead accounts for a significant share of the packet, the difference between *MSS* and the payload cannot be neglected. Therefore Mathis' model must be adapted to the needs of SCTP.

The model that we use for the following calculations can be characterized as follows. We have one client connected to one server, which runs as a discard server. The throughput is not limited by the bandwidth of the link, i.e. we don't assume a bottleneck link. The server's receiver window is sufficiently big to read all incoming data, hence flow control is not a limiting factor. The connection is already established and the state of congestion avoidance is reached. All errors can be corrected by sending fast retransmissions. Thus no timer based retransmissions will cause the connection to leave congestion avoidance and go into slow start again.

In Subsection 3.1 we pointed out, that the congestion window controls the amount of data in flight. Looking at TCP, where congestion control was first introduced, the data in flight are the user messages without the headers. In SCTP, where the messages can be bundled and thus the headers can contribute significantly to the amount of the transferred data, the question arises, whether the data in flight should be calculated with or without taking the headers into account. RFC 4960 defining SCTP does not specify whether the message specific headers have to be considered when updating the parameters for congestion control. Therefore, in the following subsection, we will derive a formula for the throughput without taking the headers into account. In Subsection 3.4 we will derive the corresponding formula with headers included and point out the differences. Information about the influence of the way the outstanding bytes are counted on the fairness towards TCP connections is given in [RTR09].

3.3 Calculating the Throughput without Taking the Headers into Account

Figure 2 shows the evolution of the SCTP congestion window in the status of congestion avoidance. In contrast to Mathis' model, cwnd is measured in bytes instead of packets. The maximum cwnd is assumed to be X bytes. For each arriving `SACK`-chunk the window grows by $1 \cdot MTU$. Thus in $\frac{X}{2 \cdot MTU}$ round trip times half the window is filled. The growth of the congestion window is stopped when a fast retransmission is triggered by the arrival of three successive `SACK`-chunks announcing a gap in the list of transmission sequence numbers (TSNs).

During this time

$$\frac{X}{2} \cdot \frac{X}{2 \cdot MTU} + \frac{X}{2} \cdot \frac{X}{2 \cdot MTU} \cdot \frac{1}{2} = \frac{3 \cdot X^2}{8 \cdot MTU} \qquad (2)$$

bytes, which is equivalent to the area below the dotted polygon, are transmitted.

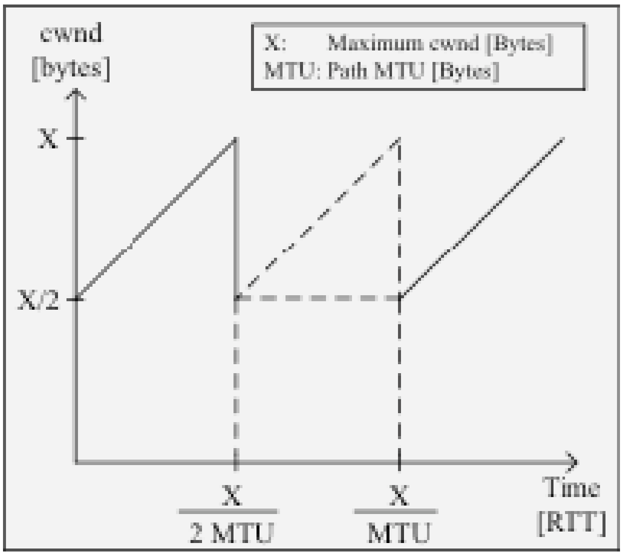

Figure 2: Evolution of a window cycle

Assuming a packet loss rate of P_P, $\frac{1}{P_P}$ packets will be transmitted, before an error occurs. As we measure in bytes the probability for a byte error can be calculated to be

$$P_B \approx \frac{P_P}{D} \tag{3}$$

with D corresponding to the DATA-chunk payload. Hence the number of error-free transmitted bytes is $\frac{D}{P_P}$. Equating (2) and $\frac{D}{P_P}$ and solving for $\frac{X}{2}$ leads to

$$\frac{X}{2} = \sqrt{\frac{2 \cdot D \cdot MTU}{3 \cdot P_P}} \tag{4}$$

The throughput is calculated as the ratio of the data per cycle to the time per cycle. The data comprises the actual payload without counting the headers.

$$\begin{aligned} Throughput &= \frac{data\ per\ cycle}{time\ per\ cycle} \\ &= \frac{D \cdot \frac{1}{P_P}}{RTT \cdot \frac{X}{2 \cdot MTU}} \\ &= \frac{D \cdot \frac{1}{P_P}}{RTT \cdot \frac{1}{MTU} \cdot \sqrt{\frac{2 \cdot D \cdot MTU}{3 \cdot P_P}}} \\ &= \frac{D \cdot \sqrt{\frac{3}{2}}}{RTT \cdot \sqrt{\frac{D}{MTU}} \cdot \sqrt{P_P}} \\ &= \frac{\sqrt{D \cdot MTU} \cdot \sqrt{\frac{3}{2}}}{RTT \cdot \sqrt{P_P}} \end{aligned} \tag{5}$$

We verified this theoretical result by comparing it to the simulation described in [RTR08]. Figure 3 shows four different tests each consisting of two graphs representing the theoretical and the simulated values. As the throughput is dependent on the payload, we chose user message sizes from 10 to 1450 bytes in steps of 10 bytes, whereas each run was performed 10 times. The figure shows the mean of these runs and the corresponding confidence intervals. Starting from the base run with an *MTU* of 1500 bytes, an *RTT* of 200 ms and a packet loss rate of 1% (second lowest graph), one of the parameters at a time has been changed to generate the other curves.

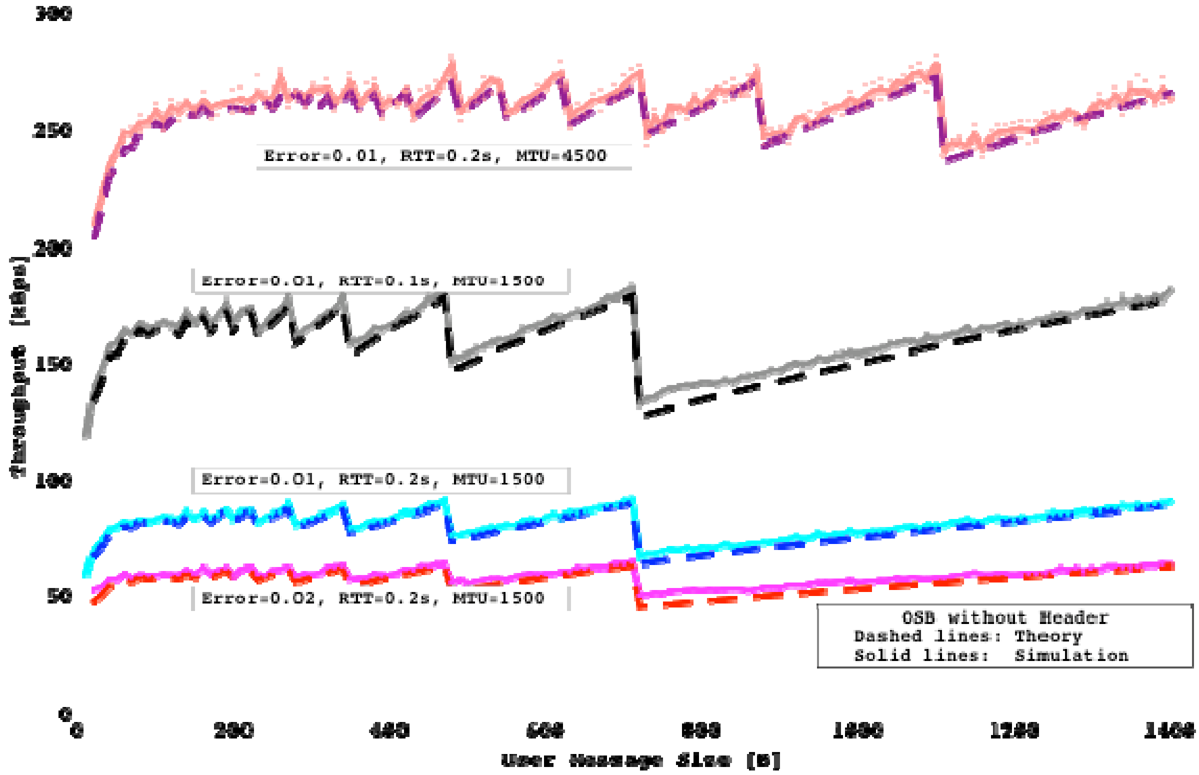

Figure 3: Comparison between simulation and theory for varying parameters

3.4 Including the Headers in the Calculation of the Data in Flight

In congestion avoidance the congestion window increases linearly by one MSS in the case of TCP, which corresponds to one MTU for SCTP. This amount is independent from the user message size. Therefore, the cwnd in Figure 4 is the same as in Figure 2. The amount of data transferred in one RTT is dependent on the size of the headers, which results in the factor *H*. *H* stands for the headers' proportion of the data.

$$H = 1 + \frac{H_{Chunk}}{UMS} \qquad (6)$$

with H_{Chunk} meaning the size of the DATA-chunk header and *UMS* the average payload of a DATA-chunk in a packet.

Following the same conclusions as in Subsection 3.3 results in Equation (7)

$$Throughput = \frac{\sqrt{D \cdot MTU} \cdot \sqrt{\frac{3}{2}}}{RTT \cdot \sqrt{P_P \cdot H}} \quad (7)$$

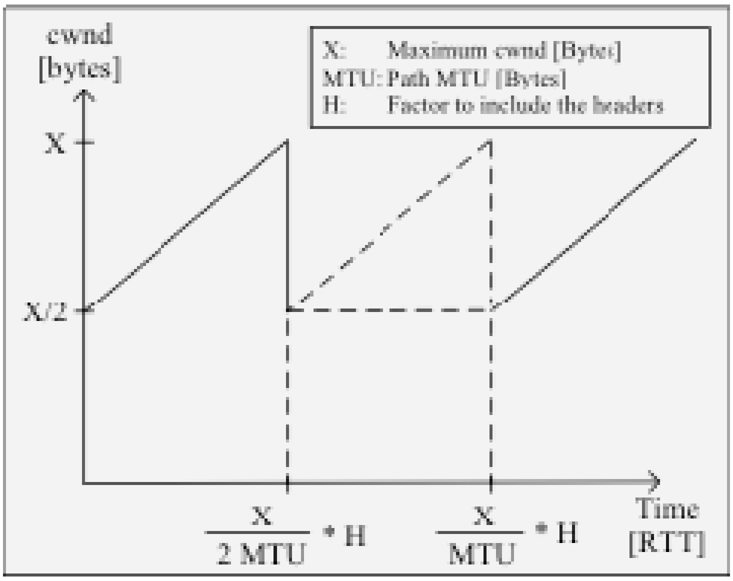

Figure 4: Evolution of a window cycle, when the headers are taken into account

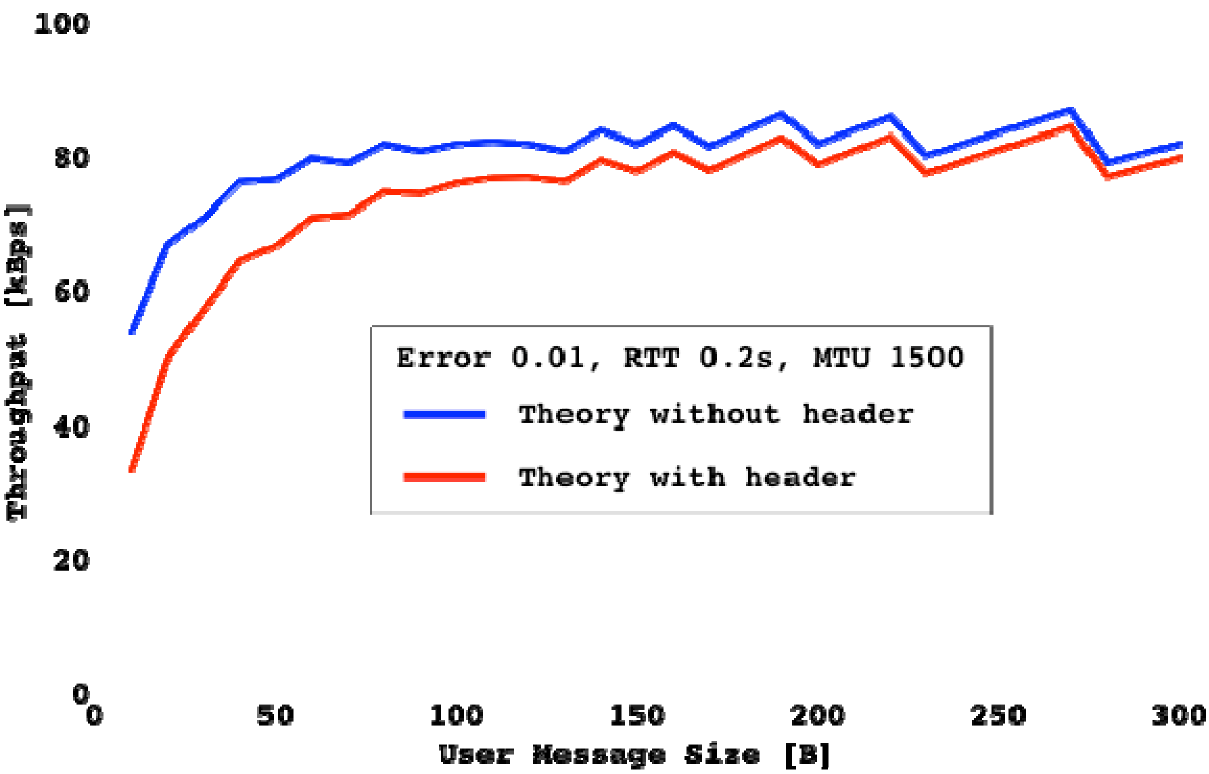

Figure 5: Comparison of Equation (5) with Equation (7)

This equation was again verified by running the same simulations, but this time, the headers were taken into account, when computing the data in flight. The graphs of the simulation and the theory matched like expected. Figure 5 shows a detailed comparison between Equation (5) and Equation (7) for a packet loss rate of 1% and a link delay of 200 ms. For message sizes less than 300 Bytes the throughput is slightly less, when the headers are taken into account.

4 Using Packet Drop Reporting to Reduce false Adjustment of the Congestion Window

As mentioned before, the congestion window reacts to a blocked link, i.e. when packets are dropped because of full router queues. When link errors occur, the resulting data corruption is discovered by a false IP checksum. In most cases the packet is dropped by the network adapter, and thus there is no chance for the transport layer to react according to that event. Welzl [Wel05] states that there is a great demand for passing erroneous packets from the link layer to the transport layer, where measurements could be taken according to the protocol and the application needs.

Packet drop reporting (PKTDROP) is a extension of SCTP [SLT08] to report packets that have been dropped by middle boxes or the end-host due to a false checksum or an exhausted receiver window. This feature is implemented in the FreeBSD 7 kernel. If both end-hosts of an association support PKTDROP, which they announce during association setup, the end-host receiving a corrupted packet will send a `PKTDROP`-chunk back. This chunk includes the complete packet that was corrupted. In case, that the resulting packet is larger than the maximum segment size, the corrupted message is truncated, which is announced by setting the T-bit and the *Truncated Length* field in the header. The receiver of the `PKTDROP`-chunk tries to figure out, which TSNs were included. To identify the TSNs the 4 byte TSN field and the 2 byte length field have to be uncorrupted. The retrieved TSNs have to be marked for retransmission to be resent as soon as possible. In addition to this faster way of retransmission, compared to the three necessary `SACK`-chunks reporting the TSN missing, the congestion window will not be decreased and the fast recovery status will not be entered. This is justified by the fact that the packet was not lost due to congestion but because of a lossy link.

4.1 Calculating the Maximum Throughput

To be able to compare the following simulation results, i.e. the achievements of PKTDROP, the possible throughput, we need a formula for the maximum throughput. Again it is dependent on the payload sizes of a packet. In contrast to the equations in Section 3, no packet error rate or delay is considered.

The theoretical throughput for SCTP is calculated as follows

$$Throughput = CPP \cdot UMS \cdot PPS \tag{8}$$

with the average user message length per packet *UMS*, the number of packets per second *PPS*, and *CPP* the number of chunks per packet, which are again calculated as

$$CPP = \left\lfloor \frac{MTU - H_{IP} - H_{SCTP}}{CL} \right\rfloor \tag{9}$$

with the IP-header length H_{IP}, the SCTP common header length H_{SCTP} and the chunk length *CL* adding up to

$$CL = UMS + P_{UMS} + H_{Chunk} \tag{10}$$

where H_{Chunk}denotes the length of the `DATA`-chunk header and P_{UMS} is the number of padding bytes.

The number of packets per second *PPS* can be computed as

$$PPS = \frac{datarate}{H_{IP} + H_{SCTP} + CPP \cdot CL} \tag{11}$$

Using equation (9), (10), and (11) results in

$$Throughput = \frac{(MTU - H_{IP} - H_{SCTP}) \cdot UMS}{UMS + P_{UMS} + H_{Chunk}} \cdot \frac{datarate}{H_{IP} + H_{SCTP} + CPP \cdot (UMS + P_{UMS} + H_{Chunk})} \tag{12}$$

For the following simulations we assume, that the host receiving the packet drop reports is able to retrieve the information, which TSN has to be retransmitted.

4.1.1 Comparing Associations with and without PKTDROP over a Lossy Link

In Figure 6 we tested a simple scenario with one client and one server connected over a lossy link with a packet error rate of 1% and an RTT of 20 ms. The throughput of an association with packet drop reporting is compared to one without it.

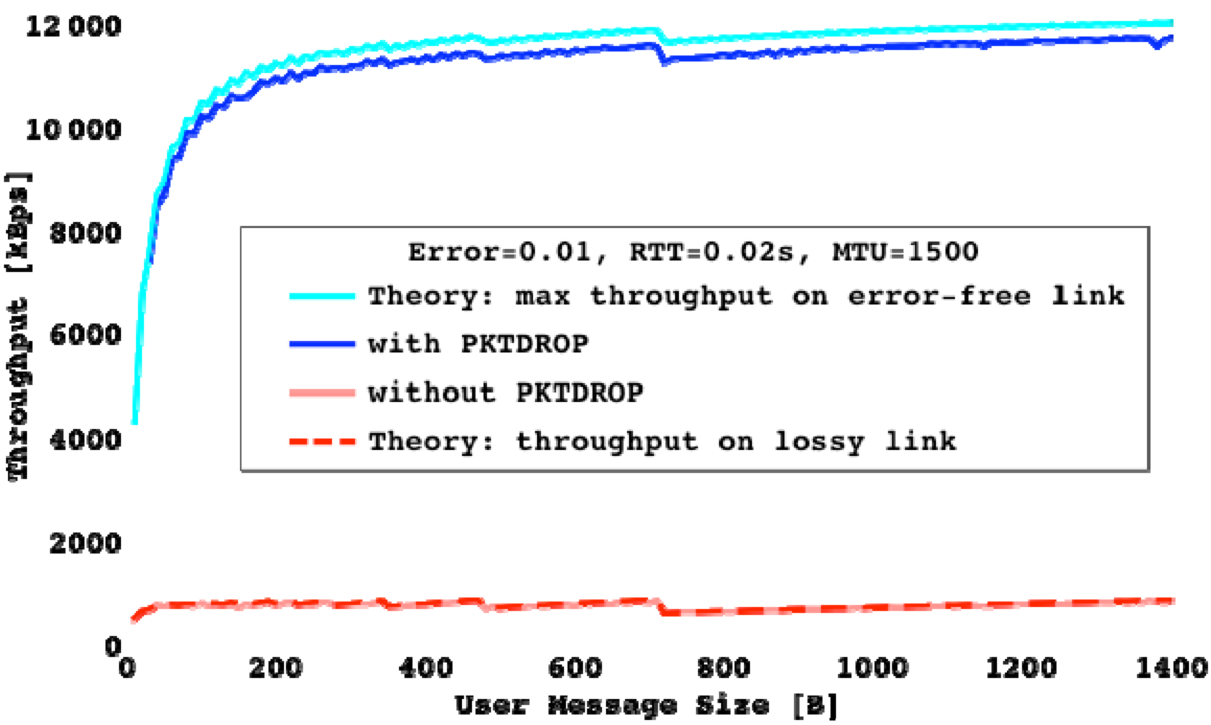

Figure 6: Comparison of an association with and without packet drop reporting

The lower dashed graph shows the theoretical throughput according to Equation (5). As before the simulated results match the theoretical ones. The graph with packet drop reporting is compared to the highest theoretical throughput on an error-free and delay-free link according to Equation (12).

The graphs in Figures 6 and 8 are displayed within confidence intervals, because they were too small to be visible.

It is obvious, that by using packet drop reporting, the negative effect of packet loss caused by corrupted packets can be almost fully compensated.

4.2 Fairness towards an Association without PKTDROP

Another scenario is shown in Figure 7. The bottleneck link between `Router 1` and `Router 2` is configured with a packet error rate of 1% and a delay of 20 ms. This scenario is used for the next simulations. `Client 1` sends data to `Server 1` and `Client 2` to `Server 2`.

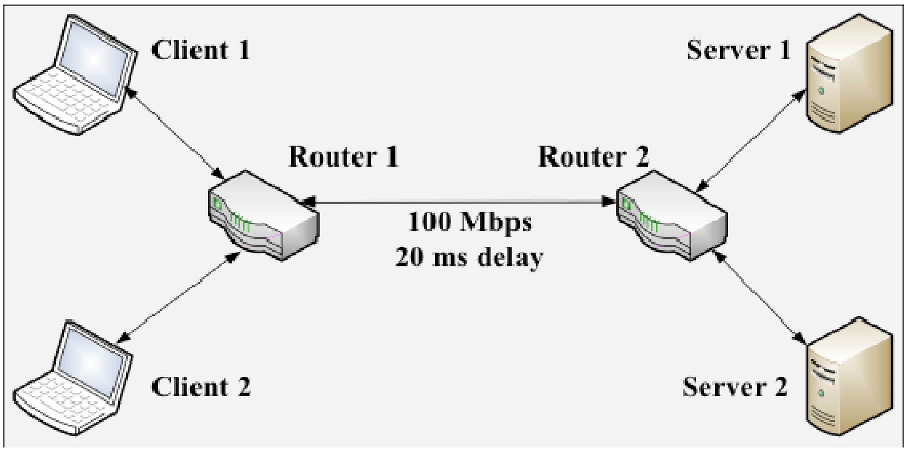

Figure 7: Scenario with bottleneck link

In the first case no end-points were configured to use packet drop reporting. The results are shown in the second lowest lines of Figure 8. Both associations achieve the same throughput, which also equals the theory of Section 3.

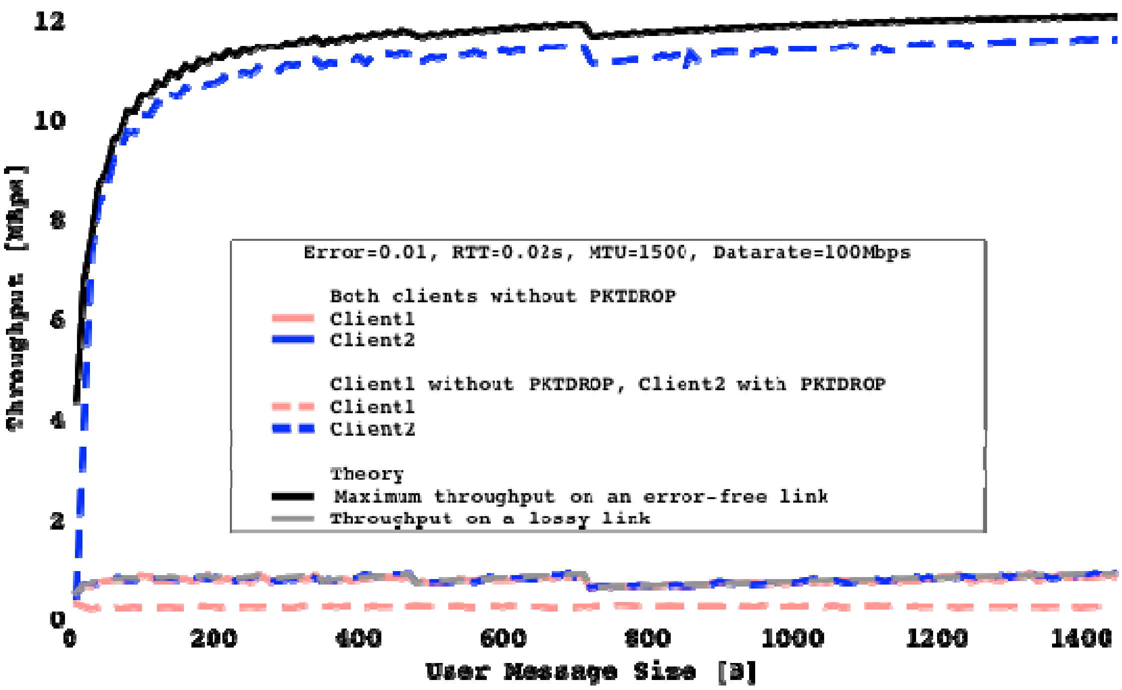

Figure 8: Throughput on a lossy bottleneck link

In the second case both `Client 2` and `Server 2` used PKTDROP. Sharing the link with an association that has to cope with many retransmissions lets the association with PKTDROP gain even more bandwidth. Thus the throughput of the association between `Client 1` and `Server 1` is reduced, which is depicted in the lowest graph of Figure 8. This is acceptable since `Client 1` is still misinterpreting packet loss as congestion indication.

4.3 Fairness Measurements when the Lossy Link is not the Bottleneck

Sometimes one link of a path is faulty, whereas the rest is error free. When packet drop reporting is provided, the connection could be more aggressive because of another retransmission behavior than other 'normal' connections. To test this scenario we isolated the association which uses PKTDROP and configured the link between `Client 2` and `Router 1` with a packet error rate of 1%. All other links were error-free. Thus the bottleneck link is shared between a 'normal' association and one with PKTDROP and a high retransmission rate.

Figure 9: Throughput of the bottleneck link, if the link between *Client* 2 and *Router 1* is configured with an error rate

Each simulation run was repeated 100 times with different seeds for the random numbers to ensure validity. Figure 9 shows the throughput on the transport layer. The black dots represent the 95% confidence intervals. As expected, both connections share the bandwidth equally. They are fair towards each other.

5 Conclusion and Outlook

In this paper we introduced a formula to predict the throughput for an SCTP association given network parameters like *RTT* and the packet loss rate. We verified this formula with simulation results.

PKTDROP, an SCTP extension, that helps recovering packets that were dropped due to a lossy link, was explained and validated. It was shown that, using this feature, the negative impact on the throughput resulting from error-prone links can be almost fully compensated.

Rüngeler, I., Tüxen, M., Rathgeb, E.P.

Further studies will focus on the evaluation of packet drop reporting in real scenarios.

6 References

[INET09] **INET Framework Documentation**; http://inet.omnetpp.org/doc/INET/neddoc/index.html, last visited: August 27, 2009

[MSM97] Mathis, M.; Semke, J.; Mahdavi, J.: **The macroscopic behavior of the TCP congestion avoidance algorithm**, ACM SIGCOMM Computer Communication Review, Volume 27, No. 3, July 1997.

[OnYo02] L. Ong, L.; Yoakum, J.: **An Introduction to the Stream Control Transmission Protocol (SCTP)**, RFC 3286, May 2002.

[RTR08] Rüngeler, I.; Tüxen, M.; Rathgeb, E.P.: **Integration of SCTP in the OMNeT++ simulation environment**, Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems, 2008.

[RTR09] Rüngeler, I.; Tüxen, M.; Rathgeb, E.P.: **Congestion and Flow Control in the Context of the Message-oriented Protocol SCTP**, IFIP Networking 2009, pp. 468-481.

[SmBe94] Smith, D.; Bellcore, R.B.: **Effects of Feedback Delay on the Performance of the Transfer-Controlled Procedure in Controlling CCS Network Overloads**, IEEE Journal on Selected Areas in Communications, Volume 12, No. 3, pp. 424-432, 1994

[SLT08] Stewart, R; Lei, P.; Tüxen, M.: **Stream Control Transmission Protocol (SCTP) Packet Drop Reporting,** draft-stewart-sctp. pktdrprep-08 (work in progress).

[Ste07] Stewart, R.: **Stream Control Transmission Protocol**, RFC 4960, September 2007.

[VaHo08] Varga, A.; Hornig, R.: **An Overview of the OMNeT++ Simulation Environment**, International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, March 2008.

[Wel05] Welzl, M.: **Passing Corrupt Data Across Network Layers: An Overview of Recent Developments and Issues**, EURASIP Journal of Applied Signal Processing, Volume 2005, No. 2, pp. 242-247.

Previously published ICB - Research Reports

2009

No 34 (June 2009)
Karastoyanova, Dimka ; Kazhamiakin, Raman; Metzger, Andreas ; Pistore, Marco (Eds.): Workshop on Service Monitoring, Adaptation and Beyond

No 33 (May 2009)
Adelsberger, Heimo; Drechsler, Andreas ; Bruckmann, Tobias ; Kalvelage, Peter ; Kinne, Sophia; Pellinger, Jan; Rosenberger, Marcel ; Trepper, Tobias: Einsatz von Social Software in Unternehmen – Studie über Umfang und Zweck der Nutzung

No 32 (April 2009)
Barth, Manfred; Gadatsch, Andreas; Kütz, Martin; Rüding, Otto; Schauer, Hanno; Strecker, Stefan: Leitbild IT-Controller/-in – Beitrag der Fachgruppe IT-Controlling der Gesellschaft für Informatik e. V.

No 31 (April 2009)
Frank, Ulrich; Strecker, Stefan: Beyond ERP Systems: An Outline of Self-Referential Enterprise Systems – Requirements, Conceptual Foundation and Design Options

No 30 (February 2009)
Schauer, Hanno; Wolff, Frank: Kriterien guter Wissensarbeit – Ein Vorschlag aus dem Blickwinkel der Wissenschaftstheorie (Langfassung)

No 29 (January 2009)
Benavides, David; Metzger, Andreas; Eisenecker, Ulrich (Eds.): Third International Workshop on Variability Modelling of Software-intensive Systems

2008

No 28 (December 2008)
Goedicke, Michael; Striewe, Michael; Balz, Moritz: „Computer Aided Assessments and Programming Exercises with JACK“

No 27 (December 2008)
Schauer, Carola: "Größe und Ausrichtung der Disziplin Wirtschaftsinformatik an Universitäten im deutschsprachigen Raum - Aktueller Status und Entwicklung seit 1992"

No 26 (September 2008)
Milen, Tilev; Bruno Müller□Clostermann: " CapSys: A Tool for Macroscopic Capacity Planning"

No 25 (August 2008)
Eicker, Stefan; Spies, Thorsten; Tschersich, Markus: "Einsatz von Multi-Touch beim Softwaredesign am Beispiel der CRC Card-Methode"

No 24 (August 2008)
Frank, Ulrich: "The MEMO Meta Modelling Language (MML) and Language Architecture – Revised Version"

No 23 (January 2008)
Sprenger, Jonas; Jung, Jürgen: "Enterprise Modelling in the Context of Manufacturing – Outline of an Approach Supporting Production Planning"

No 22 (January 2008)
Heymans, Patrick; Kang, Kyo-Chul; Metzger, Andreas, Pohl, Klaus (Eds.): "Second International Workshop on Variability Modelling of Software-intensive Systems"

2007

No 21 (September 2007)
Eicker, Stefan; Annett Nagel; Peter M. Schuler: "Flexibilität im Geschäftsprozess-management-Kreislauf"

No 20 (August 2007)
Blau, Holger; Eicker, Stefan; Spies, Thorsten: "Reifegradüberwachung von Software"

No 19 (June 2007)
Schauer, Carola: "Relevance and Success of IS Teaching and Research: An Analysis of the ‚Relevance Debate'

No 18 (May 2007)
Schauer, Carola: "Rekonstruktion der historischen Entwicklung der Wirtschaftsinformatik: Schritte der Institutionalisierung, Diskussion zum Status, Rahmenempfehlungen für die Lehre"

No 17 (May 2007)
Schauer, Carola; Schmeing, Tobias: "Development of IS Teaching in North-America: An Analysis of Model Curricula"

No 16 (May 2007)
Müller-Clostermann, Bruno; Tilev, Milen: "Using G/G/m-Models for Multi-Server and Mainframe Capacity Planning"

No 15 (April 2007)
Heise, David; Schauer, Carola; Strecker, Stefan: "Informationsquellen für IT-Professionals – Analyse und Bewertung der Fachpresse aus Sicht der Wirtschaftsinformatik"

No 14 (March 2007)
Eicker, Stefan; Hegmanns, Christian; Malich, Stefan: "Auswahl von Bewertungsmethoden für Softwarearchitekturen"

No 13 (February 2007)
Eicker, Stefan; Spies, Thorsten; Kahl, Christian: "Softwarevisualisierung im Kontext serviceorientierter Architekturen"

No 12 (February 2007)
Brenner, Freimut: "Cumulative Measures of Absorbing Joint Markov Chains and an Application to Markovian Process Algebras"

No 11 (February 2007)
Kirchner, Lutz: "Entwurf einer Modellierungssprache zur Unterstützung der Aufgaben des IT-Managements – Grundlagen, Anforderungen und Metamodell"

No 10 (February 2007)
Schauer, Carola; Strecker, Stefan: "Vergleichende Literaturstudie aktueller einführender Lehrbücher der Wirtschaftsinformatik: Bezugsrahmen und Auswertung"

No 9 (February 2007)
Strecker, Stefan; Kuckertz, Andreas; Pawlowski, Jan M.: "Überlegungen zur Qualifizierung des wissenschaftlichen Nachwuchses: Ein Diskussionsbeitrag zur (kumulativen) Habilitation"

No 8 (February 2007)
Frank, Ulrich; Strecker, Stefan; Koch, Stefan: "Open Model - Ein Vorschlag für ein Forschungsprogramm der Wirtschaftsinformatik (Langfassung)"

2006

No 7 (December 2006)
Frank, Ulrich: "Towards a Pluralistic Conception of Research Methods in Information Systems Research"

No 6 (April 2006)
Frank, Ulrich: "Evaluation von Forschung und Lehre an Universitäten – Ein Diskussionsbeitrag"

No 5 (April 2006)
Jung, Jürgen: "Supply Chains in the Context of Resource Modelling"

No 4 (February 2006)
Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part III – Results Wirtschaftsinformatik Discipline"

2005

No 3 (December 2005)
Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part II – Results Information Systems Discipline"

No 2 (December 2005)
Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part I – Research Objectives and Method"

No 1 (August 2005)
Lange, Carola: „Ein Bezugsrahmen zur Beschreibung von Forschungsgegenständen und -methoden in Wirtschaftsinformatik und Information Systems"

Research Group	Core Research Topics
Prof. Dr. H. H. Adelsberger Information Systems for Production and Operations Management	E-Learning, Knowledge Management, Skill-Management, Simulation, Artificial Intelligence
Prof. Dr. P. Chamoni MIS and Management Science / Operations Research	Information Systems and Operations Research, Business Intelligence, Data Warehousing
Prof. Dr. F.-D. Dorloff Procurement, Logistics and Information Management	E-Business, E-Procurement, E-Government
Prof. Dr. K. Echtle Dependability of Computing Systems	Dependability of Computing Systems
Prof. Dr. S. Eicker Information Systems and Software Engineering	Process Models, Software-Architectures
Prof. Dr. U. Frank Information Systems and Enterprise Modelling	Enterprise Modelling, Enterprise Application Integration, IT Management, Knowledge Management
Prof. Dr. M. Goedicke Specification of Software Systems	Distributed Systems, Software Components, CSCW
Prof. Dr. T. Kollmann E-Business and E-Entrepreneurship	E-Business and Information Management,E-Entrepreneurship/ E-Venture, Virtual Marketplaces and Mobile Commerce, Online-Marketing
Prof. Dr. B. Müller-Clostermann Systems Modelling	Performance Evaluation of Computer and Communication Systems, Modelling and Simulation
Prof. Dr. K. Pohl Software Systems Engineering	Requirements Engineering, Software Quality Assurance, Software-Architectures, Evaluation of COTS/Open Source-Components
Prof. Dr.-Ing. E. Rathgeb Computer Networking Technology	Computer Networking Technology
Prof. Dr. A. Schmidt Pervasive Computing	Pervasive Computing, Uniquitous Computing, Automotive User Interfaces, Novel Interaction Technologies, Context-Aware Computing
Prof. Dr. R. Unland Data Management Systems and Knowledge Representation	Data Management, Artificial Intelligence, Software Engineering, Internet Based Teaching
Prof. Dr. S. Zelewski Institute of Production and Industrial Information Mangement	Industrial Business Processes, Innovation Management, Information Management, Economic Analyses

For more information visit us on the Web: **http://www.icb.uni-due.de**

ISSN 1860-2770 (Print)
ISSN 1866-5101 (Online)