IPv6 Operations
Internet-Draft
Updates: 6145 (if approved)
Intended status: Standards Track
Expires: July 12, 2015

T. Anderson
Redpill Linpro

January 08, 2015

Explicit Address Mappings for Stateless IP/ICMP Translation
draft-anderson-v6ops-siit-eam-03

Abstract

This document extends the Stateless IP/ICMP Translation Algorithm (SIIT) with an Explicit Address Mapping (EAM) algorithm, and formally updates RFC 6145. The EAM algorithm facilitates stateless IP/ICMP translation between arbitrary (non-IPv4-translatable) IPv6 endpoints and IPv4.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Terminology 3
2. Problem Statement 3
3. Explicit Address Mapping Algorithm 5
3.1. Explicit Address Mapping Table 5
3.2. Explicit Address Mapping Specification 6
3.3. IP Address Translation Procedure 6
3.3.1. Address Translation Steps: IPv4 to IPv6 7
3.3.2. Address Translation Steps: IPv6 to IPv4 7
4. Lack of Checksum Neutrality 8
5. Security Considerations 8
6. IANA Considerations 8
7. Acknowledgements 8
8. References 8
8.1. Normative References 8
8.2. Informative References 9
Appendix A. Use Cases 9
A.1. 464XLAT 9
A.2. IVI 10
A.3. SIIT-DC 10
Appendix B. Example IP Address Translations 11
Author's Address 12

1. Introduction

The Stateless IP/ICMP Translation Algorithm (SIIT) [RFC6145] specifies that when translating IPv4 addresses to IPv6 and vice versa, all addresses must be translated using the algorithm specified in [RFC6052]. This document specifies an alternative to the [RFC6052] algorithm, where IP addresses are translated according to a table of Explicit Address Mappings configured on the stateless translator. This removes the previous constraint that IPv6 nodes that communicate with IPv4 nodes through SIIT must be configured with IPv4-translatable IPv6 addresses.

The Explicit Address Mapping Table does not replace [RFC6052]. For most use cases, it is expected that both algorithms are used in concert. The Explicit Address Mapping algorithm is used only when a mapping matching the address to be translated exists. If no matching mapping exists, the [RFC6052] algorithm will be used instead. Thus, when translating an individual IP packet, an SIIT implementation might translate one of the two IP address fields according to an EAM, while the other IP address field is translated according to [RFC6052].

1.1. Terminology

This document makes use of the following terms:

EAM
An Explicit Address Mapping, as specified in Section 3.2.

EAMT
The Explicit Address Mapping Table, as specified in Section 3.1.

SIIT
The Stateless IP/ICMP Translation algorithm, as specified in [RFC6145].

IPv4-converted IPv6 addresses
As defined in Section 1.3 of [RFC6052].

IPv4-translatable IPv6 addresses
As defined in Section 1.3 of [RFC6052].

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

2. Problem Statement

Section 3.2.1 of [RFC6144] notes that "stateless translation mechanisms typically put constraints on what IPv6 addresses can be assigned to IPv6 nodes that want to communicate with IPv4 destinations using an algorithmic mapping". In practice, this means that the IPv6 nodes must be configured with IPv4-translatable IPv6 addresses. For the reasons discussed below, some environments may find that the use of IPv4-translatable IPv6 addresses is not desired or even possible.

Limited availability:
The number of IPv4-translatable IPv6 addresses available to an operator is equal to the number of IPv4 addresses he assigns to

the SIIT function. IPv4 addresses are scarce, and as a result an operator might not have enough IPv4-translatable IPv6 addresses to number his entire IPv6 infrastructure.

Restricted format:
IPv4-translatable IPv6 addresses must conform to the format specified in Section 2.2 of [RFC6052]. This format is not compatible with other common IPv6 address formats, such as the EUI-64 based IPv6 address format used by IPv6 Stateless Address Autoconfiguration [RFC4862].

An operator could overcome the above two problems by building an IPv6 network using regular (non-IPv4-translatable) IPv6 addresses, and assign IPv4-translatable IPv6 addresses as secondary addresses on the nodes that want to communicate with IPv4 nodes through SIIT only. However, doing so may result in a new set of undesired properties:

Routing complexity:
The IPv4-translatable IPv6 addresses must be routed throughout the IPv6 network separately from the primary (non-IPv4-translatable) IPv6 addresses used by the nodes. It might be impossible to aggregate these routes, as two adjacent IPv4-translatable IPv6 addresses might not be assigned to two adjacent IPv6 nodes. As a result, in order to support SIIT, the IPv6 network might need to carry a large number of extraneous routes. These routes must be separately injected into the IPv6 routing topology somehow. Any intermediate devices in the IPv6 network such as a firewall might require special configuration in order to treat the IPv4-translatable IPv6 address the same as the primary IPv6 address, for example by requiring that any ACL entries involving the primary IPv6 address of a node must be duplicated.

Operational complexity:

> The IPv4-translatable IPv6 addresses must not only be assigned to the IPv6 nodes participating in SIIT; all applications and services on those nodes must also be configured to use them. For example, if the IPv6 node is a load balancer, it might require a separate Virtual Server definition using the IPv4-translatable IPv6 address in addition to one using the service's primary IPv6 address. A web server might require specific configuration to listen for connections on both the IPv4-translatable and the primary IPv6 address. A High-Availability cluster service must be set up to fail over both addresses between cluster nodes, and depending on how the IPv6 network learns the location of the IPv4-translatable IPv6 address, the fail-over mechanism used for the two addresses might be completely different. Service monitoring must be done for both the IPv4-translatable and the primary IPv6 address, and any trouble-shooting procedures must be extended to involve both addresses.

In short, the use of IPv4-translatable IPv6 addresses in parallel with regular IPv6 addresses is in many ways analogous to the use of Dual Stack [RFC4213]. While no actual IPv4 packets are used, the IPv4-translatable IPv6 addresses creates a secondary "stack" in the infrastructure that must be treated and operated separately from the primary one. This increases the complexity of the overall infrastructure, in turn increasing operational overhead, and reducing reliability. An operator who for such reasons finds the use Dual Stack unappealing, might feel the same way about using SIIT with IPv4-translatable IPv6 addresses.

3. Explicit Address Mapping Algorithm

This normative section defines the EAM algorithm. SIIT implementations are REQUIRED to support the specifications herein.

3.1. Explicit Address Mapping Table

An SIIT implementation MUST include an Explicit Address Mapping Table (EAMT). By default, the EAMT SHOULD be empty. The operator MUST be able to populate the EAMT using the implementation's normal configuration interfaces. The implementation MAY additionally support other ways of populating the EAMT.

The EAMT consists of the following columns:

> IPv4 Prefix
>
> IPv6 Prefix

SIIT implementations MAY include other columns in order to support proprietary extensions to the EAM algorithm.

Throughout this document, figures representing the EAMT contain an Index column using the pound sign as the header. This column is not a required part of this specification; it is included only as a convenience to the reader.

3.2. Explicit Address Mapping Specification

An EAM consists of an IPv4 Prefix and an IPv6 Prefix. The prefix length MAY be omitted, in which case the implementation MUST assume it to be 32 for IPv4 and 128 for IPv6. Figure 1 illustrates an EAMT containing examples of valid EAMs.

Example EAMT

#	IPv4 Prefix	IPv6 Prefix
1	192.0.2.1	2001:db8:aaaa::
2	192.0.2.2/32	2001:db8:bbbb::b/128
3	192.0.2.16/28	2001:db8:cccc::/124
4	192.0.2.128/26	2001:db8:dddd::/64
5	192.0.2.192/31	64:ff9b::/127

Figure 1

An EAM's IPv4 Prefix value MUST have an identical or smaller number of suffix bits than its corresponding IPv6 Prefix value.

Overlapping EAMs SHOULD be considered an error, and attempts to insert them into the EAMT SHOULD be blocked. The behaviour of an SIIT implementation when overlapping EAMs are present in the EAMT is left undefined.

When translating a packet between IPv4 and IPv6, an SIIT implementation MUST individually translate each IP address it encounters in the packet's IP headers (including any IP headers contained within ICMP errors) according to Section 3.3.

3.3. IP Address Translation Procedure

This section describes step-by-step how an SIIT implementation translates addresses between IPv4 and IPv6. Only the outcome of the algorithm described should be considered normative, that is, an SIIT implementation MAY implement the exact procedure differently than

what is described here, but the outcome of the algorithm MUST be the same.

For concrete examples of IP addresses translations, refer to Appendix B.

3.3.1. Address Translation Steps: IPv4 to IPv6

1. The EAMT is searched for an EAM entry containing an IPv4 Prefix identical to that of the IPv4 address being translated. The IPv4 Prefix and IPv6 Prefix values of the EAM entry found is from now on referred to as EAM4 and EAM6, respectively.

2. If no matching EAM entry is found, the EAM algorithm is aborted. The SIIT implementation MUST proceed to translate the address in accordance with [RFC6145] (and its updates).

3. The prefix bits of EAM4 are removed from IPv4 address being translated. The remaining suffix bits from the IPv4 address being translated are stored in a temporary buffer.

4. The prefix bits of EAM6 are prepended to the temporary buffer.

5. If the temporary buffer at this point does not contain a 128-bit value, it is padded with trailing zeroes so that it reaches a length of 128 bits.

6. The contents of the temporary buffer is the translated IPv6 address.

3.3.2. Address Translation Steps: IPv6 to IPv4

1. The EAMT is searched for an EAM entry containing an IPv6 Prefix identical to that of the IPv6 address being translated. The IPv4 Prefix and IPv6 Prefix values of the EAM entry found is from now on referred to as EAM4 and EAM6, respectively.

2. If no matching EAM entry is found, the EAM algorithm is aborted. The SIIT implementation MUST proceed to translate the address in accordance with [RFC6145] (and its updates).

3. The prefix bits of EAM6 are removed from IPv6 address being translated. The remaining suffix bits from the IPv6 address being translated are stored in a temporary buffer.

4. The prefix bits of EAM4 are prepended to the temporary buffer.

5. If the temporary buffer at this point does not contain a 32-bit value, any trailing bits are discarded so that the buffer is reduced to a length of 32 bits.

6. The contents of the temporary buffer is the translated IPv4 address.

4. Lack of Checksum Neutrality

When one or both of the address fields in an IP/ICMP packet are translated according to EAM, the translation can not be relied upon to be checksum neutral, even if the well-known prefix 64:ff9b::/96 is used. This consideration is discussed in more detail in Section 4.1 of [RFC6052].

5. Security Considerations

The EAM algorithm does not introduce any new security issues beyond those that are already discussed in Section 7 of [RFC6145].

6. IANA Considerations

This draft makes no request of the IANA. The RFC Editor may remove this section prior to publication.

7. Acknowledgements

This document was conceived due to comments made by Dave Thaler in the v6ops session at IETF 91 as well as e-mail discussions between Fred Baker and the author.

Valuable reviews, suggestions, and other feedback was given by Cameron Byrne, Brian E Carpenter, Alberto Leiva, and Andrew Yourtchenko.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

[RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.

8.2. Informative References

[I-D.anderson-v6ops-siit-dc]
tore, t., "SIIT-DC: Stateless IP/ICMP Translation for IPv6 Data Centre Environments", draft-anderson-v6ops-siit-dc-01 (work in progress), October 2014.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.

[RFC6219] Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, "The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition", RFC 6219, May 2011.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, April 2013.

[RFC7335] Byrne, C., "IPv4 Service Continuity Prefix", RFC 7335, August 2014.

Appendix A. Use Cases

The following subsections lists some use cases that at the time of writing leverage SIIT with the EAM algorithm.

A.1. 464XLAT

When the CLAT component in the 464XLAT [RFC6877] architecture does not have a dedicated IPv6 prefix assigned, it may instead use "one interface IPv6 address that is claimed by the CLAT". This IPv6 address might not be IPv4-translatable. If this is the case, the CLAT essentially implements the EAM algorithm using an EAMT as follows (assuming the CLAT's IPv4 address is picked from the IPv4 Service Continuity Prefix [RFC7335]):

Example EAMT for an 464XLAT CLAT

#	IPv4 Prefix	IPv6 Prefix
1	192.0.0.1/32	CLAT_claimed_IPv6_address/128

Figure 2

In this particular use case, the EAM algorithm is used to translate IPv6 destination addresses to IPv4, and conversely, IPv4 source addresses to IPv6. Other addresses are translated using [RFC6052]. Note that this is the exact opposite of the SIIT-DC use case (Appendix A.3).

A.2. IVI

IVI [RFC6219] describes a stateless translation model that embeds IPv4 addresses in a 40-bit translation prefix where bits 33-40 are required to be 1. The embedded IPv4 address is located in bits 41-72 of the IPv6 address. Bits 73-128 are required to be 0.

The location of the eight least significant IPv4 address bits makes the IVI address mapping differ from [RFC6052].

Example EAMT for IVI

#	IPv4 Prefix	IPv6 Prefix
1	0.0.0.0/0	2001:db8:ff00::/40

Figure 3

In this particular use case, all addresses are translated according to the EAM algorithm. In other words, [RFC6052] mapping is not used at all.

A.3. SIIT-DC

SIIT-DC [I-D.anderson-v6ops-siit-dc] describes the use of SIIT to facilitate connectivity from the IPv4 Internet to services hosted in an IPv6-only data centre. In order to avoid the constraints relating to the use of IPv4-translatable IPv6 addresses discussed in Section 2 the stateless IPv4/IPv6 translators are provisioned with an EAMT containing one entry per IPv6-only service that are to be made available from the IPv4 Internet, for example (assuming 2001:db8:aaaa::1 and 2001:db8:bbbb::1 are assigned to load balancers or servers that provides the IPv6-only services in question):

Example EAMT for SIIT-DC

#	IPv4 Prefix	IPv6 Prefix
1	192.0.2.1/32	2001:db8:aaaa::1/128
2	192.0.2.2/32	2001:db8:bbbb::1/128

Figure 4

In this particular use case, the EAM algorithm is used to translate IPv4 destination addresses to IPv6, and conversely, IPv6 source addresses to IPv4. Other addresses are translated using [RFC6052]. Note that this is the exact opposite of the 464XLAT use case (Appendix A.1).

Appendix B. Example IP Address Translations

Figure 5 demonstrates how a set of example IP addresses are translated given the example EAMT in Figure 1. Implementors may use the examples given to develop test cases to validate correct operation. Note that the address translations are bidirectional, so a single row in the table describes two address translations: IPv4 to IPv6, and IPv6 to IPv4.

It is also assumed that the [RFC6052] translation prefix is configured to be 64:ff9b::/96.

Example IP Address Translations

IPv4 Address	IPv6 Address	Comment
192.0.2.1	2001:db8:aaaa::	According to EAM #1
192.0.2.2	2001:db8:bbbb::b	According to EAM #2
192.0.2.16	2001:db8:cccc::	According to EAM #3
192.0.2.24	2001:db8:cccc::8	According to EAM #3
192.0.2.31	2001:db8:cccc::f	According to EAM #3
192.0.2.128	2001:db8:dddd::	According to EAM #4
192.0.2.152	2001:db8:dddd:0:6000::	According to EAM #4
192.0.2.183	2001:db8:dddd:0:dc00::	According to EAM #4
192.0.2.191	2001:db8:dddd:0:fc00::	According to EAM #4
192.0.2.193	64:ff9b::1	According to EAM #5
192.0.2.200	64:ff9b::c000:2c8	According to RFC 6052

Figure 5

Author's Address

Tore Anderson
Redpill Linpro
Vitaminveien 1A
0485 Oslo
Norway

Phone: +47 959 31 212
Email: tore@redpill-linpro.com
URI: http://www.redpill-linpro.com