Performance Analysis and Improvement of PR-SCTP in an Event Logging Context

Mohammad Rajiullah
Performance Analysis and Improvement of PR-SCTP in an Event Logging Context

Mohammad Rajiullah
Performance Analysis and Improvement of PR-SCTP in an Event Logging Context
Mohammad Rajiullah

LICENTIATE THESIS
Karlstad University Studies  I  2012:44

ISSN 1403-8099

© The author

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

Print: Universitetstryckeriet, Karlstad 2012

WWW.KAU.SE
Performance Analysis and Improvement of PR-SCTP in an Event Logging Context

MOHAMMAD RAJULLAH
Department of Computer Science, Karlstad University, Sweden

Abstract

Much of the Internet’s successes rely on the underlying connectionless, best effort Internet Protocol (IP). On top of IP, transport protocols provide several end-to-end services to applications. TCP and UDP are the most dominant transport protocols in the Internet today. IP based networks are cheaper and more flexible than circuit switched networks. Telecommunication companies therefore prefer IP based replacements to compete with Voice over IP (VoIP) applications. Due to some shortcomings in TCP and UDP, the Stream Control Transmission Protocol (SCTP) was defined for transporting telephony signaling traffic (i.e. SS7). SCTP provides advanced features such as multi-homing, multi-streaming, partial reliability and partial ordering. The partially reliable extension of SCTP, PR-SCTP, has been considered a candidate for prioritizing content sensitive traffic and trading reliability against timeliness for applications such as streaming multimedia, IPTV transmission and SIP signaling. Using PR-SCTP, an application can choose (re)transmission policy on a per message basis. A special mechanism called forward_tsn is used to provide partial reliability.

In this thesis, we investigate the applicability of PR-SCTP for event logging applications. Event logs are inherently prioritized. This makes PR-SCTP a promising candidate for transporting event logs. Our investigation, however, suggests that the performance gain of PR-SCTP can be very limited when application messages are of small sizes, have mixed reliability requirements and are bundled due to congestion control. According to our analysis, several factors influence PR-SCTP’s performance. One key factor is the inefficiency in the forward_tsn mechanism of PR-SCTP. We in our work examine the forward_tsn inefficiency in detail and propose several solutions to improve the performance. We also implement and evaluate one solution that takes advantage of the Non-Renegable Selective Acknowledgements (NR-SACKs) mechanism currently under standardization in the IETF and available in the FreeBSD operating system. Our results show a significant performance gain for PR-SCTP with NR-SACKs. In some scenarios the average message transfer delay is reduced by more than 75%. Moreover, we use real traces from syslog, which is the most common event logging application, to evaluate NR-SACK based PR-SCTP. NR-SACK based PR-SCTP significantly improves the syslog application performance as compared to SCTP, TCP and UDP.

Keywords: SCTP; PR-SCTP; NR-SACKs; event logging; performance evaluation
Acknowledgements

First and foremost, I would like to take the opportunity to thank my supervisor, Professor Anna Brunstrom, for giving me the opportunity to pursue my doctoral studies and for continuously providing me with intellectual support and encouragement throughout my work. Her deep knowledge, detailed and constructive comments and critical thinking have been a great value in my work.

I wish to express my sincere thanks to my co-supervisor, Professor Stefan Lindskog, for his detailed review and excellent advice during the preparation of this thesis.

My warm thanks are due to all my colleagues at the Computer Science Department at Karlstad University for their valuable suggestions regarding my research and help in my daily life. I am grateful to all the present and past members of the distributed systems and communications research group (DISCO). Extensive and constructive discussions in the DISCO group during reading courses have provided me a good basis in my research in networking.

I am grateful to compare business innovation center (C-BIC) for financial support during my research.

I owe my loving thanks to my wife Farhana Akter and my son Ryan Rajiullah. My wife has always been helpful and inspiring during my work. It would have been impossible to finish my work without their support. Lastly, my deepest appreciation goes to my mother, Ferdous Ara Begum, my brother, Md. Rifat Ullah, my sister, Tahira Ferdous, and my grandmother, Hasina Begum, for their unending encouragement and love for me.

In memory of my father,

Karlstad, Fall 2012

Mohammad Rajiullah
List of Appended Papers

The thesis is based on the work presented in the following four papers. Reference to the papers will be made using the Roman numbers associated with the papers.


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

Comments on my Participation

Paper I I was responsible for carrying out the experimental evaluations and for the written material. The other co-authors helped me developing the underlying ideas in the paper and gave useful comments in reviewing.

Paper II I was responsible for all the experiments in the paper. I authored most of the written material except the syslog data modeling section. Reine Lundin authored that section. The other co-authors helped me during the development of the ideas in the paper and with suggestions in reviewing.

Paper III I was responsible for carrying out all the experiments and for all the written parts. My co-author, Anna Brunstrom, collaborated with me developing the ideas in the paper. Anna also contributed constructive comments during the review process.

Paper IV I was responsible for all the experiments in the paper and the implementation of the NR-SACK based PR-SCTP optimization in the FreeBSD operating system. I authored most of the paper, except that Stefan Lindskog authored the section on event logging and Reine Lundin
authored the section describing the syslog specific application scenario. Reine also prepared the primary syslog trace that I employed in the use case specific experiments in the paper. All the co-authors helped me developing the ideas in the paper. They also contributed during the review process.

Other Publications


Contents

**INTRODUCTORY SUMMARY**

1 Introduction 3

2 Research Objectives 5

3 Related Work 6
  3.1 Partial Reliability 6
  3.2 Applications using PR-SCTP 8
  3.3 Event Logging 10
  3.4 Performance of Small Packets 10
    3.4.1 Overhead for Small Messages 10
    3.4.2 Byte Based or Packet Based Buffering 11
  3.5 NR-SACKs 12

4 Research Methodology 12

5 Main Contributions 14

6 Summary of Papers 15

7 Conclusions and Future Work 16

**PAPER I**
Priority Based Transport Service for Syslog Messages 23

1 Introduction 25

2 Background 26
  2.1 Syslog 26
  2.2 Transport Service for Syslog 28

3 PR-SCTP as a Transport Service for Syslog Messages 29

4 Performance Evaluation 31
  4.1 Experiment Setup 31
  4.2 Experimental Results 32

5 Conclusion 35

**PAPER II**
Syslog Performance: Data Modeling and Transport 39

1 Introduction 41
Introductory Summary
1 Introduction

A research experiment on packet switched networks in the early 1960s laid the dream of an ubiquitous network spanning the whole globe to serve all our communications needs that would later carry a wide range of different traffic from data and image to voice to video and virtually bring together all the people in the world. Internet is the reality of that dream. It started as ARPANET with a few low bandwidth links connecting a small number of computers to share their resources. Since then, the size of the Internet has grown, and continues to grow, exponentially.

A digital revolution through the Internet has changed many aspects of our daily lives, from the way we do business to the way we spend our leisure time. Today’s business is very dependent on the Internet, which has been the most efficient media to reach billions of consumers through a large number of commercial applications. The recent trend of using mobile devices to connect to the Internet has further increased its popularity. What is the magic in the Internet that brings the whole world of information to our fingertips? The phenomenal success of the Internet comes from the revolution in the computer networking technology that started in the 1960s. The Internet Protocol (IP) [1], which lies at the core of this networking technology, indirectly connects billions of computing devices in the Internet. IP works as the magic key to unlock the mysterious web of interconnected heterogeneous devices and networks in the Internet.

Complexity in computer networking is dealt with by layering. The IP layer provides a best effort service for internet applications. Due to its multiplexing nature, packet switching often exhibits packet loss. Higher layer applications therefore use a separate layer called the transport layer to cope with the connectionless best effort service provided by IP. The transport layer provides different end-to-end services to applications. The most common transport layer services are multiplexing, reliability, flow control, congestion control etc. Transmission Control Protocol (TCP) [2] and User Datagram Protocol (UDP) [3] are the most dominant transport layer protocols in the Internet.

TCP provides in order, flow controlled, reliable delivery, whereas UDP provides no flow controlled, unreliable delivery for application data. Specific Quality of Service (QoS) requirements of application data govern the transport layer selection. Several applications such as email, web browsing and music or movie download specifically need reliability and therefore choose TCP. In contrast, applications including DNS resolve and time server access require faster communication to a greater extent than reliability and therefore choose UDP. Besides, IP based networks are cheaper and more flexible than circuit switched networks. Telecommunication companies prefer IP based replacements to compete with Voice over IP (VoIP) applications. Although researchers in the Internet Engineering Task Force (IETF) initially experimented with both TCP and UDP as the transport for telephony signaling (i.e. SS7), shortcomings in these transports led to the development of a new transport protocol.
called the Stream Control Transmission Protocol (SCTP) in 2000 [4]. SCTP is message based. Unlike TCP or UDP, SCTP provides additional advanced features such as multi-homing, multi-streaming and partial ordering. These features were later found useful for other applications as well. Consequently, IETF standardized SCTP as a general purpose transport layer protocol.

The reliability requirement of an application is only settled in an all or nothing way by today’s transports of TCP or UDP. However, some applications require partial reliability, which falls between what TCP and UDP provide. For instance, real time multimedia applications primarily claim timeliness. Full reliability and in order delivery in TCP are expensive for these applications due to the real time requirements. UDP often becomes the resort. However, the lack of congestion control in UDP can be challenging for the stability in the Internet [5]. Furthermore, UDP is unreliable. Multimedia applications may require reliable delivery of critical data for good play back quality at the receiver. In contrast to TCP and UDP, partial reliability can offer a flexible trade-off between timeliness and reliability. Retransmissions were long considered impractical for real time multimedia. Earlier work by Dempsey [6] and Papadopoulos [7], however, establish the feasibility of retransmission based partial reliability for multimedia traffic. Using partial reliability in real time multimedia, only the most important data are considered for retransmission during loss recovery. This ensures high quality decoding at the receiver. Besides, avoiding some retransmissions is useful for an overall improvement in the experienced delay.

Partial reliability can also be beneficial for other types of time sensitive data. Time sensitive data that reach the receiver beyond its time boundary are often useless. Only those data that may still reach the receiver within the time boundary are eligible for retransmission. Overall, given that an application generates data with different priorities, prioritization during loss recovery not only reduces general data transfer delay but also ensures a reliable delivery for the most important data.

PR-SCTP [8] is a partially reliable extension to SCTP. It provides partial reliability on a per message basis. Applications can specify particular reliability requirements for each message using PR-SCTP. Real time multimedia applications have been shown to benefit from PR-SCTP [9, 10] by trading reliability for timeliness when network resources are congested. Other applications such as IPTV transmission [11] and SIP signaling [12] also show performance improvement using PR-SCTP. In addition, several IETF working groups, such as real-time communication in WEB-browser (rtcweb) [13, 14] and IP Flow Information Export (IPFIX) [15], also consider PR-SCTP for partial reliability.

Log messages generated in computer systems are inherently priority based. Syslog [16] is probably the most common event logging system. Syslog messages are prioritized on the basis of the source and the level of importance. Some messages become more important than others during loss recovery and should be treated differently for overall performance improvement. In the thesis, we propose PR-SCTP as a promising candidate for transporting event
logs. However, our investigation suggests that the performance of the existing PR-SCTP can be limited for certain traffic and network characteristics.

Firstly, if an application generates small messages, packet based buffering in the network can increase the loss rate per application byte. Besides, small messages increase the overhead. Finally, the most important limitation in PR-SCTP is an inefficiency in the key mechanism called the forward cumulative transmission sequence number or \textit{forward\_tsn} mechanism. This limitation manifests itself when for instance a large number of messages is lost in sequence. Loss recovery of these messages becomes difficult if their reliability requirements are different, meaning that only a fraction of these messages require retransmission. In such a case, PR-SCTP is hindered in its transmission due to an inefficiency in the \textit{forward\_tsn} mechanism.

In the thesis, we propose a couple of mitigations to improve the efficiency of the \textit{forward\_tsn} mechanism. One of the solutions takes advantage of non-renegable selective acknowledgements (NR-SACKs) [17, 18]. The NR-SACK mechanism is currently being standardized in the IETF and is available in the FreeBSD operating system. We have implemented NR-SACK based PR-SCTP in FreeBSD 8.2. Our evaluations show a significant performance improvement with NR-SACK based PR-SCTP as compared to existing PR-SCTP. In certain scenarios, the message transfer delay is reduced by over 75% when NR-SACK based PR-SCTP is used. In the thesis, we also evaluate NR-SACK based PR-SCTP as a transport service for syslog messages. The evaluations indicate that NR-SACK based PR-SCTP reduces the average message transfer delay as compared to TCP, SCTP and the existing PR-SCTP. Besides, unlike UDP, the proposed method is as reliable as TCP or SCTP for high priority messages.

2 Research Objectives

The objective of this thesis is to study and evaluate the possible benefits of prioritization of messages at the transport layer. The primary application is event logging.

As stated above, PR-SCTP is a standardized transport protocol that provides partial reliability. We therefore choose PR-SCTP for prioritizing application messages at the transport layer. In the thesis, we mainly focus on syslog as an event logging system. To carry out our overall research objective, we focus on a number of sub objectives.

We consider several traffic characteristics and network scenarios to evaluate and analyze the performance of PR-SCTP as a transport for syslog messages. The traffic characteristics include both fixed and variable message sizes, where the variable message size distribution is derived from real syslog traces. Network scenarios, on the other hand, include both artificial loss based and competing flow based networks. We consider message transfer delay as the main performance metrics. Our first research objective is

\textit{To analyze the applicability of PR-SCTP for syslog.}
We found that quite a few factors, most importantly application messages being small, limit PR-SCTP's performance. Our next research objective is

*To determine the factors that influence the performance of PR-SCTP for small messages.*

Lastly, our research objective is

*To enhance the performance of PR-SCTP for small messages.*

### 3 Related Work

This section describes the most relevant research related to the work presented in this licentiate thesis. Although partial reliability was standardized as PR-SCTP in 2004 [8], its origin dates back to the early 1990s. We here present some of the related work on partial reliability along with PR-SCTP. In our thesis, we particularly look into syslog as a prospective application for PR-SCTP. There are many other applications that have been shown to benefit from PR-SCTP. We summarize them here as well. We also include a subsection on event logging to clarify the usefulness of PR-SCTP as its transport. In addition, there is some background work where partial reliability has been found appropriate for syslog. We therefore also describe these here. Moreover, in our work, we find that PR-SCTP performance is influenced by message sizes. There is some related work that looks into the importance of message size from both the application and network perspectives. We present them here as well. Lastly, we mention the NR-SACK mechanism that we use to optimize the performance of PR-SCTP for small messages.

#### 3.1 Partial Reliability

Traditional transport layer services of TCP and UDP fall short because of the continuously increasing demand of multimedia oriented and heterogeneous user traffic. These traffic types are, for instance, MPEG flows with multiple frames with different QoS requirements, real time gaming traffic with both critical status update and non critical object location and state information etc. TCP or UDP is a choice between extremes for heterogeneous traffic. TCP provides reliability, flow control and in order delivery at the expense of extra delay and reduced throughput. This extra delay might be inconvenient for continuous media, video or audio, with real time requirements. However, the congestion control in TCP has been tested for more than two decades and is necessary for the stability in the internet.

In contrast, UDP provides connectionless, unreliable delivery for user data. UDP might be acceptable unless the loss in the underlying network goes beyond the application’s tolerance of loss. For instance, loss of a high priority frame in an MPEG flow can reduce the playback performance at the receiver.
Moreover, UDP does not provide congestion control. Continuous growth of applications using UDP could lead to a congestion collapse in the internet [5]. While neither TCP nor UDP is appropriate, application developers often choose to implement transport functionalities on top of UDP, i.e. by using RTP [19].

Building transport mechanisms at the application level is non trivial. An implementation of round trip time estimation, congestion control, flow control etc. will increase the complexity of an application. Moreover, the stability of some functionalities such as congestion control requires wide scale testing before deployment. One solution is to specify the reliability requirements at the application layer, whereas the transport layer should implement the requirements along with the TCP like congestion control and flow control services [20]. Applications should be able to specify partial reliability requirements according to a flexible trade-off between delay and reliability.

There was a common view that retransmission is particularly infeasible for real time multimedia transmission prior to the pioneer work of Dempsey [6] and Papadopoulos [7] that describe the feasibility of retransmissions using the right playout buffer size. Dempsey et al. [6, 21] developed a retransmission based error control service, partially error controlled connection (PECC). PECC extends the unreliable express transfer protocol (XTP) [22]. Using PECC, an application can specify its loss tolerance, which enables an XTP receiver to occasionally ask the sender for retransmissions that fulfill the minimal reliability requirements while keeping the overall delay low. Papadopoulos et al. [7] developed a similar protocol where a receiver asks for retransmissions only if delay requirements are not likely to be violated. Furthermore, Conard et al. [23] proposed POCv2 as an extension of the original partial order connection (POC) [24, 25]. Message based POCv2 allows any application to assign each application message one of the three reliability classes: reliable, partially reliable and unreliable. These messages correspond to audio clips, video frames etc. A receiver asks for retransmissions based on the reliability assignments.

Li et al. [20] proposed heterogeneous packet flows (HPF), a sender based partially reliable transport protocol. Similar to POCv2, HPF allows an application to specify particular reliability requirements for data units. However, unlike PECC or POCv2, retransmission decisions are made by the sender based on the reliability requirements of the lost data. Furthermore, HPF seeks for network assistance to drop low priority packets during congestion.

PR-SCTP [8], an extension of SCTP [4], is a sender based partially reliable protocol. Like SCTP, PR-SCTP is message based. It provides partial reliability on a per message granularity based on application level specifications. User messages are called data chunks when they are placed in an SCTP packet. On the other hand, chunks containing control information for an SCTP association are called control chunks. Constrained by the maximum transfer unit (MTU), a number of data chunks and control chunks can be bundled into an SCTP packet. Utilizing the chunk building function in SCTP, ordered, unordered, reliable, partially reliable and unreliable user messages can be multiplexed over a single SCTP association.
In PR-SCTP, applications set different reliability services for different messages. *Timed reliability* is an example of such reliability services. In this case, the application sets a lifetime value with every message before forwarding it to the PR-SCTP layer. Upon expiration of this lifetime value, PR-SCTP simply abandons the message and does not (re)transmit it. Instead, it sends a special control chunk called *forward_tsn* that tells the receiver to advance its cumulative ACK point and to no longer expect that particular abandoned message. This might be useful for time sensitive traffic that becomes useless if it is not transferred before a specific time. This also implies that, during periods of congestion, no expired messages are retransmitted. Network resources are only used for fresh and unexpired messages. Furthermore, if only important messages are given a longer lifetime, overall transfer time can be improved since only these messages will be retransmitted when network resources are scarce. Although PR-SCTP may not retransmit a lost message, it responds to the congestion signal in order to be fair with other competing flows in the network. In addition, a PR-SCTP receiver does not require any knowledge of the particular partial reliability service employed at the sender. A detailed survey of partially reliable transport protocols is given in [26].

### 3.2 Applications using PR-SCTP

PR-SCTP has been considered largely for real time multimedia streaming [9, 10, 27–38]. Both MPEG and H.264/AVC [39] encoding can be used for source coding in multimedia streaming. Target scenarios are low bit rate video conferencing, remote video surveillance, interactive gaming etc. Both MPEG and H.264/AVC encode source data into three different types of frames: I-frame (intra coded frame), P-frame (predicted frame), and B-frame (bi-directional frame). Multimedia streaming, both audio and video, generally has strict latency requirements but does not require completely reliable transport protocols. However, an I-frame contains the most important information for media playback performance at the receiver and is therefore preferred for reliable delivery. Using the timed reliability service in PR-SCTP, applications can set longer lifetimes for I-frames as compared to P- or B-frames. In consequence, unnecessary delays are saved by not considering retransmission for P- or B-frames, while media playback performance is ensured by giving reliable delivery for I-frames.

In [9], the authors evaluated PR-SCTP for multimedia transmission in a mobile network scenario. PR-SCTP performed close to UDP in terms of message transfer delay, as unreliable frames were not considered for retransmission. The delay was much lower than the case in which TCP was used because TCP retransmits all the lost frames. On the other hand, UDP may suffer from the poor media quality since no frames are retransmitted. This was evaluated in [27]. In addition, PR-SCTP performance was optimized in [29] by limiting the duplicate acknowledgement threshold to one for low intensity MPEG-4 traffic.
PR-SCTP has been proposed as the transport for VoIP traffic in [40, 41] primarily to reduce head-of-line blocking delay on a per-stream basis by partial reliability. SCTP’s multi-homing features [4] are used to support handover among multiple IP interfaces. Evaluations show that PR-SCTP can support high-quality calls during handover. A PR-SCTP based handover solution was also used in a system called messenger on-the-drive (MOD) [42] to provide vehicular communication in multi-network environments. The use of PR-SCTP has further been shown to improve the failover delay for real-time traffic as compared to basic SCTP. In this case, during failover in basic SCTP, start-up delay on the new path causes further delays to the retransmissions of the data that were lost over the old path. Furthermore, concurrent multipath transfer (CMT) [18] in SCTP is sensitive to reordering, which might find CMT a difficult choice for transmitting real-time data such as VoIP or video streaming. A large amount of data may become useless at the receiver. The authors of [43] proposed PR-SCTP based CMT (PR-CMT) with timed reliability based partially reliable service. Simulation results show an improvement of transmission efficiency as compared to basic CMT. PR-SCTP has also been proposed in [11] as the transport for Internet protocol television (IPTV) traffic. In the evaluations, PR-SCTP provides a shorter transfer completion time than TCP or fully reliable SCTP and a better reception ratio than UDP.

Furthermore, in [12], the authors proposed PR-SCTP as the transport for session initiation protocol (SIP) messages. Partially reliable and unreliable deliveries in PR-SCTP are used according to the SIP message types and their reliability requirements. For instance, SIP provisional response messages, such as “180 Ringing” and “100 Trying”, are informative and are therefore transmitted unreliably. Other messages are transmitted using partial reliability with their lifetimes set as the application layer timeout values. Simulation results show that PR-SCTP improves the performance of SIP transport under both high and low levels of SIP traffic as compared to both UDP and SCTP.

The aim of Rtcweb [13] is to enable peer-to-peer exchange of arbitrary application data among web browsers. Use cases include gaming, real-time text, real-time audio/video, file transfer, etc. Rtcweb needs PR-SCTP to provide both reliable and unreliable delivery semantics [14]. For instance, in multiplayer network gaming, position and object state information that are only consistent for a short time can be delivered unreliably or with partial reliability. However, critical state information in the game or non-real-time file transfer must be delivered reliably.

IPFIX [15] is an IETF standard for collecting Internet flow information from routers, probes and other devices. It facilitates obtaining measurements, accounting and billing information for operators. PR-SCTP is the mandatory transport for IPFIX messages, while the use of either TCP or UDP is optional. Primary reasons are multi-streaming that reduces head-of-line blocking delay [4] and per-message partial reliability. For instance, although reliable delivery is required for IPFIX messages from security and billing applications, a capacity planning-related application can use partially reliable delivery due to its loss tolerance.
3.3 Event Logging

Logs in computing systems contain entries about specific events that have occurred in a system or a network. Various applications and operating systems on servers, clients or other networking devices are common sources of logs in computer networks. Logs are used for various purposes, for instance system optimizations, trouble shooting or investigating security related issues. Logs are either stored at the source devices or sent over the network to some centralized server. Since log records contain information about the severity level of the event, they can be prioritized. Therefore, a partially reliable transport is a natural choice for log delivery across computer networks.

The syslog protocol [44] is commonly used for event logging in UNIX like operating systems. Using syslog, any machine or device can send event notification messages over networks to the syslog server. The initial syslog specification [45] includes unreliable UDP transport for sending syslog messages. However, in the more recent RFC 3195 [46], TCP has been specified for reliable log message delivery. Reliable delivery of syslog messages is important, since critical log messages may otherwise be lost in the network. Administrators may be unaware of serious problems developing in the system. However, reliable delivery has some potential drawbacks. A syslog sender or relay can be blocked once the receiver is unable to accept any messages [16]. This is particularly problematic in Unix/Linux like operating systems where a syslog originator or relay runs inside a high priority process, e.g. syslogd. Blocking such a process may cause a system wide halt. RFC 5424 [16] suggests implementing the reliable delivery for syslog messages in a way such that a sender can discard messages that may otherwise block the sender. In this case, senders or relays are recommended to discard low priority messages in favor of high priority messages.

Furthermore, an application based prioritized retransmission of syslog messages during loss recovery has been proposed in [47]. In this proposal, however, senders, do not discard any messages. Instead, high priority message losses are detected and retransmitted faster than low priority messages, which ensures timely delivery of high priority messages.

3.4 Performance of Small Packets

In our work, we see the influence of small packets on the performance of PR-SCTP. The influence of small packets has also been discussed in several existing works. First we discuss the overhead for small messages that may lead to small packets. We then discuss the implication of packet size in buffering mechanisms at the network.

3.4.1 Overhead for Small Messages

The primary target area for SCTP was telephony signaling. Byte oriented TCP is inconvenient for telephony signaling messages. These messages are small and
Performance Analysis and Improvement of PR-SCTP

need special application layer marking for correct interpretation at the receiver. SCTP was therefore designed as a message oriented protocol [4].

User messages are called data chunks when they are placed in an SCTP packet. Each SCTP packet has a 12 byte common header and each chunk has a 16 byte chunk header. Considering a 20 byte IP header, the overhead for a small user message can be substantially high. This is a classical problem and was noticed in the Tymnet network in the 1960s [48]. John Nagle offered an elegant solution to solve this small packet problem in TCP. In Nagle’s algorithm, user data are buffered until a full sized packet is formed or until all outstanding data are acknowledged. The Nagle algorithm later became a requirement for TCP [49] and is included today in most TCP implementations.

The SCTP protocol specification in RFC 4960 [4] does not specify Nagle’s algorithm for SCTP. However, the socket API extensions for SCTP [50] include a socket option, ‘SCTP_NODELAY’, to turn on/off any Nagle like algorithm to reduce the overhead from small packets. If this option is turned on, several small chunks can be bundled together in a single SCTP packet. SCTP, in addition, provides general bundling capability. For example, when user messages are queued up in the sender buffer due to congestion control, they are bundled. Although some overhead is reduced after bundling, the use of chunk headers still contributes overhead. In this case, the overhead of an SCTP packet depends on the size of the user messages and the number of user messages that are bundled together. This overhead impacts system performance. Results from [51] show the throughput increase in SCTP with the increase of user message size.

3.4.2 Byte Based or Packet Based Buffering

Internet is built on packet switching technology. Due to statistical multiplexing in switches, several packets may arrive in a switch at any point in time in a form of transient burst. Buffers are placed at switches to absorb this burst and ultimately reduce packet loss and improve resource utilization. Since buffers can accommodate only a finite number of packets, any switch or router generally either marks [52] or simply drops packets when its resources are congested. This works as a feedback to end hosts. Eventually, transport protocols at the end hosts detect the congestion and respond accordingly.

Drop-tail is the simplest and a traditional queuing mechanism in Internet routers. In this mechanism, queues are filled when congestion occurs, and packets are only dropped when all buffer space is used up. There are several shortcomings in the drop-tail queuing mechanism [53]. Advanced queue managements, such as random early detection (RED) [54], are recommended. Instead of using the instantaneous queue size as in drop-tail, the average queue size over time is used in RED before dropping or marking packets. However, drop-tail routers are still believed to exist [55], especially in access network equipment and in middle boxes like firewalls.

Regardless of the particular queuing mechanism used, some concerns are common. It is possible to measure the level of congestion in the router either
in terms of bytes or in terms of packets [53]. In several scenarios, a particular choice impacts system performance.

In byte based calculations, packet sizes are taken into account. Small packets are more likely to have space in the buffer during congestion. According to Sally Floyd’s discussion [56] cited from RFC 2309 [53], particularly in RED, queue size should be measured in packets given that the respected queue capacity is in units of packets, and in bytes given that the respected queue capacity is in units of bytes. The same source [56] recommends byte based dropping in general, since the scarce resource is link bandwidth in bytes per second. Byte based dropping reduces the drop of small packets such as control packets. However, ongoing work in the IETF [55] disapproves of the favorable treatment of small packets as it might induce DoS vulnerabilities.

3.5 NR-SACKs

SACKs allow a SCTP receiver to acknowledge out of order data [4]. The SACK mechanism is also available for TCP. SACK information is only advisory in SCTP and TCP. The information is therefore only used for selective retransmissions. The sender must keep the selectively acknowledged or sacked data until they are cumulatively acknowledged, since the receiver is allowed to discard sacked data. Discarding sacked data is known as ‘reneging’.

Unlike TCP, which only provides in order delivery, SCTP provides unordered delivery. In such a case, out of order data at the SCTP receiver can be forwarded to the receiving application. These data are non-renegable. However, the SACK mechanism in SCTP does not provide a distinction between sacked data that are renegable and sacked data that are non-renegable.

NR-SACK [17, 18] is a recently proposed acknowledgement mechanism for SCTP that can explicitly provide the renegibility information for sacked data. The NR-SACK chunk, as shown in Figure 1, is an extension to the regular SACK chunk in SCTP. Beyond the information in a regular SACK chunk, an NR-SACK chunk can selectively acknowledge out of order data as non-renegable because either the data have already been delivered to the receiver application or the receiver takes full responsibility. ‘NR Gap Ack’ Blocks in the figure can include non-renegable out of order data. Currently, the NR-SACK mechanism is being standardized in the IETF along with CMT-SCTP [18].

4 Research Methodology

Theoretical computer science basically studies what can be computed and the associated cost. On the other hand, experimental computer science uses scientific methods for its inquiries. Scientific method is not strictly defined. However, in most cases, scientific method refers to a cycle of observation, hypothesis building or description, and experimental testing for the verification of prediction.

In this iterative scientific method, a hypothesis is built as a tentative answer to the question formulated in the observation stage. This hypothesis is then
verified in the experiment stage. The scientific method as illustrated in Figure 2 has been used as a research methodology in this thesis. In this model, the observation stage consists mainly of a literature review. This phase may employ published works to identify the relevant problems in which one researcher is interested. Further, the state of the art or the highest level of development for the relevant problem is pointed out in this stage. Once the observation phase is complete and a problem statement or research question is stated, a hypothesis is formulated. This hypothesis generally delivers a feasible answer to the research questions and allows making predictions based on some assumptions about the system under consideration. The next stage is hypothesis testing. This is called experiment and performance evaluations.

There are several methods, such as real measurements, simulation, emulation or analytical methods, that are used in hypothesis testing. Measuring real systems represents the lowest possible level of abstraction but is also the most difficult technique. One of the goals of performance analysis is to characterize the system as certain parameters are varied. Since it is very difficult to change parameters in running systems, real measurements are rarely used. Moreover, the measurements themselves may perturb the system, which is also quite difficult to isolate. Therefore, other kinds of experiment techniques are preferable in many cases.

A simulation is a computer program that models important features of a system under test. Since this is a computer program, varying several parameters is comparatively easy. However, considerable effort is needed to write and debug a reasonably sized simulation program. Besides, the model used might ignore a critical behavior of the system, improperly handled initial conditions may lead to incorrect conclusions or too much simplification of assumptions may limit the accuracy of the end result. Nevertheless, simulation is quite popular because of its high degree of flexibility and its relative ease of implementation [58].
Hypothesis

Existing theories and observations

Predictions

Experiments

New theory

Not consistent, modify hypothesis

Consistent

Figure 2: Diagram describing the iterative nature of the scientific method, adapted from [57].

Since both simulation and real measurements have their downsides, a mixture of real and simulated entities called emulation is becoming more and more common. Emulation is quite frequently used in networking research. In this case, real machines with real implementations of networking stacks communicate over a simulated network. Dummynet [59] is one such emulator that can emulate different network scenarios. In our work, an emulation based experiment method is chosen. To test our hypothesis, an emulation based experiment is done where part of the underlying network characteristics is only simulated.

Finally, an analytical method can also be used for hypothesis testing. An analytical method mathematically models a system under investigation. As compared to the above mentioned methods, results from analytical models can be less accurate [58]. However, an analytical method can give quick insight into the overall behavior of the system. Besides, an easy to perform analysis can simply be used to coarsely verify the results from other methods.

5 Main Contributions

We study and evaluate the performance of PR-SCTP for syslog in different network scenarios. To do this, we analyze and model real syslog traces from an operational system as an input in our evaluations. Our evaluations suggest that PR-SCTP performance is impeded when message sizes are small. We then determine that the key mechanism in the existing PR-SCTP, the *forward_tsn* mechanism, becomes inefficient in the presence of small messages. Additionally, small messages increase overhead. Small messages may also lead to small packets. However, when packet based buffering is used in the network, every packet,
Performance Analysis and Improvement of PR-SCTP

regardless of its size, has the same risk of being dropped. In such an instance, PR-SCTP encounters a higher loss rate per application byte when network resources are shared among competing flows.

We propose two enhancements to mitigate the inefficiency in the *forward_tsn* mechanism. We implement and evaluate one of the proposed solutions that takes advantage of the NR-SACK mechanism. NR-SACKs are available in the current FreeBSD operating system. The NR-SACK mechanism is currently also being standardized in IETF as a part of concurrent multipath transfer (CMT) in SCTP [18]. In our evaluation, NR-SACK based PR-SCTP reduces the average message transfer delay by more than 75% as compared to existing PR-SCTP in some scenarios.

The further evaluation of NR-SACK based PR-SCTP for syslog shows a significant performance improvement as compared to existing PR-SCTP, TCP and SCTP. NR-SACK based PR-SCTP provides a shorter average message transfer delay. Further, in contrast to UDP, it provides reliable delivery of high priority log messages. All in all, using NR-SACK based PR-SCTP, a syslog application can have a flexible trade-off between timeliness and reliability.

6 Summary of Papers

**Paper I–Priority Based Delivery of PR-SCTP Messages in a Syslog Context**

In this paper, we discuss the problem with the existing transport services such as TCP and UDP for syslog. We also describe several features of SCTP in relation to the syslog protocol and suggest PR-SCTP as a transport alternative for syslog. In our emulation based experimental results, PR-SCTP shows better performance than TCP in terms of the average delay for message transfer. Furthermore, PR-SCTP exhibits less average packet loss than UDP. In both cases, PR-SCTP exploits priority properties of syslog messages during loss recovery. However, the study reported in this paper is quite restricted. We chose a fixed message size for syslog. Therefore, the observation of PR-SCTP performance is limited. Additionally, we only emulate an artificial loss scenario in the network without any competing traffic.

**Paper II–Syslog Performance: Data Modeling and Transport**

In this paper, we first model syslog data using real syslog traces from an operational network. The model includes several traffic parameters such as message size, interarrival time and fraction of important messages. The model is then used as an input in the performance evaluation of PR-SCTP. In the experiments, real congestion is introduced in the network by running several competing flows. Furthermore, in our previous work, we assumed a message size of 250 bytes as an approximation for syslog messages. In contrast, our study of real syslog traffic exhibits a far lower mean message size. Our evaluations show that PR-SCTP performance is heavily influenced by the syslog data size characteristics.
Paper III—On the Effectiveness of PR-SCTP in Networks with Competing Traffic

Based on the findings in paper II, a broad evaluation of PR-SCTP for different network scenarios and traffic characteristics is presented in this paper. We find that a number of factors can influence the performance of PR-SCTP. Firstly, small messages increase the overhead, which impacts the message transfer delay. Secondly, small messages can lead to small or less than full sized packets when they are bundled. When packet based buffering is used in the network, these packets may raise the loss rate per application bytes as compared to other competing flows that use full-sized packets. Lastly and most importantly, the forward_tsn mechanism in PR-SCTP becomes inefficient, particularly when messages with different reliability requirements are lost in bursts. Furthermore, we propose an enhancement of PR-SCTP that requires an extension of the existing forward_tsn chunk to mitigate the inefficiency in existing PR-SCTP.

Paper IV—Performance Analysis and Improvement of PR-SCTP for Small Messages

In [60], we propose, implement and initially evaluate an enhancement to improve the forward_tsn efficiency in the existing PR-SCTP. The enhancement takes advantage of the NR-SACK mechanism in FreeBSD OS. The NR-SACK mechanism is currently being standardized in IETF [18]. An NR-SACK chunk can selectively acknowledge out of order but non-negligible data. In this paper, we extensively evaluate and analyze NR-SACK based PR-SCTP for different network scenarios and traffic characteristics. According to our analysis, using NR-SACKs improves the forward_tsn mechanism in PR-SCTP, particularly when application messages are small, have mixed reliability requirements and are bundled due to congestion control. We further investigate syslog traces collected from a bigger operational network than described in paper II. We then perform a trace based evaluation to compare the performance of NR-SACK based PR-SCTP with existing transport protocols. Our evaluation suggests a significant improvement in terms of average message transfer delay in PR-SCTP using our NR-SACK based optimization as compared to the existing PR-SCTP, TCP and SCTP.

7 Conclusions and Future Work

PR-SCTP can improve application performance by trading timeliness for reliability when the application can tolerate some loss. In this thesis, we have investigated the applicability of PR-SCTP for syslog. In our evaluation, however, PR-SCTP exhibits performance penalty when syslog messages are small with heterogeneous reliability requirements and bundled into packets due to congestion control. The existing forward_tsn mechanism in PR-SCTP becomes inefficient when these messages, bundled in a packet, are lost. We have the-
fore proposed and implemented a solution to improve the efficiency in the forward_tsn mechanism. The proposed solution utilizes the NR-SACK mechanism of SCTP. In the evaluation, NR-SACK based PR-SCTP improves the application’s performance in general. Moreover, NR-SACK based PR-SCTP shows improved performance as compared to the existing transport protocols SCTP, TCP and UDP for syslog traffic delivery.

There is a further possibility for improving PR-SCTP performance if expired messages can be discarded from the send buffer. This can occur when several messages are queued up during a congestion period. In this case, message lifetimes may expire due to long waiting times in the queue. We are planning to modify the SCTP send buffer management functionalities as a part of our future work so that discarding messages from the send buffer is possible. Moreover, we are aiming to modify a real syslog system to use NR-SACK based PR-SCTP. In this thesis, we have considered only a timed reliability based PR-SCTP service. A mapping of reliability requirements for different types of log messages to several PR-SCTP services is also a part of future work. It might require defining and standardizing new PR-SCTP services.

In a more general context, there is a growing need for supporting the timeliness aspects of applications, such as multiplayer online gaming and video conference over the Internet. We would like to investigate the sources of latency in the end systems. Moreover, we would like to modify or develop transport layer mechanisms to support the low latency requirements in the Internet.

References


