SCTP extensions for time sensitive traffic

Vitor Basto[*] and Vasco Freitas

Departamento de Informática, Universidade do Minho, 4710-057 Braga, Portugal
e-mail: vbasto@uminho.pt

Abstract

Stream Control Transmission Protocol (SCTP) is a standard Internet protocol originally designed to transport signalling messages over IP networks. It has been developed and extended to support partially reliable and partially ordered traffic requirements. So far, in-depth unreliable ordered time sensitive traffic performance issues in SCTP have not been addressed. This paper proposes protocol extensions for such traffic and evaluates them by analytical and simulation means.

Keywords

Multimedia transport, Time sensitive traffic, SCTP

1 Introduction

SCTP (Stewart et al., 2000) follows a *partially ordered partially reliable transport protocol* (POPRTP) approach. POPRTP services have been extensively compared to transport services provided by UDP and TCP (Marasli et al., 1997) which have shown, and SCTP in particular (Conrad et al., 2001, Diaz and Owezarski, 1997), to outperform TCP in that they support a wider spectrum of transport service requirements for QoS. It has also been suggested that adaptive multimedia applications use SCTP as its transport protocol which now incorporates several of the service features required by such applications, eg, reliable ordered, reliable unordered, unreliable ordered and unreliable unordered services, to be set in a per stream basis (Stewart et al., 2000, Stewart et al., 2004). A SCTP transport association allows for the setup of multiple QoS differentiated streams. SCTP reliability mechanisms can be parameterised with an upper bound on retransmission effort or with a valid time period (*life time*) for the retransmission of application data units (ADU). Ordered delivery is only supported in the context of a stream and it is left to the applications the responsibility for inter-stream synchronisation through the use of the global connection sequencing identifier *tsn* (transmission sequence number).

Several studies have addressed SCTP reliability issues (Ladha et al., 2004), performance improvements relating to versatility and recovery (Conrad et al., 2001), prioritising algorithms for streams competing for bandwidth within a SCTP association (Heinz and Amer, 2004) and

[*] Supported by *Fundação para a Ciência e a Tecnologia* (FCT) - PRAXIS XXI/BD/9190/96 FSE III QCA. The authors acknowledge the support of Armando Caro from University of Delaware on NS-2/SCTP module issues.

congestion control strategies when SCTP is used over *differentiated services* networks (Zou et al., 2004). However, none have addressed the performance of unreliable mechanisms in SCTP.

SCTP provides for a transport service in which receiving entities can afford to be unaware of most QoS related issues (specification, adaptation, etc). The sender application may change QoS policies of which the receiver is not required to be aware of. Such a sender-centred QoS management approach allows, therefore, a simpler design of the receiver application. This paper proposes SCTP extensions for unreliable traffic, which still keep the protocol simple. Section 2 addresses QoS performance improvements through SCTP extensions for unreliable traffic. Analytic and simulation results in support of the proposals are presented in section 3 and 4 respectively. Conclusions and references to further work follow in section 5.

2 SCTP extensions for time sensitive traffic

As unreliable application data is usually associated with time sensitive (sometimes real-time) applications, consideration of *end-to-end delay* and *jitter* becomes a major issue in the design of transport protocols. With this concern in mind, some SCTP extensions were specified which take into consideration the requirements of time sensitive traffic. These specifications were then modelled and evaluated both analytically and experimentally.

Flow and congestion control mechanisms to support reliable services and network congestion avoidance are required in SCTP and both features also proved to be mandatory for both multimedia traffic support (eg reliability) and Internet traffic stability (collapse avoidance). Although these features are difficult to trade-off under time sensitive traffic, it is feasible to incorporate in the base design and future extensions of SCTP an integrated multimedia transport service (*application aware transport protocols approach*), that competes with solutions based exclusively on multiple differentiated transport connections and application layer QoS management (*network aware applications approach*).

A detailed analysis of protocol operation reveals situations of undesirable and avoidable latency. For unreliable application data, receiving entities are not able to decide on their own on discard actions. All QoS decisions are driven by sender control data which leads to transport control information having to be conveyed to the receiver as soon as possible. Receiver discard actions are then triggered by sender *forward tsn* messages. Following the reception of an amount of *missing reports* = *FAST_RTX_TRIGGER* (4 consecutive *SACK* (Stewart et al., 2000)) the sender generates *forward tsn* messages. This occurs after a round trip time (*minimum period* - *rtt*), or several *rtt* when *congestion window* < *FAST_RTX_TRIGGER*, eg, in lossy periods. This means that a discard notification of a lost unreliable ADU by a transmitter reaches the receiver entity after one or several *rtt* periods have elapsed since the loss was detected. During this time period, other (ordered delivery) ADUs from the same stream are prevented from being delivered to the application.

In order to solve the undesired discard notification latency two SCTP extensions are proposed, named *unreliable time sensitive* and *selective forward tsn*, which do not impair any other SCTP mechanism. The first one consists of removing the wait period of *FAST_RTX_TRIGGER* loss indications for the generation of *forward tsn* messages by the sender and the second extension rids of the delay in loss detection and *forward tsn* messages generation, transmission and processing in standard SCTP. Latency reduction is achieved through the new *selective forward option* (sender indication), carried on a data packet following the unreliable transmitted ADU. This option specifies the *Tsn, StreamId*, *StreamSeqNum* of the discardable ADU. The proposed changes are summarised as follows:

a) *Unreliable time sensitive* extension: when *retransmissions* = *reliability level*, the first loss (or network misorder) detected must trigger a *forward tsn* message. The sender entity does not wait for multiple loss notifications, allowing for earlier discard notifications of unreliable ADUs. A discard notification is issued after *reliability level* retransmissions only. This behaviour allows for a smoother flow of transport streams and application flow(s). Such a discard behaviour may cause extemporaneous (not duplicated) delivery to the receiving transport entity of either an ADU or a *forward tsn* message, none of them having a significant transmission or processing costs.
b) *Selective forward tsn* extension: by means of explicit discardable indication (*Tsn*, *StreamId, StreamSeqNum*) in packets following the discardable ADU, the receiver is able to immediately ignore the lost data. Discardable indication is done for ADU transmitted *reliability level* times. This procedure would improve efficiency when discardable indications are sent *FAST_RTX_TRIGGER* packets later than the ADU they refer to. The *distance* prevents loss of data and discardable indications when network burst errors occur, and early discards when data is out of order. On the other hand, if *distance* $>$ *window size avg*, a discardable indication carried by data packets is undesirably delayed by acknowledgements not being received on time to reopen the flow control window. SCTP sets minimum window values of 2 and 1 in situations of congestion and *timeout* respectively. The *distance* parameter impacts efficiency according to network error patterns. For experimental purposes we set a distance of one between the ADU and the corresponding discardable indication. The overhead introduced by *forward tsn* messages is 12 bytes per packet.

Because receivers have no means of associating detected missing ADUs to the streams they belong to, they are unable to perform early discards and therefore discard indications must be explicitly provided by the sender. The next sections show the improvements achieved by these extensions on application QoS parameters *throughput*, *end-to-end delay* and *jitter*.

3 Protocol extensions performance analysis

Most of SCTP flow and congestion control mechanisms derive from those of TCP which have been extensively studied and tested. The SCTP performance model devised for the evaluation of the proposed extensions is based upon the analytical characterisation of *TCP Reno* sender throughput in (Padhye et al., 1998), which has also been adopted for other SCTP analytical studies (Yi et al., 2003). The features of interest that were modelled are loss detection by multiple *ACK* duplicates, *timeout retransmission*, *fast retransmit* and *congestion avoidance*. The SCTP implementation on the testbed system used support these mechanisms (Caro and Iyengard, 2003). Though the model does not capture the subtleties of the *fast recovery* algorithm, as the testbed implementation of SCTP does not support it we believe this omission does not impact significantly on the results.

Assumptions made in the setting up of the analytical model in (Padhye et al., 1998), which are considered reasonable for the current scenario are:

i) Negligible time spent in slow start taking into account the duration of the connection.
ii) Losses in one round are independent from losses in other rounds. Packets in different rounds are one or more *rtt* seconds apart and are likely to find buffers in different states, independent of each other.
iii) In (Padhye et al., 1998) it is assumed that losses in the same round are correlated in that, whenever a packet is lost, all further packets in that round are also lost. This assumption is based on the behaviour of *TCP Reno* and routing paths that use *FIFO drop tail queueing*. Here, a *RED like* queue management strategy is used and SCTP behavioural differences from *TCP Reno* are considered. These assumptions have consequences on receiver throughput analysis but none on the sender throughput model.
iv) A round corresponds to the back-to-back transmission of a window of packets (a *rtt*).
v) The *rtt* is independent of the window size, which implies that the time needed to send a window of packets to the network is negligible compared to the *rtt*.
vi) Sender loss detection is done through the reception of four *ACK* (triple duplicate) with the same sequence number.
vii) $B(p)$ represents packets sent per unit of time regardless of their fate, ie, whether or not they have been received. It represents the throughput of the connection rather than its goodput.

Let p be the (symmetrical) loss, b the number of delayed *ACK*, rtt the round trip time, T_o the initial retransmission timeout period, frt the fast retransmit trigger and sgt the sack generation timeout (set at $0.2\,s$). Processing time of the receiver application is assumed negligible. E_W is the congestion window size average defined by equation (1)).

$$E_W(p,b) = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \tag{1}$$

W_{max} is the receiver buffer size (set at 65536 bytes $\approx$ 44 packets $> E_W$). If $W_{max} > E_W$, the receiver window size has a negligible impact on the long term average of throughput. Equation (2) has been used for the characterisation of sender throughput.

$$B(p,b,rtt,T_o) \approx \frac{1}{rtt\sqrt{\frac{2bp}{3}} + T_o\,Min\left(1, 3\sqrt{\frac{3bp}{8}}\right)\,p\,(1+32p^2)} \tag{2}$$

For the purpose of characterisation of transport receiver throughput, throughput is considered to be driven by the sender transport mechanisms (2) and network loss $B_R = B(p,b,rtt,T_o)(1-p)$. The SCTP extensions do not affect the behaviour of the sender transport entity, except for the generation of *[selective] forward tsn* messages. No improvement on receiver throughput is expected for *loss rates* $\leq$ *11%* if delayed *ACK* are activated or *loss rates* $\leq$ *25%* which correspond to a *congestion window avg* $\geq$ *4*. For *congestion window avg* $<$ *4 pkts*, the extensions outperform standard SCTP for both transfer (e.g. throughput) and time parameters (e.g. jitter). The reason for this has to do with the standard SCTP behaviour of waiting for *FAST_RTX_TRIGGER* missing reports before triggering the generation of *forward tsn* messages. The extensions optimise not only the average of packet blocking time but also smooths out the effects of congestion.

The initial motivation and hypothesis focused on time sensitive applications so it is in the time parameters (*end-to-end delay*, *jitter* and *drift*) that significant QoS optimisation is expected. For the purpose of end-to-end delay analysis its components are broken down into: *Application processing time*, *transmission time* and *blocking time*. Only *blocking time* caused by network data misorder and loss/retransmission events have been subject to optimisation by the proposed SCTP extensions. The elements of blocking time calculation for standard SCTP and SCTP extensions in equation (5) are defined in equation (3), the probability of having exactly *i* losses, and in equation (4), the average number of received packets per round. Independent loss of data and discard indication is assumed for the extensions, blocking events probability being p^2. Figure 1 shows the blocking time average reduction compared to that of standard SCTP.

$$\begin{cases} P_{std}(p,b,i) = \binom{E_W(p,b)}{i}\, p^i\, (1-p)^{E_W(p,b)-i} \\ P_{ext}(p,b,i) = \binom{E_W(p,b)}{i}\, (p^2)^i\, (1-p^2)^{E_W(p,b)-i} \end{cases} \tag{3}$$

$$\begin{cases} Wr_{std}(p,b) = \sum_{i=0}^{E_W(p,b)+1} P_{std}(p,b,i)\, (E_W(p,b)-i) \\ Wr_{ext}(p,b) = \sum_{i=0}^{E_W(p,b)+1} P_{ext}(p^2,b,i)\, (E_W(p,b)-i) \end{cases} \tag{4}$$

$$\begin{cases} Eblock_{std}(p,rtt,b,T_o,frt,sgt) = \dfrac{rtt\ Max[1, \frac{frt}{Wr_{std}(p,b)/b+rtt/sgt}]\, (\sum_{i=1}^{E_W(p,b)} P_{std}(p,b,i)) + p^{Wr_{std}(p,b)}\, T_o}{Wr_{std}(p,b)/2} \\ Eblock_{ext}(p,rtt,b,T_o,frt,sgt) = \dfrac{rtt\ Max[1, \frac{frt}{Wr_{ext}(p,b)/b+rtt/sgt}]\, (\sum_{i=1}^{E_W(p,b)} P_{ext}(p,b,i)) + p^{Wr_{ext}(p,b)}\, T_o}{Wr_{ext}(p,b)/2} \end{cases} \tag{5}$$

Analytical estimators for *jitter* can be derived although they might be too complex or of little significance the reasons being the infinite traffic source model assumed associated to the window based transmission control of SCTP. On the other hand, timestamps are not provided to receiving transport entities and *rtt* measurements are done once per round in SCTP. There is no mechanism or control data for *jitter* measurement purposes in SCTP. Instead, an equivalent and representative QoS parameter, *receiver inter-arrival time variation*, is measured.

In this study receiver buffering issues were also addressed. Receiver buffering events happen $p\ E_W$ times per round for standard SCTP and $p^2\ E_W$ times for the SCTP extensions. Equation (6) defines buffering estimations per buffering event for both standard and extended SCTP. Buffering reduction is exclusively due to the extensions buffering events reduction. Fig 2 shows that for higher loss rates the improvement on throughput results in a higher data buffering (per loss or buffering event).

$$E\,[RCVbuf_{avg}] = E_W + (\frac{1}{1-p} - 1)\, Wr \tag{6}$$

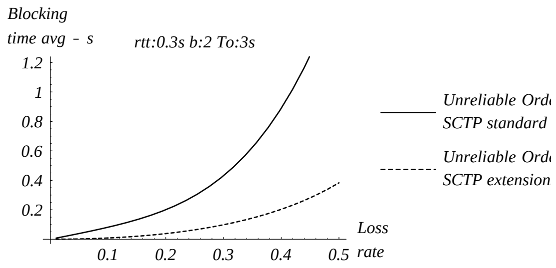

Fig. 1. Blocking time

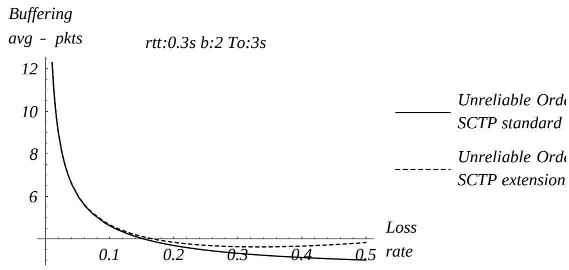

Fig. 2. Receiver buffering avg per buffering event

The analytic estimators lead to the conclusion that time and resource QoS parameters can be significantly optimised for multimedia (unreliable ordered) traffic by using these extensions.

4 Simulation results

In order to check the analytical estimators simulation was used, under the *network simulator 2.27* with SCTP (Stewart et al., 2000) and unreliable data mode extensions support of *Partial Reliable SCTP* (Stewart et al., 2004) provided by the University of Delaware (Caro and Iyengard, 2003). Transmission of data is controlled by an infinite traffic source. Traffic burstiness depends on loss rate and flow and congestion control windowing mechanisms of SCTP. Network setup targeted a broadband, wide area lossy network. Data misorder was assumed to be less than *FAST_RTX_TRIGGER* packets. The network was not subject to congestion and loss followed a uniform distribution whose pattern was setup not to match a router *drop tail queueing* strategy, instead it approximates a *RED like* queue management system. The size of application data was set smaller than or equal to the MTU size and therefore no additional delay was introduced by fragmentation. A no loss period of 30s prior to loss activation from 30s to 300s was setup for all simulations.

The analytic estimation of receiver goodput improvements were confirmed by the simulations (only time and buffering evaluations are shown). It can be stated that the *unreliable time sensitive* extension contributes more for goodput improvement than the *selective forward tsn* one. On the other hand, *jitter* reduction is mainly caused by *selective forward tsn*. Time and receiver buffering reductions were also confirmed by the simulations. For $p = 0.1$ blocking times were *Avg:0.18 Max:3.9 Std:0.3* for SCTP standard and *Avg:0.009 Max:0.3 Std:0.05* with the extensions. A reduction in interarrival time can be observed when comparing results achieved by standard SCTP in Fig 3, *Avg:0.133 Max:2.647 Std:0.276* and extended SCTP in Fig 4, *Avg:0.101 Max:1.343 Std:0.116*. Significant reductions were observed for the average (24%) and standard deviation (56%). Receiver buffer occupancy comparisons can be made by inspection of Figures 5 and 6, corresponding to standard SCTP (*Buff.Data:5026 pkts Buff.Events:1131 Buff.Avg:4.4 pkts*) and extended SCTP (*Buff.Data:344 pkts Buff.Events:252 Buff.Avg:1.4 pkts*) respectively. Because memory is a scarce resource in some systems these show that SCTP extensions impacts positively on buffer requirements and as buffer management is one of the most time-consuming tasks in protocol processing, it is believed that the SCTP extensions improve the overall transport service performance.

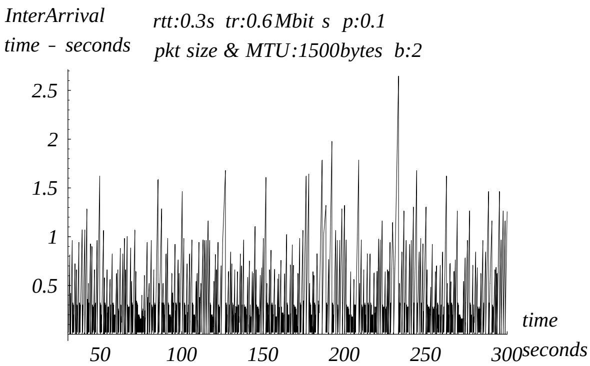

Fig. 3. SCTP inter-arrival time

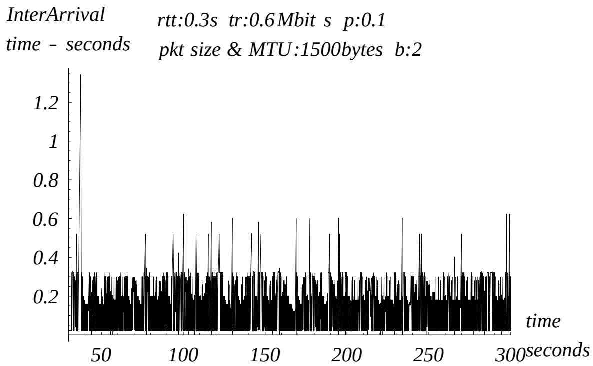

Fig. 4. Extended SCTP inter-arrival time

From such analytical and simulation analysis, it is possible to conclude that when loss approximates zero the overhead introduced by the SCTP extensions is negligible. Analytically, when loss of data occurs but a discardable notification reaches the receiver, it is considered that no blocking event takes place. Under simulation though, Fig 6 accounts for a negligible blocking.

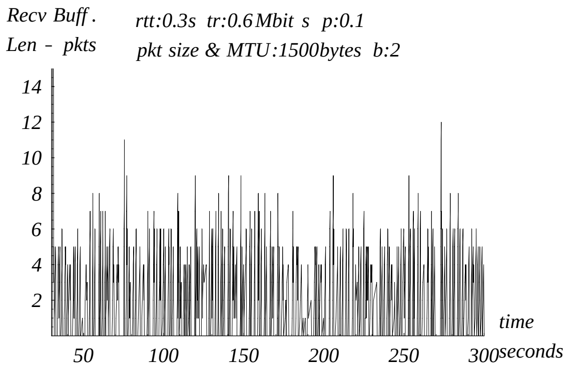

Fig. 5. SCTP receiver buffer

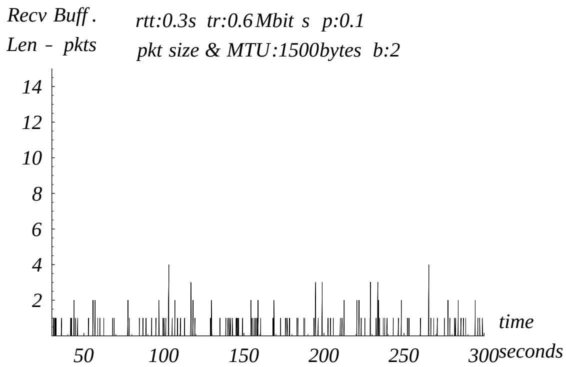

Fig. 6. Extended SCTP receiver buffer

5 Conclusions and further work

Both analytical and simulation models show that the proposed SCTP extensions improve the efficiency of time sensitive/adaptive multimedia streams over wide area (*best effort*) networks. The impact of SCTP extensions on application QoS performance parameters and resources were discussed. The main conclusion of this study is that QoS degradation, specially of time parameters, caused by unnecessary (not essential) application data can be highly reduced by simple extensions to the protocol.

For the sake of clarity and simplicity, the study focused on single, unreliable (*reliability level 0*), ordered traffic streams. But there is a potential for optimisation of multi-stream transport associations by these SCTP extensions at the level of inter-stream spatial and temporal dependencies of reliable and unreliable streams. *Drift*, being the time deviation of synchronised streams, depends strongly on *blocking time* and *jitter/inter-arrival time* and would therefore be a candidate parameter for optimisation.

Distance tunning of the proposed extensions has a major impact on the results achieved. Loss independence was assumed between ADU and their corresponding *selective forward tsn* notifications. Such assumption no longer holds when network routers adopting *FIFO drop tail queueing* policies are involved as these may cause bursty losses. Because loss independence and *distance* can in some cases be related, *distance tuning* for these specific scenarios needs further study.

References

Caro, Armando and Janardhan Iyengard (2003), NS-2 SCTP module release 3.3, Protocol Engineering Lab - University of Delaware.

Conrad, P., A. Caro and P. Amer (2001), "ReMDoR: Remote Multimedia Document Retrieval over Partial Order Transport", *ACM Multimedia* 2001, Ottawa, Canada, September 2001.

Diaz, Michel and Philippe Owezarski (1997), "From multimedia models to multimedia transport protocols", *Computer Networks and ISDN Systems*, vol 29, n. 7, pp 745-758, August 1997.

Heinz, G. and P. Amer (2004), "Priorities in SCTP Multistreaming", *SCI '04*, Orlando, FL, July 2004.

Ladha, S., S. Bauke, R. Ludwig and P. Amer (2004), "On Making SCTP Robust to Spurious Retransmissions", *ACM SIGCOMM Computer Communication Review*, Vol 34 (Issue 2), pp 123-135, April 2004.

Marasli, R., Paul D. Amer and Phillip T. Conrad (1997), "An analytic study of partially ordered transport services". *Computer Networks and ISDN Systems*, vol 29 (Issue 6), pp 675-699, May 1997.

Padhye, J., V. Firoiu, D. Towsley and J. Kurose (1998). Modeling TCP Throughput: A Simple Model and its Empirical Validation", *In Proceedings of ACM SIGCOMM*, September 1998.

Stewart, R., Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina and M. Kalla, L. Zhang, V. Paxson (2000), "Stream Control Transmission Protocol", RFC 2960, *IETF*, October 2000.

Stewart, R., M. Ramalho, Q. Xie, M. Tuexen and P. Conrad (2004), "Stream Control Transmission Protocol (SCTP) - Partial Reliability Extension", RFC 3758, *IETF*, May 2004.

Yi, Z., Tarek Saadawi and Myung Lee (2003), "Analytic model of Stream Control Transmission Protocol", *International Workshop on Performability Modeling of Computer and Communication Systems (PMCCS)*, Monticello, Illinois USA, September 2003.

Zou, J., M.U. Uyar, M.A. Fecko and S. Samtani (2004), "Preferential treatment of SCTP subflows: Analysis and simulation", *IEEE International Symposium on Computer Communications (ISCC)*, Alexandria, Egypt, pp 810-815, June/July 2004.