Network Working Group	R. Stewart	
Internet-Draft	Researcher	
Intended status: Standards Track	P. Lei	
Expires: August 20, 2009	Cisco Systems, Inc.	
	M. Tuexen	
	Muenster Univ. of Applied Sciences	
	February 16, 2009	

TOC

Stream Control Transmission Protocol (SCTP) Stream Reconfiguration draft-stewart-tsvwg-sctpstrrst-01.txt

Status of this Memo

Copyright Notice

Abstract

Many applications that desire to use SCTP have requested the ability to "reset" a stream. The intention of resetting a stream is to start the

numbering sequence of the stream back at 'zero' with a corresponding notification to the upper layer that this act as been performed. The applications that have requested this feature normally desire it so that they can "re-use" streams for different purposes but still utilize the stream sequence number for the application to track the message flows. Thus, without this feature, a new use on an old stream would result in message numbers larger than expected without a protocol mechanism to "start the streams back at zero". This documents presents also a method for resetting the transport sequence numbers and all stream sequence numbers.

Table of Contents

---

1. Introduction

TOC

Many applications that desire to use [RFC4960] (Stewart, R., “Stream Control Transmission Protocol,” September 2007.) have requested the ability to "reset" a stream. The intention of resetting a stream is to start the numbering sequence of the stream back at 'zero' with a corresponding notification to the upper layer that this act as been performed. The applications that have requested this feature normally desire it so that they can "re-use" streams for different purposes but still utilize the stream sequence number for the application to track the message flows. Thus, without this feature, a new use of an old stream would result in message numbers larger than expected without a protocol mechanism to "start the streams back at zero". This documents presents also a method for resetting the transport sequence numbers and all stream sequence numbers.
[Editors note: We probably need to add more text here]

---

2. Conventions

TOC

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] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

---

3. Data Formats

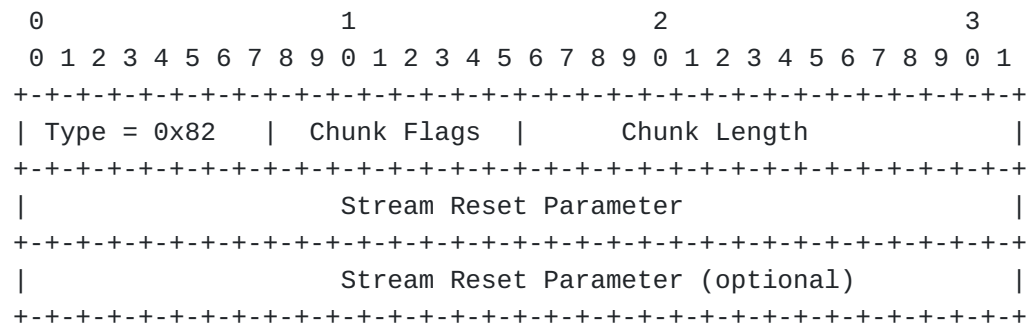
TOC

This section examines all new data formats defined by this document. All transported integer numbers are in "network byte order" a.k.a., Big Endian, unless otherwise noted.

TOC

3.1. STREAM RESET Chunk

This document adds one new chunk type to SCTP. The suggested value for this chunk is 0x82 hex or 130 decimal. The range selected by IANA must have the upper bit (or ignore bit) set and the next to highest bit (or the report bit) cleared. The chunk has the following format:



Chunk Type: 1 byte (unsigned integer) This field holds the IANA defined chunk type for the STREAM RESET chunk.

Chunk Flags: 1 byte (unsigned integer) This field is set to 0 by the sender and ignored by the receiver.

Chunk Length: 2 bytes (unsigned integer) This field holds the length of the chunk, including the Chunk Type, Chunk Flags and Chunk Length.

Stream Reset Parameter This field holds a Stream Reset Request Parameter or a Stream Reset Response Parameter.

Note each STREAM RESET chunk holds at least one parameter and at most two parameters. Only the following combinations are allowed:

1. Outgoing SSN Reset Request Parameter.
2. Incoming SSN Reset Request Parameter.
3. Outgoing SSN Reset Request Parameter, Incoming SSN Reset Request Parameter.
4. SSN/TSN Reset Request Parameter.
5. Stream Reset Response Parameter.
6. Stream Reset Response Parameter, Outgoing SSN Reset Request Parameter.
7. Stream Reset Response Parameter, Stream Reset Response Parameter.

3.2. New Parameters

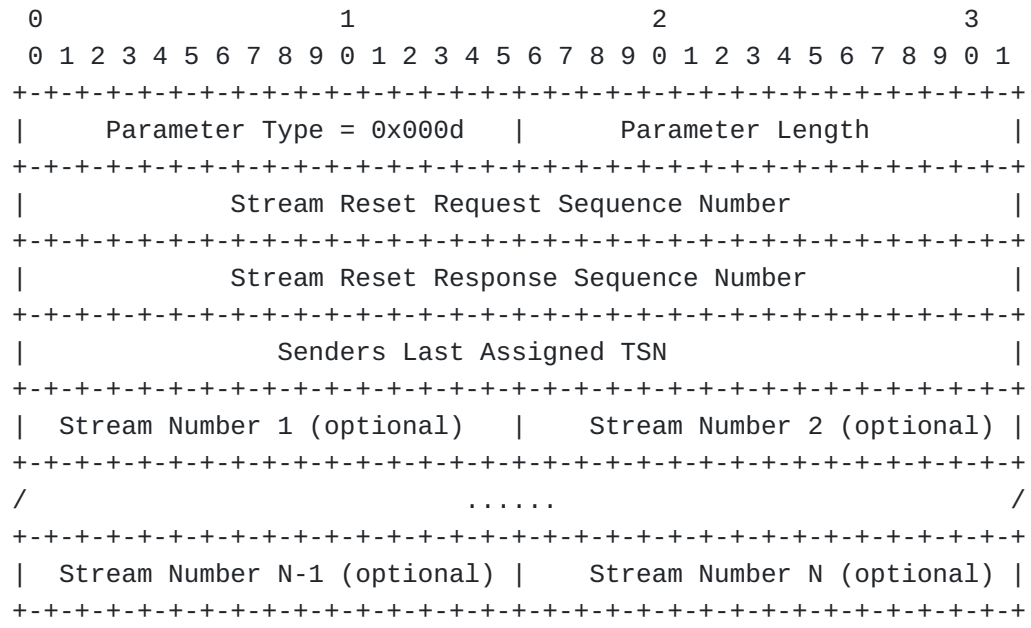
TOC

This section identifies four new parameters, their formats, and in what chunk type these parameters may appear.

3.2.1. Outgoing SSN Reset Request Parameter

TOC

This parameter is used by the sender to request some outgoing streams to be reset.



Parameter Type: 2 bytes (unsigned integer)
This field holds the IANA defined parameter type for Stream Reset Request Parameter. The suggested value of this field for IANA is 0x000d.

Parameter Length: 2 bytes (unsigned integer) This field holds the length of the parameter.

Stream Reset Request Sequence Number: 4 bytes (unsigned integer)
This field is used to identify the request. It is a monotonically increasing number that is initialized to the same value as the Initial TSN number. It is increased by 1.

Stream Reset Response Sequence Number: 4 bytes (unsigned integer)
In case that this Outgoing SSN Reset Request Parameter is sent in response to an Incoming SSN Reset Request Parameter this parameter is also an implicit response to the incoming request. Then this field holds the Stream Reset Request Sequence Number of the incoming request. In the other case it holds the next expected Stream Reset Request Sequence Number - 1.

Senders last assigned TSN: 4 bytes (unsigned integer) This value holds the next TSN minus 1, in other words the last TSN that this sender assigned.

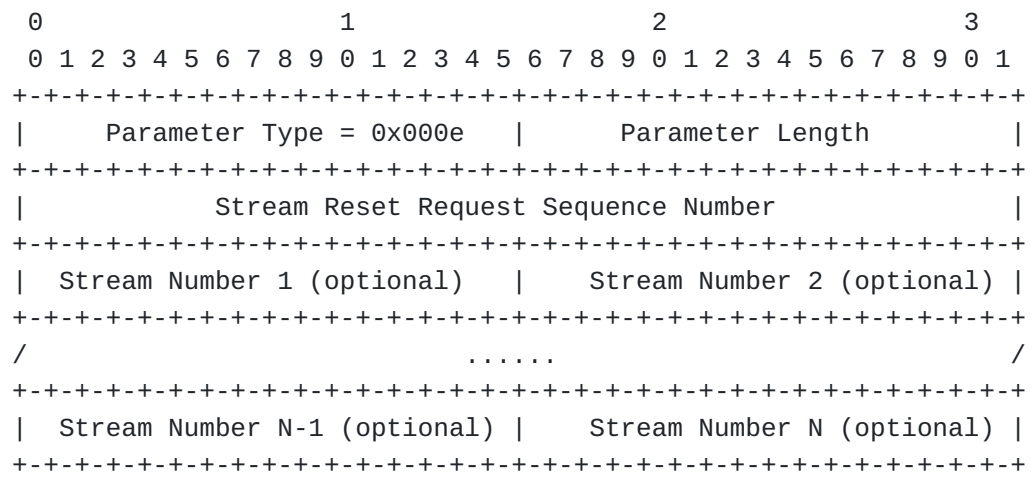
Stream Number N: 2 bytes (unsigned integer) This optional field, if included, is used to indicates specific streams that are to be reset. If no streams are listed, then ALL streams are to be reset.

This parameter can appear in a STREAM RESET chunk. This parameter MUST NOT appear in any other chunk type.

3.2.2. Incoming SSN Reset Request Parameter

TOC

This parameter is used by the sender to request that the peer requests some of its outgoing streams to be reset.



Parameter Type: 2 bytes (unsigned integer) This field holds the IANA defined parameter type for Stream Reset Request Parameter. The suggested value of this field for IANA is 0x000e.

Parameter Length: 2 bytes (unsigned integer) This field holds the length of the parameter.

Stream Reset Request Sequence Number: 4 bytes (unsigned integer) This field is used to identify the request. It is a monotonically increasing number that is initialized to the same value as the Initial TSN number. It is increased by 1.

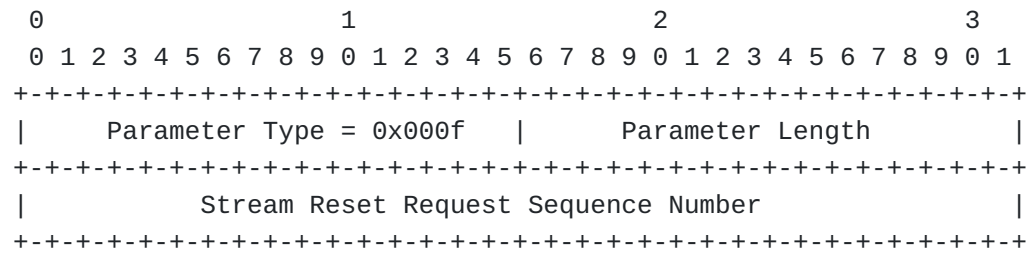
Stream Number N: 2 bytes (unsigned integer) This optional field, if included, is used to indicate specific streams that are to be reset. If no streams are listed, then ALL streams are to be reset.

This parameter can appear in a STREAM RESET chunk. This parameter MUST NOT appear in any other chunk type.

3.2.3. SSN/TSN Reset Request Parameter

TOC

This parameter is used by the sender to request to reset the TSN and SSN numbering of all streams.



Parameter Type: 2 bytes (unsigned integer) This field holds the IANA defined parameter type for Stream Reset Request Parameter. The suggested value of this field for IANA is 0x000f.

Parameter Length: 2 bytes (unsigned integer) This field holds the length of the parameter.

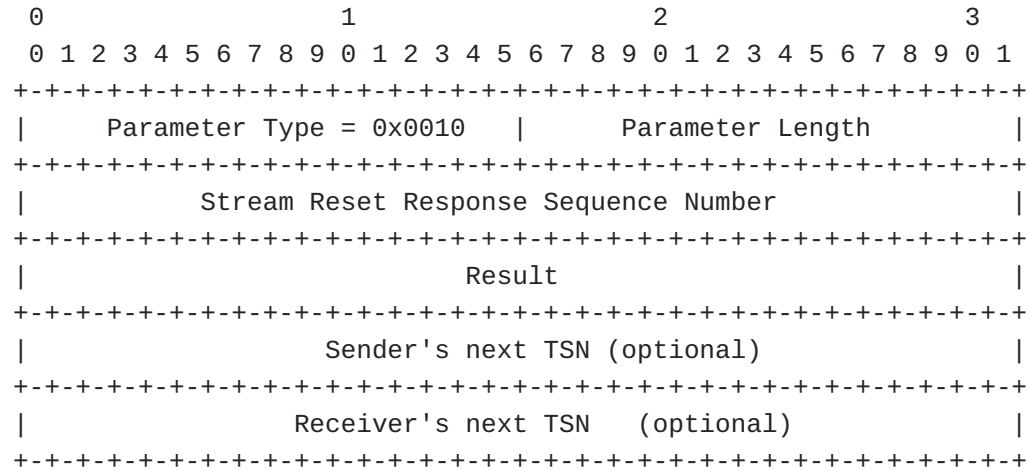
Stream Reset Request Sequence Number: 4 bytes (unsigned integer) This field is used to identify the request. It is a monotonically increasing number that is initialized to the same value as the Initial TSN number. It is increased by 1.

This parameter can appear in a STREAM RESET chunk. This parameter MUST NOT appear in any other chunk type.

3.2.4. Stream Reset Response Parameter

TOC

This parameter is used by the receiver of a stream reset request parameter to respond to the stream reset request.



Parameter Type: 2 bytes (unsigned integer)

This field holds the IANA defined parameter type for Stream Reset Response Parameter. The suggested value of this field for IANA is 0x0010.

Parameter Type Length: 2 bytes (unsigned integer) This field holds the length of the parameter.

Stream Reset Response Sequence Number: 4 bytes (unsigned integer) This value is copied from the request parameter and is used by the receiver of the Stream Reset Response Parameter to tie the response to the request.

Result: 4 bytes (unsigned integer) This value describes the result of the processing of the request. It is encoded as given by the following table

---

Result	Description
0	Nothing to do
1	Performed
2	Denied
3	Error - Wrong SSN
4	Error - Request already in progress
5	Error - Bad Sequence Number

Table 1

Sender's next TSN: 4 bytes (unsigned integer) This field holds the TSN the sender of the Response will use to send the next DATA chunk. The field is only applicable in responses to SSN/TSN reset requests.

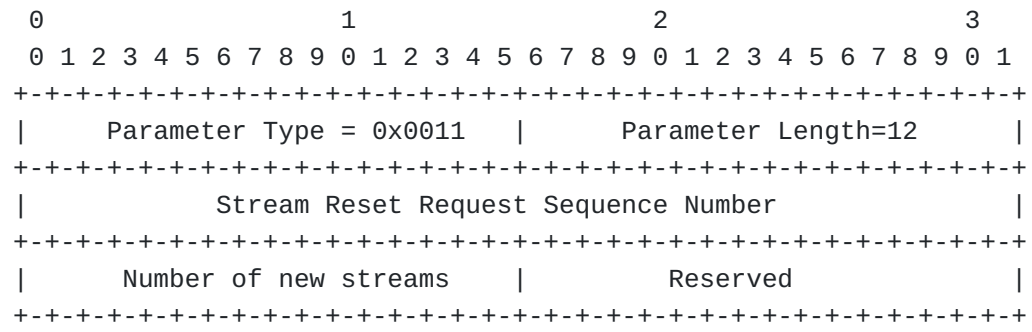
Receiver's next TSN: 4 bytes (unsigned integer) This field holds the TSN the receiver of the response must use to send the next DATA chunk. The field is only applicable in responses to SSN/TSN reset requests.

This parameter can appear in a STREAM RESET chunk. This parameter MUST NOT appear in any other chunk type.

TOC

3.2.5. Add Streams

This parameter is used by the sender to request that the peer add the requested number of streams to the association.



Parameter Type: 2 bytes (unsigned integer) This field holds the IANA defined parameter type for the Add Streams Parameter. The suggested value of this field for IANA is 0x0011.

Parameter Length: 2 bytes (unsigned integer) This field holds the length of the parameter, a fixed value of 12 MUST be found in this field.

Stream Reset Request Sequence Number: 4 bytes (unsigned integer) This field is used to identify the request. It is a monotonically increasing number that is initialized to the same value as the Initial TSN number. It is increased by 1.

Number of new streams: 2 bytes (unsigned integer) This value holds the number of additional streams the sender would like added to the association. Streams are added in order and are consecutive, e.g. if an association has four streams (0-3) and a requested is made to add 3 streams then the new streams will be 4, 5 and 6.

Reserved: 2 bytes (unsigned integer) This field is reserved. It SHOULD be set to 0 by the sender and ignored by the receiver.

This parameter can appear in a STREAM RESET chunk. This parameter MUST NOT appear in any other chunk type.

---

4. Procedures

TOC

This section defines the procedures used by both the sender and receiver of a stream reset. We also give various example stream reset scenarios.

4.1. Sender side procedures

TOC

This section describes the procedures related to the sending of Stream Reset Chunks. A Stream Reset Chunk is a composition of a Type Length Value (TLV) parameters.

4.1.1. Sender side procedures for the Stream Reset Chunk

TOC

Note that before sending a Stream Reset Chunk the sender MUST ensure that the peer advertised support for the stream reset extension. The indication for support of the extensions MUST be determined using the Supported Extensions Parameter in either the INIT or INIT-ACK. This parameter is defined in [RFC5061] (Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration," September 2007.). If the chunk value '0x82' does NOT appear in the supported extensions list of chunks, then the sender MUST NOT send any stream reset request to the peer and any request by the application for such service SHOULD be responded to with an appropriate error indicating the peer SCTP stack does not support the stream reset extension.
After packaging the Stream Reset Chunk and sending it to the peer the sender MUST start a 'Stream Reset Timer' when the STREAM RESET chunk contains at least one request parameter. If it contains no request parameter, the Stream Reset Timer MUST NOT be started. This timer MUST use the same value as SCTP's Data transmission timer (i.e. the RTO timer) and MUST use exponential backoff doubling the value at every expiration. If the timer does expire, besides doubling the value, the sender MUST retransmit the Stream Reset Chunk, increment the appropriate error counts (both for the association and the destination), and do threshold management possibly destroying the association if SCTP retransmission thresholds are surpassed.

4.1.2. Sender side procedures for the Outgoing SSN Reset Request Parameter

TOC

When an SCTP sender wants to reset the SSNs of some or all outgoing streams it can send an Outgoing SSN Reset Request Parameter if the Stream Reset Timer is not running. The following steps MUST be followed:

> **A1:** The sender MUST stop assigning new SSNs to new user data provided by the upper layer. This is because it is unknown as to

if the receiver of the request will accept or deny it and more so, a lost request might cause an out-of-sequence error in a stream that the receiver is not yet prepared to handle.

A2: The sender MUST assign the next stream reset request sequence number and put it into the Stream Reset Request Sequence Number field of the Outgoing SSN Reset Request Parameter. After assigning it the next stream reset request sequence number MUST be incremented by '1'.

A3: If this Outgoing SSN Reset Request Parameter is sent in response to an Incoming SSN Request Parameter the Stream Reset Request Sequence Number of the Incoming SSN Request Parameter is copied into the Stream Reset Response Sequence Number field of the Outgoing SSN Reset Request Parameter. If the Outgoing SSN Reset Request Parameter is sent on request of the upper layer the Stream Reset Response Sequence Number is the next expected Stream Reset Request Sequence Number of the peer - 3.

A4: The sender fills in the TSN it has assigned last.

A5: If this Outgoing SSN Reset Request Parameter is sent in response to an Incoming SSN Request Parameter the Stream Numbers are copied from the Incoming SSN Request Parameter to the Outgoing SSN Reset Request Parameter. If this Outgoing SSN Reset Request Parameter is sent on request of the upper layer and the sender wants all outgoing streams to be reset no Stream Numbers MUST be put into the Outgoing SSN Reset Request Parameter. If the sender wants only some outgoing streams to be reset these Stream Numbers MUST be filled in the Outgoing SSN Reset Request Parameter.

A6: The Outgoing SSN Reset Request Parameter is put into a STREAM RESET Chunk. It MAY be put together with an Incoming SSN Reset Request Parameter or an Stream Reset Response Parameter and MUST NOT be put together with any other parameter.

A7: The STREAM RESET Chunk is sent following the rules given in Section 4.1.1 (Sender side procedures for the Stream Reset Chunk)

4.1.3. Sender side procedures for the Incoming SSN Reset Request Parameter

TOC

When an SCTP sender wants to reset the SSNs of some or all incoming streams it can send an Incoming SSN Reset Request Parameter if the

Stream Reset Timer is not running. The following steps MUST be followed:

B1: The sender MUST assign the next stream reset request sequence number and put it into the Stream Reset Request Sequence Number field of the Incoming SSN Reset Request Parameter. After assigning it the next stream reset request sequence number MUST be incremented by '1'.

B2: If the sender wants all incoming streams to be reset no Stream Numbers MUST be put into the Incoming SSN Reset Request Parameter. If the sender wants only some incoming streams to be reset these Stream Numbers MUST be filled in the Incoming SSN Reset Request Parameter.

B3: The Incoming SSN Reset Request Parameter is put into a STREAM RESET Chunk. It MAY be put together with an Outgoing SSN Reset Request Parameter and MUST NOT be put together with any other parameter.

B4: The STREAM RESET Chunk is sent following the rules given in Section 4.1.1 (Sender side procedures for the Stream Reset Chunk)

4.1.4. Sender side procedures for the SSN/TSN Reset Request Parameter

TOC

When an SCTP sender wants to reset the SSNs and TSNs it can send a SSN/TSN Reset Request Parameter if the Stream Reset Timer is not running. The following steps MUST be followed:

C1: The sender MUST assign the next stream reset request sequence number and put it into the Stream Reset Request Sequence Number field of the SSN/TSN Reset Request Parameter. After assigning it the next stream reset request sequence number MUST be incremented by '1'.

C2: The sender MUST queue any user data.

C3: The SSN/TSN Reset Request Parameter is put into a STREAM RESET Chunk. There MUST NOT be any other parameter in this chunk.

C4: The STREAM RESET Chunk is sent following the rules given in Section 4.1.1 (Sender side procedures for the Stream Reset Chunk)

4.1.5. Sender side procedures for the Stream Reset Response Parameter

TOC

When an implementation receives a request parameter it MUST respond with a Stream Reset Response Parameter in the following manner:

D1 The Stream Reset Request Sequence number of the incoming request is copied to the Stream Reset Response Sequence Number field of the Stream Reset Response Parameter.

D2 The result of the processing of the incoming request is filled in the Result field of the Stream Reset Response Parameter

D3 If the incoming request is a SSN/TSN reset requests, the Sender's next TSN field is filled with the next TSN the sender of this Stream Reset Response Parameter will assign. For other requests the Sender's next TSN field is not filled.

D4 If the incoming request is a SSN/TSN reset request, the Receiver's next TSN field is filled with a TSN such that the sender of the Stream Reset Response Parameter can be sure it can discard received DATA chunks with smaller TSNs. A good value for this is the highest TSN it has seen plus some delta. For other requests the Sender's next TSN field is not filled.

4.1.6. Sender side procedures for addition of streams

TOC

When an SCTP sender wants to increase the number of outbound streams it is able to send to, it may add a Add Streams parameter to the STREAM RESET chunk. Upon sending the request the sender MUST await a positive acknowledgment (Success) before using any additional stream added by this request. Note that new streams are added adjacent to the previous steams with no gaps. This means that if a request is made to add 2 streams to an association that has already 5 (0-4) then the new streams, upon successful completion, are streams 5 and 6. Any new stream MUST number its first message to be stream sequence 0.

4.2. Receiver side procedures

TOC

TOC

4.2.1. Receiver side procedures for the Stream Reset Chunk

Upon reception of a Stream Reset Chunk each parameter within it should be processed. If some parameters have to be sent back, they MUST all be put into one STREAM RESET chunk. If the received STREAM RESET chunk contains at least one request parameter, a SACK chunk MUST be sent back and MAY be bundled with the STREAM RESET chunk. If the received STREAM RESET chunk contains at least one request and based on the analysis of the Stream Reset Request Sequence Numbers this is the last received STREAM RESET chunk, the same STREAM RESET chunk has to be sent back in response as earlier.

4.2.2. Receiver side procedures for the Outgoing SSN Reset Request Parameter

TOC

The decision to deny a stream reset request is an administrative decision and may be user configurable even after the association has formed. If for whatever reason the endpoint does NOT wish to reset any streams it MUST send a stream reset response as described in Section 4.1.5 (Sender side procedures for the Stream Reset Response Parameter) with an appropriate Result field.
In the case that the endpoint is willing to perform a stream reset the following steps SHOULD be followed:

E1 If the Senders Last Assigned TSN number is greater than the cumulative acknowledgment point, then the endpoint must enter "deferred reset processing". In this mode, any data arriving with a TSN number larger than the 'senders last assigned TSN' for the effected stream(s) MUST be queued locally and held until the Cumulative Acknowledgment point reaches the 'senders last assigned TSN number'. When the Cumulative Acknowledgment point reaches the last assigned TSN number then proceed to the next step. Note that the receiver of a stream reset that causes it to entered deferred reset processing does NOT withhold the stream reset acknowledgment from the peer. This also means that the receiver will need to queue up any additional stream reset requests received including the one that caused the receiver to enter deferred reset processing.

E2 If the Stream Reset Timer is running for the Stream Reset Request Sequence Number indicated in the Stream Reset Response Sequence Number field, mark the Stream Reset Request Sequence Number as acknowledged. If all Stream Reset Request Sequence

Numbers the Stream Reset Timer is running for are acknowledged, stop the Stream Reset Timer.

E3 If no Stream Numbers are listed in the parameter, then all incoming streams MUST be reset to '0' as the next expected stream sequence number. If specific Stream Numbers are listed, then only these specific streams MUST be reset to '0' and all other non-listed stream sequence numbers remain unchanged.

E4 Optionally an Upper Layer Notification SHOULD be sent to inform the local endpoint that the inbound streams have been reset.

E5 Any queued TSN's (queued at step D3) should now be released and processed normally.

E6 A Stream Reset Response Parameter is put into a STREAM RESET chunk indicating successful processing.

E7 The STREAM RESET chunk is sent after the incoming STREAM RESET chunk is processed completely.

4.2.3. Receiver side procedures for the Incoming SSN Reset Request Parameter

TOC

The decision to deny a stream reset request is an administrative decision and may be user configurable even after the association has formed. If for whatever reason the endpoint does NOT wish to reset any streams it MUST send a stream reset response as described in Section 4.1.5 (Sender side procedures for the Stream Reset Response Parameter) with an appropriate Result field.
In the case that the endpoint is willing to perform a stream reset the following steps SHOULD be followed:

F1 An Outgoing Stream Reset Request Parameter MUST be put into an STREAM RESET chunk according to Section 4.1.2 (Sender side procedures for the Outgoing SSN Reset Request Parameter).

F2 The STREAM RESET chunk is sent after the incoming STREAM RESET chunk is processed completely.

TOC

4.2.4. Receiver side procedures for the SSN/TSN Reset Request Parameter

The decision to deny a stream reset request is an administrative decision and may be user configurable even after the association has formed. If for whatever reason the endpoint does NOT wish to reset any streams it MUST send a stream reset response as described in Section 4.1.5 (Sender side procedures for the Stream Reset Response Parameter) with an appropriate Result field.
In the case that the endpoint is willing to perform a SSN/TSN reset the following steps SHOULD be followed:

> **G1** Compute an appropriate value for the Receiver's next TSN, the TSN the peer should use to send the next DATA chunk. Note that an appropriate value should be larger than the highest TSN last received plus a delta of at least 500 additional TSN's.
>
> **G2** Compute an appropriate value for the local endpoints next TSN, i.e. the receiver of the SSN/TSN reset chunks next TSN to be assigned. Note that an appropriate value should be larger than the endpoints current next TSN to send by at least one TSN.
>
> **G3** Do the same processing as if a SACK chunk with no gap report and a cumulative TSN ACK of Sender's next TSN - 1 was received.
>
> **G4** Do the same processing as if an FWD-TSN chunk with all streams affected and a new cumulative TSN ACK of Receiver's next TSN - 1 was received.
>
> **G5** All incoming and outgoing streams MUST be reset to '0' as the next expected and outgoing stream sequence numbers, respectively.
>
> **G6** A Stream Reset Response Parameter is put into a STREAM RESET chunk indicating successful processing.
>
> **G7** The STREAM RESET chunk is sent after the incoming STREAM RESET chunk is processed completely.

4.2.5. Receiver side procedures for addition of streams

TOC

When an SCTP endpoint receives a stream reset request adding additional streams, it MUST send a response parameter either acknowledging or rejecting the request. If the response is successful the receiver MUST add the requested number of inbound streams to the association, initializing the next expected stream sequence number to be 0.

4.2.6. Receiver side procedures for the Stream Reset Response Parameter

TOC

On receipt of a Stream Reset Response Parameter the following MUST be performed:

H1 If the Stream Reset Timer is running for the Stream Reset Request Sequence Number indicated in the Stream Reset Response Sequence Number field, mark the Stream Reset Request Sequence Number as acknowledged. If all Stream Reset Request Sequence Numbers the Stream Reset Timer is running for are acknowledged, stop the Stream Reset Timer. If the timer was not running for the Stream Reset Request Sequence Number, the processing of the Stream Reset Response Parameter is complete.

H2 If the Result field does not indicate successful processing an Upper Layer Notification SHOULD be sent to inform the local endpoint of the failure to reset its outbound streams. Afterwards processing of this response is complete.

H3 If the request was an Outgoing Stream Reset Request the affected streams should now be reset and all queued data should be processed now and assigning of stream sequence numbers is allowed again. Optionally an Upper Layer Notification SHOULD be sent to inform the local endpoint that the outbound streams have been reset.

H4 If the request was a SSN/TSN Reset Request new DATA should be sent from Receiver's next TSN and beginning with stream sequence number '0' for all outgoing streams. All incoming streams are also reset to '0' as the next expected stream sequence number. The peer will send DATA chunks starting with Sender's next TSN.

4.3. Various Examples of the Stream Reset procedures

TOC

The following example illustrates an Endpoint A resetting all streams in both directions.

```
     E-A                                                    E-Z
     ----------[STR_RESET(IN-REQ:X|OUT-REQ:X+1,Y-3)]------->
     <-[STR_RESET(RESP:Y|OUT-REQ:Y+1,X+1))]---------
     -------[STR_RESET(RESP:Y)]----------------->
```

The following example illustrates an Endpoint A resetting stream 1 and 2 for just its outgoing streams.

```
     E-A                                                    E-Z
     -------[STR_RESET(OUT-REQ:X/1,2]------------------>
     <---[STR_RESET(RESP:X/1,2)]------------
```

The following example illustrates an Endpoint A resetting stream 1 and 2 for just its incoming streams.

```
     E-A                                            E-Z
     ------[STR_RESET(IN-REQ:X/1,2]----------->
     <---[STR_RESET(RESP:X/1,2]-------
```

The following example illustrates an Endpoint A requesting the streams and TSN's be reset. At the completion E-A has the new sending TSN (selected by the peer) of B and E-Z has the new sending TSN of A (also selected by the peer).

```
     E-A                                            E-Z
     ------[STR_RESET(TSN-REQ:X]----------->
     <---[STR_RESET(RESP:X/S-TSN=A, R-TSN=B]-------
```

---

5. Security Considerations

TOC

Having the ability to reset a stream should not pose any additional security risk to SCTP. An attacker that can successfully inject a stream reset would also be able to inject data or other malicious information into an association such as an ABORT.

---

6. Iana Considerations

TOC

This document defines one new chunk type and four new parameter types. This document recommends the values of 0x82 for the chunk type and 0x000d, 0x000e, 0x000f, 0x0010 and 0x0011 for the new parameter types. However IANA may assign any free chunk or parameter type as long it is from the same chunk or parameter pool. In the case of the chunk, it MUST be from the pool of chunks with the upper two bits set to '10'. In the case of the parameters, it MUST be from the pool whose upper bits are set to '00'.

---

7. Acknowledgments

TOC

The authors wish to thank Paul Aitken and Irene Ruengeler for there invaluable comments.

---

8. Normative References

TOC

[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC4960]	Stewart, R., “Stream Control Transmission Protocol,” RFC 4960, September 2007 (TXT).
[RFC5061]	Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, “Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration,” RFC 5061, September 2007 (TXT).

Authors' Addresses

TOC

	Randall R. Stewart
	Researcher
	Chapin, SC 29036
	USA
Phone:	
Email:	randall@lakerest.net
	Peter Lei
	Cisco Systems, Inc.
	8735 West Higgins Road
	Suite 300

	Chicago, IL 60631
	USA
Phone:	
Email:	peterlei@cisco.com
	Michael Tuexen
	Muenster Univ. of Applied Sciences
	Stegerwaldstr. 39
	48565 Steinfurt
	Germany
Email:	tuexen@fh-muenster.de