

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: April 21, 2016

K. Nielsen  
R. De Santis  
Ericsson  
A. Brunstrom  
Karlstad University  
M. Tuexen  
Muenster Univ. of Appl. Science  
R. Stewart  
Netflix, Inc.  
October 19, 2015

SCTP Tail Loss Recovery Enhancements  
draft-nielsen-tsvwg-sctp-tlr-02.txt

#### Abstract

Loss Recovery by means of T3-Retransmission has significant detrimental impact on the delays experienced through an SCTP association. The throughput achievable over an SCTP association also is negatively impacted by the occurrence of T3-Retransmissions. The present SCTP Fast Recovery algorithms as specified by [RFC4960] are not able to adequately or timely recover losses in certain situations, thus resorting to loss recovery by lengthy T3-Retransmissions or by non-timely activation of Fast Recovery. In this document we specify a number of enhancements to the SCTP Loss Recovery algorithms which amends some of these deficiencies with a particular focus on Loss Recovery for drops in Traffic Tails. The enhancements supplement the existing algorithms of [RFC4960] with proactive probing and timer driven activation of the Fast Retransmission algorithm as well as a number of enhancements of the Fast Retransmission algorithm in itself are specified.

#### Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 21, 2016.

#### Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1.	Introduction . . . . .	3
1.1.	The SCTP TLR Function . . . . .	4
1.1.1.	Dependencies . . . . .	5
1.2.	Relation to other work . . . . .	5
1.2.1.	Early Retransmit and RTO Restart . . . . .	5
1.2.2.	TCP applicability . . . . .	6
1.2.3.	Packet Re-ordering . . . . .	6
1.2.4.	Congestion Control . . . . .	7
1.2.5.	CMT-SCTP Applicability . . . . .	7
2.	Conventions and Terminology . . . . .	8
3.	Description of Algorithms . . . . .	9
3.1.	SCTP Scoreboard and miss indication Counting Enhancement . . . . .	9
3.1.1.	Multi-Path Considerations . . . . .	11
3.2.	RFC6675 nextseg() Tail Loss Enhancements for SCTP FR . . . . .	11
3.2.1.	Multi-Path Considerations . . . . .	14
3.3.	SCTP-TLR Description . . . . .	15
3.3.1.	Principles . . . . .	15
3.3.2.	SCTP - TLR Statemachine . . . . .	19
3.3.3.	TLPP Transmission Rules . . . . .	24
3.3.4.	Masking of TLPP Recovered Losses . . . . .	28
3.3.5.	Elimination of unnecessary DELAY-ACK delays . . . . .	30
4.	Confirmation of support for Immediate SACK . . . . .	31
5.	Socket API Considerations . . . . .	31
6.	Security Considerations . . . . .	31
7.	Acknowledgements . . . . .	32
8.	IANA Considerations . . . . .	32
9.	Discussion and Evaluation of function . . . . .	32
10.	References . . . . .	32
10.1.	Normative References . . . . .	32

10.2. Informative References . . . . .	33
Appendix A. Unambiguous SACK . . . . .	35
A.1. TSN Retransmission ID in Data Chunk Header . . . . .	35
A.1.1. Sender side behaviour . . . . .	36
A.1.2. Receiver side behaviour . . . . .	36
A.2. Unambiguous SACK Chunk . . . . .	36
A.2.1. Receiver side behaviour . . . . .	40
A.3. Unambiguous SACK return . . . . .	40
A.4. Negotiation . . . . .	41
Authors' Addresses . . . . .	41

## 1. Introduction

Loss Recovery by means of T3-Retransmission has significant impact on the delays experienced through, as well as, the throughput achievable over an SCTP association. Loss Recovery by Fast Retransmission operation in many situations is superior to T3-Retransmission from both a latency and a throughput perspective.

The present SCTP Fast Retransmission algorithm, as specified by [RFC4960], is driven uniquely by exceed of a DupTresh number of miss indication counts stemming for returned SACKs, and it is as such not able to adequately or timely recover losses in traffic tails where a sufficient number of such SACKs may not be generated, there resorting to loss recovery by T3-Retransmissions or by non-timely activation of Fast Recovery. Non-timely activation here refer to the situation where activation of Fast Recovery for packets lost within one data burst needs to await arrival of SACKs from a subsequent data burst.

By drop in traffic tails (or tail drops) we refer generally and specifically to the following situations:

1. Drops of the last SCTP packets of an SCTP association or more generally drop of packets in the end of an SCTP association which are not preceded by more than DupThresh number of packets which are not dropped.
2. Drops among packets sent in a the end of bursts spaced by pauses of time equal to or greater than the T3-timeout (approximately). It is noted that such bursts (pauses in between bursts) may result from application limitations, from congestion control limitations or from receiver side limitations.
3. Drops among packets sent so sparsely that each dropped packet constitutes a tail drop in that DupThresh number of packets would not be sent (would not be available for sent) prior to expiry of the T3-timeout.

It shall be noted that while the above traffic drop criteria describe drops among the forward data packets only, then drops among forward data packets combined with drops of the returned SACKs may together result in that an insufficient number of SACKs be returned to traffic sender for that the Fast Retransmission algorithm be activated prior to T3-timeout occurring. The tail traffic situations for which SCTP Fast Retransmission is not able to recover the losses is thus in general broader than the exact situations listed above. The improvements specified include enhancement of SCTP to deduce the miss indication counts from enhanced scoreboard information thus removing some of the vulnerability of the present SCTP miss indication counting to loss of SACKs.

### 1.1. The SCTP TLR Function

The function proposed for enhancements of the SCTP Loss Recovery operation for Traffic Tail Losses is divided in two parts:

- o Enhancements of SCTP Fast Retransmission (SCTP FR) algorithm by means of the following Tail Loss Recovery improving functions inspired by or specified by [RFC6675] for TCP:
  - \* miss indication counting for a missing (non-SACK'ed) TSN will be based on augmented scoreboard information such that the miss indications will be based not on the number of returned SACKs but on the number of SACK'ed SCTP packets carrying data chunks of higher TSNs. The mechanism is specified both in terms of packets, the book-keeping of which requires new logic, as well as in terms of a less implementation demanding byte based variant following the Islost() approach of [RFC6675]. We shall refer to this improvement as Extended miss indication Counting.
  - \* Fast Recovery operation is extended to include the "last resort" retransmission, Nextseg 3) and Nextseg 4), operations of [RFC6675], thus supporting conditional proactive fast retransmissions of missing, but not yet classified as lost, TSNs within the Fast Recovery Exit Point.
- o New SCTP Tail Loss Recovery State machine with proactive timer driven activation of (the enhanced) Fast Recovery operation. Timer driven activation of Fast Recovery is initiated for outstanding data whenever a certain time, shorter than the T3 timeout, has elapsed from the transmittal of the lowest outstanding TSN and network responsiveness, in form of SACKs of packets ahead of the TSN, has been proven since the transmittal of the lowest outstanding TSN. The SCTP TLR mechanism implements a new timer, the Tail Loss Probe timer (PTO), and it works in parts by:

- \* Forced activation of Fast Recovery when network responsiveness has been proven, and the PTO timer has kicked, since transmittal of the lowest outstanding TSN, but additional traffic sent (SACKs of TSNs ahead of the TSN) has not served to activate Fast Recovery based on the Extended Mis Indication Counting.
- \* Probing for network responsiveness, by transmittal of a TLR probe packet, when no network responsiveness information (no SACKs have been received for any packets ahead of line of the TSN) is available at expiration of the PTO timer relative to the lowest outstanding TSN
- \* Activation for T3-retransmission Loss Recovery only when the network remains unresponsive (no SACKs are received) also after transmittal, and subsequently timeout, of a TLR probe packet.

#### 1.1.1. Dependencies

The SCTP TLR procedures proposed apply as add-on supplements to any SCTP implementation based on [RFC4960]. The SCTP TLR procedures in their core are sender-side only and do not impact the SCTP receiver.

Exploitation of SCTP immediate SACK feature, [RFC7053], and usage of new (to be defined) Unambiguous Selective Acknowledgement feature of SCTP require support in both sender and receiver of these SCTP extensions.

#### 1.2. Relation to other work

##### 1.2.1. Early Retransmit and RTO Restart

It is noted that the Early Retransmit algorithm, [RFC5827], addresses activation of Fast Recovery for a particular subset of the tail drop situations in target of the SCTP TLR function. The solution proposed embeds (as a special case) the Early Retransmits algorithm in the delayed variant, experienced with for TCP in [DUKKIPATI02] in which Early Retransmission is only activated provided a certain time has elapsed since the lowest outstanding TSN was transmitted. The delay adds robustness towards spurious retransmissions caused by "mild" packet re-ordering as documented for TCP in [DUKKIPATI02].

It is further noted that depending on the exact situation (e.g., drop pattern, congestion window and amount of data in flight) then T3-retransmission procedures need not be inferior to Fast Retransmission procedures. Rather in some situations T3-retransmission will indeed be superior as T3-retransmissions allow for ramp up of the congestion window during the recovery process.

The changes proposed in this document focus on improving the Loss Recovery operation of SCTP by enforcing timely activation of (improved) Fast Retransmission algorithms. With the purpose to reduce the latency of the TCP and SCTP Loss Recovery operation [HURTIG] has taken the alternative approach of accelerating the activation of T3-retransmission processes when Fast Recovery is not able to kick in to recover the loss. [HURTIG] only addresses a subset of the Tail loss scenarios in scope in the work presented here. The ideas of [HURTIG] for accurate RTO restart are drawn on in the solution proposed here for accurate restart of the new tail loss probe timer (PTO-timer) as well as for accurate set of the T3-timer under certain conditions thus harvesting some of the same latency optimizations as [HURTIG]. The same approach has recently been exploited for TCP by the invention of the TLPR function by the authors of [Rajiullah].

### 1.2.2. TCP applicability

SCTP Loss Recovery operation in its core is based on the design of Loss Recovery for TCP with SACK enabled. The enhancements of SCTP Tail Loss Recovery proposed here are applicable for TCP.

Note: The - to be determined - exploitation of SCTP immediate SACK feature, [RFC7053], and the - to be determined - usage of new unambiguous selective acknowledgement feature of SCTP may not be readably applicable to TCP at present. ISSUE: Need to follow up on [zimmermann02], [zimmermann03],

It is noted that while the SCTP TLR algorithms and SCTP TLR state machine defined is inspired by the timer driven tail loss probe approach specified in [DUKKIPATI01] for TCP, then the solution defined here differs in the approach taken. The approach here is a clean state approach defining a new comprehensive SCTP TLR state machine as an add-on to the (at least conceptually) existing Fast Recovery and T3-Retransmission SCTP state machines of SCTP. Thereby the SCTP TLR algorithm is able to address all tail loss patterns, whereas the approach of [DUKKIPATI01] relies on a number of experimental mechanisms ([DUKKIPATI02], [MATHIS], [RFC5827]) defined for TCP in IETF or in Research with ad hoc extension to support selected tail loss patterns by addition of the tail loss probe mechanism and the therefrom driven activation of the mechanisms.

### 1.2.3. Packet Re-ordering

The solution proposed is an enhancement of the existing mis indication counting based Fast Recovery operation of SCTP, [RFC4960], and as such the solution inherits the fundamental vulnerability to

packet re-ordering that the SCTP Fast Retransmission algorithm of [RFC4960] embeds.

For deployment of SCTP in environments where the Fast Retransmission algorithm of [RFC4960] gives rise to spurious entering of Fast Recovery it would be relevant to look into remedies which may detect such and undo the effects of such. Possibly following the approaches taken for TCP (and SCTP) in this area.

OPEN ISSUE: In severe packet re-ordering situations where the second packet of two subsequently sent packets outrace the first packet in arrival with more than PTO time, then such may trick the SCTP TLR function to enter spurious Fast Recovery. It is conjectured that the this situation does not significantly increase the vulnerability of Loss Recovery to packet-reordering. To be determined and evaluated.

#### 1.2.4. Congestion Control

In its very nature of prompting for activation of Fast Recovery instead of T3-Retransmission Recovery then the benefit of the solution proposed versus the existing solution of [RFC4960] will depend on the CC operation not only during the recovery process but also after exit of the recovery process. In this context it is noted that the prior approach taken for TCP, [DUKKIPATI01], has been documented for a TCP implementation running CUBIC, e.g., see [zimmermann01], whereas SCTP runs a CC algorithm more similar to TCP Reno CC as defined by [RFC5681].

The solution at present is defined within the constraints of existing Congestion Control principles of STCP as defined by [RFC4960]. It is anticipated that Congestion Control improvements are desirable for SCTP in general as well as for the functions defined here in particular.

#### 1.2.5. CMT-SCTP Applicability

The SCTP TLR specification in this document applies to a SCTP implementation following the [RFC4960] principles of using one shared SACK clock spanning the data transfer over multiple paths. It is noted that in its nature of maintaining the common SACK clock principles of [RFC4960] then the SCTP TLR mechanism specified here retains some of the vulnerabilities from [RFC4960] to spurious (or delayed) entering of Fast Recovery operation caused by path changes in inhomogeneous environments (change of data transfer among paths of significantly different RTTs). The validity of this choice is motivated by that concurrent data transfer on multiple paths is the exception case in [RFC4960] MH SCTP and remains the exception also with the enhancements of [RFC4960] specified here.

It is envisaged that the SCTP TLR mechanism specified is readably applicable also to a SCTP implementation supporting concurrent multi path transfer in line with the specification of [CMT-SCTP]. Though is it emphasized that SCTP-TLR, when applied to [CMT-SCTP], needs some adjustments as it should be applied in a split manner following the principles of SFR of [CMT-SCTP].

## 2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

For the purposes of defining the SCTP TLR function, we use the following terms and concepts:

"DupThresh": The number of miss indication counts on an outstanding TSN at the reach of which SCTP declares the TSN as lost and enters Fast Recovery for the TSN if not in Fast Recovery already.

"Flight size": At any given time we define the "Flight size" to be the number of bytes that a SCTP sender considers to be in flight in the network from the sender to the receiver. It is noted that the bytes of a message, which is considered lost and which has not been retransmitted, is not contained in the Flight size. Further it is noted that the bytes of a message which has been retransmitted (once) will count either once or twice in the Flight size depending on whether SCTP considers the first transmission of the message as having been lost (dropped) in the network.

"Outstanding TSN": A TSN (and the associated DATA chunk) that has been sent by the SCTP sender for which it has not yet received an acknowledgement and which the SCTP sender has not abandoned (e.g., abandoned as a result of [RFC3758]).

"highTSN": The highest outstanding TSN at this point in time.

"lowTSN": The lowest outstanding TSN at this point in time.

"Scoreboard": An SCTP sender need maintain a data structure to store various information on a per outstanding TSN basis. This includes the selective acknowledgment information, miss indication counts, bytes counts and other information defined [RFC4960], in this document and in other SCTP specifications. This data structure we refer to as "scoreboard". The specifics of the scoreboard data structure are out of scope for this document (as

long as the implementation can perform all functions required by this specification).

### 3. Description of Algorithms

#### 3.1. SCTP Scoreboard and miss indication Counting Enhancement

Entering of Fast Recovery in SCTP, as specified by [RFC4960]), is driven by miss indication counts. When a TSN has received DupThresh=3 miss indication counts, the TSN is declared lost and will be eligible for fast retransmission via Fast Recovery procedure.

miss indication counts are in RFC4960 SCTP driven entirely by receipt of SACKs in accordance with the Highest TSN Newly Acknowledged algorithm (section 7.2.4 of [RFC4960]):

Highest TSN Newly Acknowledged (HTNA): For each incoming SACK, miss indications are incremented only for missing TSNs prior to the highest TSN newly acknowledged in the SACK. A newly acknowledged DATA chunk is one not previously acknowledged in a SACK.

An evident issue with the HTNA algorithm is that it is vulnerable to loss of SACKs. In many situations loss of SACKs will result only in a slight delayed entering of Fast Recovery for a dropped TSN, but generally, then by relying on HTNA algorithm only, loss of SACKs will further broaden the traffic tails situations where Fast Recovery either not be activated in a timely manner or not be activated at all due to the receipt of an insufficient number SACKs only.

In order to make SCTP Fast Recovery more robust towards drop of SACKs, the following extension of the HTNA algorithm SHOULD be supported by an SCTP implementation:

Newly Acked Packets ahead-of-line (NAPahol): For each incoming SACK, miss indications are incremented only for missing TSNs prior to the highest TSN newly acknowledged in the SACK. A newly acknowledged DATA chunk is one not previously acknowledged in a SACK. For each missing TSN thus potentially eligible for additional miss indication counts, the number of miss indications to be given shall follow the number of newly acknowledged packets ahead of line of the packet of the missing TSN.

The solution is robust towards split SACK. The solution requires for the SCTP implementation to keep track of the relationship in between data chunks (TSN numbers) and packets. One solution is for the SCTP implementation to maintain a packet id as a monotonically incrementing packet sequence number to map chunks to packets and for

each outstanding chunk to keep state of the packet id that the chunk was sent in as well as (incrementally updated) the packet ids of up to DupThresh-1 (=2) packets ahead of line for which chunks have been SACKed.

For accurate PTO-timer management, using the restart principles of [HURTIG] and [Rajiullah], see Section 3.3, an SCTP TLR implementation is required to keep track of the time at which packets/TSNs are transmitted (or strictly speaking to be able to deduce the time since a packet/a TSN was last transmitted). An implementation may exploit timestamps for the generation of (part of) the packet id as well as for the mentioned time management thereby limiting the additional overhead required for the packet id storage.

As an alternative to the above accurate packet counting then an SCTP implementation MAY, to reduce implementation complexity, instead support the following bytes counting based extension of the RFC4960 HTNA algorithm:

Highest Bytes Newly Acknowledged (HBNA): For each incoming SACK, miss indications are incremented only for missing TSNs prior to the highest TSN newly acknowledged in the SACK. A newly acknowledged DATA chunk is one not previously acknowledged in a SACK. For each missing TSN thus eligible for additional miss indication counts, the number of miss indications to be given shall follow the number of newly acknowledged bytes in the SACK ahead of line of the missing TSN in the following manner Add-miss indication-count(TSN) = Ceiling((Newly bytes ahead of line(TSN))/PMTU).

The HBNA approach as specified above is vulnerable to split of SACK. An implementation choice which is robust to split of SACK is to recalculate the total amount of selectively acknowledged bytes ahead of line of an outstanding TSN and update the miss indication count of the TSN as Ceiling((Selectively Acked bytes ahead of line (TSN))/PMTU). This more robust implementation choice however demands either for maintain of additional state per TSN, namely the Selectively Acked bytes ahead of line (TSN) or for extensive repeated computations. Risk of split SACK may not be weighty enough to worth such implementation complexity.

The HBNA approach follows the approach taken for TCP, Islost(), in [RFC6675]. It is noted, however, that due to the message based approach of SCTP, then a byte based approach generally will be less accurate as a measure for the number of packet received ahead of line than it is for byte stream based TCP.

### 3.1.1.1. Multi-Path Considerations

In multi-homed [RFC4960] SCTP, data that potentially will be subject to fast retransmission may be in flight on multiple paths. This (exception) situation can occur as a result of a change of the data transfer path, which may come about, e.g., as a result of a switchback operation performed autonomously by SCTP or as a result of a management operation setting a new primary path. The situation can also occur as a result of destination directed data transfer where the destination address specified is different from the present data transfer path destination. In an [RFC4960] SCTP implementation, SACKs of data sent on one path will increase the miss indication counts of data with lower TSN in flight on a different path. As such SACKs of data sent on one path may actually result in generation of (potentially spurious) loss event reactions on a different path. This fundamental aspect of [RFC4960] miss indication counting is not changed in this document. Meaning that it is not intended for the miss indication counting improvements defined above, i.e., the NAPahol and the HBNA mechanisms, to discriminate among the paths on which the SACK'ed data contributing to the miss indication counting has been sent.

### 3.2. RFC6675 nextseg() Tail Loss Enhancements for SCTP FR

The Fast Retransmission algorithm for TCP as specified in [RFC6675] implements some differences compared to the Fast Retransmission algorithm specified for SCTP by [RFC4960]. Of particular significance for recovery of losses in traffic tail scenarios are the fact that the [RFC6675] algorithm, once Fast Recovery has been activated, takes two "last resort" retransmission measures, step 3) and step 4) of Nextseg() of [RFC6675]. These measures facilitate the recovery of losses in situations where only an insufficient number of SACKs would be able to be generated to complete the Fast Recovery process without resorting to T3-timeout. For SCTP Fast Recovery we formulate the equivalent measures as follows:

Last Resort Retransmission: If the following conditions are met:

- \* there are no outstanding TSN's eligible for fast retransmission due to DupThresh or more miss indications
- \* there is no new data available for transmission

then an outstanding TSN less than or equal to the Fast Recovery Exit Point, for which there exists SACKs of chunks ahead of line of the TSN, may be retransmitted provided the CWND allow. The bytes of a TSN which is retransmitted in this manner are not subtracted from the Flight size prior to this action be taken nor

as a result of this action. If the miss indication count of the TSN subsequently reaches the DupThresh value, the bytes of the TSN shall be subtracted from the Flight size. Once acknowledged the remaining contribution of this TSN in the Flight size (whether it be there counted once or twice at this point in time) is subtracted. A TSN which is retransmitted in this manner will be marked as ineligible for a subsequent fast retransmit (see considerations on Multiple Fast Retransmission operation in Section 3.3.1.3).

An SCTP implementation which implements the Unambiguous SACK feature of Appendix A may implement a more accurate calculation of the flightsize when doing Last Resort Retransmission. That is, instead of subtracting the contribution from the retransmitted TSN from the flightsize once the acknowledgement of the TSN arrives, the SCTP implement may distinguish where the acknowledgment is for the original TSN or for the retransmitted TSN and in case the acknowledgement is not for the retransmitted TSN, SCTP should delay the subtract of the bytes of the retransmitted TSN from the flightsize until either an acknowledgement of the retransmitted TSN is received (see Appendix A) or until PTO2-T\_latest(TSN) time has elapsed (see Section 3.3.1).

Rescue: If all of the following conditions are met:

- \* there are no outstanding TSN's eligible for fast retransmission due to DupThresh or more miss indications
- \* there is no new data available for transmission and no data is outstanding on the association beyond the Fast Recovery Exit Point
- \* there are no outstanding TSNs eligible for Last Resort Retransmission
- \* the cumack has progressed since this entering of Fast Recovery

and there exist non-SACKed, non fast retransmitted TSNs, within the Fast Recovery Exit point, then for this entry of Fast Recovery, conditionally to that the CWND allows, we allow for fast retransmission of one packet of consecutive outstanding non fast retransmitted TSNs up to PMTU size, the highest TSN of which MUST be the highest outstanding TSN within the Fast Recovery Point. The bytes of a TSN which is retransmitted in this manner are not subtracted from the Flight size prior to this action be taken nor as a result of this action. If the miss indication count of the TSN subsequently reaches the DupThresh value, the bytes of the TSN shall be subtracted from the Flight size. Once acknowledged the

remaining contribution of this TSN in the Flight size (whether it be there counted once or twice at this point in time) is subtracted. A TSN which is retransmitted in this manner will be marked as ineligible for a subsequent fast retransmit (see considerations on Multiple Fast Retransmission operation in Section 3.3.1.3).

An implementation of the Rescue operation may be accomplished by maintain of an RescueRTX parameter as described for TCP in [RFC6675].

An SCTP implementation which implements the Unambiguous SACK feature of Appendix A may implement a more accurate calculation of the flightsize when performing Rescue operation. That is, instead of subtracting the contribution from the retransmitted TSN from the flightsize once the acknowledgement of the TSN arrives, the SCTP implement may distinguish where the acknowledgment is for the original TSN or for the retransmitted TSN and in case the acknowledgement is not for the retransmitted TSN, SCTP should delay the subtract of the bytes of the retransmitted TSN from the flightsize until either an acknowledgement of the retransmitted TSN is received (see Appendix A) or until  $PTO2-T_{latest}(TSN)$  time has elapsed (see Section 3.3.1).

DISCUSSION: [RFC4960] in addition to the HTNA algorithm demand for additional miss indication counting to be performed during Fast Recovery according to the following prescription (section 7.2.4 of [RFC4960]):

- (#) If an endpoint is in Fast Recovery and a SACK arrives that advances the Cumulative TSN Ack Point, the miss indications are incremented for all TSNs reported missing in the SACK.

It is noted that under special circumstances then (#) makes SCTP Fast Recovery complete in situations where TCP Fast Recovery would only complete by virtue of the measure 3) or 4) of [RFC6675] and as such these measures are more critically demanded for TCP Fast Recovery operation than for the SCTP Fast Recovery operation. However as documented by (OPEN ISSUE: to be filled in) the Last Resort Retransmission operation and the Rescue operation also for SCTP significantly improve the Loss Recovery operation; the latency of the individual loss recovery operation as well as the ability of the operation to complete without resort to T3-timeout. Consequently this document prescribes for SCTP TLR to implement these procedures. Conversely even when the measures 3) and 4) of [RFC6675] are implemented, (#) gives benefits in terms of releasing flight size space allowing Fast Recovery to progress.

As the algorithm extension is limited by the existing congestion control algorithm of SCTP, these extensions of SCTP Fast Recovery do not compromise the TCP fairness of the SCTP Fast Recovery Operation.

### 3.2.1. Multi-Path Considerations

In multi-homed [RFC4960] SCTP, data that potentially will be subject to Fast Retransmission may be in flight on multiple paths. This (exception) situation in particular can occur as a result of a change of the data transfer path as a result of a switchback operation to a primary path. Here SACKs of data sent on one path (e.g., the new data transfer path) may result in generation of (potentially spurious) loss event reactions on a different path (the prior data transfer path). The [RFC4960] miss indication counting based on a common SACK clock is not changed in this document, nevertheless the protocol operation, here the operation of the Last Resort Retransmission and the Rescue operation in this situation, need to be specified.

The specification in this document is based on the following fundamental goals:

- o an [RFC4960] SCTP implementation must appropriately react to loss events observed by means of miss indication counting, by performing appropriate adjustments of CWND and sstresh, on all paths where such loss events are observed.
- o The observation of a loss event on one path should not for [RFC4960] SCTP MH impact the congestion control operation on a different path.

For the implementation of the Last Resort Retransmission and the Rescue operations for [RFC4960] MH SCTP then the following specifications are given:

- o For a TSN to be eligible for Last Resort Retransmission a loss event MUST have been observed on the path on which this TSN is in flight.
- o For a TSN to be eligible for the Rescue operation a loss event MUST have been observed on the path on which this TSN is in flight.

An implementation of the above may be accomplished by the implementation of a Fast Recovery state and Fast Recovery Exit point on a per path basis with the following particulars:

- o A path enters the Fast Recovery State based on loss event observation of TSNs in flight on the path.
- o When a loss event is observed on a path the Fast Recovery Exit point on the path is set to the highest TSN in flight of the path.
- o Fast Retransmission of TSNs in flight on the path terminates once the Fast Recovery Exit Point on the path has been reached (i.e., has been cumulative SACK'ed) at which point the Fast Recovery process on the path is terminated.
- o The eligibility of a TSN for the Last Resort Retransmission and the Rescue operation shall follow the prescriptions given above with adherence to the Fast Recovery Exit point set on the path on which the TSN is in flight.

The data retransmission process of data chunks in itself is prescribed to happen on the present data transfer path of the association regardless of which path the data chunks were in flight on when they became eligible for Fast Retransmission. This follows [RFC4960] and the preceding [CARO02].

With the above per path modelling of the Fast Recovery operation, SCTP may have multiple fast recovery exit points at any given time (though at most one per path) and the fast recovery operation may terminate at different times on the different paths. Further it is noted that a path may be in Fast Recovery even if no data is in flight on the path or even if the only data in flight on the path is beyond the Fast Recovery Exit Point of the path. The latter can occur in the very peculiar case where fast retransmission of data declared lost on the path happens on a different path as well as that the user performs a data directed data transfer on the path in question.

An SCTP implementation fulfilling the goals described above may also be achieved by other means than by maintain of a per path Fast Recovery Exit point. For example it might be achieved by maintain of a common association Fast Recovery Point spanning multiple paths, but still the implementation must ensure appropriate per destination address congestion control operation.

### 3.3. SCTP-TLR Description

#### 3.3.1. Principles

The SCTP TLR function is based on the following principles.

### 3.3.1.1. Retransmission Timers Management

This document is specified as if there is a single retransmission timer per destination transport address, but implementations MAY have a retransmission timer for each DATA chunk.

This document specifies usage of new PTO timer for SCTP TLR. The document is specified as if the PTO timer functions are implemented by means of the existing retransmission timer of [RFC4960] SCTP, i.e., under certain conditions the retransmission-timer is activated with special PTO values rather than with the standard T3-timer value. The document is specified as if there is a single PTO timer per destination transport address, equivalently a single PTO timer per path. Implementations MAY choose to implement a PTO timer per DATA chunk.

For an outstanding TSN we define the time  $T_{latest}(TSN)$  to be the time that has elapsed since the TSN was last sent. When a TSN is first sent, or when it is retransmitted,  $T_{latest}(TSN)=0$ . An SCTP TLR implementation must be able to deduce this value for any outstanding TSN.

### 3.3.1.2. Timer driven entering of Fast Recovery

Timer driven entering of Fast Recovery in SCTP TLR is based on the following principles:

- o Maintain of a Tail Loss Probe Timer (PTO) which in certain situations (generally when retransmission is not performed) is running on a path. At any given time the value of the PTO timer is related to the lowest TSN in flight on the path. The PTO timer value used will depend on the situation:

By default the following timer value is used:

PTO1:  $PTO = \text{MIN}(RTO, 1.5 * SRTT + \text{MAX}(RTTVAR, DELAY\_ACK))$

Whereas the following value is used:

PTO2:  $PTO = \text{MIN}(RTO, 1.5 * SRTT + RTTVAR)$

when it is known that subsequent SACKs not acknowledging the TSN for which the PTO is running will be (or will have been) returned immediately. For more details see Section 3.3.2.

By design the probe timer is kept lower or equal to the RTO, thereby aiming to prevent a potential unnecessary and damaging RTO, as well as generally larger than an anticipated RTT

thereby preventing that it kicks in prematurely. I.e., the timer only kicks in at a time where one would have expected to have received a SACK of the lowest TSN in flight were there no problems.

A minimal PTO value, `PTO_MIN`, is applied to the above formulas (particularly important for `PTO2`). I.e., the effective `PTO1` = `MAX(PTO_MIN, PTO1)` and the effective `PTO2` = `MAX(PTO_MIN, PTO2)`. The suggested value of `PTO_MIN` is 10 msec. In the following when referring to `PTO1` and `PTO2` we refer to the effective `PTO1` and `PTO2` values.

For an SCTP implementation which performs RTT measurements during the association set-up, the PTO set on the path on which the first data chunk is sent shall be initialized from the RTT measured on the path during the association set-up. If no such RTT measurement is performed or is available on the particular path in question, the PTO shall be initialized as `RTO_INIT`.

- o PTO timer driven transmittal of Tail Loss Probe Packet: Once data is outstanding on a path and the PTO timer of the path kicks and no SACKs of any chunks with higher TSN number have arrived, a probe packet, denoted a Tail Loss Probe Packet (TLPP), is sent to probe for network responsiveness (i.e., for SACK of the TLPP) in order to potentially drive proactive entering of Fast Recovery.
  - \* For a SCTP sender that supports the Immediate SACK feature, [RFC7053], the I-bit MUST be set on chunks sent in a TLPP packet.
- o PTO timer driven entering of Fast Recovery: Process is enforced when network responsiveness is proven (SACK of later sent data than lowest TSN in flight on the path is available) and (at least) PTO time has elapsed since transmittal of this lowest TSN in flight on the path.

Comment: The lowest outstanding TSN on an association may under special circumstances not be in flight on any path of the association. This can happen when the lowest outstanding TSN has been declared lost but the transmittal of the TSN is prevented due to congestion window limitations (e.g., during Fast Recovery). In this case, as well as generally for TSNs that are being retransmitted due to fast retransmission or T3-timeout, no PTO timer is running on the TSN. Conversely when the lowest outstanding TSN on a path is not subject to Fast Recovery or T3-Recovery, then this lowest outstanding TSN is also in flight on the path.

### 3.3.1.3. Fast-Recovery and Loss Detection

Fast Recovery and miss indication counting for the SCTP TLR function MUST embed the enhancements described in Section 3.2. In addition SCTP TLR implements the following loss detection during Fast Recovery:

- o If in Fast Recovery, then an outstanding TSN in flight on the path, with TSN lower than the Fast Recovery Exit Point on the path, is declared lost when the following conditions are satisfied:
  - \* The TSN has not been fast retransmitted.
  - \*  $T_{\text{latest}}(\text{TSN}) > \text{PTO2}$ .
  - \* The TSN is lower than the highest outstanding SACK'ed TSN.

When declared lost by this procedure the TSN is subtracted from the flight size as well as it becomes eligible for fast retransmission as if it had been declared lost by reach of Dupthresh miss indication counts.

Such loss detection during SCTP TLR Fast Recovery shall at a minimum be done at receipt of SACK as well as at times where the possibility to transmit new data is being evaluated. An implementation maintaining PTO timers on a per data chunk basis may make further evaluation based on timer expiration.

Following [RFC4960] it is assumed that a data chunk should only be fast retransmitted once. I.e., subsequent retransmissions of the data chunk must proceed as T3-retransmission. An SCTP TLR implementation MAY possibly implement Multiple Fast Retransmission operation following the principles described in [CAR001] extended to include the Last Resort Retransmission and Rescue operations. Such however is not covered by the specification given here.

### 3.3.1.4. T3-Recovery

[RFC4960] does not explicitly specify for an T3-Recovery phase to be supported for SCTP, nor does [RFC4960] explicitly demand for that a data chunk which has been T3-retransmitted cannot undergo fast retransmission. It can be an advantage that a lost T3-retransmitted data chunk may be recovered by timely fast retransmission rather than by a subsequently, potentially back-off'ed T3-retransmission. For [RFC4960] MH SCTP, however, reliable implementation of such fast recovery of lost T3-retransmitted data is difficult to achieve given the usage of one common SACK clock as new data on one path may trick

spurious fast retransmission of data that has been/is being T3-retransmitted on a different path. Here it is important to emphasize that concurrent T3-retransmission and new data transmission on different paths is the standard operation of MH SCTP [RFC4960]. (Though implementations might possibly mitigate such effects by only sending new data after completion of the T3-retransmission operation as well as the implementation of SCTP-PF, [SCTP-PF], would further decrease the likelihood of such concurrent data transfer occurring.)

In this document we assume that an SCTP implementation follows either of the following implementation choices:

- o A data chunk which has undergone T3-retransmission cannot subsequently be subject to Fast Retransmission whether such entering of Fast Recovery be driven alone by miss indication counting or by the SCTP TLR mechanism. This implementation choice corresponds to implementing a T3-Recovery phase for SCTP equivalent with the RTO-recovery phase of TCP.
- o A data chunk, which has undergone T3-retransmission, will be eligible for subsequent Fast Retransmission if such is driven by miss indication counts from SACKs of new data chunks sent after all data outstanding for T3-retransmission have been sent and the new data is sent on the same path as the T3-retransmission data.

One implementation choice may be to follow the first implementation choice for SCTP MH and the second implementation choice for SCTP SH. Regardless of this implementation choice then in SCTP TLR a data chunk that has been subject to T3-retransmission SHOULD NOT be subject to the timer driven entering of Fast Recovery specified below. The motivation for this choice is that the SRTT may not be appropriately refreshed during the T3-retransmission process. OPEN ISSUE/TO DO: Ideally the PTO timer used after the exit of the T3-recovery phase should be updated based on a fresh RTT measurement. E.g., from the last acknowledged TSN. If no new SRTT calculation is made based on a scheduled RTT measurement, then the PTO timer values could be made sure to be appropriately adjusted, if necessary, by a last measured RTT by  $1,5 * SRTT + RTTVAR \rightarrow \text{MAX}(1 * 5 \text{ RTT}, 1,5 * SRTT + RTTVAR)$ .

### 3.3.2. SCTP - TLR Statemachine

The SCTP Tail Loss Recovery function defines 3 states: The SCTP TLR OPEN state, the SCTP TLR PROBE WAIT state and the SCTP TLR DELAY WAIT state. At any given time the SCTP transmission logic for the lowest outstanding TSN on a path will be in one of these 3 states or the TSN is sought being recovered by means of Fast Recovery or T3-Recovery.

Figure 1 illustrates the states and the state transitions.

(to be inserted)

Figure 1, Enhanced Loss Recovery State Machine Diagram

In the following we describe the states and the actions taken.

#### 3.3.2.1. SCTP TLR OPEN STATE

This is the state the SCTP transmission logic is in on any path when no TSN is outstanding on the association as well as it is the state when SCTP sends the first data on a path after idle/no TSN outstanding. It also more generally is the state the transmission logic is in when there are no gaps in the SACK scoreboard beyond the lowest outstanding TSN on the path.

In this state SCTP is not performing Fast Recovery nor T3-Recovery on the lowest TSN outstanding on the path and no SACKs of any chunks with higher TSN number have arrived. In this state, when SCTP has outstanding data on the path, a PTO timer is running relative to the lowest TSN outstanding on the path.

The PTO set on a (new) lowest outstanding TSN on the path in this state will follow PTO1 when less than 2 packets are outstanding beyond the TSN at the time when the timer is set and follow PTO2 when 2 or more packets are outstanding beyond the TSN when the PTO timer is set or when the Immediate SACK feature is known to be supported by both sender and receiver (see Section 4) and the I-bit has been set on the TSN or on an outstanding TSN of higher number.

In the OPEN state the following may happen:

- o A SACK commutatively acknowledging the lowest outstanding TSN and resulting in no gaps in the SACK scoreboard may arrive. In this case the state remains in OPEN state. If there still is outstanding data on the path, the PTO timer is set on the new lowest outstanding TSN. The PTO timer value set will be the value  $PTO - T_{latest}(TSN)$  where the PTO value is calculated either from PTO1 or PTO2 according to the evaluation criteria given above.
- o A SACK with gap(s) may arrive, thus proving network responsiveness while still not cumulatively acknowledging all lower (than the SACK'ed gap) outstanding TSNs on the path. The SACK may or may not move the cumulative ACK point. This indicates that either

packets are being re-ordered or the (new) lowest outstanding TSN on the path has been lost.

- \* If the SACK makes the miss indication count on the (new) lowest outstanding TSN reach Dupthresh the SCTP OPEN state is terminated and Fast Recovery is started.
- \* If Dupthresh miss indication count is not reached on the (new) lowest outstanding TSN, the state will now transit to SCTP TLR DELAY WAIT state for potential entering of SCTP TLR driven Fast Recovery if the PTO timer kicks prior to the (new) lowest outstanding TSN has been acknowledged or for potential later entering of Fast Recovery by reach of Dupthresh miss indication counts. When transiting to SCTP TLR DELAY WAIT the PTO timer relative to the (new) lowest outstanding TSN is reset to  $PTO2 - T\_latest(TSN)$ . In case  $PTO2 - T\_latest(TSN) \leq 0$ , the DELAY WAIT state is immediately terminated, the packet containing the lowest outstanding TSN is declared lost, and Fast Recovery is started.
- o The PTO timer relative to the lowest outstanding TSN may kick, in which case SCTP TLR will send a TLPP, reset the PTO timer relative to the lowest outstanding TSN to a T3 timer and transit to SCTP TLR PROBE WAIT state to await either the kick of the T3 relative to the lowest outstanding TSN (network is persistently unresponsive) or proof of network responsiveness and potential entering of SCTP TLR driven Fast Recovery unless the network responsiveness proof comes in form of cumulative acknowledgement of the TSN. The T3-value set relative to the lowest outstanding TSN when sending the TLPP probe and entering this state shall be:
  - \*  $MAX(PTO1, RTO - T\_latest(TSN))$ , when receiver side support for Immediate SACK has not been confirmed for the association, see Section 4.
  - \*  $MAX(PTO2, RTO - T\_latest(TSN))$ , when receiver side support for Immediate SACK has been confirmed for the association, see Section 4, and the SCTP sender itself deploys the Immediate SACK feature.

For further details on the TLPP transmission see Section 3.3.3.

#### 3.3.2.2. SCTP TLR PROBE WAIT STATE

In this state the lowest outstanding TSN has remained unSACK'ed for more than PTO time and no indication (no SACK of higher outstanding TSNS have been received) thus resulting in the transmittal of a TLPP to probe for the network responsiveness.

The T3-value set relative to the lowest outstanding TSN when sending the TLPP probe and entering this state is:

- o  $\text{MAX}(\text{PTO1}, \text{RTO} - \text{T\_latest}(\text{TSN}))$ , when receiver side support for Immediate SACK has not been confirmed for the association, see Section 4.
- o  $\text{MAX}(\text{PTO2}, \text{RTO} - \text{T\_latest}(\text{TSN}))$ , when receiver side support for Immediate SACK has been confirmed for the association, see Section 4, and the SCTP sender itself deploys the Immediate SACK feature.

For further details on the TLPP transmission see Section 3.3.3. Observe that in some special cases no TLPP is sent even if this state is entered and conceptually is handled as if a TLPP has been sent.

In the PROBE WAIT state the following may happen:

- o SACKs may arrive that makes the miss indication count on the lowest outstanding TSN/lowest TSN in flight reach Dupthresh in which case the PROBE WAIT state is terminated and Fast Recovery is started.
- o A SACK cumulatively acknowledging all holes including the lowest outstanding TSN may bring the SCTP TLR STM state back to SCTP TLR OPEN state. In this case a new PTO timer will be started on the new lowest outstanding TSN following the PTO timer setting in the SCTP TLR OPEN state. In this situation "PTO restart principles" (i.e., yielding  $\text{PTO} - \text{T\_latest}(\text{TSN})$ ) shall not be deployed. Spurious entering of PROBE WAIT state can happen if the PTO is too short, in such a situation it would not be prudent to deploy PTO restart principles when returning to OPEN state. OPEN ISSUE: Possibly PTO restart principles shall be refrained from until new RTT measurements are available.
- o A SACK may arrive for a higher outstanding TSN with lowest outstanding TSN on the path remaining unSACK'ed. This will result in declaration of the packet of the lowest outstanding TSN as lost and will make SCTP enter Fast Recovery.
- o A SACK may arrive that acknowledges the lowest outstanding TSN, but also data of higher TSN than the new lowest outstanding TSN are acknowledged in the SACK. In this case there is indication that either packet re-ordering has occurred or the new lowest outstanding TSN has been lost. The state will now transit to SCTP TLR DELAY WAIT state for potential entering of SCTP TLR driven Fast Recovery if the PTO timer kicks prior to the new lowest outstanding TSN has been acknowledged. The PTO timer set on the

new lowest outstanding TSN will be  $PTO2 - T\_latest(TSN)$ . In case  $PTO2 - T\_latest(TSN) \leq 0$ , the DELAY WAIT state is immediately terminated, the packet containing the lowest outstanding TSN is declared lost, and Fast Recovery is started.

- o The T3-timer may kick. In this case the PROBE WAIT state will be terminated and T3-recovery will start on non-SACK'ed outstanding data.

### 3.3.2.3. SCTP TLR DELAY WAIT STATE

In this state network responsiveness has been received (in form of a SACK of higher TSN than the lowest outstanding TSN) and the PTO timer relative to the lowest outstanding TSN is running for potential entering of SCTP TLR driven Fast Recovery.

The PTO set on a new lowest outstanding TSN in this state will be according to PTO2 in form of  $PTO2 - T\_latest(TSN)$ .

In the DELAY WAIT state the following may happen:

- o SACKs may arrive that will make the miss indication count on the lowest TSN in flight reach Dupthresh, the DELAY WAIT state is terminated and SCTP enters Fast Recovery.
- o The PTO timer relative to the lowest outstanding TSN may kick. This will result in declaration of packet of the lowest outstanding TSN as lost and will make SCTP enter Fast Recovery.
- o A SACK cumulatively acknowledging all holes including the lowest outstanding TSN may arrive and bring the SCTP TLR STM state back to SCTP TLR OPEN state and the PTO timer will be restarted on the new lowest outstanding TSN. The PTO timer value set will be the value  $PTO - T\_latest(TSN)$  where the PTO value is calculated either from PTO1 or PTO2 according to the evaluation criteria given for the OPEN state.
- o A SACK may arrive that acknowledges the lowest outstanding TSN, but also data of higher TSN than the new lowest outstanding TSN are acknowledged in the SACK. In this case there is indication that either packet re-ordering has occurred or the new lowest outstanding TSN has been lost. The state will remain in SCTP TLR DELAY WAIT state for potential entering of SCTP TLR driven Fast Recovery if the PTO timer kicks prior to the new lowest outstanding TSN has been acknowledged. The PTO timer set on the new lowest outstanding TSN will be  $PTO2 - T\_latest(TSN)$ . In case  $PTO2 - T\_latest(TSN) \leq 0$ , the DELAY WAIT state is terminated, the

packet containing the lowest outstanding TSN is declared lost and Fast Recovery is started.

- o A SACK may arrive that does not acknowledge the lowest outstanding TSN and still do not make the miss indication count reach the Dupthresh value. In this situation no changes are done to the PTO timer running and the state will remain in SCTP TLR DELAY WAIT state for potential entering of SCTP TLR driven Fast Recovery if the PTO timer kicks prior to the lowest outstanding TSN has been acknowledged.

#### 3.3.2.4. Exit of Loss Recovery

After exit of Fast Recovery or completion of T3-retransmission then if data is outstanding a PTO timer is started relative to the lowest outstanding TSN on the path and the state transits to either SCTP TLR OPEN state or to SCTP TLR DELAY Wait state depending on the status of the SACK scoreboard (i.e., do gaps exist or not). The PTO timer set will follow the rules described above. PTO-restart principles shall not be deployed in this situation as fresh RTT measurements might not be available. OPEN ISSUE: Possibly PTO restart principles shall be refrained from until new RTT measurements are available.

#### 3.3.2.5. RTO-Restart Principles for the T3-timer

When the lowest TSN in flight on a path is undergoing Fast Recovery or T3-retransmission a T3-timer is running on the path (relative to this lowest TSN in flight). For SCTP TLR the RTO-restart principles as of [HURTIG] SHOULD unconditionally be applied to the T3-timer. Thus the T3-timer set on a path in this case SHOULD be the value RTO-T\_latest(TSN) relative to the lowest TSN in flight on the path.

#### 3.3.3. TLPP Transmission Rules

The transmission of a Tail Loss Probe Packet (TLPP), done just prior to entering the SCTP TLR PROBE WAIT state from SCTP OPEN, is governed by the following details:

- o TLPP of new data is always preferred if such is available for transmission. If such exists, the TLPP sent is chosen as the lowest unsent TSNs that fit into one packet
- o Alternatively if no new data is available for transmission, either due to application or receiver side limitations, the presently outstanding packet with highest TSN number is chosen as the TLPP.
- o TLPP of retransmission data counts twice in the in-flight until acknowledged or detected as lost.

- o The transmittal of a TLPP of sub-PMTU size is not blocked by Nagle-like bundling.

The highest (new) outstanding TSN is chosen for probing in order to best possibly interface with standard Fast Recovery, i.e., to create a loss pattern situation that corresponds best possibly with how Fast Recovery algorithm retransmits, and is invoked to retransmit, lost packets.

TLPP Transmission conditions:

A TLPP is not sent unconditionally when SCTP enters PROBE WAIT state on a path.

No explicit limit is applied to the number of TLPP probe packets (i.e., the number of unacknowledged packets sent as TLPP) that may be outstanding at any given time but the number of such will in most situations be effectively limited to a very few (very often only one) by the following rules based on latency and congestion control principles; Generally a TLPP will not be allowed to breach the CWND more than once per RTT and further a TLPP is omitted to be sent if an already outstanding packet is considered to serve "good enough" from a network probing perspective. In addition special considerations are given for the transmittal of a TLPP consisting of retransmission data to ease loss masking detection (see Section 3.3.4). It is further noted that the frequency of TLPP transmittal is limited by how often a transition can happen out of and back into the PROBE WAIT state.

The conditional transmission of a TLPP is specified as follows:

- o If the highest outstanding TSN has been sent only a little while ago, this TSN effectively serves as a probe and no TLPP need to be send. This condition aims to prevent unnecessary retransmission of just sent data and unnecessary transmittal of small sub-PMTU packets of new data. The exact condition to apply is:

\* If  $T\_Latest(highTSN) < \gamma * SRTT$

then no TLPP is sent.  $\gamma = 1/2$  is recommended. A special condition arise when little data is outstanding and the SACK of the outstanding data may be lost by a single loss of SACK. In this case the transmittal of a TLPP packet will make the SACK return be robust toward single loss of SACK. For added robustness to SACK return an SCTP TLR implementation MAY disregard the above condition if only 2 packets are outstanding.

- o If no TLPP is outstanding, a probe is sent unconditionally of CWND.
- o If a TLPP is outstanding, a probe is sent conditionally to that there is room in CWND. Otherwise no TLPP is sent. I.e., the CWND is not breached when a TLPP is outstanding.
- o If no new data exists, a probe of retransmission data is sent conditional to whether a TLPP of retransmission data is already outstanding. I.e.,:
  - \* If no TLPP of retransmission data is outstanding, send TLPP consisting of highest outstanding TSN.
  - \* If a TLPP of retransmission data is outstanding, no TLPP is sent.

The above rules on probes of retransmission data are defined to ease the detection of TLPP recovered losses by the algorithm described in Section 3.3.4.

#### 3.3.3.1. Multi-Path Considerations for TLPP Transmission

In multi-homed [RFC4960] SCTP, multiple paths may have a PTO timer running on data in flight. E.g., two paths may be in SCTP OPEN state and SCTP will have two PTO timers running, each relative to the lowest outstanding TSN on the respective path. This (exception) situation in particular can occur as a result of a change of the data transfer path as a result of a switchback operation to a primary path. The handling of TLPP transmission for SCTP MH is described in the following. The underlying philosophy of the solution is, as far as possible, to have the SCTP TLR probing mechanism be undertaken on, and by, the data transfer path. Thus best possibly avoiding conflicts that may arise due to concurrent data transfers on multiple paths. As follows:

- o When the PTO timer kicks on a path in SCTP OPEN state and the TLPP selected by the rules above consists of new data, then if the path is the present data transfer path of the association the TLPP will be sent and in this case the TLPP is sent on the data transfer path of the association. When in this situation the path is not the present data transfer path of the association, then
  - \* if there is no outstanding data on the present data transfer path, the TLPP of new data is sent there.
  - \* if there is outstanding data on the data transfer path, the TLPP is not sent. Instead the potential transmittal of a TLPP

is deferred to be driven by a later kick of the PTO timer on the data transfer path.

The first situation that data is available for transmittal on the data transfer path but has not been sent, is an unlikely situation, but it might possibly occur in some implementations.

- o When the PTO timer kicks on a path in SCTP OPEN state and the TLPP selected by the rules above consist of retransmission of the presently highest outstanding TSNs on the association, then if and only if these TSNs are outstanding on the path in question is the TLPP allowed to be sent. The following guidelines are given for the path selection for the TLPP:
  - \* An SCTP implementation which does not implement the Unambiguous SACK feature of Appendix A should send the TLPP on the path on which the TNSs are presently outstanding (i.e., on the path on which the PTO kicked).
  - \* An SCTP implementation which implements the Unambiguous SACK feature of Appendix A may send the TLPP on the data transfer path of the association.

The reason a TLPP of retransmitted data in the first case above is sent on the path on which the data was first sent, even if this path is not the present data transfer path (special corner case with change of data transfer path or destination adders directed data transfer), is that the TLPP Loss Mask Detection mechanism, see Section 3.3.4 could not infer on which path to perform a congestion window reduction if the TLPP and original data is sent on different paths. An SCTP implementation which implements the Unambiguous SACK feature of Appendix A can better distinguish the SACK of the original TSN and the retransmitted TSN and can therefore operate differently. The choice of sending the TLPP on the data transfer path may be motivated by that the Fast Recovery procedure, which the SACK of the TLPP may result in, would use the data transfer path. On the other hand then differences in the RTT on the different paths may make it suboptimal to send the TLPP on the data transfer path as well as it can give rise to potential uncertainty in the TLPP Loss Recovery Mask detection and reaction process (see Section 3.3.4).

It is emphasized that the deferral of the transmission of a TLPP does not prevent entering of the PROBE WAIT state on the path where the PTO kicked.

### 3.3.4. Masking of TLPP Recovered Losses

If a single SCTP packet is lost, there is a risk that the TLPP packet itself might repair the loss if that particular lost packet is used as probe. The masking problem is only present if the TLPP is based on retransmission data. The TLPP might mask the loss and thus interfere with the congestion control principle that requires for CWND halving when a loss is detected.

At present the solution in this document operates with the algorithm defined for this purpose in [DUKKIPATI01] with adjustment to SCTP to rely on the D-SACK (duplicate TSN received) information available from SCTP SACK or alternatively to the information available from the Unambiguous SACK information of Appendix A. The solution operates with a conceptual TLPP Retransmission Episode. As follows:

- o Once a TLPP packet consisting of retransmission data is sent a TLPP Retransmission Episode is started.
- o A TLPP Retransmission Episode is abruptly terminated if Fast Recovery or T3-Recovery is entered.
- o For an SCTP implementation which does not implement the Unambiguous SACK feature of Appendix A, as well as for an SCTP association where the Unambiguous SACK feature of Appendix A is not in use, the TLPP Retransmission Episode terminates when an incoming SACK cumulatively acknowledges a sequence number higher than the sequence number of the TLPP probe with retransmission data. If at this time in stage the number of times the TLPP TSN has been received, according to the D-SACK information received, is lower than the number of times the TLPP TSN has been sent, CWND halving is done on the unique path on which the retransmission TLPP TSN has been sent. Further at this stage in time the contribution from the TSN is subtracted from the flight size in accordance to the number of times the TSN has been sent.
- o For an SCTP implementation which implements the Unambiguous SACK feature of Appendix A the following actions are taken at the time of acknowledgement of the TSN used as TLPP:
  - \* If the TLPP TSN is first cumulatively acknowledged in a SACK with CUMACK TSN = TLPP TSN and with no SACK (or CUMACK) of higher TSNs, then from the Unambiguous SACK information SCTP sender can classify to be in the following cases:
    - + The original TSN has not (yet) been received, the retransmission TSN (the TLPP) has been received.

- In this case the original TSN is judged as lost, CWND halving is performed on the path on which the original TSN was sent and the sent TSNs are subtracted from the flight size(s). This concludes the TLPP Retransmission Episode.
- + Both the original transmission as well as the retransmission (the TLPP) have been received.
  - In this case the sent TSNs are subtracted from the flight size(s). This concludes the TLPP Retransmission Episode.
- + The original TSN has been received, the retransmission TSN (the TLPP) has not yet been received:
  - In this case a special timer is started with value  $PTO - T\_latest(TSN)$  and the bytes of the retransmitted TSN (the TLPP) remains in the flightsize of the path on which it was sent until either of the following happens - whichever happens first:
    - o Unambiguous SACK of the TSN is received in which case the TSN is subtracted from the flightsize and the timer is stopped. This concludes the TLPP Retransmission Episode.
    - o A SACK of a higher TSN than the TLPP arrives with unambiguous SACK information indicating that the TLPP has not been received. Now marking is made on the path so that, if when the timer kicks, the TSN has still not been acknowledged, the TSN is judged as lost, CWND halving is done and the TSN is subtracted from the flightsize. This then concludes the TLPP Retransmission Episode.
    - o The timer kicks, the TSN is subtracted from the flightsize (but no CWND halving is done). This concludes the TLPP Retransmission Episode.
- \* If the TLPP TSN is first cumulatively acknowledged in a SACK with highest SACK'ed (or CUMACK'ed)  $TSN > TLPP\ TSN$ , then from the Unambiguous SACK information SCTP sender can classify the same cases as above and take corresponding actions. One additional situation can arise in this situation:
  - + Only one of the transmissions of the TSN has been received, but no clear Unambiguous SACK indication of which that was received is available from the SACK. This uncertainty can

only result from situations where SACKs are lost, potentially in combination with that more data chunks than the TSN it self were outstanding at the time when the TLPP was sent and some of this data arrived later at the receiver than the original TSN or the TLPP.

- In this case the original TSN is judged as having been received and it is subtracted on the flightsize of the path on which it was sent. The timer PTO-T\_latest(TSN) is set and handling of potential CWND reduction caused by loss of the TLPP is handled following the principles described above.

DISCUSSION of Unambiguous SACK Case Handling: CWND halving is not prescribed to be done for a potential lost retransmitted TSN used as TLPP in all cases above as there is no guarantee that a SACK confirming a potential arrival of the retransmitted TSN will arrive in time (i.e., this SACK may be lost). CWND halving is done if SACK of a higher TSN number than the TLPP number has arrived, PTO time has elapsed since the transmittal of the TLPP and the TLPP in it self cannot be determined to be received from the Unambiguous SACK information.

### 3.3.5. Elimination of unnecessary DELAY-ACK delays

The negative impact of DELAY\_ACK on the loss recovery delay is partially mitigated by setting of the I-bit on TLPP.

#### OPEN ISSUES:

- o It is to be determined if the Immediate SACK feature shall be relied on more aggressively. Possible options are:
  - \* Immediate SACK flag to be set on all retransmitted TSNs.
  - \* Immediate SACK flag to be set on all TSNs that are sent where the transmittal of an immediate following subsequent packet cannot be foreseen. This effectively would result in that the I-bit is set on a sent TSN whenever either of the following is true:
    - + no more chunks can be sent right after this chunk due to CWND limitations.
    - + no more chunks can be sent right after this due to RCV window limitations

- + no more chunks can be sent right after this as no more chunks are available in the SND buffer.
- + no more chunks can be sent right after this due to Nagle. (May depend on the exact Nagle-like implementation).

For the second choice it would be relevant to use PTO1 setting for the PTO timer on all TSNS sent with the I-bit set, when the receiver is known to support the Immediate SACK feature. The downside of this choice is that it very severely limits the effectiveness of the DELAY\_ACK feature.

- o Ideally the PTO timer relative to the lowest outstanding TSN should be adjusted to follow PTO2 when a subsequent packet is transmitted. The downside of this choice is the implementation impacts of such detailed - potentially per packet transmission - logic. To be elaborated further.

#### 4. Confirmation of support for Immediate SACK

Confirmation of receiver support of the Immediate SACK function, [RFC7053] is established by an SCTP TLR sender by the following means:

- o In case the data chunk of [RFC4960] is in use on the association, confirmation of [RFC7053] support by the SCTP receiver is assumed if SCTP TLR sender receives a data chunk with the I-bit flag set.
- o [TO DE CONFIRMED:] In case the I-data chunk of [SCTP-IDATA] is in use on the association, SCTP sender can by [SCTP-IDATA] assume that SCTP receiver supports [RFC7053].

#### 5. Socket API Considerations

This section will describe how the socket API defined in [RFC6458] is extended to provide a way for the application to control the retransmission algorithms in operation in the SCTP layer.

Socket option for control of the features is yet to be defined.

Please note that this section is informational only.

#### 6. Security Considerations

There are no new security considerations introduced by the functions defined in this document.

## 7. Acknowledgements

The author acknowledges Henrik Jensen for his very significant contribution for the definition of, the implementation of and the experiments with function.

The work heavily draws on prior art work done for TCP, [DUKKIPATI01] in particular. The contributors of that work should be credited for many of the ideas put forward here for SCTP.

## 8. IANA Considerations

This document does not create any new registries or modify the rules for any existing registries managed by IANA.

## 9. Discussion and Evaluation of function

Experiments in progress. Details to be filled in.

Right now we use this section to retain a number of issues that are to further elaborated on:

- o A significant number of spurious TLR probes have been observed in tests. It is to be determined if this is a fact of the function or whether it may be improved with adjustment of the PTO timer calculations.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <<http://www.rfc-editor.org/info/rfc4960>>.
- [RFC5061] Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", RFC 5061, DOI 10.17487/RFC5061, September 2007, <<http://www.rfc-editor.org/info/rfc5061>>.

- [RFC5062] Stewart, R., Tuexen, M., and G. Camarillo, "Security Attacks Found Against the Stream Control Transmission Protocol (SCTP) and Current Countermeasures", RFC 5062, DOI 10.17487/RFC5062, September 2007, <<http://www.rfc-editor.org/info/rfc5062>>.
- [RFC7053] Tuexen, M., Ruengeler, I., and R. Stewart, "SACK-IMMEDIATELY Extension for the Stream Control Transmission Protocol", RFC 7053, DOI 10.17487/RFC7053, November 2013, <<http://www.rfc-editor.org/info/rfc7053>>.
- [SCTP-IDATA]  
R. Stewart et al, , "Stream Schedulers and User Message Interleaving for the Stream Control Transmission Protocol draft-ietf-tsvwg-sctp-ndata-04.txt", IETF Work In Progress , 07 2015.

## 10.2. Informative References

- [CARO01] A. Caro et al, , "Retransmission Policies with Transport Layer Multihoming", ICON , 2003.
- [CARO02] A. Caro et al, , "Retransmission Schemes for End-to-end Failover with Transport Layer Multihoming", GLOBECOM , 11 2004.
- [CMT-SCTP]  
Amer et al., P., "Load Sharing for the Stream Control Transmission Protocol (SCTP) draft-tuexen-tsvwg-sctp-multipath-10.txt", IETF Work In Progress , 5 2015.
- [DUKKIPATI01]  
Dukkipati, N., Cardwell, N., Cheng, Y., and M. Mathis, "Tail Loss Probe (TLP): An Algorithm for Fast Recovery of Tail", Work Expired , 2 2013.
- [DUKKIPATI02]  
Dukkipati, N., Mathis, M., Cheng, Y., and M. Ghobadi, "Proportional Rate Reduction for TCP", Proceedings of the 11th ACM SIGCOMM Conference on Internet Measurement , 11 2011.
- [HURTIG] P. Hurtig et al., , "TCP and SCTP RTO Restart, draft-ietf-tcpm-rtorestart-08", IETF Work In Progress , 3 2015.
- [MATHIS] Mathis, M., "FACK", ACM SIGCOMM Computer Communication Review 26,4, 10 1996.

- [Rajiullah] M. Rajiullah et al., , "An Evaluation of Tail Loss Recovery Mechanisms for TCP", ACM SIGCOMM Computer Communication Review 45,1, 1 2015.
- [RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, DOI 10.17487/RFC3758, May 2004, <<http://www.rfc-editor.org/info/rfc3758>>.
- [RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <<http://www.rfc-editor.org/info/rfc5681>>.
- [RFC5827] Allman, M., Avrachenkov, K., Ayesta, U., Blanton, J., and P. Hurtig, "Early Retransmit for TCP and Stream Control Transmission Protocol (SCTP)", RFC 5827, DOI 10.17487/RFC5827, May 2010, <<http://www.rfc-editor.org/info/rfc5827>>.
- [RFC6458] Stewart, R., Tuexen, M., Poon, K., Lei, P., and V. Yasevich, "Sockets API Extensions for the Stream Control Transmission Protocol (SCTP)", RFC 6458, DOI 10.17487/RFC6458, December 2011, <<http://www.rfc-editor.org/info/rfc6458>>.
- [RFC6675] Blanton, E., Allman, M., Wang, L., Jarvinen, I., Kojo, M., and Y. Nishida, "A Conservative Loss Recovery Algorithm Based on Selective Acknowledgment (SACK) for TCP", RFC 6675, DOI 10.17487/RFC6675, August 2012, <<http://www.rfc-editor.org/info/rfc6675>>.
- [SCTP-PF] Y. Nishida et al, , "SCTP-PF: Quick Failover Algorithm in SCTP, draft-ietf-tsvwg-sctp-failover-13.txt", IETF Work In Progress , 09 2015.
- [zimmermann01] Zimmermann, A., "CUBIC for Fast Long-Distance Networks, draft-ietf-tcpm-cubic-00", IETF Work In Progress , 6 2015.
- [zimmermann02] Zimmermann, A., "The TCP Echo and TCP Echo Reply Option, draft-zimmermann-tcpm-echo-option-00", IETF Work In Progress , 6 2015.

[zimmermann03]

Zimmermann, A., "Using the TCP Echo Option for Spurious Retransmission Detection, draft-zimmermann-tcpm-spurious-rxmit-00", IETF Work In Progress , 7 2015.

#### Appendix A. Unambiguous SACK

When receiving a SACK of a TSN it is not possible to unambiguously determine if the receiver hereby acknowledges the first transmission of the TSN or possible subsequent retransmissions of the TSN, when such multiple transmissions of the same TSN have been made. The duplicate TSN information in the SCTP SACK chunk does help to provide information about how many times the same TSN has been received at the received side, but still it is not possible to unequivocally link the SACK information to the different transmissions of the same TSN. An additional source of ambiguity comes from the fact that packets may be duplicated in the network.

Unambiguous SACK information is generally beneficial for many SCTP protocol aspects, e.g., for improved RTT measurements, for more accurate loss detection, maintain of flightsize and congestion control operation.

Providing full accurate SACK information from receiver to sender side requires a reliable (and ordered) SACK feedback channel thus overcoming the information gap that may arise from loss (or from re-ordering) of SACKs. The establishment of such a reliable feedback Chanel is not proposed but the proposal implements measures that allow for some robustness towards information loss due to SACK loss.

NOTE for AUTHORS: The solution is independent from a potential split of the SACK TSN Gap information in SACK and NR-SACK gaps respectively following [CMT-SCTP].

##### A.1. TSN Retransmission ID in Data Chunk Header

It is a prerequisite that the SCTP association deploy, and has negotiated usage of, the new I-data chunk of [SCTP-IDATA].

We define a new 4-bit Retransmission ID (RTX ID) in the I-data Chunk header. The 4 bits consume 4 bits of the new reserved 16-bit field of the I-data chunk header. See Figure 1.

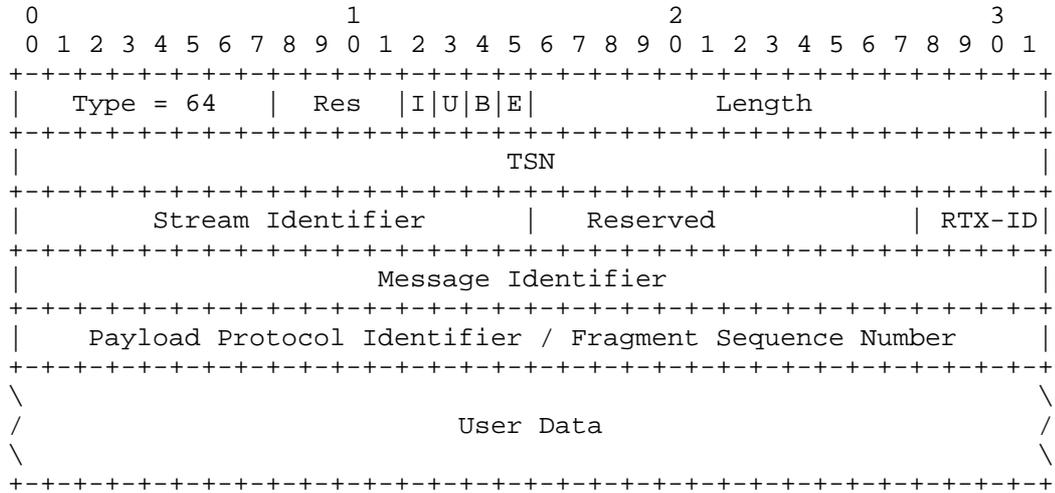


Figure 1: RTX-ID in I-DATA chunk format

A.1.1. Sender side behaviour

New data MUST be sent with RTX-ID =0. Whenever SCTP retransmits a data chunk it SHOULD step up the RTX ID. The highest RXT ID = 15 is used for all retransmissions of the same TSN beyond the 15-th retransmission or when the RTX ID last used fort his TSN is 15. An SCTP sender MAY step the RTX ID up with more than one count when retransmitting a TSNs in order to have all TSNs within the SCTP packet use the one and the same RTX ID.

A.1.2. Receiver side behaviour

An SCTP receiver supporting this feature MUST process the RTX ID for all received TSNs in accordance with the prescriptions for Unambiguous SACK return below.

A.2. Unambiguous SACK Chunk

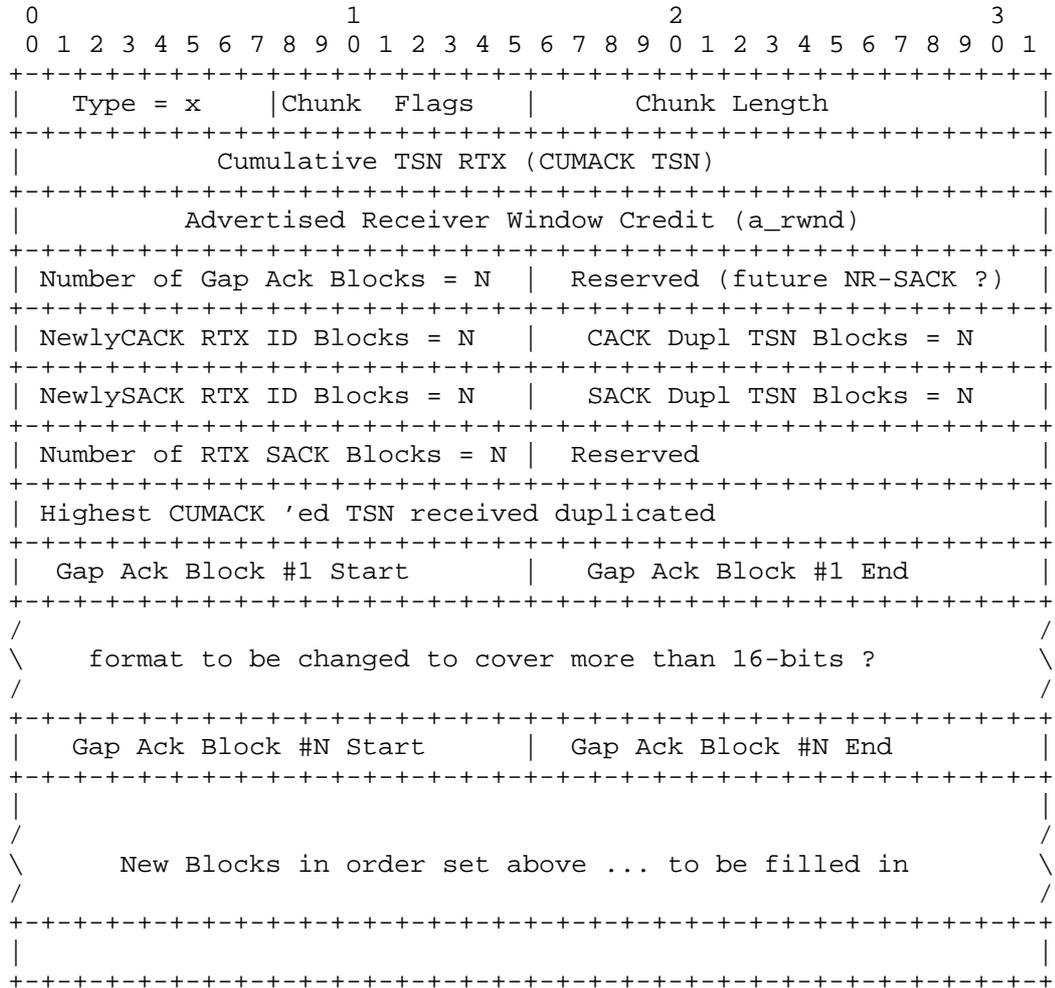


Figure 2: Unambiguous SACK chunk format

Newly CACK RTX ID block:

This block provides information on the newly acknowledged TSNs that were cumulatively acked in this SACK and for which the following hold:

- \* The TSN is newly acked in this SACK. I.e., the TSN has not been received before (or if it has been received before it was since renege).

- \* The newly acknowledged TSN was received with RTX ID different from zero.

The RTX ID received with the TSN is returned in this block. The information returned in a CACK RTX ID block is a consecutive range of TSN fulfilling the above for which identical RTX ID has been received. Proposed format is off-set from CUMACK TSN (lower than CUMACK TSN), length of range and RTX ID.

#### Newly SACK RTX ID block:

This block provides information on the newly acknowledged TSNs that were selectively acknowledged in this SACK and for which the following hold:

- \* The TSN is newly acked in this SACK. I.e., the TSN has not been received before (or if it has been received before, it was since renege).
- \* The newly acknowledged TSN was received with RTX ID different from zero.

The RTX ID received with the TSN is returned in this block. The information returned in a SACK RTX ID block is a consecutive range of TSN fulfilling the above for which identical RTX ID has been received. Proposed format is off-set from CUMACK TSN (higher than CUMACK TSN), length of range and RTX ID - OR alternatively format of present SACK blocks with off set bounded by 16-bit to CUMACK TSN.

#### Newly CACK Dupl TSN block:

This block provides information on the TSNs received since last returned SACK for which following hold:

- \* The TSN is lower than or equal to the CUMACK TSN.
- \* The TSN is a duplicate. Meaning that a data chunk with same TSN, but possibly different RTX ID, has been received.

The RTX ID received with the TSN is returned in this block. The information returned in a CACK Dupl TSN block is a consecutive range of TSN fulfilling the above for which identical RTX ID has been received. Proposed format is off-set from CUMACK TSN (lower than CUMACK TSN), length of range and RTX ID. The RTX ID may be zero.

#### Newly SACK Dupl TSN block:

This block provide information on the TSNs received since last returned SACK for which the following hold:

- \* The TSN is higher than the CUMACK TSN.
- \* The TSN is a duplicate. Meaning that a data chunk with same TSN, but possibly different RTX ID, has been received.

The RTX ID received with the TSN is returned in this block. The information returned in a SACK Dupl TSN block is a consecutive range of TSN fulfilling the above for which identical RTX ID has been received. Proposed format is off-set from CUMACK TSN (higher than CUMACK TSN), length of range and RTX ID - OR - format of present SAC blocks with off set bounded by 16-bit to CUMACK TSN. The RTX ID may be zero.

Together with the existing SACK information, the Newly CACK/SACK RTX ID and the CACK/SACK Dupl TSN blocks provide unambiguous SACK information for all received TSNs differentiating on the RTX ID received with the TSN. The information may be partially lost from the receiver to the sender if a SACK is lost. The RTX SACK Block and the Highest CUMACK Received Duplicated information is returned in order to provide means to recover part of the information that can be lost when a SACK is lost.

RTX SACK block:

This block provides information on the TSNs for which the following hold:

- \* The TSN has been received and has been selectively acked in prior SACKs (OPEN: alternatively in SACKs including this one).
- \* The TSN is higher than the CUMACK TSN.
- \* The TSN has been received only with RTX IDs different from zero.

The information returned in an RTX block is a consecutive range of TSN fulfilling the above. Proposed format is off-set from CUMACK TSN (higher than CUMACK TSN) and length of range - OR - format of present SACK blocks with off set - bounded by 16-bit to CUMACK TSN.

Highest CUMACK'ed TSN received Duplicated:

Here the highest TSNs that fulfill the following condition is inserted:

- \* The TSN has been received duplicated
- \* The TSN is lower than or equal to the CUMACK TSN.

When no duplicates have been seen or when no duplicates have been seen in last  $2^{31}$  window of TSNs that have been cumulatively acknowledged, CUMACK TSN +1 is returned.

By means of the RTX SACK block an SCTP sender may recover the information that a SACK'ed TSN does not represent the original TSN first sent. I.e., the TSN sent with RTX ID = 0.

By means of the "Highest CUMACK'ed TSN received Duplicated" an SCTP receiver may recover the information that more than one incarnation of a TSN has been received when the SACK, which cumulatively acknowledged the arrival of the different incarnations of the TSN, in it self was lost. The particular example of special interest is the case where the one and the same SACK would contain information on receipt of both the original TSN and a spurious retransmission of the TSN. Such can happen in scenarios where DELAY\_ACK handling at the receiver side delays the return of SACK information and a SACK is lost, even if the original data and the spurious retransmission data was sent with reasonable spacing in time.

#### A.2.1. Receiver side behaviour

The RTX SACK Block and the Highest CUMACK information to be returned in SACKs demand for an SCTP receiver to keep track (state) of the following information on a per association basis:

- o A list (or ranges) of TSNs that have been SACK'ed, but not yet cumulatively acknowledged and for which RTX ID = 0 has not been seen. It is noted that the TSN data chunk itself may have been delivered to the application.
- o The highest TSN lower than CUMACK TSN for which a duplicate has been received.

#### A.3. Unambiguous SACK return

Whenever Unambiguous SACKs are in use on an association and SCTP receives a valid data chunk with RTX-ID different from zero it shall not delay the return of the Unambiguous SACK. Otherwise Unambiguous SACKs are returned at any time when an [RFC4960] implementation would return a SACK.

A window opener MUST include Unambiguous SACK information.

## A.4. Negotiation

An SCTP receiver MUST NOT send an Unambiguous SACK chunk unless both peers have indicated its support of the Unambiguous SACK feature within the Supported Extensions Parameter as defined in [RFC5061]. If Unambiguous SACK has been negotiated on an association, Unambiguous SACKs MUST be returned whenever a SCTP receiver would return SACK information. If Unambiguous SACK has not been negotiated on an association, the RTX-ID field in the chunk header of incoming data chunks MUST be ignored and [RFC4960] SACK format and return policies MUST be adhered to.

## Authors' Addresses

Karen E. E. Nielsen  
Ericsson  
Kistavaegen 25  
Stockholm 164 80  
Sweden

Email: karen.nielsen@tieto.com

Rafaele De Santis  
Ericsson  
xx  
xx xx  
Italy

Email: rafaele.de.santis@ericsson.com

Anna Brunstrom  
Karlstad University  
Universitetsgatan 2  
Karlstad 651 88  
Sweden

Email: anna.brunstrom@kau.se

Michael Tuexen  
Muenster Univ. of Appl. Science  
Stegerwaldstrasse 39  
Steinfurt 48565  
Germany

Email: tuexen@fh-muenster.de

Randall Stewart  
Netflix, Inc.  
xx  
Chapin 29036 SC  
United States

Email: randall@lakerest.net