Abstract

This document describes extensions to the Stream Control Transmission Protocol (SCTP) [RFC2960] and the "Dynamic Address Reconfiguration" [draft-ietf-tsvwg-addip-sctp]. In this proposal, we propose changes to the way ASCONF’s are handled and bundled to allow multiple ASCONF’s to be sent during times of disconnected connectivity. This then allows a method which allows the retransmit of multiple ASCONF's...
chunk within a single packet to better support reliable handover.

Table of Contents

1. Introduction ................................................. 3
2. Conventions .................................................. 4
3. The Cumulative ASCONF chunks ............................... 5
   3.1. ASCONF-ACK for Cumulative ASCONF chunk ............... 6
4. Procedures ..................................................... 7
   4.1. Procedure of the ASCONF sender in detail .............. 8
   4.2. Procedure of the ASCONF receiver in detail ........... 8
   4.3. Procedure of the ASCONF-ACK receiver in detail ...... 10
5. Acknowledgments ............................................... 10
6. References .................................................... 10
Authors’ Addresses ............................................... 12
Intellectual Property and Copyright Statements ............... 13
1. Introduction

SCTP is a protocol which supports assigning multiple addresses for both ends of an association. To support the dynamic reconfiguration of the addresses for each end, an internet-draft "Dynamic Address Reconfiguration (ADD-IP)" [draft-ietf-tsvwg-addip-sctp] is proposed. By adopting this proposal, each end can reconfigure its addresses one by one by sending an "Address Configuration (ASCONF) chunk" and receiving an ASCONF-ACK chunk for the receipt of an ASCONF chunk.

The current ADD-IP draft allows an endpoint of an association to have only one outstanding ASCONF chunk at any time. Before a subsequent ASCONF can be sent, the sender of ASCONF must wait to receive an ASCONF-ACK.

Because of this limitation, there are some situations that an association may enter in which it will end up causing an association failure event or it will take much longer time than expected, even though there are available pairs of addresses. The following are typical cases where such an event may occur:

- In the case that all addresses lose reachability and transition through multiple separate address reconfiguration events requiring the sending of multiple ASCONF’s over a short period of time.

This situation may happen when a wireless access, with both IPv4 and IPv6 addresses, first loses one of its addresses, followed by a subsequent address loss, followed by an acquisition of yet a new address. When the interface which these addresses are attached goes down, and a lease time of Router Advertisement (RA) or DHCP is expired, these addresses become invalid one by one. As the first IP address is not the last remaining IP address, the endpoint first tries to send an ASCONF to delete the address. But it fails because no valid source IP address exists and it enters to retransmission phase.

The host cannot send the second ASCONF chunk that may add the new addresses because it has not yet received the ASCONF_ACK for the first ASCONF. It takes a long time before the retransmission of ASCONF occurs. Even when it is retransmitted, the ASCONF chunk is ignored by the peer if the new address is used as the source IP address of the ASCONF chunk, because the new address may not be used as source IP address of a chunk except the case that the chunk is to add the new address. Other old addresses might be unreachable, and they should not be used as the source IP address.

As the result, the association will fail and enter into the CLOSED state.
- In the case that the newly acquired address is not reachable to the peer when there is no valid address in the association. This transition may happen for just a short segment of time and after a few moments before connectivity is restored.

This situation happens when the last remaining valid IP address turns unavailable as a host loses its wireless access and get another new address but its network is disconnected or under NAT, or unable to connect to the peer because of some filters, etc.

The ADD-IP draft targets to support this type of transition, and Address Parameter which indicates one of the IP addresses of the association is introduced to look for the association which the ASCONF chunk belongs to. The source IP address of usual SCTP packet must be an address of the association. But on getting a new address, the ADD-IP draft allows the end-host to send an ASCONF chunk from the new address which is not in the association only when one of the IP address of the association is listed in Address Parameter, and the source IP address of the packet is the address which the ASCONF chunk is going to add.

When such transition occurs, but the new address is not able to send an ASCONF chunk to the peer because its network is disconnected or under NAT, or unable to connect to the peer because of some filters, etc, the ASCONF chunk is stored in queue for retransmission.

However, if the new address is the only address that the end-host has, the retransmission will fail forever and the association will enter into the CLOSED state.

Even though the end-host gets another new address later, the end-host cannot use this address for the source address of the ASCONF stored in queue because the source IP address must be same to the address going to add.

To postpone sending ASCONF and pre-check whether the new address has reachability to the peer may partially solve the second problem. But this strategy will increase handover delay, and of course, there is no reliable way of detecting the reachability judging from an IP address by itself.

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
3. The Cumulative ASCONF chunks

An Ordinary ASCONF Chunk packet format is as follows:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type = 0xC1   |  Chunk Flags  |      Chunk Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Serial Number                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Address Parameter                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   ASCONF Parameter #1                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   ASCONF Parameter #N                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

There is a rule that ASCONF chunk MUST NOT be sent out while there is an unacknowledged ASCONF chunk which is previously outstanding.

This document changes those procedures with the following new requirements:

- The end-host MAY send an ASCONF at any time when new ASCONF is created and appended to the unacknowledged ASCONF chunks. The retransmission timer for the previous ASCONF is stopped, and another timer is started.
- These additional ASCONF chunks MUST be bundled and in strict numerically ascending order.
- Each chunk added to the packet to be retransmitted MUST assume that the processing of the other chunks is successfully done for the purposes of specifying the Address Parameter (used for lookup of the association).

The following example illustrates a Cumulative ASCONF packet which consists of an ASCONF(Serial: N) with 3 unacknowledged ASCONF chunks (Serial: N-3 to N-1):
These 4 ASCONF chunks MUST be packed together in a message and the serial number of these packets MUST be in ascending order as illustrated above.

The format of each ASCONF chunk has not changed from the ADD-IP draft. The ASCONF chunks (Serial N-3 to N-1) MUST not be modified and must remain the same as when they were first transmitted.

When creating the ASCONF chunk for Serial N, the chunk has the same format which ADD-IP draft describes. The only difference is that if the Address Parameter is not set to 0, it MUST be one of the IP addresses of the association assuming that all of the former ASCONFs are applied.

3.1. ASCONF-ACK for Cumulative ASCONF chunk

The receiver of Cumulative ASCONF chunk must return an ASCONF-ACK for each of the ASCONF chunks in the order received from the Cumulative ASCONF packet.

The packet containing the ASCONF-ACK chunks MUST be returned to the source IP address of the sender of Cumulative ASCONF chunk.

Each ASCONF-ACK chunks MUST be listed and bundled together in a single reply packet by ascending order of the serial number from the beginning of the packet.
The following is an example of an ordinary ASCONF-ACK 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 2 3 4 5 6 7 8 9 0 1
+---------------------------------------------------------------+
| Type = 0x80   |  Chunk Flags  |      Chunk Length             |
+---------------------------------------------------------------+
|                       Serial Number                          |
+---------------------------------------------------------------+
|                       ASCONF Parameter Response#1             |
|                       +---------------------------------+           |
|                       | \                                   |           |
|                       |   ....                              |           |
|                       | \                                   |           |
+---------------------------------------------------------------+
|                       ASCONF Parameter Response#N              |
+---------------------------------------------------------------+
```

The Cumulative ASCONF-ACK has a format as follows

```
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 2 3 4 5 6 7 8 9 0 1
+---------------------------------------------------------------+
\                       ASCONF-ACK chunk (Serial: N-3)                |
/   ....                          |
\                       ASCONF-ACK chunk (Serial: N-2)                |
/   ....                          |
\                       ASCONF-ACK chunk (Serial: N-1)                |
/   ....                          |
\                       ASCONF-ACK chunk (Serial: N)                 |
/   ....                          |
+---------------------------------------------------------------+
```

4. Procedures
4.1. Procedure of the ASCONF sender in detail

In the following discussion, a ASCONF chunk with serial(N) is described as ASCONF(N), and ASCONF-ACK(N) is in a same manner.

1. On sending an ASCONF chunk, the sender MUST look into ASCONF output queue to determine if there are any unacknowledged ASCONF chunks. If no ASCONF chunks are present then, simply follow the procedure of current ADD-IP draft.

   However, if one or more ASCONF chunks are found outstanding, go to step 2 to send all ASCONFs bundled together.

2. When creating a Cumulative ASCONF chunk, all unacknowledged ASCONF chunks MUST be bundled together in a single packet.

   Those unacknowledged ASCONF chunks MUST NOT be modified or changed they MUST remain with the same information from when it was created.

   ASCONF chunks in a Cumulative ASCONF chunk must be bundled in a single packet.

3. These chunks MUST be listed in ascending order of their Serial Number.

4. The size of Cumulative ASCONF chunk SHOULD NOT be larger than MTU size. If it exceeds the size of MTU, the end-host SHOULD wait for ASCONF-ACK.

5. The Address Parameter must be set to 0 or one of the IP addresses of the association assuming that all of the ASCONFs previously sent were applied successfully.

6. The sender MUST NOT modify the ASCONF packet once it is sent.

7. On sending a Cumulative ASCONF chunk, the retransmission timer of ASCONF chunk should be restarted.

4.2. Procedure of the ASCONF receiver in detail

In general, the receiver of ASCONF chunk behave in the same manner as specified in the ADD-IP draft.

Upon receiving an ASCONF chunk, the endpoint performs the following procedure to find the association state information:
L1) Use the source and destination addresses and port numbers to attempt to identify the association (i.e. use the same method defined in RFC2960 [RFC2960] used for all other SCTP chunks). If found proceed to rule L4.

L2) If the association is not found, use the address found in the Address Parameter TLV combined with the port number found in the SCTP common header. If found proceed to rule L4.

L2-ext) If more than one ASCONF chunks are packed together, use the address found in the Address Parameter TLV of the each ASCONF chunk. If found, proceed to rule L4.

L3) If neither L1 or L2 locates the association, treat the chunk as an Out Of The Blue chunk as defined in RFC2960 [RFC2960].

L4) Follow the normal rules to validate the SCTP verification tag found in RFC2960 [RFC2960].

After identification and verification of the association, the following should be performed to properly process the ASCONF Chunk:

M1) If the value found in the serial number of the first ASCONF chunk is equal to the (‘Peer-Serial-Number’ + 1), the endpoint clears any old cached ASCONF-ACK responses and then proceeds with the same procedures described in the ADD-IP draft. After processing the receiver should cache the ASCONF-ACK response in case the peer retransmits the ASCONF (i.e. the response is lost).

M2) If the value found in the serial number is less than the (‘Peer-Serial-Number’ + 1), simply skip to the next ASCONF, and include in the outbound response packet the previously cached ASCONF-ACK response that was sent and saved that matches the serial number of the ASCONF.

M3) Then, process each ASCONF one by one. If the Serial Number of the ASCONF is less than the (‘Peer-Serial-Number’ + 1), then simply skip it, as above, adding to the response packet the previously saved ASCONF-ACK response.

M4) When the serial number matches the next one expected, process the ASCONF as described in the ADD-IP draft and then, after processing the ASCONF chunk, append an ASCONF-ACK to the response packet and cache a copy of it (in the event it later needs to be retransmitted). If an error occurs on processing the ASCONF chunk, the end-host MUST stop processing the following ASCONF chunks and return the response packet immediately.

M5) When all processes are ASCONF chunks are processed for packet containing the Cumulative ASCONF, send back the single response packet with all of the ASCONF-ACK chunks for all the serials in Cumulative ASCONF chunk. The destination address of the ASCONF-ACK chunk MUST be the source address of the Cumulative ASCONF packet.
M6) While processing the Cumulative ASCONF packet, if the response Cumulative ASCONF-ACK packet will exceed the PMTU of the return path, the receiver MUST stop adding additional ASCONF-ACK’s into the response packet but MUST continue to process all of the ASCONF chunks, saving ASCONF-ACK responses in its cached copy. The sender of the ASCONF will later retransmit the ASCONF’s that were not responded to at which time the cached copies of the responses that would NOT fit in the MTU can be sent to the peer. The cached copies of the responses can be sent to the peer when the sender of the ASCONF retransmit the unacknowledged ASCONF chunks.

4.3. Procedure of the ASCONF-ACK receiver in detail

The receiver of the Cumulative ASCONF-ACK packet should process each separate ASCONF-ACK chunk as described in the current ADD-IP specification.

5. Acknowledgments

The authors wish to thank Randall Stewart, Yasuo Okabe and Motonori Nakamura for giving us invaluable comments and encouraging us at any time.

The authors also wish to thank to the members of WIDE Project SCTP working group for their contribution to the demonstration.

6. References


[I-D.ietf-tsvwg-addip-sctp]
Authors’ Addresses

Shin Maruyama
Kyoto University.
Yoshida-Honmachi
Sakyo-ku
Kyoto, Kyoto 606-8501
JAPAN

Phone: +81-75-753-7468
Email: marushin@net.ist.i.kyoto-u.ac.jp

Masahiro Kozuka
Kyoto University
Yoshida-Honmachi
Sakyo-ku
Kyoto, Kyoto 606-8501
JAPAN

Phone: +81-75-753-7468
Email: ma-kun@kozuka.jp