Computer Networks 57 (2013) 3967–3986

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

# Performance analysis and improvement of PR-SCTP for small messages

Mohammad Rajiullah *, Reine Lundin, Anna Brunstrom, Stefan Lindskog

*Department of Computer Science, Karlstad University, Universitetsgatan 2, SE-651 88 Karlstad, Sweden*

ARTICLE INFO

*Article history:*
Received 12 October 2012
Received in revised form 10 September 2013
Accepted 26 September 2013
Available online 6 October 2013

*Keywords:*
PR-SCTP
SCTP
NR-SACKs
Event logging
Syslog
Performance evaluation

ABSTRACT

PR-SCTP, a partially reliable extension of SCTP, provides a flexible QoS trade-off between timeliness and reliability for application traffic. However, the performance of PR-SCTP can be affected by certain traffic characteristics and network scenarios. We previously proposed an NR-SACK based optimization for PR-SCTP. In this work, we extensively evaluate and analyze the performance of PR-SCTP with NR-SACKs using different network scenarios. Moreover, we compare the performance of our NR-SACK based PR-SCTP with existing transport protocols for syslog traffic. In the evaluation, we use real syslog traces from an operational syslog system. The results indicate that NR-SACK based PR-SCTP significantly improves the overall message transfer delay as compared to SCTP and TCP.



## 1. Introduction

The transport layer provides several end-to-end communication services such as multiplexing, reliability and flow control for applications. TCP and UDP are the most common protocols at this layer. However, to facilitate Internet telephony, a new transport protocol called the Stream Control Transmission Protocol (SCTP) [55] was defined by the IETF Signaling Transport (SIGTRAN) working group. Although SCTP was initially designed to address TCP's shortcomings [26] in transporting telephony signaling (i.e. SS7) messages over IP-based networks, IETF later standardized it as a general purpose transport protocol for the Internet. Different OS kernel implementations such as FreeBSD, MAC OS X, Linux and Windows (SctpDrv4) include SCTP. Services such as flow control, congestion control and reliability in SCTP overlap with SACK TCP [39]. However, unlike TCP, SCTP supports additional advanced transport services like multi-streaming, multi-homing, partial ordering and partial reliability that application developers can utilize to define future applications.

TCP and UDP serve disjoint sets of applications. They have an all or nothing approach to the reliability requirements of applications. TCP gives a fully reliable service for applications such as email, file transfer and web browsing that do not require timeliness. In contrast, UDP is commonly used for DNS resolution and time services that do not impose reliability but rely heavily on fast transport. However, a different set of applications including for instance real time multimedia distribution require services that fall between what TCP and UDP provide. Since TCP places reliability over timeliness, loss recovery during congestion periods can increase the end-to-end delay, which may violate the real time requirements of these applications. In consequence, applications that favor timeliness over reliability rely on UDP. However, the lack of congestion control in UDP may lead to congestion collapse in the Internet if the usage of these applications grows continuously [24].

* Corresponding author. Tel.: 46 54-700 17 58.
*E-mail addresses:* mohammad.rajiullah@kau.se (M. Rajiullah), reine.lundin@kau.se (R. Lundin), anna.brunstrom@kau.se (A. Brunstrom), stefan.lindskog@kau.se (S. Lindskog).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.09.018

Congestion control mechanisms are often implemented at the application layer. However, congestion control itself is not a trivial problem to solve and requires extensive testing and maturity before being deployed [25]. Furthermore, UDP provides no guarantees for the delivery of data. This is again problematic for applications such as real time multimedia that often use high compression, i.e., low bit rate encoding. For example, the Real-time Transport protocol (RTP) [51] on top of UDP does not retransmit lost packets. In this case, loss of important data may result in poor play back quality at the receiver. All these shortcomings in TCP and UDP lead to the development of partially reliable transport protocols that give a flexible trade-off between timeliness and reliability for an application. With partial reliability it is possible for an application to define reliability requirement per application data unit. This is useful for any application that generates data of heterogeneous priorities. If some of the data has relatively higher importance than the rest, overall end-to-end delay can be reduced if only important data are considered during loss recovery or when network resources are congested. For instance, time sensitive data are only useful at the receiver if they are delivered before their deadline. In this case, avoiding retransmission of expired data can improve the application performance. All in all, if the loss rate in the network does exceed the applications' sensitivity to loss, partial reliability can improve application performance [37].

PR-SCTP [53] is a partially reliable extension of SCTP. By using PR-SCTP, it is possible to give varying levels of reliability according to the heterogeneous QoS requirements of individual application data units. In this case, a specific reliability level dictates when PR-SCTP should stop (re)transmission of the associated data. An application may thus trade timeliness for reliability, since this reliability level is specified at the application layer. This could be particularly useful when, for instance, the availability of network resources is dynamic. Several applications, including real time audio or video streaming [6,42,59,60], IPTV transmission [33] and SIP signaling [61], have been shown to benefit from PR-SCTP. Currently, several IETF working groups such as Real-Time Communication in WEB-browsers (rtcweb) [2,32] and IP Flow Information Export (IPFIX) [8] are considering PR-SCTP for partial reliability.

In contrast to previous work on PR-SCTP, our research shows that the improvement when using PR-SCTP can be limited in several circumstances. In our earlier work [47,48], we showed that packet based buffering at the network may degrade application performance when application message sizes are small. In this case, the loss rate per application byte is increased, which degrades application throughput. Moreover, when small messages are bundled into a SCTP packet, for instance, due to congestion control, loss of this packet introduces bursty loss. In this particular instance, the key mechanism in PR-SCTP, the Forward Cumulative TSN (*forward_tsn*) mechanism, becomes inefficient. We identified this problem in [47] and the results of our further experiments with various traffic characteristics in [48] verified the inefficiency. We therefore proposed an optimization of PR-SCTP that uses Non-Renegable Selective Acknowledgement (NR-SACKs) [3,63]. Results of our small scale experiments in [48] suggested a significant improvement of application performance when using NR-SACKs in PR-SCTP.

In this work, we extensively evaluate and analyze the performance of NR-SACK based PR-SCTP for different network scenarios. Moreover, we evaluate NR-SACK based PR-SCTP as a transport for the syslog system [20]. In the evaluations, we use real syslog traces from an operational network. Our evaluations show that the modified PR-SCTP can better trade timeliness for reliability as compared to SCTP, TCP and UDP.

The remainder of the paper is organized as follows. The background to partial reliability and related work on PR-SCTP are described in Section 2. In Section 3, the *forward_tsn* inefficiency is described using an illustrative experiment. Our proposed NR-SACK based optimization for solving the inefficiency in the *forward_tsn* mechanism is described in Section 4. The implementation and performance evaluation of our solution are also described in this section. Section 5 discusses our study on syslog and the performance evaluation of NR-SACK based PR-SCTP for syslog. Finally, concluding remarks are given in Section 6.

## 2. Related work

In this section, we describe some background to partial reliability along with PR-SCTP. We also mention some of the applications that have shown to benefit from PR-SCTP. Finally, we mention the NR-SACK mechanism.

### 2.1. Partial reliability

Traditional transport layer services provided by TCP and UDP fall short of the special QoS requirements of applications such as distributed multimedia networking. For instance, an MPEG flow with frames with different importance requires different QoS for each frame. TCP guarantees full reliable and in order delivery of all packets in a flow, whereas UDP guarantees neither reliability nor in order delivery of packets. Applications that require full reliability for only some of their data find TCP's full reliability and in order delivery unnecessary at the expense of extra delay, although TCP like congestion control is important for the stability of the Internet. The same application cannot resort to UDP because of faster delivery, since no reliability at all is guaranteed. Instead, many applications solve the delivery by implementing transport layer functionalities on top of UDP. However, Li et al. [35] argued that the transport mechanisms should still be at the transport layer given that there are more flexible mechanisms available to specify policies and requirements than in the application layer.

Furthermore, Conrad in [15], argued that the implementation of transport layer functionalities in user land is not trivial; especially the implementation of congestion control may exceed the complexity of the application itself. In this case, each new application may need to reinvent the wheel from the transport layer. It necessitates a new transport layer that has, for instance, today's TCP like congestion control, flow control and avoids the extra delay of fully reliable service, but guarantees minimal reliability