On the Relation Between SACK Delay and SCTP Failover Performance for Different Traffic Distributions

Johan Eklund, Anna Brunstrom  
Department of Computer Science  
Karlstad University, Sweden  
Email: {Johan.Eklund, Anna.Brunstrom}@kau.se

Karl-Johan Grinnemo  
TietoEnator AB  
S-651 15 Karlstad, Sweden  
Email: Karl-Johan.Grinnemo@tietoenator.com

Abstract—The Stream Control Transmission Protocol (SCTP) is an important component in the ongoing evolution towards IP in the fixed and mobile telephone networks. It is the transport protocol being used in the ongoing deployment of IETF’s signaling transport (SIGTRAN) architecture for tunneling of traditional telephony signaling traffic over IP. Further, SCTP represents an alternative for future SIP signaling traffic. Key to the success of SCTP is its ability to recover from network failures, in particular failed network paths. SCTP includes multihoming and a failover mechanism which should swiftly shift from a failed or unavailable network path to a backup path. However, several studies have shown that SCTP’s failover performance is dependent on factors both related to protocol parameters and network conditions. This paper complements these studies by providing a comprehensive evaluation of the impact of SACK delay under various traffic distributions. The results show a clear relation between the traffic distribution and the impact of the SACK delay on SCTP failover performance. Severe negative effects are observed for low intensity traffic composed of individual signaling messages. On the other hand, our results show limited impact of SACK delay for high intensity and bursty traffic. Furthermore, the results show a limited increase in network traffic by reducing the SACK delay at low traffic intensities and even less impact on network traffic at high traffic intensities. Based on these results we recommend a decrease of the SCTP SACK timer to a small value in signaling scenarios.

I. INTRODUCTION

Telephony customers expect sustainable connections without delay or any other interference during their calls. To properly negotiate call performance during set up and to tear down a call, signaling traffic is sent over the network. The transport of the signaling messages is crucial for the call performance. In most of today’s telecom networks the signaling transport is carried out by the Signaling System No.7 (SS7). The current trend of substituting SS7 for IP, makes robustness a challenge, since IP serves as an unreliable best effort protocol. To address this challenge the Internet Engineering Task Force (IETF) formed the signaling transport (SIGTRAN) working group, which defined an architecture for transport of traditional telephony traffic (PSTN) over IP [11]. The transport protocol Stream Control Transmission Protocol (SCTP) [15] [16], is a core component in this architecture. SCTP has also been pointed out as an interesting alternative to TCP for SIP traffic [14]. Furthermore, SCTP has been selected as the transport protocol to be used for the transport of Diameter messages in the IP multimedia subsystem (IMS) [1] of the future all-IP mobile core network.

Since the transition from the traditional TDM-based signaling to IP will not happen overnight, the two networks will have to cooperate and the success of SCTP relies on its ability to recover from network failures and to offer a service comparable to the performance of the TDM network. To discover a network failure, SCTP includes a failover mechanism which should swiftly shift from a failed or unavailable network path to a backup path. As shown by Grinnemo [3], the requirements on link failure detection and response give an upper limit on the SCTP failover time in between 1.4 and 2.9 seconds for SS7 application traffic, which is the target traffic in this study. Although there are no explicit recommendations in SS7 on the maximum acceptable transfer time for a message, studies made on SS7 application protocols, such as the ISDN user part (ISUP) [7] and the Transaction Capabilities Application Part (TCAP) [6], suggest that the maximum transfer time for a message is in the range of 600 ms - 1000 ms [3]. In case this time is exceeded the signaling application performance may deteriorate.

Several studies have, however, shown that SCTP’s failover performance is dependant on several factors. Jungmaier et al. [9] studied failover performance for traffic comprising small independent messages. They recommended a more strict tuning of the SCTP failover parameters compared to the recommendations in [17], to detect path failure earlier and make the protocol comply with the application demands. The parameters they pointed out were the maximum value for the retransmission timeout (RTO_{max}) and the maximum number of allowed consecutive retransmissions, Path.Max.Retrans (PMR). Further, a similar study by Grinnemo and Brunstrom [4] found 3 to be the maximum acceptable value for PMR to have SCTP comply with the SS7 signaling application demands. However, none of these studies considered the
impact of the selective acknowledgment\(^1\) (SACK) delay [17], a mechanism to reduce the network traffic by holding the acknowledgments for a specified time before transmission, which is standard in most implementations today.

In an earlier study [8] by the first two authors of this paper a negative impact of the SACK delay on failover performance was observed. The results pointed out the RTO at the time of failure to have a substantial impact on the failover performance. Further, the study found no interaction effects between the configuration of the SACK delay and the PMR. However, that study was performed with a limited, single static traffic distribution, which motivates this study where we use a more varied traffic scenario. This paper presents a more balanced and complete view of the impact of the SACK delay on the failover performance under different traffic conditions.

The results show a clear relation between the impact of the SACK delay on the failover performance and the traffic distribution. For low intensity traffic with independent messages sent over the network, which could represent signaling traffic in the network during lightly loaded periods, the SACK delay indeed shows a severe negative impact on the failover performance. This negative effect is seen on both the failover time and on the maximum transfer time for a message. These results support the results in [8]. On the other hand, for more intense traffic and for bursty traffic, this negative impact is not seen. Furthermore, the results indicate that a reduction of the SACK timer to a value close to zero implies hardly no increase in network traffic if traffic intensity is high and only a limited increase in traffic if traffic intensity is low. Disabling SACK delay results in a higher improvement but also results in more network traffic.

The remainder of this paper is organized as follows. In Section 2, SCTP and its failover mechanism is further described. In Section 3, the experimental parameters are presented and motivated together with a description of the experimental setup. Section 4 presents and analyzes the results achieved during the experiments. Finally, Section 5 concludes the paper.

II. BACKGROUND ON SCTP AND SCTP FAILOVER

SCTP is a reliable transport protocol, initially developed to meet the telephony signaling requirements, concerning robustness and timing. The original motivation behind the development of SCTP was in the SIGTRAN architecture to serve as a transport protocol to tunnel traditional SS7 signaling traffic over IP [11]. Still, during the development process, SCTP has evolved to become a general purpose transport protocol. The protocol is specified in RFC4960 [15].

SCTP inherits most of its features from the predominant reliable transport protocol on the Internet, the Transmission Control Protocol (TCP) [12]. For example, the congestion control of SCTP is similar to TCP congestion control [2]. Furthermore, SCTP employs a SACK scheme similar to SACK TCP [10]. Further, the recommendation in RFC 4960 is to set the SACK timer to 200 ms to reduce the network traffic. This means that if one packet arrives to the sender the SACK is delayed for a maximum of 200 ms or until a second packet arrives. One extension in SCTP is the multihoming facility, where more than one interface could be used in the same session. Multihoming was introduced as a way to enhance end-to-end robustness for a session. An illustration of a multihomed scenario is shown in Fig. 1: A terminal, called Source in the figure, sends data to the Destination over a dual homed session.

In a multihomed session all data is, under normal conditions, sent on the path designated as primary. All other paths serve as backup paths, where only so called heartbeats are sent at regular intervals to probe reachability. The way SCTP detects a path failure is by keeping track of missing SACKs at the sender. The reason for a missing SACK may be a path failure or congestion in the network. The challenge for the SCTP failover mechanism is to distinguish between these two reasons, and decide when to abandon the primary path and continue the transfer on the alternate path.

To decide when to switch over to an alternate path, the sending SCTP host keeps an error counter to count the number of consecutive missing SACKs. In case of a missing SACK, either a fast retransmit will be triggered, or the retransmission timer will expire. In case of a retransmission timeout, the error counter for the transfer is incremented by one, and the data not yet acknowledged is retransmitted on one of the alternate paths. At this retransmission, several non full size packets may be bundled into one packet. New traffic continues to be sent on the primary path.

The differentiation between congestion and path failure is solved by holding a discrete control parameter, Path.Max.Retrans (PMR), for each destination. A path is considered unavailable and abandoned if the error counter exceeds the value of PMR. From this point on the transfer of messages continues on the alternate path.

An illustration of a failover scenario in a dual-homed session is shown in Fig. 2, where the primary path is seen to the left and the alternate path to the right. Packets are sent from the sender to the receiver, and SACKs are sent back. After the link failure, some more packets are sent on the primary path, but they never reach the destination. Eventually, the retransmission timer times out (Timeout 1). All not yet acknowledged data is retransmitted on the alternate path, and new data is sent on the

\[^1\]With selective acknowledgments, the data receiver can inform the sender about all segments that have arrived successfully, so the sender needs to retransmit only the segments that have actually been lost.
primary path. After PMR+1 consecutive timeouts the primary path is abandoned and all data is sent on the alternate path. The failover time is the time elapsed between the link failure on the primary path and the failover to a new path occurs.

III. INVESTIGATION OF FAILOVER PERFORMANCE

In most of today’s implementations of SCTP, the value of the SACK delay is set to 200 ms, the value recommended in RFC4960. The motivation behind this is to reduce the network traffic. To evaluate the impact of this parameter on the SCTP failover performance, we conducted experiments in an emulated network. The scenarios used for the investigation in this paper do not represent specific application scenarios, but instead illustrate a range of plausible scenarios. Signaling applications usually generate small individual messages to be sent over the network. Many signaling applications are request-response type applications, where messages are sent from one host, awaiting response before the next message is transmitted. In a request-response scenario traffic is bidirectional. As the acknowledgement is piggybacked on the response, the SACK timer will have marginal impact on the transmission time for this type of traffic. This is particularly the case if the processing of the request at the receiver in this scenario is quick.

However, not all signaling applications are captured by this scenario. For example TCAP [6], one of the SS7 applications, is a multi-purpose request-response protocol, with considerable processing time before a response is sent. Furthermore, not all SS7 applications are transaction oriented, e.g., ISUP [7], why this investigation is representative for that type of applications and for transaction-oriented applications with long processing times. Further, the scenario in this study could be representative for a SCTP session between gateways in a core signaling network where a path failure has to be detected and recovered before signaling endpoint retransmissions.

The network used in the experiments is depicted in Fig. 3. A dual-homed SCTP association was set up between the source and the destination. Further, the characteristics of both the primary and the alternate paths were emulated by machines running Dummynet [13], denoted “Network Emulator” in Fig. 3.

![Failover scenario](image)

![Network Setup](image)

The Source and the Destination machines were two PCs which both ran the Linux 2.6.16 kernel. An application on the source machine served as traffic generator, and an application on the destination machine as traffic sink. The experiments were managed by the Admin machine, which also logged traffic. Initially, all data was sent on the primary path. When traffic had stabilized, a failure was emulated, and the failover procedure started. The primary path never became available again during the same experimental run. Both the time it took to recover from a path failure, the failover time, and the maximum message transfer time (MMTT), metrics of great importance for signaling, were measured during the experiments.

It is not possible to exactly measure the failover time due to the lack of an exact knowledge of when the failure occurs. In our study, we measured the failover time as the time between the command was issued by the Admin machine to take down the primary path, until the traffic generator was notified about a failover to the alternate path. The transfer time for each message was measured by time-stamping its departure from the traffic generator and its arrival at the traffic sink. The MMTT in a test run was then derived from these message transfer times.

Since our interest was on the impact of the SACK delay on the failover performance, we considered this parameter under a number of traffic conditions. The experiment was run with three SACK delay settings. We used the default value of 200 ms [17]. Further, we ran the experiment without SACK delay, and finally, we used a SACK delay of 40 ms, which has been recommended by some telecommunication companies.

Two types of traffic were considered; exponentially distributed traffic and exponentially distributed burst traffic. Both types of traffic used a fixed message size of 250 bytes. The

2From this point on control messages, Heartbeats, are sent on the primary path to control reachability. As soon as a Heartbeat is acknowledged on the primary path the transfer is switched back and continued on the primary path.

3The clocks of the two machines were synchronized with a common time-server using NTP.
messages/message bursts were generated at different mean intervals, varying from 5 ms to 80 ms. This was done to see the impact of the SACK delay at different traffic intensities. The single message traffic pattern was chosen to imitate signaling traffic during varying traffic loads; the low-traffic intensity modeling periods with low customer activity. For completeness, and to check if the results found for single messages also hold for other types of traffic, the experiments were run with messages sent in bursts. However, this traffic was not sent with the highest intensity (5 ms mean burst interval) to avoid queuing in the network. The bursts had a uniformly distributed length of 1 to 5 messages.

The presented results is for a one-way-delay of 40 ms, which represents a plausible intra-continental session. Since the recommendations in RFC 4960 have been shown to be too liberal for signaling traffic [9], the (RTO\textsubscript{min}) was set to 80 ms which is low enough never to be reached with a one-way link delay of 40 ms. The (RTO\textsubscript{max}) was kept at the value of the RFC (60000 ms), to not disable the dynamic aspects of the congestion control. Further, the wide range of the RTO timer gave the possibility to notice if the traffic pattern and the SACK delay had some impact on the failover performance. Since signaling traffic is usually sent over a logically separate network, and to point out the impact of the SACK delay, no traffic but the evaluated traffic was sent over the network in the experiment. Further, a bandwidth of 5 Mbps was used, high enough not to have a limiting impact on the performance.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>EXPERIMENTAL PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message size</strong></td>
<td>250 Bytes</td>
</tr>
<tr>
<td><strong>Mean burst interval (ms)</strong></td>
<td>5, 10, 20, 40, 60, 80</td>
</tr>
<tr>
<td><strong>Link Delay (ms)</strong></td>
<td>40</td>
</tr>
<tr>
<td><strong>Bandwidth</strong></td>
<td>5 Mbps</td>
</tr>
<tr>
<td><strong>SACK delay (ms)</strong></td>
<td>0, 40, 200</td>
</tr>
<tr>
<td><strong>Burst size (messages)</strong></td>
<td>varied 1-5</td>
</tr>
<tr>
<td><strong>RTO\textsubscript{init} (ms)</strong></td>
<td>3000</td>
</tr>
<tr>
<td><strong>RTO\textsubscript{min} (ms)</strong></td>
<td>80</td>
</tr>
<tr>
<td><strong>RTO\textsubscript{max} (ms)</strong></td>
<td>60000</td>
</tr>
<tr>
<td><strong>PMR</strong></td>
<td>2, 3, 4, 5</td>
</tr>
</tbody>
</table>

The PMR parameter was varied between 2 and 5. However, the focus is on a PMR value of 2, which has been shown to be a reasonable value to prevent spurious failovers [4] [5] and to have the protocol comply with the signaling application demands. All parameters used in the study are shown in Table 1.

IV. EXPERIMENTAL RESULTS

Each experiment was repeated 40 times and the graphs found in this section show the mean values of these 40 repetitions together with the 95% confidence intervals.

A. Exponentially distributed traffic

As a traffic pattern representative for signaling traffic, we injected individual messages into the network at different,
exponentially distributed, intervals. The results from these experiments can be seen in Fig. 4, where Fig. 4(a) shows the mean failover time as a function of the mean interval between the messages. In the figure it is seen that there is no difference between 200 ms SACK delay, 40 ms SACK delay and no SACK delay when the mean message interval is small, however, a significant difference is shown already when the mean message interval is 10 ms. The difference between no SACK delay and 200 ms SACK delay increases constantly with increasing mean message intervals, while the graph representing a SACK delay of 40 ms grows in conformity with the graph representing 200 ms SACK delay until a mean message interval of 20 ms. From this point on, this graph shows a more conservative increase and from the point of 40 ms mean message interval the difference between no SACK delay and 40 ms SACK delay is almost constant. Disabling the SACK delay keeps the failover time almost constant independent of message interval. It is also clearly seen in the figure that a high SACK delay means an increasing variation between the different repetitions, resulting in larger confidence intervals.

In Fig. 4(b), the RTO times at the moment of failure as a function of mean message intervals are shown. The relation between the failover time and the RTO times at the time of failure is easy to see when comparing Fig. 4(a) and Fig. 4(b) as the shapes of the graphs have strong similarities. This relation was also observed in [8]. A high RTO at the time of failure will mean a long failover time. Further, it is obvious in the figure that a high SACK delay will increase the risk of a long RTO at the time of failure.

The reason behind the different shapes of the graphs in fig. 4(b), is found in the different impact of the SACK delay at different traffic intensities. If the traffic intensity is high, then packets arrive at the destination almost continuously. This means that the SACK delay is almost never activated, since for every second packet arriving a SACK is generated. If, on the other hand, the traffic is more moderate, then packets arrive at the destination less frequently. If the SACK delay is high, this causes the destination to hold the SACK for a packet until the next packet arrives or the SACK timer expires. This means a longer RTT for these messages, which results in a higher RTO. If the SACK delay is not used, then a SACK is generated for every packet which keeps the RTO-timer, and thus the failover time, constant irrespective of message interval.

When looking carefully at figure 4(a), it is seen that, when no SACK delay is used, the failover times are higher as the average message intervals are short (5 ms), compared to a longer message interval (10 ms). This difference is not found in fig. 4(b). This increase in failover time is not caused by a changed RTO time, but by the traffic intensity itself. If the traffic intensity is high, that means that new messages are queued at the sender during the failover procedure. This prolongs the failover procedure, since all not-yet-acknowledged messages have to be retransmitted on the alternate path before new data is transmitted. Thus, this increase in failover time is not related to the SACK delay.

Fig. 4(c) presents the MMTT for a message as a function of mean message intervals. Also these results follow almost the same pattern as the results above. It is seen that running the system without SACK delay consistently performs best. Further, if the SACK delay is high the mean MMTT as well as the size of the confidence interval, increases rapidly with increasing inter-message intervals. This increase is not seen when the SACK delay is set to 40 ms or 0 ms. The graphs representing the lower SACK delays also show a slight decrease as the message intervals increase. This decrease is due to lower bandwidth utilization. In this figure the impact of message queuing during the failover procedure at high traffic load, noticed in fig. 4(a), is even more prominent. Too intense signaling traffic over the same SCTP session may cause longer MMTTs in a failure situation.

In a scenario where individual signaling messages are sent over the network, it is from these results evident that the default SACK delay of 200 ms may have a severe negative impact on the failover time and on the MMTT in case of network failure. Using the default SACK delay may jeopardize the possibility to meet the demands from signaling applications in cases of light traffic load. A compromise of 40 ms SACK delay improves performance but still has a negative impact compared to no SACK delay that in the experiments have shown to stabilize at a constant level as the message interval increases.

B. Exponentially Distributed Burst Traffic

To give a complete view of the impact of SACK delay for different traffic distributions and to verify if the results found for exponentially distributed traffic were generally applicable, a set of experiments was conducted with exponentially distributed burst traffic with bursts of varying size. The results from this scenario is found in Fig. 5.

When examining these results, a consistent improvement is still found when running the experiment without SACK delay in terms of both failover time, (Fig. 5(a)) and MMTT, (Fig. 5(c)). However, the improvement compared to the results when using SACK delays of 40 ms and 200 ms is limited. The failover times and the MMTT’s stay in the same magnitude for all SACK delays as the mean burst intervals exceed 10 ms. Further, it is seen that the confidence intervals for this type of traffic remains small as the burst intervals increase, irrespective of SACK delay. The impact of the SACK delay is thus much less prominent for this traffic distribution.

The reason behind the significant increase of the MMTT for high intensity burst traffic, i.e. a mean burst interval of 10 ms, is the same as for high intensity message traffic. As the failover procedure continues, no traffic reaches the destination without retransmission. With high intensity traffic, this means queuing of messages at the sender before transmission. This effect is of course more prominent with this traffic distribution, where message bursts, instead of individual messages, are generated at a given average rate. This effect is, as mentioned in the previous subsection, not related to the SACK delay, but to traffic dimensioning.
Fig. 5. Failover Performance. Exponentially distributed burst traffic

The reason behind the different behavior found for bursty transmissions compared to individual messages is found in the way the RTT measurements were performed in the system. RTT measurements were performed only once per round trip. The RTT timer was typically started by the first packet in a burst. As the data reached the destination the SACK was sent from the receiver almost directly as the second packet arrived. Consequently, for bursts larger than one the SACK delay never timed out for the message for which the RTT timer was started. This almost immediate generation of a SACK as the burst arrived eliminates the impact of the SACK delay. In bursts of odd number larger than one (size three or five) the last packet did not generate an immediate SACK. Instead the SACK was delayed until either the SACK timer timed out or until the first packet in the next burst arrived to the destination. This delay did, however, not affect the RTO since the RTT was only calculated on the first message in the burst. Thus, the RTO-timer remained almost constant, irrespective of burst interval and SACK delay, which is clearly seen in Fig. 5(b).

The graphs in Fig 5(b) are, nevertheless, not exactly the same. The major reason for this minor difference between the results for the three SACK delays is that a small portion of the bursts was of size one. The SACKs for these bursts were delayed, which in some cases influenced the RTO.

C. Cost for reduction of the SACK timer

Disabling or reducing the value of the SACK timer may create extra network traffic due to extra acknowledgements compared to the default timer of 200 ms. To quantify the extra network traffic, we have analyzed the log file from a sample run of every experiment, and calculated the number of SACKs received in relation to the number of data packets sent on the primary path before failure. This was executed for every SACK delay and for every mean burst interval used in the experiment. The results from these calculations are found in Table II and Table III.

<table>
<thead>
<tr>
<th>Mean burst interval (ms)</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
</tr>
</thead>
<tbody>
<tr>
<td>SACK Delay (ms)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>0.49</td>
<td>0.48</td>
<td>0.48</td>
<td>0.48</td>
<td>0.47</td>
<td>0.45</td>
</tr>
<tr>
<td>40</td>
<td>0.49</td>
<td>0.49</td>
<td>0.51</td>
<td>0.58</td>
<td>0.57</td>
<td>0.60</td>
</tr>
<tr>
<td>0</td>
<td>0.97</td>
<td>0.97</td>
<td>0.96</td>
<td>0.93</td>
<td>0.90</td>
<td>0.86</td>
</tr>
</tbody>
</table>

The results for single messages are shown in Table II. Here it is seen that disabling the SACK delay results in approximately one, SACK per data packet. Disabling the SACK delay is expected to generate exactly one SACK per data packet. The reason the rate is lower than one is that after failure, but before failover, more data is sent on the primary path, but the SACKs are never received. This reduction becomes slightly more prominent as the mean message intervals increase since this results in fewer messages being transmitted before failure, why the previously mentioned impact is greater. If the default
A SACK delay of 200 ms is used, it is seen that the SACK rate in relation to the number of data packets is approximately 0.5, irrespective of traffic intensity. This means that almost every SACK sent back to the source of the traffic acknowledges 2 data packets, and the SACK timer almost never times out.

When looking at the number of SACKs generated in relation to the number of data packets when SACK delay is set to 40 ms, it follows that the rate of SACKs follows the trend for a SACK timer of 200 ms provided the intra-message intervals are low. This is intuitive, since close time gaps between the arrival of messages at the receiver never causes the SACK timer to time out. As the message intervals increases, the SACK timer of 40 ms times out more frequently, which results in an increasing rate of SACKs in relation to generated data messages.

Table III shows the results from the burst traffic. Here, it is clearly seen that a SACK delay of 40 ms results in approximately the same rate of SACKs as for a SACK delay of 200 ms. Furthermore, it is seen that for both these SACK delays the rate of generated SACKs is approximately 0.5, irrespective of traffic intensity. These results are intuitive since a burst with more than one message will generate a SACK immediately as the burst reaches the destination, why the delayed SACK is less of an issue in these situations. Further, in the table it is seen that also for this type of traffic, disabling the SACK delay will result in approximately one SACK per data packet. These results are in line with the results found for individual messages above and indicates that a reduction of SACK delay from 200 ms to 40 ms comes at a very low cost.

The results in this subsection imply that also the relative cost in extra network traffic of reducing the SACK timer is dependent on both the traffic pattern and the traffic intensity. Furthermore, the results show that the extra cost from reducing the SACK timer only occurs in the scenarios when there is a possible gain in failover time by reducing the SACK delay, i.e. as the traffic intensity is low.

D. Impact of PMR

Although not considered in detail here, additional experiments were conducted with different PMR values. The motivation behind this was to verify that the presented results were representative for other values of PMR as well.

In earlier studies [4], [8], [9], it has been shown that the PMR has a direct impact on both the failover time and on the MMTT. This impact on the failover performance is clearly seen also in this study. In Figure 6, the failover times for a mean message/burst interval of 20 ms are shown as a function of different PMR values. In Fig. 6(a), the results representing transfer of individual messages are shown. Here it is seen that the SACK delay has a significant impact on the failover time, irrespective of PMR. Further, the graphs in Fig. 6(b) show the results for bursty traffic. These results verify the limited impact of the SACK delay for this traffic type. To further verify the results in the study, we also conducted experiments with different link delays. Although not shown here, the results from these experiments confirm the results concerning the impact of SACK delay in relation to the traffic patterns presented in this study. The results follow the same trend irrespective of link delay.
V. CONCLUSIONS

In this study the focus has been on investigating the relation between the SACK delay and different traffic distributions on the SCTP failover performance. The results show that using the default value of the SACK delay may have a severe negative impact, on both the failover time and on the maximum message transfer time in some cases. Furthermore, it is seen that the impact of the SACK delay on the failover performance is heavily dependant on the traffic pattern and on the traffic distribution. For low-intensity signaling traffic, consisting of small individual messages, the negative impact of a long SACK delay is significant. This negative impact of SACK delay is, however, not seen for more intense traffic of individual messages or for bursty traffic. Furthermore, the results show that a reduction of the SACK delay from 200 ms to 40 ms only results in a marginal increase in network traffic and the increase is seen only as the traffic intensity is low.

In a failure situation a large SACK delay may jeopardize the protocol’s ability to fulfill the application timing demands and thus decline customer satisfaction. A reduction of the default SACK delay value is therefore desirable. We have, in this study, shown that reducing the SACK delay to 40 ms comes at a very low cost in terms of extra network traffic. Furthermore, we see that extra network traffic is generated only in the cases where a possible improvement in failover performance is seen.

Based on these results, we recommend reducing the SACK delay to a lower value compared to the default of 200 ms. It is desirable not to disable the SACK delay completely, since this always results in an increase of network traffic, still a reduction of the SACK timer to a value close to zero gives a performance gain at limited cost.

ACKNOWLEDGMENT

The authors would like to thank the Flux Research group at the University of Utah for providing the Emulab [18] testbed. The work has been supported by grants from the Knowledge Foundation of Sweden with TietoEnator and Ericsson as industrial partners.

REFERENCES