SCTP Unreliable Data Mode Extension
<draft-xie-usctp-sigtran-01.txt>

Status of This Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026 [RFC2026]. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This document describes an extension to the Stream Control Transmission Protocol (SCTP) [RFC2960] to provide unreliable data transfer services. The benefits of this extension includes unified congestion control over reliable and unreliable data streams, single association for multi-content data services, link level fault tolerance for unreliable data transfer, unreliable data stream multiplexing, etc.

TABLE OF CONTENTS
1. Introduction.................................................. 2
2. Conventions.................................................. 3
3. Unreliable Data Extension Design............................ 3
3.1 New INIT and INIT-ACK parameters.......................... 3
3.1.1 Unreliable Streams Parameter Definition................ 3
3.1.2 Partial Checksum Parameter Definition................... 5
3.2 New chunk definitions................................... 5
3.2.1 Forward Cumulative TSN Chunk Definition (FORWARD TSN).... 5
3.2.2 Partial Checksum Data Chunk (P-DATA).................... 6
4. Unreliable Stream Operations................................ 8
4.1 Initialization of Unreliable Streams...................... 8
4.2 Send Unreliable Data.................................... 9
4.3 Receive Unreliable Data................................ 10
4.4 Usage of the P-DATA chunk............................... 10

Xie, et al [Page 1]
1. Introduction

Taking advantage of the extensibility of SCTP, this document adds unreliable data transfer services to SCTP and an optional method to send SCTP Data Chunks with limited checksum coverage. The design presented here allows the co-existence of unreliable data streams and reliable streams in a single SCTP association.

The following are some advantages for integrating unreliable data services into SCTP:

1) Unreliable extension to SCTP (U-SCTP) supports congestion control and congestion avoidance over unreliable data traffic; this is very desirable since it is much friendlier towards the network than UDP.

2) Some applications services can greatly benefit from U-SCTP by using a single SCTP association to carry both reliable content (e.g., text, billing, accounting, set-up information, etc.) and unreliable content (e.g., Fiber channel, SCSI over IP, etc.).

3) U-SCTP allows the use of a unified congestion control across both reliable and unreliable traffic between two endpoints. This has the potential for better utilization of network resources, achieving similar objectives of the Endpoint Congestion Management (ecm) Working Group.

4) Taking advantage of SCTP data chunk bundling function, sending multiple unreliable data streams across a single SCTP association creates a very efficient and effective way of data multiplexing.

5) U-SCTP gives even the unreliable data traffic "link-level" fault tolerance, taking advantage of SCTP multi-homing ability. This is not possible with UDP.

6) U-SCTP can achieve either ordered or unordered unreliable data transfer, while UDP is incapable of controlling the order of data delivery.

7) An application can control its retransmission policies if retransmission is deemed needed.

8) Some applications may find it desirable to limit the coverage of the Adler32 checksum over the actual data chunks.
2. Conventions

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, NOT RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in RFC 2119 [RFC2119].

3. Unreliable Data Extension Design

With the present extension, an SCTP data sender will be allowed to designate a sub-set of its outbound streams to be unreliable streams. The user data chunks sent to an unreliable stream will share the same TSN space, the same congestion control/avoidance treatment, and the same transmission priority as those sent to a reliable stream, but they will not be retransmitted if they are found missing at the data receiver.

3.1 New INIT and INIT-ACK parameters

The following new optional parameter, are being added to the INIT and INIT ACK chunks. At the initialization of the association, the sender of the INIT or INIT ACK chunk may include these parameters to indicate its ability to support these features.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Status</th>
<th>Type Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unreliable Streams</td>
<td>Optional</td>
<td>0xC000</td>
</tr>
<tr>
<td>Partial Checksum support</td>
<td>Optional</td>
<td>0xC004</td>
</tr>
</tbody>
</table>

3.1.1 Unreliable Streams Parameter Definition

The Unreliable Streams parameter is added to the INIT and INIT ACK chunks. At the initialization of the association, the sender of the INIT or INIT ACK chunk shall include this optional parameter to inform its peer that it is able to support unreliable streams and to designate its unreliable outbound streams. If no streams are marked as unreliable but the sender does support the unreliable streams option the sender SHOULD include a parameter with no u-stream ranges and a fixed Parameter Length of 4.
Type: 16 bit u_int
0xC000, indicating Unreliable Streams parameter

Length: 16 bit u_int

Indicate the size of the parameter in octets, including the Type, Length, u-stream start, and u-stream end fields.

u-stream start: 16 bit u_int, and
u-stream end: 16 bit u_int

Each pair of u-stream start and u-stream end fields defines one or more unreliable outbound streams, starting from stream number US and ending with stream number UE. The union of all the pairs together defines the complete sub-set of all unreliable outbound streams.

The following are some examples of unreliable stream designation (assuming OS = 10):

Example 1: (assuming OS = 10)

```
+-------------+-----------+---------------------
| type=0xC000 | length=8  |
+-------------+-----------+---------------------
| u-start= 3  | u-end= 5  | 0 - 2    reliable
|             |           | 3 - 5    unreliable
|             |           | 6 - 9    reliable
```

Example 2: (assuming OS = 10)

```
+-------------+-----------+---------------------
| type=0xC000 | length=12 |
+-------------+-----------+---------------------
| u-start= 3  | u-end= 5  | 0 - 2    reliable
|             |           | 3 - 9    unreliable
| u-start= 6  | u-end= 9  |
+-------------+-----------+---------------------
```

Example 3: (assuming OS = 10)

```
+-------------+-----------+---------------------
| type=0xC000 | length=12 |
+-------------+-----------+---------------------
| u-start= 9  | u-end= 9  | 0       unreliable
|             |           | 1 - 8    reliable
| u-start= 0  | u-end= 0  | 9       unreliable
+-------------+-----------+---------------------
```

Example 4: (assuming OS = 10)

```
+-------------+-----------+---------------------
| type=0xC000 | length=8  |
+-------------+-----------+---------------------
```
Example 5: (assuming OS = 10)

|type=0xC000 | length=4  |

0 - 9      reliable

3.1.2 Partial Checksum Parameter Definition

The Partial Checksum Parameter is added to the INIT and INIT ACK chunks. At the initialization of the association, the sender of the INIT or INIT ACK chunk shall include this optional parameter to inform its peer that it is able to support the new Data Chunk type 193 (see section 3.2.2).

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Parameter Type = 0xc004    |     Parameter Length=4        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

3.2 New chunk definitions

The following new control chunks, are being added to support two new features in SCTP. The FORWARD TSN supports the unreliable stream. The Partial Checksum Data Chunk will support Data chunks that are not completely covered by the Adler32 checksum.

| 0 1 1 0 0 0 0 0 0 | Chunk Flags |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      New Cumulative TSN                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Chunk Flags:

Set to all zeros on transmit and ignored on receipt.

New Cumulative TSN: 32 bit u_int
This indicates the new cumulative received TSN to the data receiver. Upon the reception of this value, the data receiver shall consider
any missing TSNs earlier than this value received and stop reporting
them as gaps in the subsequent SACKs.

3.2.2 Partial Checksum Data Chunk (P-DATA)

The following format MUST be used for the P-DATA chunk:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 0 0 0 0 1 1| Reserved|U|B|E|    Length                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              TSN                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Stream Identifier S      |   Stream Sequence Number n    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Checksum Coverage       |  Payload Protocol Identifier  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                     \                     |
/                 User Data (seq n of Stream S)                 /
\                     \                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Reserved: 5 bits
Should be set to all ‘0’s and ignored by the receiver.

U bit: 1 bit
The (U)nordered bit, if set to ‘1’, indicates that this is an
unordered P-DATA chunk, and there is no Stream Sequence Number assigned
to this P-DATA chunk. Therefore, the receiver MUST ignore the Stream
Sequence Number field.

After re-assembly (if necessary), unordered P-DATA chunks MUST be
dispatched to the upper layer by the receiver without any attempt to
re-order.

If an unordered user message is fragmented, each fragment of the
message MUST have its U bit set to ‘1’.

B bit: 1 bit
The (B)eginning fragment bit, if set, indicates the first fragment of
a user message.

E bit: 1 bit
The (E)nding fragment bit, if set, indicates the last fragment of a
user message.

An unfragmented user message shall have both the B and E bits set
to ‘1’. Setting both B and E bits to ‘0’ indicates a middle fragment of
a multi-fragment user message, as summarized in the following table:
<table>
<thead>
<tr>
<th>B E</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td>First piece of a fragmented user message</td>
</tr>
<tr>
<td>0 0</td>
<td>Middle piece of a fragmented user message</td>
</tr>
<tr>
<td>0 1</td>
<td>Last piece of a fragmented user message</td>
</tr>
<tr>
<td>1 1</td>
<td>Unfragmented Message</td>
</tr>
</tbody>
</table>

Table 1: Fragment Description Flags

When a user message is fragmented into multiple chunks, the TSNs are used by the receiver to reassemble the message. This means that the TSNs for each fragment of a fragmented user message MUST be strictly sequential.

Length: 16 bits (unsigned integer)

This field indicates the length of the P-DATA chunk in bytes from the beginning of the type field to the end of the user data field excluding any padding. A P-DATA chunk with no user data field will have Length set to 16 (indicating 16 bytes).

TSN : 32 bits (unsigned integer)

This value represents the TSN for this P-DATA chunk. The valid range of TSN is from 0 to 4294967295 (2**32 - 1). TSN wraps back to 0 after reaching 4294967295.

Stream Identifier S: 16 bits (unsigned integer)

Identifies the stream to which the following user data belongs.

Stream Sequence Number n: 16 bits (unsigned integer)

This value represents the stream sequence number of the following user data within the stream S. Valid range is 0 to 65535.

When a user message is fragmented by SCTP for transport, the same stream sequence number MUST be carried in each of the fragments of the message.

Checksum Coverage: 16 bits (unsigned integer)

This field contains a integer that is used to determine how much of this P-DATA chunk is covered by the Adler32 checksum calculation. The value in the Checksum Coverage MUST NOT be smaller than the size of the P-DATA header, i.e. 16 octets and MUST NOT be larger than the value contained in the Length field. The data covered by the checksum is defined as starting from the first transmitted byte in the
P-DATA Chunk (i.e. the Chunk Type Byte) for exactly Checksum Coverage bytes in length. For example if a value of "18" appears in the Checksum Coverage field, the Checksum covers the header and 2 bytes of the data payload. The minimum value of the Checksum Coverage field is 16 (i.e. covering the Chunk header).

Payload Protocol Identifier: 16 bits (unsigned integer)

This value represents an application (or upper layer) specified protocol identifier. This value is passed to SCTP by its upper layer and sent to its peer. This identifier is not used by SCTP but can be used by certain network entities as well as the peer application to identify the type of information being carried in this P-DATA chunk. This field must be sent even in fragmented P-DATA chunks (to make sure it is available for agents in the middle of the network).

The value 0 indicates no application identifier is specified by the upper layer for this payload data.

User Data: variable length

This is the payload user data. The implementation MUST pad the end of the data to a 4 byte boundary with all-zero bytes. Any padding MUST NOT be included in the length field. A sender MUST never add more than 3 bytes of padding.

4. Unreliable Stream Operations

In this section, we first defines the procedures for opening unreliable streams in an SCTP association. Then, we will discuss procedures for sending and receiving unreliable SCTP data chunks.

4.1 Initialization of Unreliable Streams

If the SCTP data sender plans to send unreliable data, at the initialization of the association it must include the Unreliable Streams parameter in its INIT or INIT ACK chunk to indicate to its peer which of its outbound streams are going to be used as unreliable streams.

Upon the reception of the Unreliable Streams parameter, the data receiver shall determine and record the mode (reliable or unreliable) of each inbound stream, as it allocates resource for its inbound streams.

Note, if the endpoint does not support unreliable inbound streams, it SHOULD treat the Unreliable Streams parameter as an invalid or unrecognized parameter, following the rules defined in Section 5.1 of [RFC2960]. Upon reception of the operational error indicating that the peer does not support unreliable streams, the initiator of the unreliable streams may choose either (1) to give up the initiation process by discarding the INIT-ACK or (2) to continue the initiation procedures and reporting to its upper layer that no unreliable streams
are available.
Initiation of streams as reliable and/or unreliable may be under the control of the SCTP user. Hence, the ULP primitive "ASSOCIATE" (see Section 10.1 of [RFC2960]) should contain the optional U-stream-start and U-stream-end values.

4.2 Send Unreliable Data

During the lifetime of the association, any user data sent to an unreliable stream is considered as unreliable user data and will automatically be transmitted in unreliable mode.

The SCTP data sender shall handle user data sent to an unreliable stream the same way as it handles user data sent to a reliable stream (i.e., the same timer rules, congestion control rules, failure detection rules, RTO control rules, etc.), with the following exceptions:

A1) before retransmitting a DATA chunk (due to either a T3-rxt timer expiration as defined in 6.3.3 of [RFC2960] or a 4th missing indication as defined in 7.2.4 of [RFC2960]), the SCTP data sender MUST check whether the DATA chunk is being transmitted on an unreliable stream. If so it does the following:

B1) Check the unreliable retransmission threshold value for the DATA chunk. This value may be set by the SCTP user to 0 (for complete unreliability) or 1 (for an attempt of 1 retransmission). The ULP primitive "DATA" (see Section 10.1 of RFC2960) should contain an optional unreliable retransmission parameter to achieve this ULP control.

B2) If the value is 0, the sender MUST NOT retransmit the data chunk. Instead, the sender MUST mark the data chunk as being finally acked and advance its cumulative TSN accordingly.

B3) If the value is 1, and the sender has made one previous retransmission, the sender MUST NOT retransmit the data chunk. Instead, the sender MUST mark the data chunk as being finally acked and advance its cumulative TSN accordingly.

B4) If the value is 1, and the sender has never attempted to retransmit this chunk, the sender MUST retransmit the Chunk and mark it as being retransmitted. The rules for retransmission as defined in [RFC2960] should be used for destination selection and error reporting.

A2) whenever the data sender receives a SACK from the data receiver that carries a cumulative TSN which is earlier than the sender’s own cumulative TSN (indicating that the receiver is still waiting for some missing unreliable data chunks to advance its cumulative TSN), the sender MUST send the data receiver a FORWARD TSN chunk containing its own latest cumulative TSN.
A sender MUST NOT use the forward TSN for any other purposes other than the above scenario.

4.3 Receive Unreliable Data

Regardless whether a DATA chunk arrives for a reliable stream or an unreliable stream, the receiver MUST perform the same TSN handling (e.g., duplicate detection, gap detection, SACK generation, cumulative TSN advancement, etc.) as defined in [RFC2960].

However, whenever a FORWARD TSN chunk arrives the data receiver MUST update its cumulative TSN to the value carried in the FORWARD TSN chunk, and MUST stop reporting any un-received TSN before the new cumulative TSN as missing.

When receiving a FORWARD TSN chunk, any chunks in the reassembly queue should be discarded if they hold a value smaller than or equal to the TSN value found in the FORWARD TSN chunk.

When unreliable DATA chunks arrive with the 'U' bit set to '0' (i.e. are considered ordered) and are out of order, the receiver must hold the chunks for reordering. However it is possible that the DATA chunk being waited upon is one that will not be retransmitted. Therefore when a FORWARD TSN chunk arrives, the receiver MUST examine all of its unreliable stream queues and examine if the forward TSN now brings the cumulative ACK point past any stream DATA held for re-ordering. If the cumulative TSN point does pass any held DATA chunks, then those chunks should be made available to the upper layer, since the earlier stream sequence numbers will not be retransmitted.

4.4 Usage of the P-DATA chunk.

For some applications it is beneficial to NOT discard a SCTP packet due to an error within the Data portion. For these types of applications this new optional chunk type is being added.

All rules defined in [RFC2960] for DATA Chunks MUST be followed for the P-DATA chunk with the following exceptions:

E1) The Payload Protocol Identifier (PPI) field is limited to 16 bits. If the ULP presents a PPI that is larger than 16 bits for transmission, the upper 16 bits MUST be discarded.

E2) The Checksum Coverage field defines how much of this P-DATA chunk the Adler-32 checksum is to cover. During Checksum computation the sender and receiver MUST use this field to determine how much of the PDATA chunk to add to the Checksum of the SCTP packet. After summing the specified amount of data to the checksum, the checksum routine MUST skip to the next chunk (if this is a bundled chunk) and NOT include the rest of the data in the P-DATA chunk in its checksum computation.

E3) The Checksum Coverage field MUST be at least 16 bytes and MUST NOT be larger than the size of the Chunk length field. If the
Checksum Coverage value is invalidated by this rule, the P-DATA chunk MUST be silently discarded.

E4) A sender MUST NOT send a PDATA chunk unless the ‘Partial Checksum support’ parameter was seen in the INIT or INIT-ACK.

It is important to note that this Chunk uses the SAME TSN values as the normal DATA Chunk type. The sender and receiver do NOT hold a
separate TSN sequence spaces. The two chunks types use the same TSN sequence space. In effect the PDATA chunk is treated in all considerations to be a Data Chunk, with all of the normal Data Chunk rules for Congestion Control effecting this chunk. The only difference in treatment of this chunk comes during the calculation of the Adler32 checksum.

This Chunk MAY be used with either a reliable or un-reliable stream, no restrictions are placed on its usage except those listed above. Use of the P-DATA chunk may be under the control of the SCTP user. Hence, the ULP primitive "DATA" (see section 10.1 of RFC2960) should contain an optional Checksum Coverage value.

5. Other Issues

5.1 Unreliable Data Stream Multiplexing

Sometimes, it is desirable to aggregate different real time media streams (e.g., RTP streams) and send them over a single communication connection. And normally, unreliable transport is preferred for these types of media streams.

With U-SCTP this is easily achieved by assigning each different media stream to a different unreliable SCTP stream and enabling the SCTP data bundling to perform the multiplexing.

The implementation of the data sender MAY use a bundling timer with a time interval adjusted to the timing characteristics of the specific media type in order to achieve the optimal multiplexing efficiency.

5.2 Fault Tolerant Unreliable Data Transfer

When the data receiver is multi-homed, unreliable data transfer using U-SCTP will obtain the same fault tolerance benefit as that of the reliable data services across an SCTP association.

This is because the data sender still follows the same failure detection rules and still counts the omitted retransmission against the association and the destination transport address to which the unreliable DATA chunk was originally sent. Thus, when failure occurs, the data sender will detect the failure and shift the unreliable data services to an alternate destination address, following the same procedures as defined in Section 8 of [RFC2960] for reliable data transfer.

5.3 Unreliable Data Out-of-order Detection

Detecting out-of-order data in an unreliable stream is useful for some applications (e.g. Fiber channel or SCSI over IP). With U-SCTP this becomes possible - the upper layer simply needs to examine the the stream sequence number of the delivered data chunks to detect any missing data or out-of-order data. This detection only works when the DATA chunks are sent in order, i.e. their "U" bit MUST NOT be
6. Acknowledgements

The authors would like to thank Scott Bradner for his comments.

7. Authors' Addresses

Qiaobing Xie                            Tel: +1-847-632-3028
Motorola, Inc.                          EMail: qxie1@email.mot.com
1501 W. Shure Drive, #2309
Arlington Heights, IL 60004
USA

Randall R. Stewart                     Tel: +1-815-477-2127
Cisco Systems, Inc.                     EMail: rrs@cisco.com
8725 West Higgins Road
Suite 300                               Chicago, Ill 60631

Chip Sharp                             Tel: +1-919-392-3121
Cisco Systems Inc.                      EMail: chsharp@cisco.com
7025 Kit Creek Road
Research Triangle Park, NC  27709
USA

Ian Rytina                             Tel: +61-3-9301-6164
Ericsson Australia                     EMail: ian.rytina@ericsson.com
37/360 Elizabeth Street
Melbourne, Victoria 3000
Australia

8. References


This Internet Draft expires in 6 months from November, 2000.