IPv6 Operations
Internet-Draft
Intended status: Standards Track
Expires: July 29, 2015

T. Anderson
Redpill Linpro

January 25, 2015

SIIT-DC: Dual Translation Mode
draft-ietf-v6ops-siit-dc-2xlat-00

Abstract

This document describes an extension of the Stateless IP/ICMP Translation for IPv6 Data Centre Environments architecture (SIIT-DC), which allows applications, protocols, or nodes that are incompatible with IPv6, SIIT-DC and/or Network Address Translation in general to operate correctly in an SIIT-DC environment. This is accomplished by introducing a new component called an Edge Translator, which reverses the translations made by an SIIT-DC Gateway. The application or device is thus provided with seemingly native IPv4 connectivity.

The reader is expected to be familiar with the SIIT-DC architecture described in I-D.ietf-v6ops-siit-dc.

Table of Contents

1. Introduction

SIIT-DC [I-D.ietf-v6ops-siit-dc] describes an architecture where IPv4-only users can access IPv6-only services through a stateless translator called an SIIT-DC Gateway. This approach has certain limitations, however. In particular, the following cases will work poorly or not at all:

- Application protocols that do not support NAT (i.e., the lack of end-to-end transparency of IP addresses).

- Devices which cannot connect to IPv6 networks at all, or which can only connect such networks if they also provide IPv4 connectivity (i.e., dual-stacked networks).

- Application software which makes use of legacy IPv4-only APIs, or otherwise makes assumptions that IPv4 connectivity is available.

By extending the SIIT-DC architecture with a new component called an Edge Translator (ET), all of the above can be made to work correctly in an otherwise IPv6-only network environment using SIIT-DC.

The purpose of the Edge Translator is to reverse the IPv4-to-IPv6 packet translations previously done by the SIIT-DC Gateway for traffic arriving from IPv4 clients and forward this as "native" IPv4 to the application software or device. In the reverse direction, IPv4 packets transmitted by the application software or device is intercepted by the Edge Translator, which will translate them to IPv6 before they are forwarded to the SIIT-DC Gateway, which in turn will reverse the translations and forward them to the IPv4 End User. In short, the device or application software is provided with "virtual" IPv4 Internet connectivity that retains end-to-end transparency for the IPv4 addresses.

2. Terminology

This document makes use of the following terms:

Edge Translator (ET)
: A device or logical function that provides "native" IPv4 connectivity to IPv4-only devices or application software. It is very similar in function to an SIIT-DC Gateway, but is typically located close to the IPv4-only component(s) it is supporting rather than on the network border.

IPv4 Service Address
: A public IPv4 address with which IPv4-only clients will communicate. This communication will be translated to IPv6 by the SIIT-DC Gateway and back to IPv4 again by the Edge Translator.

SIIT-DC Gateway
: A device or a logical function that translates between IPv4 and IPv6 in accordance with [I-D.ietf-v6ops-siit-dc].

Static Address Mapping
: A bi-directional mapping between an IPv4 Service Address and an IPv6 Service Address configured in the SIIT-DC Gateway. When translating between IPv4 and IPv6, the SIIT-DC Gateway changes the address fields in the translated packet's IP header according to any matching Static Address Mapping.

Translation Prefix
: An IPv6 prefix into which the entire IPv4 address space is mapped. This prefix is routed to the SIIT-DC Gateway's IPv6 interface. It is either an Network-Specific Prefix or a Well-Known Prefix as specified in [RFC6052]. When translating between IPv4 and IPv6,

the SIIT-DC Gateway will prepend or strip the Translation Prefix from the address fields in the translated packet's IP header, unless a Static Address Mapping exists for the IP address in question.

XLAT
Used in figures to indicate where the Stateless IP/ICMP Translation [RFC6145] algorithm is used to translate IPv4 packets to IPv6 and vice versa.

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].

3. Edge Translator Description

An Edge Translator (ET) is at its core an implementation of the Stateless IP/ICMP Translation algorithm [RFC6145], with the Static Address Mapping extension described in Section 5.2 of [I-D.ietf-v6ops-siit-dc]. It provides virtual IPv4 connectivity for application software or devices which require this to operate correctly in an SIIT-DC environment.

Inbound IPv4 packets destined for an IPv4 Service Address is first translated to IPv6 by an SIIT-DC Gateway. The resulting IPv6 packets are subsequently forwarded to the ET handling the IPv6 Service Address they are addressed to. The ET then translates them back to IPv4 before forwarding them to the IPv4 application software or device. In the other direction, the exact same translations happen, only in reverse. This process provides end-to-end transparency of IPv4 addresses.

An ET may handle an arbitrary number of IPv4 Service Addresses. All the Static Address Mappings configured in the SIIT-DC Gateway(s) that involve the IPv4 Service Addresses handled by an ET MUST be duplicated in that ET's configuration.

An ET may be implemented in two distinct ways; as a software-based service residing inside an otherwise IPv6-only host, or as a network-based service that provides an isolated IPv4 network segment to which devices which require IPv4 can connect. In both cases native IPv6 connectivity may be provided simultaneously with the virtual IPv4 connectivity. Thus, dual-stack connectivity is facilitated in case the device or application software support it.

The choice between a host- or network-based ET is made on a per-service or -device basis. An arbitrary number of each type of ET may co-exist in an SIIT-DC architecture.

This section describes the different approaches and discusses which approach fits best for the various use cases.

3.1. Host-Based Edge Translator

Overview of a Host-based Edge Translator

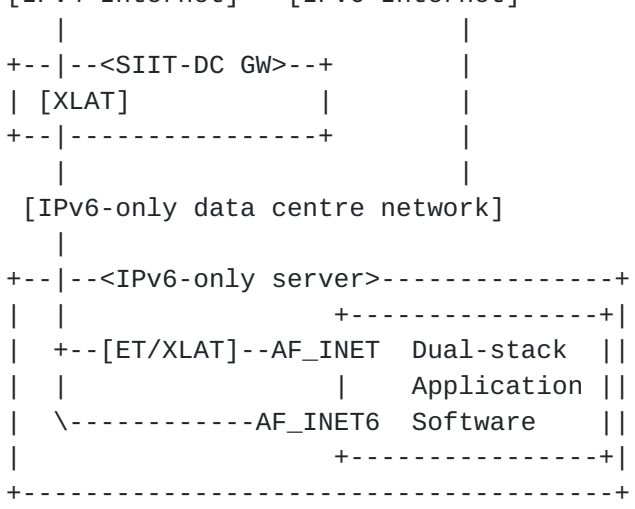

Figure 1

A host-based Edge Translator is typically implemented as a logical software function that runs inside the operating system of a host or server. It provides software applications running on the same host with IPv4 connectivity. The IPv4 Service Address it handles is considered local, allowing application software running on the same host to use traditional IPv4-only API calls, e.g., to create AF_INET sockets that listens for and accepts incoming connections to its IPv4 Service Address. An ET could accomplish this by creating an virtual network adapter to which it assigns the IPv4 Service Address and points a default IPv4 route.

As shown in Figure 1, if the application software supports dual-stack operation, IPv6 clients will be able to communicate with it directly using native IPv6. Neither the SIIT-DC Gateway nor the ET will intercept this communication. Support for IPv6 in the application software is however not a requirement; the application software may opt not to establish any IPv6 sockets. Foregoing IPv6 in this manner will simply preclude connectivity to the service from IPv6-only clients; connectivity to the service from IPv4 clients (through the SIIT-DC Gateway) will work in the exact same manner in both cases.

The ET requires a dedicated IPv6 Service Address for each IPv4 Service Address it has configured. The IPv6 network must forward

traffic to these IPv6 Service Addresses to the host, whose operating system must in turn forward them to the ET. This document does not explore the multitude of ways this could be accomplished, however considering that the IPv6 protocol is designed for having multiple addresses assigned to a single node, one particularly straight-forward way would be to assign the ET's IPv6 Service Addresses as secondary IPv6 addresses on the host itself so that it the upstream router learns of their location using the IPv6 Neighbor Discovery Protocol [RFC4861].

3.2. Network-Based Edge Translator

Overview of a Basic Network-based Edge Translator

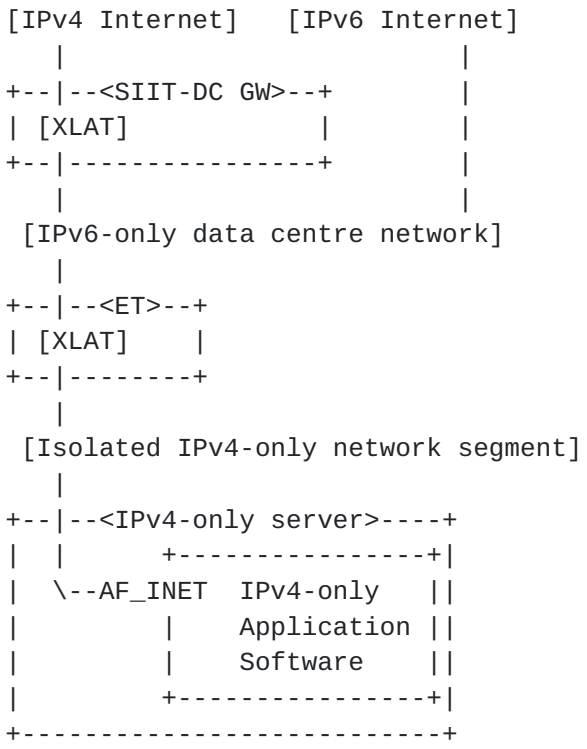

Figure 2

A network-based Edge Translator performs the exact same as a host-based ET does, only that instead of assigning the IPv4 Service Addresses to an internal-only virtual network adapter, traffic destined for them are forwarded onto a network segment to which hosts that require IPv4 connectivity connect to. The ET also functions as the default IPv4 router for the hosts on this network segment.

Each host on the IPv4 network segment must acquire and assign an IPv4 Service Address to a local network interface. This document does not attempt to explore all the various methods by which this can be

accomplished, however one relatively straight-forward possibility would be to ensure the IPv4 Service Address(es) can be enclosed in an IPv4 prefix. The ET will then claim one address in this prefix for itself (used as the IPv4 default router address), and could assign the IPv4 Service Address(es) to the host(s) using DHCPv4. For example, if the IPv4 Service Addresses are 192.0.2.26 and 192.0.2.27, the ET would configure the address 192.0.2.25/29 on its IPv4-facing interface and would add the two IPv4 Service Addresses to its DHCPv4 pool.

One disadvantage of this method is that IPv4 communication between the IPv4 hosts and other services made available through SIIT-DC using the method described in Section 6 becomes impossible, if those other services are assigned IPv4 Service Addresses that also are covered by the same IPv4 prefix (e.g., 192.0.2.28). This is because the IPv4 nodes will mistakenly believe they have an on-link route to the entire prefix, and attempt to resolve the addresses using ARP (instead of forwarding them to the ET for translation to IPv6). This problem could however be overcome by avoiding assigning IPv4 Service Addresses which overlaps with an IPv4 prefix handled by an ET (at the expense of wasting some potential IPv4 Service Addresses), or by ensuring that they are only assigned to services which do not need to communicate with the IPv4 host(s) behind the ET.

Another way to avoid the problem is to use a private unrouted IPv4 network that does not encompass the IPv4 Service Addresses as the IPv4, and instead assign the IPv4 Service Addresses as secondary addresses on the servers. The ET must then route each IPv4 Service Address to its respective server using the server's private on-link IPv4 address as the next-hop. This approach would ensure there are no overlaps, but on the other hand it would preclude the use of DHCPv4 for assigning the IPv4 Service Addresses, as well as create a need to ensure that the IPv4 application software is selecting the IPv4 Service Address (as opposed to its private on-link IPv4 address) as its source address when initiating outbound connections.

The basic ET illustrated in Figure 2 establishes an IPv4-only network segment behind itself. This is fine if the devices it provides IPv4 access have no support for IPv6 whatsoever; however if they are dual-stack capable, it is would not be ideal to take away their IPv6 connectivity. While it is recommended to use a host-based ET in this case, appropriate implementations of a host-based ET might not be available for every device. If the application protocol does not work correctly in a NAT environment, standard SIIT-DC cannot be used either. Thus, a network-based ET is the only solution.

The operator could avoid breaking the hosts' IPv4 connectivity by connecting the ET's IPv4 and IPv6 interfaces to the same network

segment, or by using a single dual-stacked interface instead. The latter alternative is shown in Figure 3. This could be thought of as an "ET on a stick". IPv6 traffic between the network and the hosts will bypass the ET entirely. IPv4 traffic from the hosts will be routed directly to the ET (because it's their default IPv4 router), and translated to IPv6 before its being transmitted to the upstream default IPv6 router. The ET could attract inbound traffic to its IPv6 Service Addresses by responding to the upstream router's IPv6 Neighbor Discovery [RFC4861] messages for them.

A Network-based Edge Translator "on a stick"

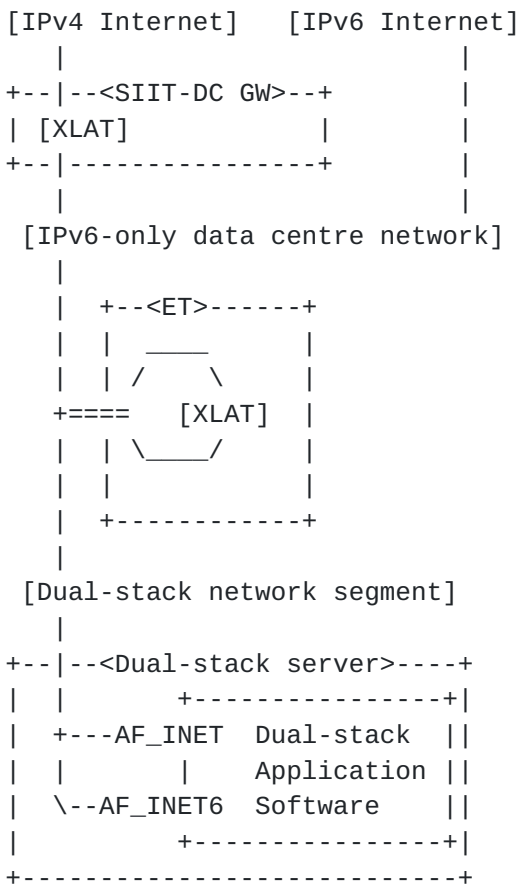

Figure 3

Yet another variation would be to implement the ET so that it transparently passes IPv6 traffic between its downstream and upstream network ports unmodified, e.g., using Layer-2 bridging. Packets sent to its own IPv6 Service Addresses from the upstream network are intercepted (e.g, by responding to IPv6 Neighbor Discovery [RFC4861] messages for them) and routed through the translation function, and forwarded out its downstream interface. The downstream network segment is thus becomes dual-stacked. This model is shown in Figure 4.

A Transparent Network-based Edge Translator

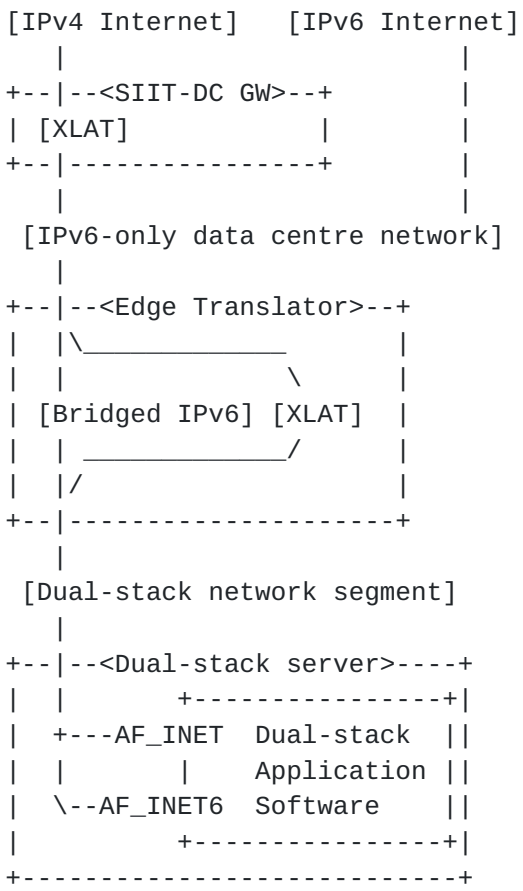

Figure 4

4. Detailed Topology Example

The following figure shows how an application (that is presumably incompatible with standard SIIT-DC) is being made available to the IPv4 Internet on the IPv4 address 192.0.2.4. The application will be able to know that this is its local address and thus be able to provide correct references to it in application payload.

The figure also shows how the same application is available over IPv6 on its IPv6 Service Address 2001:db8:12:34::3. This is included in order to illustrate how native IPv6 connectivity is not impacted by the Edge Translator, and also to illustrate how the address assigned to the ET (2001:db8:12:34::4) is separate from the primary IPv6 address of the server. It is however important to note that the application in question does not have to be dual-stack capable at all. IPv4-only applications would also be able to operate behind an ET in the exact same manner.

Note that the figure below could be considered a more detailed view of Customer A's FTP server from the example topology figure in Appendix A of [I-D.ietf-v6ops-siit-dc]. Both figures intentionally use the exact same example IP addresses and prefixes.

SIIT-DC Host Architecture with Edge Translation

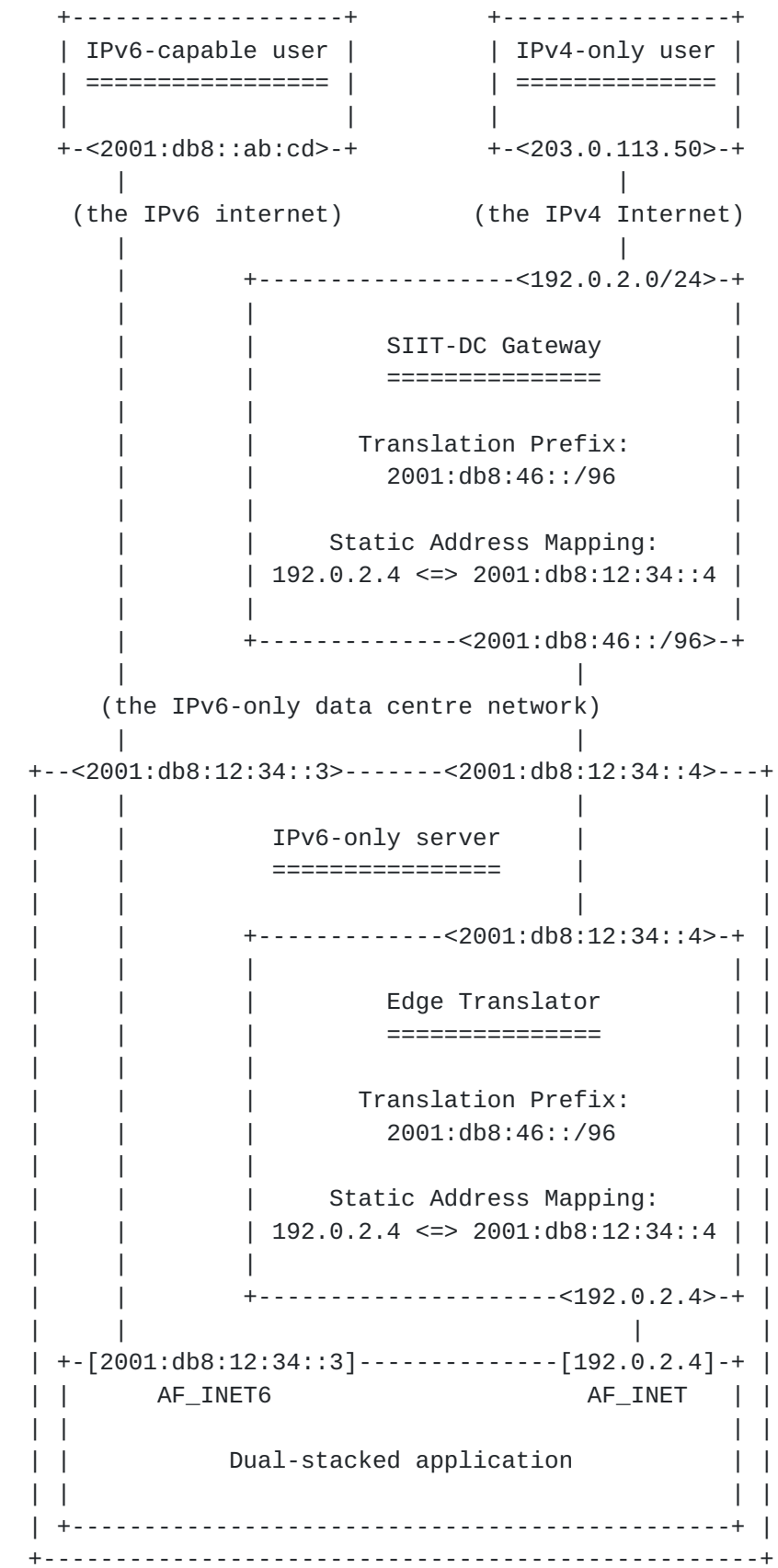
IPv6-capable user
2001:db8::ab:cd
IPv4-only user
203.0.113.50
(the IPv6 internet)
(the IPv4 Internet)
192.0.2.0/24
SIIT-DC Gateway
Translation Prefix:
2001:db8:46::/96
Static Address Mapping:
192.0.2.4 <=> 2001:db8:12:34::4
2001:db8:46::/96
(the IPv6-only data centre network)
2001:db8:12:34::3
2001:db8:12:34::4
IPv6-only server
2001:db8:12:34::4
Edge Translator
Translation Prefix:
2001:db8:46::/96
Static Address Mapping:
192.0.2.4 <=> 2001:db8:12:34::4
192.0.2.4
[2001:db8:12:34::3]
[192.0.2.4]
AF_INET6
AF_INET
Dual-stacked application

Figure 5

5. Deployment Considerations

5.1. IPv6 Path MTU

The IPv6 Path MTU between the Edge Translator and the SIIT-DC Gateway will typically be larger than the default value defined in Section 4 of [RFC6145] (1280), as it will typically contained within a single administrative domain. Therefore, it is recommended that the IPv6 Path MTU configured in the ET is raised accordingly. It is RECOMMENDED that the ET and the SIIT-DC Gateway use identical configured IPv6 Path MTU values.

5.2. IPv4 MTU

In order to avoid IPv6 fragmentation, an Edge Translator should ensure that the IPv4 MTU used by applications or hosts is equal to the configured IPv6 Path MTU - 20, so that an maximum-sized IPv4 packet can fit in an unfragmented IPv6 packet. This ensures that the application may do its part in avoiding IP-level fragmentation from occurring, e.g., by segmenting/fragmenting outbound packets at the application layer, and advertising the maximum size its peer may use for inbound packets (e.g., through the use of the TCP MSS option).

A host-based ET could accomplish this by configuring this MTU value on the virtual network adapter, while a network-based ET could do so by advertising the MTU to its downstream hosts using the DHCPv4 Interface MTU Option [RFC2132].

5.3. IPv4 Identification Header

If the generation of IPv6 Atomic Fragments is disabled, the value of the IPv4 Identification header will be lost during the translation. Conversely, enabling the generation of IPv6 Atomic Fragments will ensure that the IPv4 Identification Header will carried end-to-end. Note that for this to work bi-directionally, IPv6 Atomic Fragment generation must be enabled on both the SIIT-DC Gateway(s) and on the Edge Translator.

Note that apart from certain diagnostic tools, there are few (if any) application protocols that make use of the IPv4 Identification header. Therefore, the loss of the IPv4 Identification value will therefore generally not cause any problems.

IPv6 Atomic Fragments and their impact on the IPv4 Identification header is further discussed in Section 4.8.2 of [I-D.ietf-v6ops-siit-dc].

6. Intra-DC IPv4 Communication

While SIIT-DC is primarily intended to facilitate communication between IPv4-only nodes on the Internet and services hosted in an IPv6-only network, it is also possible to facilitate communication between an IPv4-only service or application running behind an Edge Translator and another service/application made available over IPv4 through SIIT-DC. This other service/application may be a IPv6-only service, or it may also be an IPv4-only service running behind another ET.

Facilitating such communication requires that another Static Address Mapping is configured in the ET (one for each service it wants to communicate to). If there are two ETs involved, both of them must be configured in the same fashion for bi-directional communication to work. The following two subsections contain examples that demonstrate how this may be set up.

Note that for the intra-DC communication described in this section, the SIIT-DC Gateway is not involved at all. Therefore there is no requirement that the Static Address Mappings in question are also configured on the SIIT-DC Gateway. It is also possible to use private [RFC1918] IPv4 addresses, in order to reduce the need for publicly routable IPv4 addresses. However, if the IPv4-only application(s) are also to be made available to the IPv4 Internet through an SIIT-DC Gateway, it is highly recommended that the Static Address Mappings configured in the ET match those configured in the SIIT-DC Gateway. Otherwise one end up in the situation where a service is reached using different IPv4 addresses depending on whether one connects to it from the IPv4 Internet or from another IPv4-only application residing in the same data centre. While it may still work, the overall architecture gets significantly more complex.

Finally, if both services/applications support IPv6, it is highly recommended that IPv6 is used for all internal communications. The approach described in this section should only be used if one or both of the services or applications only supports IPv4, making native IPv6 communication impossible.

6.1. Between IPv4-Only and IPv6-Only Services

This section demonstrates how an IPv4-only service/application "A" running behind an ET can communicate with an IPv6-only service "B".

Intra-DC IPv4-only to IPv6-only Overview

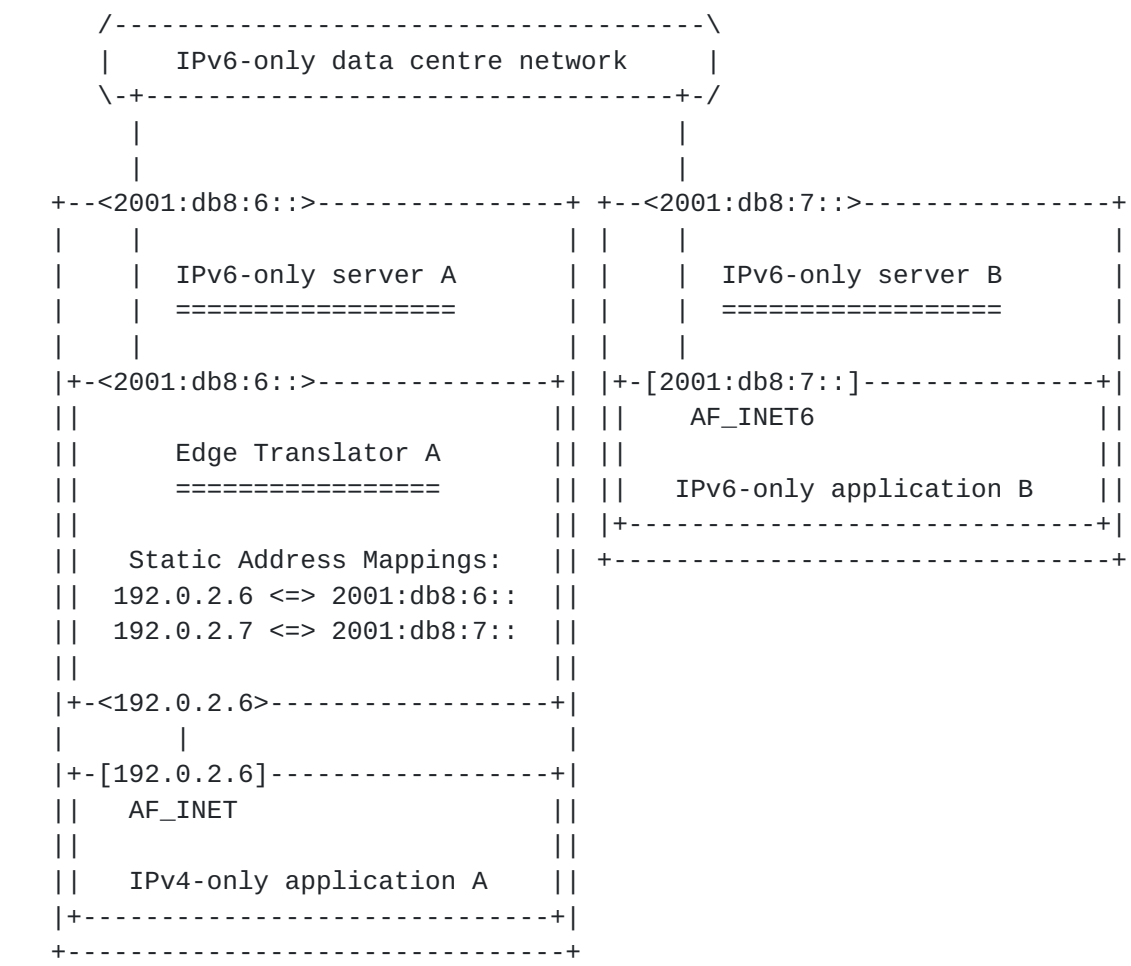

Figure 6

In this example, the IPv4-only application on server "A" is listening on the IPv4 address 192.0.2.6, which is made available to the IPv6 network on the IPv6 address 2001:db8:6:: (by the ET). The IPv6-only application on server "B" is only listening on the IPv6 address 2001:db8:7::, and has no knowledge of IPv4.

In order to facilitate communication between the two application, another Static Address Mapping must be configured in the ET on server "A". This provides an IPv4 address (192.0.2.7) that the IPv4-only application can communicate with, which represents the IPv6 address used by application "B" (2001:db8:7::).

The following figure shows the packet translations step by step, for a packet sent by the IPv4-only application "A" to the IPv6-only application "B". For traffic in the opposite direction, you may read the figure from the bottom up and swap the Src/Dst addresses.

Intra-DC IPv4-only to IPv6-only Packet Flow

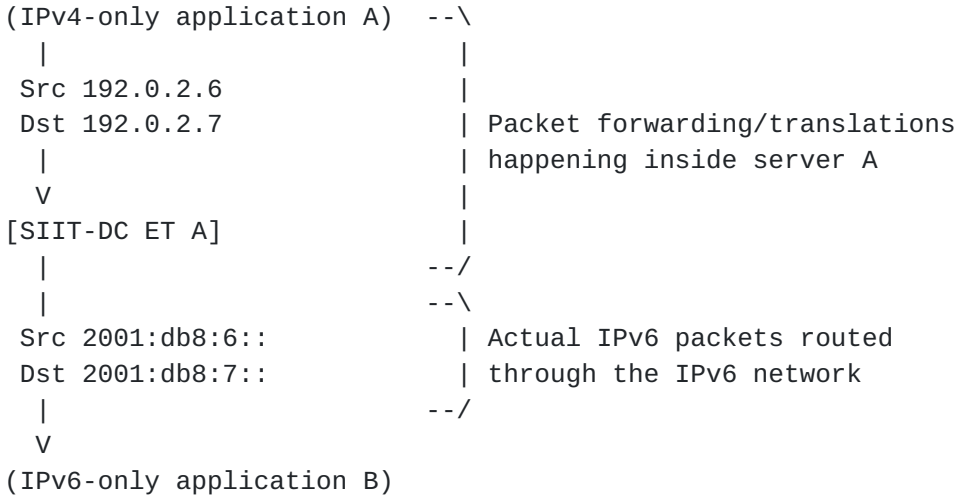

Figure 7

6.2. Between Two IPv4-Only Services

This section demonstrates how an IPv4-only service/application "A" running behind an ET can communicate with an IPv4-only service/application "B" running behind another ET.

Intra-DC IPv4-only to IPv6-only Overview

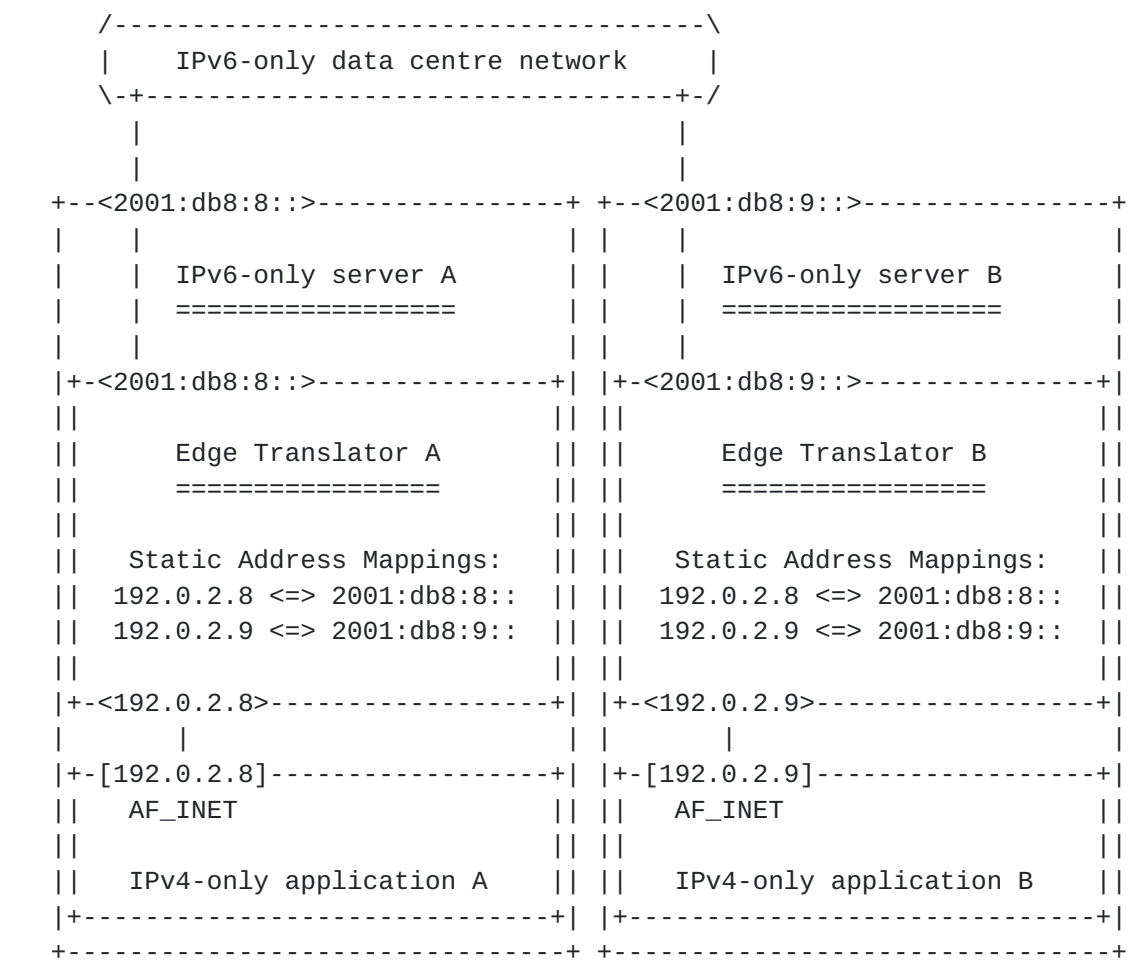

Figure 8

In this example, the IPv4-only application on server "A" is listening on the IPv4 address 192.0.2.8, which is made available to the IPv6 network on the IPv6 address 2001:db8:8:: (by the ET). In the same fashion, the IPv4-only application on server "B" is listening on the IPv4 address 192.0.2.9 and is made available by its ET on the IPv6 address 2001:db8:9::.

In order to facilitate communication between the two application, a second Static Address Mapping must be configured in the ET on both servers. This provides each application with an IPv4 address that represents the other application. Thus bi-directional communication between the two applications can commence.

The following figure shows the packet translations step by step, for a packet sent by the IPv4-only application "A" to the IPv4-only application "B". For traffic in the opposite direction, you may read the figure from the bottom up and swap the Src/Dst addresses.

Intra-DC IPv4-only to IPv4-only Packet Flow

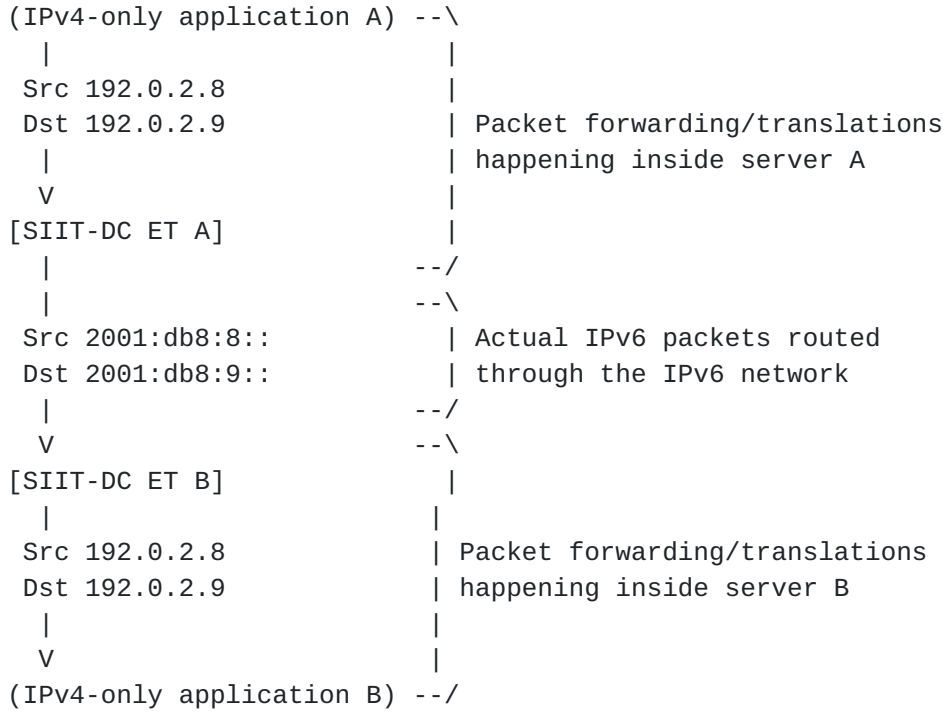

Figure 9

7. Acknowledgements

The author would like to especially thank the authors of 464XLAT [RFC6877]: Masataka Mawatari, Masanobu Kawashima, and Cameron Byrne. The architecture described by this document is merely an adaptation of their work to a data centre environment, and could not have happened without them.

The author would like also to thank the following individuals for their contributions, suggestions, corrections, and criticisms: Fred Baker, Tobias Brox, Ray Hunter, Shucheng LIU (Will), Andrew Yourtchenko.

8. IANA Considerations

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

9. Security Considerations

This section discusses security considerations specific to the use of an Edge Translator. See the Security Considerations section in [I-D.ietf-v6ops-siit-dc] for additional security considerations applicable to the SIIT-DC architecture in general.

9.1. Address Spoofing

If the ET receives an IPv4 packet from the application from a different source address than the one it has a Static Address Mapping for, the both the source and destination addresses will be rewritten according to [RFC6052]. After undergoing the reverse translation in the SIIT-DC Gateway, the resulting IPv4 packet routed to the IPv4 network will have a spoofed IPv4 source address. The ET should therefore ensure that ingress filtering (cf. BCP38 [RFC2827]) is used on the ET's IPv4 interface, so that such packets are immediately discarded.

If the ET receives an IPv6 packet with both the source and destination address equal to the one it has a Static Address Mapping for, the resulting packet would appear to the application as locally generated, as both the source address and the destination address will be the same address as the one configured on the virtual IPv4 interface. This could trick the application into thinking this packet came from a trusted source, and give elevated privileges accordingly. To prevent this, the ET should discard any received IPv6 packets that have a source address that is equal either to either the IPv4 (after undergoing [RFC6052] translation) or the IPv6 address in the Static Address Mapping.

10. References

10.1. Normative References

[I-D.ietf-v6ops-siit-dc]
tore, t., "SIIT-DC: Stateless IP/ICMP Translation for IPv6 Data Centre Environments", draft-ietf-v6ops-siit-dc-00 (work in progress), December 2014.

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

10.2. Informative References

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[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.

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

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