Network Working Group R. R. Stewart

Internet-Draft Netflix, Inc.

Intended status: Standards Track M. Tüxen

Expires: 29 January 2021 I. Rüngeler

Münster Univ. of Appl. Sciences

28 July 2020

Stream Control Transmission Protocol (SCTP) Network Address Translation

Support

draft-ietf-tsvwg-natsupp-20

Abstract

The Stream Control Transmission Protocol (SCTP) provides a reliable

communications channel between two end-hosts in many ways similar to

the Transmission Control Protocol (TCP). With the widespread

deployment of Network Address Translators (NAT), specialized code has

been added to NAT functions for TCP that allows multiple hosts to

reside behind a NAT function and yet share a single IPv4 address,

even when two hosts (behind a NAT function) choose the same port

numbers for their connection. This additional code is sometimes

classified as Network Address and Port Translation (NAPT).

This document describes the protocol extensions required for the SCTP

endpoints and the mechanisms for NAT functions necessary to provide

similar features of NAPT in the single point and multipoint

traversal scenario.

Finally, a YANG module for SCTP NAT is defined.

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 https://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 29 January 2021.

Stewart, et al. Expires 29 January 2021 [Page 1]

Internet-Draft SCTP NAT Support July 2020

Copyright Notice

Copyright (c) 2020 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 (https://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

2. Conventions . . . . . . . . . . . . . . . . . . . . . . . . . 5

3. Terminology . . . . . . . . . . . . . . . . . . . . . . . . . 5

4. Motivation . . . . . . . . . . . . . . . . . . . . . . . . . 6

4.1. SCTP NAT Traversal Scenarios . . . . . . . . . . . . . . 6

4.1.1. Single Point Traversal . . . . . . . . . . . . . . . 7

4.1.2. Multi Point Traversal . . . . . . . . . . . . . . . . 7

4.2. Limitations of Classical NAPT for SCTP . . . . . . . . . 8

4.3. The SCTP-Specific Variant of NAT . . . . . . . . . . . . 8

5. Data Formats . . . . . . . . . . . . . . . . . . . . . . . . 13

5.1. Modified Chunks . . . . . . . . . . . . . . . . . . . . . 13

5.1.1. Extended ABORT Chunk . . . . . . . . . . . . . . . . 13

5.1.2. Extended ERROR Chunk . . . . . . . . . . . . . . . . 13

5.2. New Error Causes . . . . . . . . . . . . . . . . . . . . 14

5.2.1. VTag and Port Number Collision Error Cause . . . . . 14

5.2.2. Missing State Error Cause . . . . . . . . . . . . . . 14

5.2.3. Port Number Collision Error Cause . . . . . . . . . . 15

5.3. New Parameters . . . . . . . . . . . . . . . . . . . . . 16

5.3.1. Disable Restart Parameter . . . . . . . . . . . . . . 16

5.3.2. VTags Parameter . . . . . . . . . . . . . . . . . . . 16

6. Procedures for SCTP Endpoints and NAT Functions . . . . . . . 18

6.1. Association Setup Considerations for Endpoints . . . . . 18

6.2. Handling of Internal Port Number and Verification Tag

Collisions . . . . . . . . . . . . . . . . . . . . . . . 19

6.2.1. NAT Function Considerations . . . . . . . . . . . . . 19

6.2.2. Endpoint Considerations . . . . . . . . . . . . . . . 20

6.3. Handling of Internal Port Number Collisions . . . . . . . 20

6.3.1. NAT Function Considerations . . . . . . . . . . . . . 20

6.3.2. Endpoint Considerations . . . . . . . . . . . . . . . 21

6.4. Handling of Missing State . . . . . . . . . . . . . . . . 21

6.4.1. NAT Function Considerations . . . . . . . . . . . . . 21

6.4.2. Endpoint Considerations . . . . . . . . . . . . . . . 22

Stewart, et al. Expires 29 January 2021 [Page 2]

Internet-Draft SCTP NAT Support July 2020

6.5. Handling of Fragmented SCTP Packets by NAT Functions . . 23

6.6. Multi Point Traversal Considerations for Endpoints . . . 24

7. Various Examples of NAT Traversals . . . . . . . . . . . . . 24

7.1. Single-homed Client to Single-homed Server . . . . . . . 24

7.2. Single-homed Client to Multi-homed Server . . . . . . . . 26

7.3. Multihomed Client and Server . . . . . . . . . . . . . . 28

7.4. NAT Function Loses Its State . . . . . . . . . . . . . . 31

7.5. Peer-to-Peer Communication . . . . . . . . . . . . . . . 33

8. SCTP NAT YANG Module . . . . . . . . . . . . . . . . . . . . 38

8.1. Tree Structure . . . . . . . . . . . . . . . . . . . . . 38

8.2. YANG Module . . . . . . . . . . . . . . . . . . . . . . . 39

9. Socket API Considerations . . . . . . . . . . . . . . . . . . 41

9.1. Get or Set the NAT Friendliness (SCTP\_NAT\_FRIENDLY) . . . 42

10. IANA Considerations . . . . . . . . . . . . . . . . . . . . . 42

10.1. New Chunk Flags for Two Existing Chunk Types . . . . . . 42

10.2. Three New Error Causes . . . . . . . . . . . . . . . . . 44

10.3. Two New Chunk Parameter Types . . . . . . . . . . . . . 45

10.4. One New URI . . . . . . . . . . . . . . . . . . . . . . 45

10.5. One New YANG Module . . . . . . . . . . . . . . . . . . 45

11. Security Considerations . . . . . . . . . . . . . . . . . . . 45

12. Normative References . . . . . . . . . . . . . . . . . . . . 46

13. Informative References . . . . . . . . . . . . . . . . . . . 48

Acknowledgments . . . . . . . . . . . . . . . . . . . . . . . . . 49

Authors' Addresses . . . . . . . . . . . . . . . . . . . . . . . 50

1. Introduction

Stream Control Transmission Protocol (SCTP) [RFC4960] provides a

reliable communications channel between two end-hosts in many ways

similar to TCP [RFC0793]. With the widespread deployment of Network

Address Translators (NAT), specialized code has been added to NAT

functions for TCP that allows multiple hosts to reside behind a NAT

functions using internal addresses (see [RFC6890]) and yet share

single IPv4 address, even when two hosts (behind a NAT function)

choose the same port numbers for their connections. This additional

code is sometimes classified as Network Address and Port Translation

(NAPT). Please note that this document focuses on the case where the

NAT function maps a single or multiple internal addresses to a single

external address and vice versa. To date, specialized code for SCTP

has not yet been added to most NAT functions so that only a

translation of IP addresses is supported. The end result of this is

that only one SCTP-capable host can successfully operate behind such

a NAT function and this host can only be single-homed. The only

alternative for supporting legacy NAT functions is to use UDP

encapsulation as specified in [RFC6951].

The NAT function in the document refers to NAPT functions described

in Section 2.2 of [RFC3022], NAT64 [RFC6146], or DS-Lite AFTR [RFC6333].

Stewart, et al. Expires 29 January 2021 [Page 3]

Internet-Draft SCTP NAT Support July 2020

This document specifies procedures allowing a NAT function to support

SCTP by providing similar features to those provided by a NAPT for

TCP, UDP, and ICMP. The document also specifies a set

of data formats for SCTP packets and a set of SCTP endpoint

procedures to support NAT traversal. An SCTP implementation

supporting these procedures can assure that in both single-homed and

multi-homed cases a NAT function will maintain the appropriate state

without the NAT function needing to change port numbers.

It is possible and desirable to make these changes for a number of

reasons:

\* It is desirable for SCTP internal end-hosts on multiple platforms

to be able to share a NAT function's external IP address in the

same way that a TCP session can use a NAT function.

\* If a NAT function does not need to change any data within an SCTP

Packet, it will reduce the processing burden of NAT'ing SCTP by not

needing to execute the CRC32c checksum required by SCTP.

\* Not having to touch the IP payload makes the processing of ICMP

messages by NAT functions easier.

An SCTP-aware NAT function will need to follow these procedures for

generating appropriate SCTP packet formats.

When considering this feature it is possible to have multiple levels

of support. At each level, the Internal Host, Remote Host, and NAT

function may or may not support the features described in this

document. The following table illustrates the results of the various

combinations of support and if communications can occur between two

endpoints.

Stewart, et al. Expires 29 January 2021 [Page 4]

Internet-Draft SCTP NAT Support July 2020

+===============+==============+=============+===============+

| Internal Host | NAT Function | Remote Host | Communication |

+===============+==============+=============+===============+

| Support | Support | Support | Yes |

+---------------+--------------+-------------+---------------+

| Support | Support | No Support | Limited |

+---------------+--------------+-------------+---------------+

| Support | No Support | Support | None |

+---------------+--------------+-------------+---------------+

| Support | No Support | No Support | None |

+---------------+--------------+-------------+---------------+

| No Support | Support | Support | Limited |

+---------------+--------------+-------------+---------------+

| No Support | Support | No Support | Limited |

+---------------+--------------+-------------+---------------+

| No Support | No Support | Support | None |

+---------------+--------------+-------------+---------------+

| No Support | No Support | No Support | None |

+---------------+--------------+-------------+---------------+

Table 1: Communication possibilities

From the table it can be seen that when a NAT function does not

support the extension no communication can occur. This assumes that

the NAT function does not handle SCTP packets at all and all SCTP

packets sent externally from behind a NAT function are discarded by

the NAT function. In some cases, where the NAT function supports the

feature but one of the two hosts does not support the feature,

communication may occur but in a limited way. For example only one

host may be able to have a connection when a collision case occurs.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and

"OPTIONAL" in this document are to be interpreted as described in BCP

14 [RFC2119] [RFC8174] when, and only when, they appear in all

capitals, as shown here.

3. Terminology

This document uses the following terms, which are depicted in

Figure 1. Familiarity with the terminology used in [RFC4960] and

[RFC5061] is assumed.

Internal-Address (Int-Addr)

An internal address that is known to the internal host.

Stewart, et al. Expires 29 January 2021 [Page 5]

Internet-Draft SCTP NAT Support July 2020

Internal-Port (Int-Port)

The port number that is in use by the host holding the Internal-

Address.

Internal-VTag (Int-VTag)

The SCTP Verification Tag (VTag) (see Section 3.1 of [RFC4960])

that the internal host has chosen for its communication. The VTag

is a unique 32-bit tag that must accompany any incoming SCTP

packet for this association to the Internal-Address.

Remote-Address (Rem-Addr)

The address that an internal host is attempting to contact.

Remote-Port (Rem-Port)

The port number of the peer process at the Remote-Address.

Remote-VTag (Rem-VTag)

The Verification Tag (VTag) (see Section 3.1 of [RFC4960]) that

the host holding the Remote-Address has chosen for its

communication. The VTag is a unique 32-bit tag that must

accompany any incoming SCTP packet for this association to the

Remote-Address.

External-Address (Ext-Addr)

An external address assigned to the NAT function, that it uses as

a source address when sending packets towards a Remote-Address.

Internal Network | External Network

|

Internal | External Remote

+--------+ Address | Address /--\/--\ Address +--------+

| SCTP | +-----+ / \ | SCTP |

|endpoint|=========| NAT |=======| Network |==========|endpoint|

| A | +-----+ \ / | B |

+--------+ Internal | \--/\--/ Remote +--------+

Internal Port | Port Remote

VTag | VTag

Figure 1: Basic Network Setup

4. Motivation

4.1. SCTP NAT Traversal Scenarios

This section defines the notion of single and multipoint NAT

traversal.

Stewart, et al. Expires 29 January 2021 [Page 6]

Internet-Draft SCTP NAT Support July 2020

4.1.1. Single Point Traversal

In this case, all packets in the SCTP association go through a single

NAT function, as shown in Figure 2.

Internal Network | External Network

|

+--------+ | /--\/--\ +--------+

| SCTP | +-----+ / \ | SCTP |

|endpoint|=========| NAT |========= | Network | ========|endpoint|

| A | +-----+ \ / | B |

+--------+ | \--/\--/ +--------+

|

Figure 2: Single NAT Scenario

A variation of this case is shown in Figure 3, i.e., multiple NAT functions

in the forwarding path between two endpoints.

Internal | External : Internal | External

| : |

+--------+ | : | /--\/--\ +--------+

| SCTP | +-----+ : +-----+ / \ | SCTP |

|endpoint|==| NAT |=======:=======| NAT |==| Network |==|endpoint|

| A | +-----+ : +-----+ \ / | B |

+--------+ | : | \--/\--/ +--------+

| : |

Figure 3: Serial NAT Functions scenario

Although one of the main benefits of SCTP multi-homing is redundant

paths, in the single point traversal scenario the NAT function

represents a single point of failure in the path of the SCTP multi-

homed association. However, the rest of the path may still benefit

from path diversity provided by SCTP multi-homing.

The two SCTP endpoints in this case can be either single-homed or

multi-homed. However, the important thing is that the NAT function

in this case sees all the packets of the SCTP association.

4.1.2. Multipoint Traversal

This case involves multiple NAT functions and each NAT function only

sees some of the packets in the SCTP association. An example is

shown in Figure 4.

Stewart, et al. Expires 29 January 2021 [Page 7]

Internet-Draft SCTP NAT Support July 2020

Internal | External

+------+ /---\/---\

+--------+ /=======|NAT A |=========\ / \ +--------+

| SCTP | / +------+ \/ \ | SCTP |

|endpoint|/ ... | Network |===|endpoint|

| A |\ \ / | B |

+--------+ \ +------+ / \ / +--------+

\=======|NAT B |=========/ \---\/---/

+------+

|

Figure 4: Parallel NAT Functions Scenario

This case does not apply to a single-homed SCTP association (i.e.,

both endpoints in the association use only one IP address). The

advantage here is that the existence of multiple NAT traversal points

can preserve the path diversity of a multi-homed association for the

entire path. This in turn can improve the robustness of the

communication.

4.2. Limitations of Classical NAPT for SCTP

Using classical NAPT may result in changing one of the SCTP port

numbers during the processing which requires the recomputation of the

transport layer checksum by the NAPT fucntion. Whereas for UDP and TCP

this can be done very efficiently, for SCTP the checksum (CRC32c)

over the entire packet needs to be recomputed (see Appendix B of

[RFC4960] for details of the CRC32c computation). This would

considerably add to the NAT computational burden, however hardware

support may mitigate this in some implementations.

An SCTP endpoint may have multiple addresses but only has a single

port number to use. To make multipoint traversal work, all the NAT

functions involved must recognize the packets they see as belonging

to the same SCTP association and perform port number translation in a

consistent way. One possible way of doing this is to use a pre-

defined table of port numbers and addresses configured within each NAT

function. Other mechanisms could make use of NAT to NAT

communication. Such mechanisms have not been deployed on a wide

scale base and thus are not a recommended solution. Therefore an

SCTP variant of NAT function has been developed (Section 4.3).

4.3. The SCTP-Specific Variant of NAT

In this section it is allowed that there are multiple SCTP capable

hosts behind a NAT function that share one Exernal-Address.

Furthermore, this section focuses on the single point traversal

Scenario (Section 4.1.1).

Stewart, et al. Expires 29 January 2021 [Page 8]

Internet-Draft SCTP NAT Support July 2020

The modification of outgoing SCTP packets from an internal host is simple: the

source address of the packets has to be replaced with the External-

Address.

It may also be necessary to establish some state in the NAT

function to later handle incoming packets. Typically, the NAT function has to maintain a NAT

binding table of Internal-VTag , Internal-Port, Remote-VTag, Remote-

Port, Internal-Address, and whether the restart procedure is disabled

or not. An entry in that NAT binding table is called a NAT-State

control block. The function Create() obtains the just mentioned

parameters and returns a NAT-State control block. A NAT function MAY

allow creating NAT-State control blocks via a management interface.

For SCTP packets coming from the external realm of the NAT function, the destination

address of the packets has to be replaced with the Internal-Address

of the host to which the packet has to be delivered. The lookup of

the Internal-Address is based on the Remote-VTag, Remote-Port,

Internal-VTag and the Internal-Port.

The entries in the NAT binding table need to fulfill some uniqueness

conditions. There must not be more than one entry NAT binding table

with the same pair of Internal-Port and Remote-Port. This rule can

be relaxed, if all NAT binding table entries with the same Internal-

Port and Remote-Port have the support for the restart procedure

enabled. In this case there must be no more than one entry with the

same Internal-Port, Remote-Port, and Remote-VTag and no more than one

NAT binding table entry with the same Internal-Port, Remote-Port, and

Int-VTag.

The processing of outgoing SCTP packets containing an INIT chunk is

described in Figure 5. The scenario shown is valid for

all message flows in this section.

Stewart, et al. Expires 29 January 2021 [Page 9]

Internet-Draft SCTP NAT Support July 2020

/--\/--\

+--------+ +-----+ /EXternal\ +--------+

| Host A | <------> | NAT | <------> | Network | <------> | Host B |

+--------+ +-----+ \ / +--------+

\--/\---/

INIT[Initiate-Tag]

Int-Addr:Int-Port ------> Rem-Addr:Rem-Port

Rem-VTag=0

Create(Initiate-Tag, Int-Port, 0, Rem-Port, Int-Addr,

RestartSupported)

Returns(NAT-State control block)

Translate To:

INIT[Initiate-Tag]

Ext-Addr:Int-Port ------> Rem-Addr:Rem-Port

Rem-VTag=0

Figure 5: XXXXX

Normally a NAT binding table entry will be created.

However, it is possible that there is already a NAT binding table

entry with the same Remote-Port, Internal-Port, and Internal-VTag but

different Internal-Address. In this case the packet containing the

INIT chunk MUST be dropped by the NAT and a packet containing an

ABORT chunk SHOULD be sent to the SCTP host that originated the

packet with the M bit set and an appropriate error cause (see

Section 5.1.1 for the format). The source address of the packet

containing the ABORT chunk MUST be the destination address of the

packet containing the INIT chunk.

If an outgoing SCTP packet contains an INIT or ASCONF chunk and a

matching NAT binding table entry is found, the packet is processed as

a normal outgoing packet.

It is also possible that a connection to Remote-Address and Remote-

Port exists without an Internal-VTag conflict but there exists a NAT

binding table entry with the same port numbers but a different

Internal-Address. In such a case the packet containing the INIT

chunk MUST be dropped by the NAT function and a packet containing an

ABORT chunk SHOULD be sent to the SCTP host that originated the

packet with the M bit set and an appropriate error cause (see

Section 5.1.1 for the format).

The processing of outgoing SCTP packets containing no INIT chunks is

depicted in Figure 6.

Stewart, et al. Expires 29 January 2021 [Page 10]

Internet-Draft SCTP NAT Support July 2020

/--\/--\

+--------+ +-----+ / \ +--------+

| Host A | <------> | NAT | <------> | Network | <------> | Host B |

+--------+ +-----+ \ / +--------+

\--/\---/

Int-Addr:Int-Port ------> Rem-Addr:Rem-Port

Rem-VTag

Translate To:

Ext-Addr:Int-Port ------> Rem-Addr:Rem-Port

Rem-VTag

Figure 6: XXXX

The processing of incoming SCTP packets containing an INIT ACK chunk

is described in Figure 7. The Lookup() function getting

as input the Internal-VTag, Internal-Port, Remote-VTag, and Remote-

Port, returns the corresponding entry of the NAT binding table and

updates the Remote-VTag by substituting it with the value of the

Initiate-Tag of the INIT ACK chunk. The wildcard character signifies

that the parameter's value is not considered in the Lookup() function

or changed in the Update() function, respectively.

/--\/--\

+--------+ +-----+ / \ +--------+

| Host A | <------> | NAT | <------> | Internet | <------> | Host B |

+--------+ +-----+ \ / +--------+

\--/\---/

INIT ACK[Initiate-Tag]

Ext-Addr:Int-Port <---- Rem-Addr:Rem-Port

Int-VTag

Lookup(Int-VTag, Int-Port, \*, Rem-Port)

Update(\*, \*, Initiate-Tag, \*)

Returns(NAT-State control block containing Int-Addr)

INIT ACK[Initiate-Tag]

Int-Addr:Int-Port <------ Rem-Addr:Rem-Port

Int-VTag

Figure 7: XXXX

In the case Lookup fails, the SCTP packet is dropped. If it

succeeds, the Update routine inserts the Remote-VTag (the Initiate-

Tag of the INIT ACK chunk) in the NAT-State control block.

Stewart, et al. Expires 29 January 2021 [Page 11]

Internet-Draft SCTP NAT Support July 2020

The processing of incoming SCTP packets containing an ABORT or

SHUTDOWN COMPLETE chunk with the T-Bit set is illustrated in Figure 8.

/--\/--\

+--------+ +-----+ / \ +--------+

| Host A | <------> | NAT | <------> | Internet | <------> | Host B |

+--------+ +-----+ \ / +--------+

\--/\---/

Ext-Addr:Int-Port <------ Rem-Addr:Rem-Port

Rem-VTag

Lookup(\*, Int-Port, Rem-VTag, Rem-Port)

Returns(NAT-State control block containing Int-Addr)

Int-Addr:Int-Port <------ Rem-Addr:Rem-Port

Rem-VTag

Figure 8: XXX

For an incoming packet containing an INIT chunk a table lookup is

made only based on the addresses and port numbers. If an entry with

a Remote-VTag of zero is found, it is considered a match and the

Remote-VTag is updated. If an entry with a non-matching Remote-VTag

is found or no entry is found, the incoming packet is silently dropped. If an

entry with a matching Remote-VTag is found, the incoming packet is

forwarded. This allows the handling of INIT collision through NAT

functions.

The processing of other incoming SCTP packets is described in Figure 9.

/--\/--\

+--------+ +-----+ / \ +--------+

| Host A | <------> | NAT | <------> | Internet | <------> | Host B |

+--------+ +-----+ \ / +--------+

\--/\---/

Ext-Addr:Int-Port <------ Rem-Addr:Rem-Port

Int-VTag

Lookup(Int-VTag, Int-Port, \*, Rem-Port)

Returns(NAT-State control block containing Internal-Address)

Int-Addr:Int-Port <------ Rem-Addr:Rem-Port

Int-VTag

Figure 9 : XXXX

Stewart, et al. Expires 29 January 2021 [Page 12]

Internet-Draft SCTP NAT Support July 2020

5. Data Formats

This section defines the formats used to support NAT traversal.

Sections 5.1 and 5.2 describe chunks and error causes sent by

NAT functions and received by SCTP endpoints. Section 5.3 describes

parameters sent by SCTP endpoints and used by NAT functions and SCTP

endpoints.

5.1. Modified Chunks

This section presents existing chunks defined in [RFC4960] for which

additional flags are specified by this document.

5.1.1. Extended ABORT 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 = 6 | Reserved |M|T| Length |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

\ \

/ zero or more Error Causes /

\ \

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

The ABORT chunk is extended to add the new 'M bit'. The M bit

indicates to the receiver of the ABORT chunk that the chunk was not

generated by the peer SCTP endpoint, but instead by a middle box.

[NOTE to RFC-Editor: Assignment of M bit to be confirmed by IANA.]

5.1.2. Extended ERROR 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 = 9 | Reserved |M|T| Length |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

\ \

/ zero or more Error Causes /

\ \

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

The ERROR chunk defined in [RFC4960] is extended to add the new 'M

bit'. The M bit indicates to the receiver of the ERROR chunk that

the chunk was not generated by the peer SCTP endpoint, but instead by

a middle box (e.g., NAT).

Stewart, et al. Expires 29 January 2021 [Page 13]

Internet-Draft SCTP NAT Support July 2020

[NOTE to RFC-Editor: Assignment of M bit to be confirmed by IANA.]

5.2. New Error Causes

This section defines the new error causes added by this document.

5.2.1. VTag and Port Number Collision Error Cause

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

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Cause Code = 0x00B0 | Cause Length = Variable |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

\ Chunk /

/ \

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Cause Code: 2 bytes (unsigned integer)

This field holds the IANA defined cause code for the 'VTag and

Port Number Collision' Error Cause. IANA is requested to assign

the value 0x00B0 for this cause code.

Cause Length: 2 bytes (unsigned integer)

This field holds the length in bytes of the error cause. The

value MUST be the length of the Cause-Specific Information plus 4.

Chunk: variable length

The Cause-Specific Information is filled with the chunk that

caused this error. This can be an INIT, INIT ACK, or ASCONF

chunk. Note that if the entire chunk will not fit in the ERROR

chunk or ABORT chunk being sent then the bytes that do not fit are

truncated.

[NOTE to RFC-Editor: Assignment of cause code to be confirmed by

IANA.]

5.2.2. Missing State Error Cause

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

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Cause Code = 0x00B1 | Cause Length = Variable |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

\ Incoming Packet /

/ \

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Cause Code: 2 bytes (unsigned integer)

Stewart, et al. Expires 29 January 2021 [Page 14]

Internet-Draft SCTP NAT Support July 2020

This field holds the IANA defined cause code for the 'Missing

State' Error Cause. IANA is requested to assign the value 0x00B1

for this cause code.

Cause Length: 2 bytes (unsigned integer)

This field holds the length in bytes of the error cause. The

value MUST be the length of the Cause-Specific Information plus 4.

Incoming Packet: variable length

The Cause-Specific Information is filled with the IPv4 or IPv6

packet that caused this error. The IPv4 or IPv6 header MUST be

included. Note that if the packet will not fit in the ERROR chunk

or ABORT chunk being sent then the bytes that do not fit are

truncated.

[NOTE to RFC-Editor: Assignment of cause code to be confirmed by

IANA.]

5.2.3. Port Number Collision Error Cause

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

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Cause Code = 0x00B2 | Cause Length = Variable |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

\ Chunk /

/ \

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Cause Code: 2 bytes (unsigned integer)

This field holds the IANA defined cause code for the 'Port Number

Collision' Error Cause. IANA is requested to assign the value

0x00B2 for this cause code.

Cause Length: 2 bytes (unsigned integer)

This field holds the length in bytes of the error cause. The

value MUST be the length of the Cause-Specific Information plus 4.

Chunk: variable length

The Cause-Specific Information is filled with the chunk that

caused this error. This can be an INIT, INIT ACK, or ASCONF

chunk. Note that if the entire chunk will not fit in the ERROR

chunk or ABORT chunk being sent then the bytes that do not fit are

truncated.

[NOTE to RFC-Editor: Assignment of cause code to be confirmed by

IANA.]

Stewart, et al. Expires 29 January 2021 [Page 15]

Internet-Draft SCTP NAT Support July 2020

5.3. New Parameters

This section defines new parameters and their valid appearance

defined by this document.

5.3.1. Disable Restart Parameter

This parameter is used to indicate that the restart procedure is

requested to be disabled. Both endpoints of an association MUST

include this parameter in the INIT chunk and INIT ACK chunk when

establishing an association and MUST include it in the ASCONF chunk

when adding an address to successfully disable the restart procedure.

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 = 0xC007 | Length = 4 |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Parameter Type: 2 bytes (unsigned integer)

This field holds the IANA defined parameter type for the Disable

Restart Parameter. IANA is requested to assign the value 0xC007

for this parameter type.

Parameter Length: 2 bytes (unsigned integer)

This field holds the length in bytes of the parameter. The value

MUST be 4.

[NOTE to RFC-Editor: Assignment of parameter type to be confirmed by

IANA.]

This parameter MAY appear in INIT, INIT ACK and ASCONF chunks and

MUST NOT appear in any other chunk.

5.3.2. VTags Parameter

This parameter is used to help a NAT function to recover from state

loss.

Stewart, et al. Expires 29 January 2021 [Page 16]

Internet-Draft SCTP NAT Support July 2020

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

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Parameter Type = 0xC008 | Parameter Length = 16 |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ASCONF-Request Correlation ID |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Internal Verification Tag |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Remote Verification Tag |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Parameter Type: 2 bytes (unsigned integer)

This field holds the IANA defined parameter type for the VTags

Parameter. IANA is requested to assign the value 0xC008 for this

parameter type.

Parameter Length: 2 bytes (unsigned integer)

This field holds the length in bytes of the parameter. The value

MUST be 16.

ASCONF-Request Correlation ID: 4 bytes (unsigned integer)

This is an opaque integer assigned by the sender to identify each

request parameter. The receiver of the ASCONF Chunk will copy

this 32-bit value into the ASCONF Response Correlation ID field of

the ASCONF ACK response parameter. The sender of the packet

containing the ASCONF chunk can use this same value in the ASCONF

ACK chunk to find which request the response is for. Note that

the receiver MUST NOT change this 32-bit value.

Internal Verification Tag: 4 bytes (unsigned integer)

The Verification Tag that the internal host has chosen for its

communication. The Verification Tag is a unique 32-bit tag that

must accompany any incoming SCTP packet for this association to

the Internal-Address.

Remote Verification Tag: 4 bytes (unsigned integer)

The Verification Tag that the host holding the Remote-Address has

chosen for its communication. The VTag is a unique 32-bit tag

that must accompany any incoming SCTP packet for this association

to the Remote-Address.

[NOTE to RFC-Editor: Assignment of parameter type to be confirmed by

IANA.]

This parameter MAY appear in ASCONF chunks and MUST NOT appear in any

other chunk.

Stewart, et al. Expires 29 January 2021 [Page 17]

Internet-Draft SCTP NAT Support July 2020

6. Procedures for SCTP Endpoints and NAT Functions

When an SCTP endpoint is behind an SCTP-aware NAT, a number of

problems may arise as it tries to communicate with its peers:

\* IP addresses can not be included in the SCTP packet. This is

discussed in Section 6.1.

\* More than one host behind a NAT function could select the same

VTag and source port number when communicating with the same peer server. This

creates a situation where the NAT function will not be able to

tell the two associations apart. This situation is discussed in

Section 6.2.

\* When an SCTP endpoint is a server communicating with multiple

peers and the peers are behind the same NAT function, then the two

endpoints cannot be distinguished by the server. This case is

discussed in Section 6.3.

\* A restart of a NAT function during a conversation could cause a

loss of its state. This problem and its solution is discussed in

Section 6.4.

\* NAT functions need to deal with SCTP packets being fragmented at

the IP layer. This is discussed in Section 6.5.

\* An SCTP endpoint can be behind two NAT functions in parallel

providing redundancy. The method to set up this scenario is

discussed in Section 6.6.

Each of these mechanisms requires additional chunks and parameters,

defined in this document, and modified handling procedures from those

specified in [RFC4960] as described below.

6.1. Association Setup Considerations for Endpoints

The association setup procedure defined in [RFC4960] allows multi-

homed SCTP endpoints to exchange its IP-addresses by using IPv4 or

IPv6 address parameters in the INIT and INIT ACK chunks. However,

this does not work when NAT functions are present.

Every association setup from a host behind a NAT function MUST NOT

use multiple internal addresses. The INIT chunk MUST NOT contain an

IPv4 Address parameter, IPv6 Address parameter, or Supported Address

Types parameter. The INIT ACK chunk MUST NOT contain any IPv4

Address parameter or IPv6 Address parameter using non-global

addresses. The INIT chunk and the INIT ACK chunk MUST NOT contain

any Host Name parameters.

Stewart, et al. Expires 29 January 2021 [Page 18]

Internet-Draft SCTP NAT Support July 2020

If the association should finally be multi-homed, the procedure in

Section 6.6 MUST be used.

The INIT and INIT ACK chunk SHOULD contain the Disable Restart

parameter defined in Section 5.3.1.

6.2. Handling of Internal Port Number and Verification Tag Collisions

Consider the case where two hosts in the Internal-Address space want

to set up an SCTP association with the same service provided by some

remote hosts. This means that the Remote-Port is the same.

If they both choose the same Internal-Port and Internal-VTag, the NAT

function cannot distinguish between incoming packets anymore.

However, this is unlikely. The Internal-VTags are chosen at random

and if the Internal-Ports are also chosen from the ephemeral port

range at random this gives a 46-bit random number that has to match.

A NAPT function can control the port number and therefore avoid

collisions deterministically.

The same can happen with the Remote-VTag when a packet containing an

INIT ACK chunk or an ASCONF chunk is processed by the NAT function.

6.2.1. NAT Function Considerations

If the NAT function detects a collision of internal port numbers and

verification tags, it SHOULD send a packet containing an ABORT chunk

with the M bit set if the collision is triggered by a packet

containing an INIT or INIT ACK chunk. If such a collision is

triggered by a packet containing an ASCONF chunk, it SHOULD send a

packet containing an ERROR chunk with the M bit. The M bit is a new

bit defined by this document to express to SCTP that the source of

this packet is a "middle" box, not the peer SCTP endpoint (see

Section 5.1.1). If a packet containing an INIT ACK chunk triggers

the collision, the corresponding packet containing the ABORT chunk

MUST contain the same source and destination address and port numbers

as the packet containing the INIT ACK chunk. If a packet containing

an INIT chunk or an ASCONF chunk, the source and destination address

and port numbers MUST be swapped.

The sender of the packet containing an ERROR or ABORT chunk MUST

include the error cause with cause code 'VTag and Port Number

Collision' (see Section 5.2.1).

Stewart, et al. Expires 29 January 2021 [Page 19]

Internet-Draft SCTP NAT Support July 2020

6.2.2. Endpoint Considerations

The sender of the packet containing the INIT chunk or the receiver of

a packet containing the INIT ACK chunk, upon reception of a packet

containing an ABORT chunk with M bit set and the appropriate error

cause code for colliding NAT binding table state is included, SHOULD

reinitiate the association setup procedure after choosing a new

initiate tag, if the association is in COOKIE-WAIT state. In any

other state, the SCTP endpoint MUST NOT respond.

The sender of packet containing the ASCONF chunk, upon reception of a

packet containing an ERROR chunk with M bit set, MUST stop adding the

path to the association.

6.3. Handling of Internal Port Number Collisions

When two SCTP hosts are behind an SCTP-aware NAT it is possible that

two SCTP hosts in the Internal-Address space will want to set up an

SCTP association with the same server running on the same host in the

Internet. If the two hosts choose the same internal port, this is

considered an internal port number collision.

For the NAT function, appropriate tracking may be performed by

assuring that the VTags are unique between the two hosts.

6.3.1. NAT Function Considerations

The NAT function, when processing the packet containing the INIT ACK

chunk, should note in its NAT binding table that the association

supports the disable restart extension. This note is used when

establishing future associations (i.e. when processing a packet

containing an INIT chunk from an internal host) to decide if the

connection should be allowed. The NAT function does the following

when processing a packet containing an INIT chunk:

\* If the packet containing the INIT chunk is originating from an

internal port to an remote port for which the NAT function has no

matching NAT binding table entry, it MUST allow the packet

containing the INIT chunk creating an NAT binding table entry.

\* If the packet containing the INIT chunk matches an existing NAT

binding table entry, it MUST validate that the disable restart

feature is supported and, if it does, allow the packet containing

the INIT chunk to be forwarded.

\* If the disable restart feature is not supported, the NAT function

SHOULD send a packet containing an ABORT chunk with the M bit set.

Stewart, et al. Expires 29 January 2021 [Page 20]

Internet-Draft SCTP NAT Support July 2020

The 'Port Number Collision' error cause (see Section 5.2.3) MUST be

included in the ABORT chunk sent in response to the packet containing

an INIT chunk.

If the collision is triggered by a packet containing an ASCONF chunk,

a packet containing an ERROR chunk with the 'Port Number Collision'

error cause SHOULD be sent in response to the packet containing the

ASCONF chunk.

6.3.2. Endpoint Considerations

For the remote SCTP server on the Internet this means that the

Remote-Port and the Remote-Address are the same. If they both have

chosen the same Internal-Port the server cannot distinguish between

both associations based on the address and port numbers. For the

server it looks like the association is being restarted. To overcome

this limitation the client sends a Disable Restart parameter in the

INIT chunk.

When the server receives this parameter it does the following:

\* It MUST include a Disable Restart parameter in the INIT ACK to

inform the client that it will support the feature.

\* It MUST disable the restart procedures defined in [RFC4960] for

this association.

Servers that support this feature will need to be capable of

maintaining multiple connections to what appears to be the same peer

(behind the NAT function) differentiated only by the VTags.

6.4. Handling of Missing State

6.4.1. NAT Function Considerations

If the NAT function receives a packet from the internal network for

which the lookup procedure does not find an entry in the NAT binding

table, a packet containing an ERROR chunk SHOULD be sent back with

the M bit set. The source address of the packet containing the ERROR

chunk MUST be the destination address of the incoming SCTP packet.

The verification tag is reflected and the T bit is set. Such a

packet containing an ERROR chunk SHOULD NOT be sent if the received

packet contains an ABORT, SHUTDOWN COMPLETE or INIT ACK chunk. A

packet containing an ERROR chunk MUST NOT be sent if the received

packet contains an ERROR chunk with the M bit set. In any case, the

packet SHOULD NOT be forwarded to the remote address.

Stewart, et al. Expires 29 January 2021 [Page 21]

Internet-Draft SCTP NAT Support July 2020

When sending a packet containing an ERROR chunk, the error cause

'Missing State' (see Section 5.2.2) MUST be included and the M bit of

the ERROR chunk MUST be set (see Section 5.1.2).

If the NAT device receives a packet for which it has no NAT binding

table entry and the packet contains an ASCONF chunk with the VTags

parameter, the NAT function MUST update its NAT binding table

according to the verification tags in the VTags parameter and the

optional Disable Restart parameter.

6.4.2. Endpoint Considerations

Upon reception of this packet containing the ERROR chunk by an SCTP

endpoint the receiver takes the following actions:

\* It SHOULD validate that the verification tag is reflected by

looking at the VTag that would have been included in the outgoing

packet. If the validation fails, discard the incoming packet

containing the ERROR chunk.

\* It SHOULD validate that the peer of the SCTP association supports

the dynamic address extension. If the validation fails, discard

the incoming packet containing the ERROR chunk.

\* It SHOULD generate a packet containing a new ASCONF chunk

containing the VTags parameter (see Section 5.3.2) and the Disable

Restart parameter (see Section 5.3.1) if the association is using

the disable restart feature. By processing this packet the NAT

function can recover the appropriate state. The procedures for

generating an ASCONF chunk can be found in [RFC5061].

The peer SCTP endpoint receiving such a packet containing an ASCONF

chunk SHOULD either add the address and respond with an

acknowledgment, if the address is new to the association (following

all procedures defined in [RFC5061]). Or, if the address is already

part of the association, the SCTP endpoint MUST NOT respond with an

error, but instead SHOULD respond with packet containing an ASCONF

ACK chunk acknowledging the address and take no action (since the

address is already in the association).

Note that it is possible that upon receiving a packet containing an

ASCONF chunk containing the VTags parameter the NAT function will

realize that it has an 'Internal Port Number and Verification Tag

collision'. In such a case the NAT function SHOULD send a packet

containing an ERROR chunk with the error cause code set to 'VTag and

Port Number Collision' (see Section 5.2.1).

Stewart, et al. Expires 29 January 2021 [Page 22]

Internet-Draft SCTP NAT Support July 2020

If an SCTP endpoint receives a packet containing an ERROR chunk with

'Internal Port Number and Verification Tag collision' as the error

cause and the packet in the Error Chunk contains an ASCONF with the

VTags parameter, careful examination of the association is required.

The endpoint does the following:

\* It MUST validate that the verification tag is reflected by looking

at the VTag that would have been included in the outgoing packet.

If the validation fails, it MUST discard the packet.

\* It MUST validate that the peer of the SCTP association supports

the dynamic address extension. If the peer does not support it,

the NAT function MUST discard the incoming packet containing the

ERROR chunk.

\* If the association is attempting to add an address (i.e. following

the procedures in Section 6.6) then the endpoint MUST NOT consider

the address part of the association and SHOULD make no further

attempt to add the address (i.e. cancel any ASCONF timers and

remove any record of the path), since the NAT function has a VTag

collision and the association cannot easily create a new VTag (as

it would if the error occurred when sending a packet containing an

INIT chunk).

\* If the endpoint has no other path, i.e. the procedure was executed

due to missing a state in the NAT function, then the endpoint MUST

abort the association. This would occur only if the local NAT

function restarted and accepted a new association before

attempting to repair the missing state (Note that this is no

different than what happens to all TCP connections when a NAT

function looses its state).

6.5. Handling of Fragmented SCTP Packets by NAT Functions

SCTP minimizes the use of IP-level fragmentation. However, it can

happen that using IP-level fragmentation is needed to continue an

SCTP association. For example, if the path MTU is reduced and there

are still some DATA chunk in flight, which require packets larger

than the new path MTU. If IP-level fragmentation can not be used,

the SCTP association will be terminated in a non-graceful way.

Therefore, a NAT function MUST be able to handle IP-level fragmented

SCTP packets. The fragments may arrive in any order.

Stewart, et al. Expires 29 January 2021 [Page 23]

Internet-Draft SCTP NAT Support July 2020

When an SCTP packet can not be forwarded by the NAT function due to

MTU issues and the IP header forbids fragmentation, the NAT MUST send

back a "Fragmentation needed and DF set" ICMPv4 or PTB ICMPv6 message

to the internal host. This allows for a faster recovery from this

packet drop.

6.6. Multi Point Traversal Considerations for Endpoints

If a multi-homed SCTP endpoint behind a NAT function connects to a

peer, it MUST first set up the association single-homed with only one

address causing the first NAT function to populate its state. Then

it SHOULD add each IP address using packets containing ASCONF chunks

sent via their respective NAT functions. The address to add is the

wildcard address and the lookup address SHOULD also contain the VTags

parameter and optionally the Disable Restart parameter.

7. Various Examples of NAT Traversals

Please note that this section is informational only.

The addresses being used in the following examples are IPv4 addresses

for private-use networks and for documentation as specified in

[RFC6890]. However, the method described here is not limited to this

NAT44 case.

The NAT binding table entries shown in the following examples do not

include the flag indicating whether the restart procedure is

supported or not. This flag is not relevant for these examples.

7.1. Single-homed Client to Single-homed Server

The internal client starts the association with the remote server via

a four-way-handshake. Host A starts by sending a packet containing

an INIT chunk.

/--\/--\

+--------+ +-----+ / \ +--------+

| Host A | <------> | NAT | <------> | Internet | <------> | Host B |

+--------+ +-----+ \ / +--------+

\--/\---/

+---------+--------+----------+--------+-----------+

NAT | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

INIT[Initiate-Tag = 1234]

10.0.0.1:1 ------> 203.0.113.1:2

Rem-VTtag = 0

Figure XXX: xxxx

Stewart, et al. Expires 29 January 2021 [Page 24]

Internet-Draft SCTP NAT Support July 2020

A NAT binding tabled entry is created, the source address is

substituted and the packet is sent on:

NAT function creates entry:

+---------+--------+----------+--------+-----------+

NAT | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 1234 | 1 | 0 | 2 | 10.0.0.1 |

+---------+--------+----------+--------+-----------+

INIT[Initiate-Tag = 1234]

192.0.2.1:1 ------------------------> 203.0.113.1:2

Rem-VTtag = 0

Host B receives the packet containing an INIT chunk and sends a

packet containing an INIT ACK chunk with the NAT's Remote-address as

destination address.

/--\/--\

+--------+ +-----+ / \ +--------+

| Host A | <------> | NAT | <------> | Internet | <------> | Host B |

+--------+ +-----+ \ / +--------+

\--/\---/

INIT ACK[Initiate-Tag = 5678]

192.0.2.1:1 <----------------------- 203.0.113.1:2

Int-VTag = 1234

NAT function updates entry:

+---------+--------+----------+--------+-----------+

NAT | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 1234 | 1 | 5678 | 2 | 10.0.0.1 |

+---------+--------+----------+--------+-----------+

INIT ACK[Initiate-Tag = 5678]

10.0.0.1:1 <------ 203.0.113.1:2

Int-VTag = 1234

The handshake finishes with a COOKIE ECHO acknowledged by a COOKIE

ACK.

Stewart, et al. Expires 29 January 2021 [Page 25]

Internet-Draft SCTP NAT Support July 2020

/--\/--\

+--------+ +-----+ / \ +--------+

| Host A | <------> | NAT | <------> | Internet | <------> | Host B |

+--------+ +-----+ \ / +--------+

\--/\---/

COOKIE ECHO

10.0.0.1:1 ------> 203.0.113.1:2

Rem-VTag = 5678

COOKIE ECHO

192.0.2.1:1 -----------------------> 203.0.113.1:2

Rem-VTag = 5678

COOKIE ACK

192.0.2.1:1 <----------------------- 203.0.113.1:2

Int-VTag = 1234

COOKIE ACK

10.0.0.1:1 <------ 203.0.113.1:2

Int-VTag = 1234

7.2. Single-homed Client to Multi-homed Server

The internal client is single-homed whereas the remote server is

multi-homed. The client (Host A) sends a packet containing an INIT

chunk like in the single-homed case.

+--------+

/--\/--\ /-|Router 1| \

+------+ +-----+ / \ / +--------+ \ +------+

| Host | <-----> | NAT | <-> | Internet | == =| Host |

| A | +-----+ \ / \ +--------+ / | B |

+------+ \--/\--/ \-|Router 2|-/ +------+

+--------+

+---------+--------+----------+--------+-----------+

NAT | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

INIT[Initiate-Tag = 1234]

10.0.0.1:1 ---> 203.0.113.1:2

Rem-VTag = 0

NAT function creates entry:

Stewart, et al. Expires 29 January 2021 [Page 26]

Internet-Draft SCTP NAT Support July 2020

+---------+--------+----------+--------+-----------+

NAT | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 1234 | 1 | 0 | 2 | 10.0.0.1 |

+---------+--------+----------+--------+-----------+

INIT[Initiate-Tag = 1234]

192.0.2.1:1 --------------------------> 203.0.113.1:2

Rem-VTag = 0

The server (Host B) includes its two addresses in the INIT ACK chunk.

+--------+

/--\/--\ /-|Router 1| \

+------+ +-----+ / \ / +--------+ \ +------+

| Host | <-----> | NAT | <-> | Internet | == =| Host |

| A | +-----+ \ / \ +--------+ / | B |

+------+ \--/\--/ \-|Router 2|-/ +------+

+--------+

INIT ACK[Initiate-tag = 5678, IP-Addr = 203.0.113.129]

192.0.2.1:1 <-------------------------- 203.0.113.1:2

Int-VTag = 1234

The NAT function does not need to change the NAT binding table for

the second address:

+---------+--------+----------+--------+-----------+

NAT | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 1234 | 1 | 5678 | 2 | 10.0.0.1 |

+---------+--------+----------+--------+-----------+

INIT ACK[Initiate-Tag = 5678]

10.0.0.1:1 <--- 203.0.113.1:2

Int-VTag = 1234

The handshake finishes with a COOKIE ECHO acknowledged by a COOKIE

ACK.

Stewart, et al. Expires 29 January 2021 [Page 27]

Internet-Draft SCTP NAT Support July 2020

+--------+

/--\/--\ /-|Router 1| \

+------+ +-----+ / \ / +--------+ \ +------+

| Host | <-----> | NAT | <-> | Internet | == =| Host |

| A | +-----+ \ / \ +--------+ / | B |

+------+ \--/\--/ \-|Router 2|-/ +------+

+--------+

COOKIE ECHO

10.0.0.1:1 ---> 203.0.113.1:2

Rem-VTag = 5678

COOKIE ECHO

192.0.2.1:1 --------------------------> 203.0.113.1:2

Rem-VTag = 5678

COOKIE ACK

192.0.2.1:1 <-------------------------- 203.0.113.1:2

Int-VTag = 1234

COOKIE ACK

10.0.0.1:1 <--- 203.0.113.1:2

Int-VTag = 1234

7.3. Multihomed Client and Server

The client (Host A) sends a packet containing an INIT chunk to the

server (Host B), but does not include the second address.

+-------+

/--| NAT 1 |--\ /--\/--\

+------+ / +-------+ \ / \ +--------+

| Host |=== ====| Internet |====| Host B |

| A | \ +-------+ / \ / +--------+

+------+ \--| NAT 2 |--/ \--/\--/

+-------+

+---------+--------+----------+--------+-----------+

NAT 1 | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

INIT[Initiate-Tag = 1234]

10.0.0.1:1 --------> 203.0.113.1:2

Rem-VTag = 0

NAT function 1 creates entry:

Stewart, et al. Expires 29 January 2021 [Page 28]

Internet-Draft SCTP NAT Support July 2020

+---------+--------+----------+--------+-----------+

NAT 1 | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 1234 | 1 | 0 | 2 | 10.0.0.1 |

+---------+--------+----------+--------+-----------+

INIT[Initiate-Tag = 1234]

192.0.2.1:1 ---------------------> 203.0.113.1:2

Rem-VTag = 0

Host B includes its second address in the INIT ACK.

+-------+

/--------| NAT 1 |--------\ /--\/--\

+------+ / +-------+ \ / \ +--------+

| Host |=== ====| Internet |===| Host B |

| A | \ +-------+ / \ / +--------+

+------+ \--------| NAT 2 |--------/ \--/\--/

+-------+

INIT ACK[Initiate-Tag = 5678, IP-Addr = 203.0.113.129]

192.0.2.1:1 <----------------------- 203.0.113.1:2

Int-VTag = 1234

NAT function 1 does not need to update the NAT binding table for the

second address:

+---------+--------+----------+--------+-----------+

NAT 1 | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 1234 | 1 | 5678 | 2 | 10.0.0.1 |

+---------+--------+----------+--------+-----------+

INIT ACK[Initiate-Tag = 5678]

10.0.0.1:1 <-------- 203.0.113.1:2

Int-VTag = 1234

The handshake finishes with a COOKIE ECHO acknowledged by a COOKIE

ACK.

Stewart, et al. Expires 29 January 2021 [Page 29]

Internet-Draft SCTP NAT Support July 2020

+-------+

/--------| NAT 1 |--------\ /--\/--\

+------+ / +-------+ \ / \ +--------+

| Host |=== ====| Internet |===| Host B |

| A | \ +-------+ / \ / +--------+

+------+ \--------| NAT 2 |--------/ \--/\--/

+-------+

COOKIE ECHO

10.0.0.1:1 --------> 203.0.113.1:2

Rem-VTag = 5678

COOKIE ECHO

192.0.2.1:1 ------------------> 203.0.113.1:2

Rem-VTag = 5678

COOKIE ACK

192.0.2.1:1 <------------------ 203.0.113.1:2

Int-VTag = 1234

COOKIE ACK

10.0.0.1:1 <------- 203.0.113.1:2

Int-VTag = 1234

Host A announces its second address in an ASCONF chunk. The address

parameter contains an undefined address (0) to indicate that the

source address should be added. The lookup address parameter within

the ASCONF chunk will also contain the pair of VTags (remote and

internal) so that the NAT function may populate its NAT binding table

entry completely with this single packet.

+-------+

/--------| NAT 1 |--------\ /--\/--\

+------+ / +-------+ \ / \ +--------+

| Host |=== ====| Internet |===| Host B |

| A | \ +-------+ / \ / +--------+

+------+ \--------| NAT 2 |--------/ \--/\--/

+-------+

ASCONF [ADD-IP=0.0.0.0, INT-VTag=1234, Rem-VTag = 5678]

10.1.0.1:1 --------> 203.0.113.129:2

Rem-VTag = 5678

NAT function 2 creates a complete entry:

Stewart, et al. Expires 29 January 2021 [Page 30]

Internet-Draft SCTP NAT Support July 2020

+---------+--------+----------+--------+-----------+

NAT 2 | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 1234 | 1 | 5678 | 2 | 10.1.0.1 |

+---------+--------+----------+--------+-----------+

ASCONF [ADD-IP, Int-VTag=1234, Rem-VTag = 5678]

192.0.2.129:1 -------------------> 203.0.113.129:2

Rem-VTag = 5678

ASCONF ACK

192.0.2.129:1 <------------------- 203.0.113.129:2

Int-VTag = 1234

ASCONF ACK

10.1.0.1:1 <----- 203.0.113.129:2

Int-VTag = 1234

7.4. NAT Function Loses Its State

Association is already established between Host A and Host B, when

the NAT function loses its state and obtains a new external address.

Host A sends a DATA chunk to Host B.

/--\/--\

+--------+ +-----+ / \ +--------+

| Host A | <----------> | NAT | <----> | Internet | <----> | Host B |

+--------+ +-----+ \ / +--------+

\--/\--/

+---------+--------+----------+--------+-----------+

NAT | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

DATA

10.0.0.1:1 ----------> 203.0.113.1:2

Rem-VTag = 5678

The NAT function cannot find an entry in the NAT binding table for

the association. It sends a packet containing an ERROR chunk with

the M-Bit set and the cause "NAT state missing".

Stewart, et al. Expires 29 January 2021 [Page 31]

Internet-Draft SCTP NAT Support July 2020

/--\/--\

+--------+ +-----+ / \ +--------+

| Host A | <----------> | NAT | <----> | Internet | <----> | Host B |

+--------+ +-----+ \ / +--------+

\--/\--/

ERROR [M-Bit, NAT state missing]

10.0.0.1:1 <---------- 203.0.113.1:2

Rem-VTag = 5678

On reception of the packet containing the ERROR chunk, Host A sends a

packet containing an ASCONF chunk indicating that the former

information has to be deleted and the source address of the actual

packet added.

/--\/--\

+--------+ +-----+ / \ +--------+

| Host A | <----------> | NAT | <----> | Internet | <----> | Host B |

+--------+ +-----+ \ / +--------+

\--/\--/

ASCONF [ADD-IP, DELETE-IP, Int-VTag=1234, Rem-VTag = 5678]

10.0.0.1:1 ----------> 203.0.113.129:2

Rem-VTag = 5678

+---------+--------+----------+--------+-----------+

NAT | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 1234 | 1 | 5678 | 2 | 10.0.0.1 |

+---------+--------+----------+--------+-----------+

ASCONF [ADD-IP, DELETE-IP, Int-VTag=1234, Rem-VTag = 5678]

192.0.2.2:1 -----------------> 203.0.113.129:2

Rem-VTag = 5678

Host B adds the new source address to this association and deletes

all other addresses from this association.

Stewart, et al. Expires 29 January 2021 [Page 32]

Internet-Draft SCTP NAT Support July 2020

/--\/--\

+--------+ +-----+ / \ +--------+

| Host A | <----------> | NAT | <----> | Internet | <----> | Host B |

+--------+ +-----+ \ / +--------+

\--/\--/

ASCONF ACK

192.0.2.2:1 <----------------- 203.0.113.129:2

Int-VTag = 1234

ASCONF ACK

10.1.0.1:1 <---------- 203.0.113.129:2

Int-VTag = 1234

DATA

10.0.0.1:1 ----------> 203.0.113.1:2

Rem-VTag = 5678

DATA

192.0.2.2:1 -----------------> 203.0.113.129:2

Rem-VTag = 5678

7.5. Peer-to-Peer Communications

If two hosts, each of them behind a NAT function, want to communicate

with each other, they have to get knowledge of the peer's external

address. This can be achieved with a so-called rendezvous server.

Afterwards the destination addresses are external, and the

association is set up with the help of the INIT collision. The NAT

functions create their entries according to their internal peer's

point of view. Therefore, NAT function A's Internal-VTag and

Internal-Port are NAT function B's Remote-VTag and Remote-Port,

respectively. The naming (internal/remote) of the verification tag

in the packet flow is done from the sending host's point of view.

Stewart, et al. Expires 29 January 2021 [Page 33]

Internet-Draft SCTP NAT Support July 2020

Internal | External External | Internal

| |

| /--\/---\ |

+--------+ +-------+ / \ +-------+ +--------+

| Host A |<--->| NAT A |<-->| Internet |<-->| NAT B |<--->| Host B |

+--------+ +-------+ \ / +-------+ +--------+

| \--/\---/ |

NAT Binding Tables

+---------+--------+----------+--------+-----------+

NAT A | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

+---------+--------+----------+--------+-----------+

NAT B | Int | Int | Rem | Rem | Int |

| v-tag | port | v-tag | port | Addr |

+---------+--------+----------+--------+-----------+

INIT[Initiate-Tag = 1234]

10.0.0.1:1 --> 203.0.113.1:2

Rem-VTag = 0

NAT function A creates entry:

+---------+--------+----------+--------+-----------+

NAT A | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 1234 | 1 | 0 | 2 | 10.0.0.1 |

+---------+--------+----------+--------+-----------+

INIT[Initiate-Tag = 1234]

192.0.2.1:1 ----------------> 203.0.113.1:2

Rem-VTag = 0

NAT function B processes the packet containing the INIT chunk, but

cannot find an entry. The SCTP packet is silently discarded and

leaves the NAT binding table of NAT function B unchanged.

+---------+--------+----------+--------+-----------+

NAT B | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

Stewart, et al. Expires 29 January 2021 [Page 34]

Internet-Draft SCTP NAT Support July 2020

Now Host B sends a packet containing an INIT chunk, which is

processed by NAT function B. Its parameters are used to create an

entry.

Internal | External External | Internal

| |

| /--\/---\ |

+--------+ +-------+ / \ +-------+ +--------+

| Host A |<--->| NAT A |<-->| Internet |<-->| NAT B |<--->| Host B |

+--------+ +-------+ \ / +-------+ +--------+

| \--/\---/ |

INIT[Initiate-Tag = 5678]

192.0.2.1:1 <-- 10.1.0.1:2

Rem-VTag = 0

+---------+--------+----------+--------+-----------+

NAT B | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 5678 | 2 | 0 | 1 | 10.1.0.1 |

+---------+--------+----------+--------+-----------+

INIT[Initiate-Tag = 5678]

192.0.2.1:1 <--------------- 203.0.113.1:2

Rem-VTag = 0

NAT function A processes the packet containing the INIT chunk. As

the outgoing packet containing an INIT chunk of Host A has already

created an entry, the entry is found and updated:

Stewart, et al. Expires 29 January 2021 [Page 35]

Internet-Draft SCTP NAT Support July 2020

Internal | External External | Internal

| |

| /--\/---\ |

+--------+ +-------+ / \ +-------+ +--------+

| Host A |<--->| NAT A |<-->| Internet |<-->| NAT B |<--->| Host B |

+--------+ +-------+ \ / +-------+ +--------+

| \--/\---/ |

VTag != Int-VTag, but Rem-VTag == 0, find entry.

+---------+--------+----------+--------+-----------+

NAT A | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 1234 | 1 | 5678 | 2 | 10.0.0.1 |

+---------+--------+----------+--------+-----------+

INIT[Initiate-tag = 5678]

10.0.0.1:1 <-- 203.0.113.1:2

Rem-VTag = 0

Host A sends a packet containing an INIT ACK chunk, which can pass

through NAT function B:

Stewart, et al. Expires 29 January 2021 [Page 36]

Internet-Draft SCTP NAT Support July 2020

Internal | External External | Internal

| |

| /--\/---\ |

+--------+ +-------+ / \ +-------+ +--------+

| Host A |<--->| NAT A |<-->| Internet |<-->| NAT B |<--->| Host B |

+--------+ +-------+ \ / +-------+ +--------+

| \--/\---/ |

INIT ACK[Initiate-Tag = 1234]

10.0.0.1:1 --> 203.0.113.1:2

Rem-VTag = 5678

INIT ACK[Initiate-Tag = 1234]

192.0.2.1:1 ----------------> 203.0.113.1:2

Rem-VTag = 5678

NAT function B updates entry:

+---------+--------+----------+--------+-----------+

NAT B | Int | Int | Rem | Rem | Int |

| VTag | Port | VTag | Port | Addr |

+---------+--------+----------+--------+-----------+

| 5678 | 2 | 1234 | 1 | 10.1.0.1 |

+---------+--------+----------+--------+-----------+

INIT ACK[Initiate-Tag = 1234]

192.0.2.1:1 --> 10.1.0.1:2

Rem-VTag = 5678

The lookup for COOKIE ECHO and COOKIE ACK is successful.

Stewart, et al. Expires 29 January 2021 [Page 37]

Internet-Draft SCTP NAT Support July 2020

Internal | External External | Internal

| |

| /--\/---\ |

+--------+ +-------+ / \ +-------+ +--------+

| Host A |<--->| NAT A |<-->| Internet |<-->| NAT B |<--->| Host B |

+--------+ +-------+ \ / +-------+ +--------+

| \--/\---/ |

COOKIE ECHO

192.0.2.1:1 <-- 10.1.0.1:2

Rem-VTag = 1234

COOKIE ECHO

192.0.2.1:1 <------------- 203.0.113.1:2

Rem-VTag = 1234

COOKIE ECHO

10.0.0.1:1 <-- 203.0.113.1:2

Rem-VTag = 1234

COOKIE ACK

10.0.0.1:1 --> 203.0.113.1:2

Rem-VTag = 5678

COOKIE ACK

192.0.2.1:1 ----------------> 203.0.113.1:2

Rem-VTag = 5678

COOKIE ACK

192.0.2.1:1 --> 10.1.0.1:2

Rem-VTag = 5678

8. SCTP NAT YANG Module

This section defines a YANG module for SCTP NAT.

The terminology for describing YANG data models is defined in

[RFC7950]. The meaning of the symbols in tree diagrams is defined in

[RFC8340].

8.1. Tree Structure

This module augments NAT YANG module [RFC8512] with SCTP specifics.

The module supports both classical SCTP NAT (that is, rewrite port

numbers) and SCTP-specific variant where the ports numbers are not

altered. The YANG "feature" is used to indicate whether SCTP-

specific variant is supported.

Stewart, et al. Expires 29 January 2021 [Page 38]

Internet-Draft SCTP NAT Support July 2020

The tree structure of the SCTP NAT YANG module is provided below:

module: ietf-nat-sctp

augment /nat:nat/nat:instances/nat:instance

/nat:policy/nat:timers:

+--rw sctp-timeout? uint32

augment /nat:nat/nat:instances/nat:instance

/nat:mapping-table/nat:mapping-entry:

+--rw int-VTag? uint32 {sctp-nat}?

+--rw rem-VTag? uint32 {sctp-nat}?

Concretely, the SCTP NAT YANG module augments the NAT YANG module

(policy, in particular) with the following:

\* The sctp-timeout is used to control the SCTP inactivity timeout.

That is, the time an SCTP mapping will stay active without SCTP

packets traversing the NAT. This timeout can be set only for

SCTP. Hence, "/nat:nat/nat:instances/nat:instance/nat:policy/

nat:transport-protocols/nat:protocol-id" MUST be set to '132'

(SCTP).

In addition, the SCTP NAT YANG module augments the mapping entry with

the following parameters defined in Section 3. These parameters

apply only for SCTP NAT mapping entries (i.e.,

"/nat/instances/instance/mapping-table/mapping-entry/transport-

protocol" MUST be set to '132');

\* The Internal Verification Tag (Int-VTag)

\* The Remote Verification Tag (Rem-VTag)

8.2. YANG Module

<CODE BEGINS> file "ietf-nat-sctp@2020-07-13.yang"

module ietf-nat-sctp {

yang-version 1.1;

namespace "urn:ietf:params:xml:ns:yang:ietf-nat-sctp";

prefix nat-sctp;

import ietf-nat {

prefix nat;

reference

"RFC 8512: A YANG Module for Network Address Translation

(NAT) and Network Prefix Translation (NPT)";

}

organization

"IETF TSVWG Working Group";

Stewart, et al. Expires 29 January 2021 [Page 39]

Internet-Draft SCTP NAT Support July 2020

contact

"WG Web: <https://datatracker.ietf.org/wg/tsvwg/>

WG List: <mailto:tsvwg@ietf.org>

Author: Mohamed Boucadair

<mailto:mohamed.boucadair@orange.com>";

description

"This module augments NAT YANG module with Stream Control

Transmission Protocol (SCTP) specifics. The extension supports

both a classical SCTP NAT (that is, rewrite port numbers)

and a, SCTP-specific variant where the ports numbers are

not altered.

Copyright (c) 2020 IETF Trust and the persons identified as

authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or

without modification, is permitted pursuant to, and subject

to the license terms contained in, the Simplified BSD License

set forth in Section 4.c of the IETF Trust's Legal Provisions

Relating to IETF Documents

(http://trustee.ietf.org/license-info).

This version of this YANG module is part of RFC XXXX; see

the RFC itself for full legal notices.";

revision 2020-07-13 {

description

"Initial revision.";

reference

"RFC XXXX: Stream Control Transmission Protocol (SCTP)

Network Address Translation Support";

}

feature sctp-nat {

description

"This feature means that SCTP-specific variant of NAT

is supported. That is, avoid rewriting port numbers.";

reference

"Section 4.3 of RFC XXXX.";

}

augment "/nat:nat/nat:instances/nat:instance"

+ "/nat:policy/nat:timers" {

when "/nat:nat/nat:instances/nat:instance"

+ "/nat:policy/nat:transport-protocols"

+ "/nat:protocol-id = 132";

description

Stewart, et al. Expires 29 January 2021 [Page 40]

Internet-Draft SCTP NAT Support July 2020

"Extends NAT policy with a timeout for SCTP mapping

entries.";

leaf sctp-timeout {

type uint32;

units "seconds";

description

"SCTP inactivity timeout. That is, the time an SCTP

mapping entry will stay active without packets

traversing the NAT.";

}

}

augment "/nat:nat/nat:instances/nat:instance"

+ "/nat:mapping-table/nat:mapping-entry" {

when "nat:transport-protocol = 132";

if-feature "sctp-nat";

description

"Extends the mapping entry with SCTP specifics.";

leaf int-VTag {

type uint32;

description

"The Internal Verification Tag that the internal

host has chosen for this communication.";

}

leaf rem-VTag {

type uint32;

description

"The Remote Verification Tag that the remote

peer has chosen for this communication.";

}

}

}

<CODE ENDS>

9. Socket API Considerations

This section describes how the socket API defined in [RFC6458] is

extended to provide a way for the application to control NAT

friendliness.

Please note that this section is informational only.

A socket API implementation based on [RFC6458] is extended by

supporting one new read/write socket option.

Stewart, et al. Expires 29 January 2021 [Page 41]

Internet-Draft SCTP NAT Support July 2020

9.1. Get or Set the NAT Friendliness (SCTP\_NAT\_FRIENDLY)

This socket option uses the option\_level IPPROTO\_SCTP and the

option\_name SCTP\_NAT\_FRIENDLY. It can be used to enable/disable the

NAT friendliness for future associations and retrieve the value for

future and specific ones.

struct sctp\_assoc\_value {

sctp\_assoc\_t assoc\_id;

uint32\_t assoc\_value;

};

assoc\_id

This parameter is ignored for one-to-one style sockets. For one-

to-many style sockets the application may fill in an association

identifier or SCTP\_FUTURE\_ASSOC for this query. It is an error to

use SCTP\_{CURRENT|ALL}\_ASSOC in assoc\_id.

assoc\_value

A non-zero value indicates a NAT-friendly mode.

10. IANA Considerations

[NOTE to RFC-Editor: "RFCXXXX" is to be replaced by the RFC number

you assign this document.]

[NOTE to RFC-Editor: The requested values for the chunk type and the

chunk parameter types are tentative and to be confirmed by IANA.]

This document (RFCXXXX) is the reference for all registrations

described in this section. The requested changes are described

below.

10.1. New Chunk Flags for Two Existing Chunk Types

As defined in [RFC6096] two chunk flags have to be assigned by IANA

for the ERROR chunk. The requested value for the T bit is 0x01 and

for the M bit is 0x02.

This requires an update of the "ERROR Chunk Flags" registry for SCTP:

ERROR Chunk Flags

Stewart, et al. Expires 29 January 2021 [Page 42]

Internet-Draft SCTP NAT Support July 2020

+==================+=================+===========+

| Chunk Flag Value | Chunk Flag Name | Reference |

+==================+=================+===========+

| 0x01 | T bit | [RFCXXXX] |

+------------------+-----------------+-----------+

| 0x02 | M bit | [RFCXXXX] |

+------------------+-----------------+-----------+

| 0x04 | Unassigned | |

+------------------+-----------------+-----------+

| 0x08 | Unassigned | |

+------------------+-----------------+-----------+

| 0x10 | Unassigned | |

+------------------+-----------------+-----------+

| 0x20 | Unassigned | |

+------------------+-----------------+-----------+

| 0x40 | Unassigned | |

+------------------+-----------------+-----------+

| 0x80 | Unassigned | |

+------------------+-----------------+-----------+

Table 2

As defined in [RFC6096] one chunk flag has to be assigned by IANA for

the ABORT chunk. The requested value of the M bit is 0x02.

This requires an update of the "ABORT Chunk Flags" registry for SCTP:

ABORT Chunk Flags

Stewart, et al. Expires 29 January 2021 [Page 43]

Internet-Draft SCTP NAT Support July 2020

+==================+=================+===========+

| Chunk Flag Value | Chunk Flag Name | Reference |

+==================+=================+===========+

| 0x01 | T bit | [RFC4960] |

+------------------+-----------------+-----------+

| 0x02 | M bit | [RFCXXXX] |

+------------------+-----------------+-----------+

| 0x04 | Unassigned | |

+------------------+-----------------+-----------+

| 0x08 | Unassigned | |

+------------------+-----------------+-----------+

| 0x10 | Unassigned | |

+------------------+-----------------+-----------+

| 0x20 | Unassigned | |

+------------------+-----------------+-----------+

| 0x40 | Unassigned | |

+------------------+-----------------+-----------+

| 0x80 | Unassigned | |

+------------------+-----------------+-----------+

Table 3

10.2. Three New Error Causes

Three error causes have to be assigned by IANA. It is requested to

use the values given below.

This requires three additional lines in the "Error Cause Codes"

registry for SCTP:

Error Cause Codes

+=======+================================+===========+

| Value | Cause Code | Reference |

+=======+================================+===========+

| 176 | VTag and Port Number Collision | [RFCXXXX] |

+-------+--------------------------------+-----------+

| 177 | Missing State | [RFCXXXX] |

+-------+--------------------------------+-----------+

| 178 | Port Number Collision | [RFCXXXX] |

+-------+--------------------------------+-----------+

Table 4

Stewart, et al. Expires 29 January 2021 [Page 44]

Internet-Draft SCTP NAT Support July 2020

10.3. Two New Chunk Parameter Types

Two chunk parameter types have to be assigned by IANA. It is

requested to use the values given below. IANA should assign these

values from the pool of parameters with the upper two bits set to

'11'.

This requires two additional lines in the "Chunk Parameter Types"

registry for SCTP:

Chunk Parameter Types

+==========+==========================+===========+

| ID Value | Chunk Parameter Type | Reference |

+==========+==========================+===========+

| 49159 | Disable Restart (0xC007) | [RFCXXXX] |

+----------+--------------------------+-----------+

| 49160 | VTags (0xC008) | [RFCXXXX] |

+----------+--------------------------+-----------+

Table 5

10.4. One New URI

An URI in the "ns" subregistry within the "IETF XML" registry has to

be assigned by IANA ([RFC3688]):

URI: urn:ietf:params:xml:ns:yang:ietf-nat-sctp

Registrant Contact: The IESG.

XML: N/A; the requested URI is an XML namespace.

10.5. One New YANG Module

An YANG module in the "YANG Module Names" subregistry within the

"YANG Parameters" registry has to be assigned by IANA ([RFC6020]):

Name: ietf-nat-sctp

Namespace: urn:ietf:params:xml:ns:yang:ietf-nat-sctp

Maintained by IANA: N

Prefix: nat-sctp

Reference: RFCXXXX

11. Security Considerations

State maintenance within a NAT function is always a subject of

possible Denial Of Service attacks. This document recommends that at

a minimum a NAT function runs a timer on any SCTP state so that old

association state can be cleaned up.

Stewart, et al. Expires 29 January 2021 [Page 45]

Internet-Draft SCTP NAT Support July 2020

Generic issues related to address sharing are discussed in [RFC6269]

and apply to SCTP as well.

For SCTP endpoints not disabling the restart procedure, this document

does not add any additional security considerations to the ones given

in [RFC4960], [RFC4895], and [RFC5061].

SCTP endpoints disabling the restart procedure, should monitor the

status of all associations to mitigate resource exhaustion attacks by

establishing a lot of associations sharing the same IP addresses and

port numbers.

In any case, SCTP is protected by the verification tags and the usage

of [RFC4895] against off-path attackers.

For IP-level fragmentation and reassembly related issues see

[RFC4963].

The YANG module specified in this document defines a schema for data

that is designed to be accessed via network management protocols such

as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer

is the secure transport layer, and the mandatory-to-implement secure

transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer

is HTTPS, and the mandatory-to-implement secure transport is TLS

[RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341]

provides the means to restrict access for particular NETCONF or

RESTCONF users to a preconfigured subset of all available NETCONF or

RESTCONF protocol operations and content.

All data nodes defined in the YANG module that can be created,

modified, and deleted (i.e., config true, which is the default) are

considered sensitive. Write operations (e.g., edit-config) applied

to these data nodes without proper protection can negatively affect

network operations. An attacker who is able to access the SCTP NAT

function can undertake various attacks, such as:

\* Setting a low timeout for SCTP mapping entries to cause failures

to deliver incoming SCTP packets.

\* Instantiating mapping entries to cause NAT collision.

12. Normative References

Stewart, et al. Expires 29 January 2021 [Page 46]

Internet-Draft SCTP NAT Support July 2020

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119,

DOI 10.17487/RFC2119, March 1997,

<https://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,

DOI 10.17487/RFC3688, January 2004,

<https://www.rfc-editor.org/info/rfc3688>.

[RFC4895] Tuexen, M., Stewart, R., Lei, P., and E. Rescorla,

"Authenticated Chunks for the Stream Control Transmission

Protocol (SCTP)", RFC 4895, DOI 10.17487/RFC4895, August

2007, <https://www.rfc-editor.org/info/rfc4895>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol",

RFC 4960, DOI 10.17487/RFC4960, September 2007,

<https://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,

<https://www.rfc-editor.org/info/rfc5061>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for

the Network Configuration Protocol (NETCONF)", RFC 6020,

DOI 10.17487/RFC6020, October 2010,

<https://www.rfc-editor.org/info/rfc6020>.

[RFC6096] Tuexen, M. and R. Stewart, "Stream Control Transmission

Protocol (SCTP) Chunk Flags Registration", RFC 6096,

DOI 10.17487/RFC6096, January 2011,

<https://www.rfc-editor.org/info/rfc6096>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,

and A. Bierman, Ed., "Network Configuration Protocol

(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,

<https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure

Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,

<https://www.rfc-editor.org/info/rfc6242>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF

Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,

<https://www.rfc-editor.org/info/rfc8040>.

Stewart, et al. Expires 29 January 2021 [Page 47]

Internet-Draft SCTP NAT Support July 2020

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC

2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,

May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration

Access Control Model", STD 91, RFC 8341,

DOI 10.17487/RFC8341, March 2018,

<https://www.rfc-editor.org/info/rfc8341>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol

Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,

<https://www.rfc-editor.org/info/rfc8446>.

[RFC8512] Boucadair, M., Ed., Sivakumar, S., Jacquenet, C.,

Vinapamula, S., and Q. Wu, "A YANG Module for Network

Address Translation (NAT) and Network Prefix Translation

(NPT)", RFC 8512, DOI 10.17487/RFC8512, January 2019,

<https://www.rfc-editor.org/info/rfc8512>.

13. Informative References

[DOI\_10.1145\_1496091.1496095]

Hayes, D., But, J., and G. Armitage, "Issues with network

address translation for SCTP", ACM SIGCOMM Computer

Communication Review Vol. 39, pp. 23-33,

DOI 10.1145/1496091.1496095, December 2008,

<https://doi.org/10.1145/1496091.1496095>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7,

RFC 793, DOI 10.17487/RFC0793, September 1981,

<https://www.rfc-editor.org/info/rfc793>.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network

Address Translator (Traditional NAT)", RFC 3022,

DOI 10.17487/RFC3022, January 2001,

<https://www.rfc-editor.org/info/rfc3022>.

[RFC4963] Heffner, J., Mathis, M., and B. Chandler, "IPv4 Reassembly

Errors at High Data Rates", RFC 4963,

DOI 10.17487/RFC4963, July 2007,

<https://www.rfc-editor.org/info/rfc4963>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful

NAT64: Network Address and Protocol Translation from IPv6

Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146,

April 2011, <https://www.rfc-editor.org/info/rfc6146>.

Stewart, et al. Expires 29 January 2021 [Page 48]

Internet-Draft SCTP NAT Support July 2020

[RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and

P. Roberts, "Issues with IP Address Sharing", RFC 6269,

DOI 10.17487/RFC6269, June 2011,

<https://www.rfc-editor.org/info/rfc6269>.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-

Stack Lite Broadband Deployments Following IPv4

Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011,

<https://www.rfc-editor.org/info/rfc6333>.

[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,

<https://www.rfc-editor.org/info/rfc6458>.

[RFC6890] Cotton, M., Vegoda, L., Bonica, R., Ed., and B. Haberman,

"Special-Purpose IP Address Registries", BCP 153,

RFC 6890, DOI 10.17487/RFC6890, April 2013,

<https://www.rfc-editor.org/info/rfc6890>.

[RFC6951] Tuexen, M. and R. Stewart, "UDP Encapsulation of Stream

Control Transmission Protocol (SCTP) Packets for End-Host

to End-Host Communication", RFC 6951,

DOI 10.17487/RFC6951, May 2013,

<https://www.rfc-editor.org/info/rfc6951>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language",

RFC 7950, DOI 10.17487/RFC7950, August 2016,

<https://www.rfc-editor.org/info/rfc7950>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams",

BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018,

<https://www.rfc-editor.org/info/rfc8340>.

Acknowledgments

The authors wish to thank Mohamed Boucadair, Gorry Fairhurst, Bryan

Ford, David Hayes, Alfred Hines, Karen E. E. Nielsen, Henning

Peters, Maksim Proshin, Timo Voelker, Dan Wing, and Qiaobing Xie for

their invaluable comments.

In addition, the authors wish to thank David Hayes, Jason But, and

Grenville Armitage, the authors of [DOI\_10.1145\_1496091.1496095], for

their suggestions.

The authors also wish to thank Mohamed Boucadair for contributing the

text related to the YANG module.

Stewart, et al. Expires 29 January 2021 [Page 49]

Internet-Draft SCTP NAT Support July 2020

Authors' Addresses

Randall R. Stewart

Netflix, Inc.

Chapin, SC 29036

United States of America

Email: randall@lakerest.net

Michael Tüxen

Münster University of Applied Sciences

Stegerwaldstrasse 39

48565 Steinfurt

Germany

Email: tuexen@fh-muenster.de

Irene Rüngeler

Münster University of Applied Sciences

Stegerwaldstrasse 39

48565 Steinfurt

Germany

Email: i.ruengeler@fh-muenster.de

Stewart, et al. Expires 29 January 2021 [Page 50]