IPv6 Operations
Internet-Draft
Intended status: Standards Track
Expires: April 08, 2015

T. Anderson
Redpill Linpro

October 05, 2014

SIIT-DC: Stateless IP/ICMP Translation for IPv6 Data Centre Environments
draft-anderson-v6ops-siit-dc-01

Abstract

This document describes SIIT-DC, an extension to Stateless IP/ICMP Translation (SIIT) [RFC6145] that makes it ideally suited for use in IPv6 data centre environments. SIIT-DC simultaneously facilitates IPv6 deployment and IPv4 address conservation. The overall SIIT-DC architecture is described, as well as guidelines for operators. Finally, the normative implementation requirements are described, as a list of additions and changes to SIIT [RFC6145].

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 http://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 April 08, 2015.

Copyright Notice

Table of Contents

1. Introduction

SIIT-DC is an extension of SIIT [RFC6145] that provides a network-centric stateless translation service that allows a data centre operator or Internet Content Provider (ICP) to run a data centre network, servers, and applications using exclusively IPv6, while at the same time ensuring that end users that have only IPv4 connectivity will be able to continue to access the services and applications.

1.1. Motivation and Goals

Historically, dual stack [RFC4213] has been the recommended way to transition from an IPv4-only environment to one capable of serving IPv6 users. For data centre operators and Internet content providers, dual stack operation has a number of disadvantages compared to single stack operation. In particular, running two protocols rather than one results in increased complexity and operational overhead, with a very low expected return on investment in the short to medium term, as there are practically no end-users who have only connectivity to the IPv6 Internet. Furthermore, the dual stack approach does not in any way help with the depletion of the IPv4 address space.

Therefore, a better approach is needed. The design goals are:

- Promote the deployment of native IPv6 services (cf. [RFC6540]).
- Provide IPv4 service availability for legacy users with no loss of performance or functionality.
- To ensure that that the legacy users' IPv4 addresses remain available to the servers and applications.
- To conserve and maximise the utilisation of IPv4 addresses.
- To avoid introducing more complexity than absolutely necessary, especially on the servers and applications.
- To be easy to scale and deploy in a fault-tolerant manner.

The following subsections elaborates on how SIIT-DC meets these goals.

1.1.1. Single Stack IPv6 Operation

SIIT-DC allows an operator to build their applications on an IPv6-only foundation. IPv4 end-user connectivity becomes a service provided by the network, which systems administration and application development staff do not need to concern themselves with.

Obviously, this will promote universal IPv6 deployment for all of the provider's services and applications.

It is worth noting that SIIT-DC requires no special support or change from the underlying IPv6 infrastructure, it will work with any kind of IPv6 network. Traffic between IPv6-enabled end users and IPv6-enabled services will always be native, and SIIT-DC will not be involved in it at all.

1.1.2. Stateless Operation

Unlike other solutions that provide either dual stack availability to single-stack services (e.g., Stateful NAT64 [RFC6146] and Layer-4/7 proxies), or that provide conservation of IPv4 addresses (e.g., NAPT44 [RFC3022]), a SIIT-DC Gateway does not keep any state between each packet in a single connection or flow. In this sense it operates exactly like a normal IP router, and has similar scaling properties - the limiting factors are packets per second and bandwidth. The number of concurrent flows and flow initiation rates are irrelevant for performance.

This not only allows individual SIIT-DC Gateways to easily attain "line rate" performance, it also allows for per-packet load balancing between multiple SIIT-DC Gateways using Equal-Cost Multipath Routing [RFC2991]. Asymmetric routing is also acceptable, which makes it easy to avoid sub-optimal traffic patterns; the prefixes involved may be anycasted from all the SIIT-DC Gateways in the provider's network, thus ensuring that the most optimal path through the network is used, even where the optimal path in one direction differs from the optimal path in the opposite direction.

Finally, stateless operation means that high availability is easily achieved. If an SIIT-DC Gateway should fail, its traffic can be re-routed onto another SIIT-DC Gateway using a standard IP routing protocol. This does not impact existing flows any more than what any other IP re-routing event would.

1.1.3. IPv4 Address Conservation

In most parts of the world, it is difficult or even impossible to obtain generously sized IPv4 allocations from the Regional Internet

Registries. The resulting scarcity in turn impacts individual end users and operators, which might be forced to purchase IPv4 addresses from other operators in order to cover their needs. This process can be risky to business continuity, in the case no suitable block for sale can be located, and/or turn out to be prohibitively expensive. Even so, a data centre operator will find that providing IPv4 service is essential, as a large share of the Internet users still does not have IPv6 connectivity.

A key goal of SIIT-DC is to help reduce a data centre operator's IPv4 address requirement to the absolute minimum, by allowing the operator to remove them entirely from components that do not need to communicate with endpoints in the IPv4 Internet. One example would be servers that are operating in a supporting/backend role and only communicates with to other servers (database servers, file servers, and so on). Another example would be the network infrastructure itself (router-to-router links, loopback addresses, and so on). Furthermore, as LAN prefix sizes must always be rounded up to the nearest power of two (or larger, if one reserves space for future growth), even more IPv4 addresses will often end up being wasted without even being used.

With SIIT-DC, the operator can remove these valuable IPv4 addresses from his backend servers and network infrastructure, and reassign them to the SIIT-DC service as IPv4 Service Addresses. There is no requirement that IPv4 Service Addresses are assigned in an aggregated manner, so there is nothing lost due to infrastructure overhead; every single IPv4 address assigned to SIIT-DC can be used an IPv4 Service Address.

1.1.4. No Loss of End User's IPv4 Source Address

SIIT-DC will map the entire end-user's IPv4 source address into an predefined IPv6 translation prefix. This ensures that there is no loss of information; the end-user's IPv4 source address remains available to the server/application, allowing it to perform tasks like Geo-Location, logging, abuse handling, and so forth.

1.1.5. Compatible with Standard IPv6 Implementations

Except for the introduction of the SIIT-DC Gateways themselves, no change to the network, servers, applications, or anything else is required in order to support SIIT-DC. SIIT-DC is practically invisibible from the point of view of the the IPv4 clients, the IPv6 servers, the IPv6 data centre network, and the IPv4 Internet. SIIT-DC interoperates with all standards-compliant IPv4 or IPv6 stacks.

1.1.6. No Architectural Dependency on IPv4

SIIT-DC will allow an ICP or data centre operator to build infrastructure and applications entirely on IPv6. This means that when the day comes to discontinue support for IPv4, no change needs to be made to the overall architecture - it's only a matter of shutting off the SIIT-DC Gateways. Therefore, by deploying native IPv6 along with SIIT-DC, operators will avoid future migration or deployment projects relating to IPv6 roll-out and/or IPv4 sun-setting.

2. Terminology

This document makes use of the following terms:

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.

IPv4 Service Address Pool One or more IPv4 prefixes routed to the SIIT-DC Gateway's IPv4 interface. IPv4 Service Addresses are allocated from this pool. Note that this does not necessarily have to be a "pool" per se, as it could also be one or more host routes (whose prefix length is equal to /32). The primary purpose of using a pool rather than host routes is to facilitate IPv4 route aggregation and ease provisioning of new IPv4 Service Addresses.

IPv6 Service Address A public IPv6 address assigned to a server or application in the IPv6 network. IPv6-only and dual stacked clients communicates with this address directly without invoking SIIT-DC. IPv4-only clients also communicates with this address through the SIIT-DC Gateway and via an IPv4 Service Address.

SIIT-DC Host Agent A logical function very similar to an SIIT-DC Gateway that resides on a server and provides virtual IPv4 connectivity to applications, by reversing the translations done by the SIIT-DC Gateway. It is an optional component of the SIIT-DC architecture, that may be used to increase application support. See [I-D.anderson-v6ops-siit-dc-2xlat].

SIIT-DC Gateway A device or a logical function that translates between IPv4 and IPv6 in accordance with Section 5.

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 prepends or strips 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.

3. Architectural Overview

This section describes the basic SIIT-DC architecture.

SIIT-DC Architecture

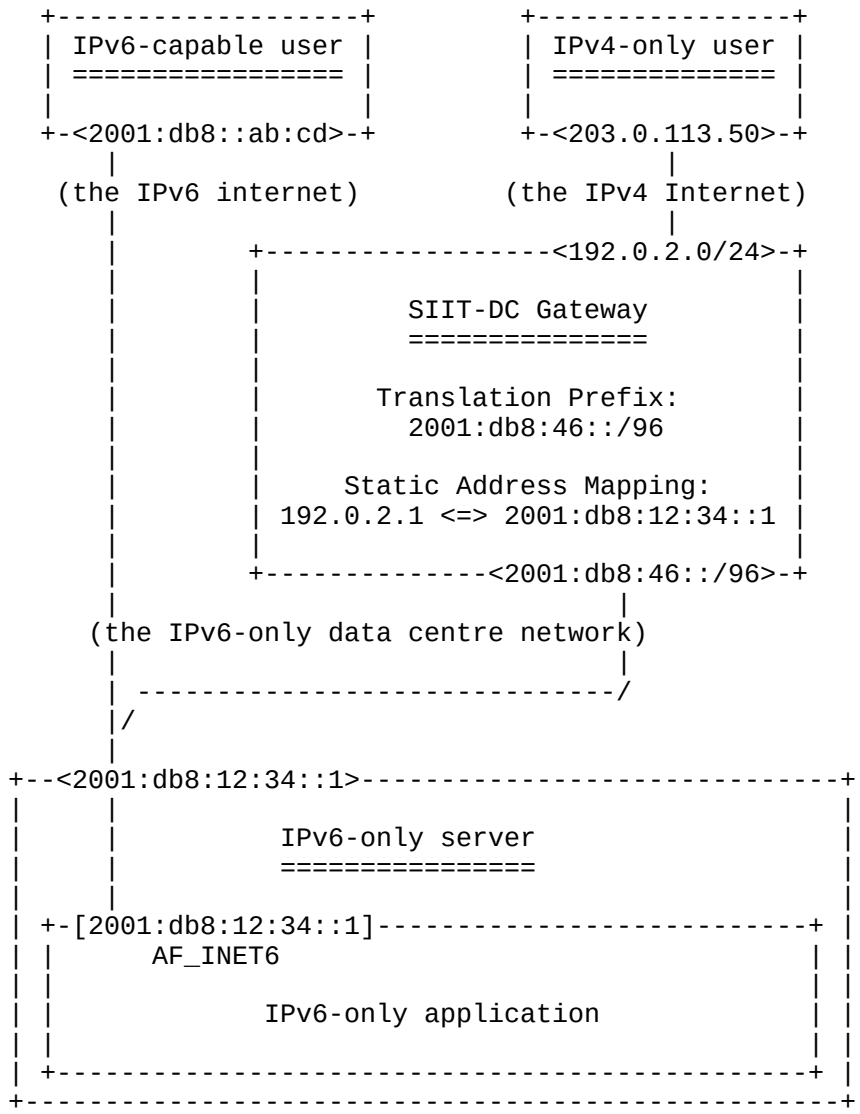

Figure 1

In this example, 192.0.2.0/24 is allocated as an IPv4 Service Address Pool. Individual IPv4 Service Addresses are assigned from this pool. The provider must route this prefix to the SIIT-DC Gateway's IPv4 interface. Note that there are no restrictions on how many IPv4 Service Address Pools are used or their prefix length, as long as they are all routed to the SIIT-DC Gateway's IPv4 interface.

The Static Address Mapping list is used when translating an IPv4 Service Address (here 192.0.2.1) to its corresponding IPv6 Service Address (here 2001:db8:12:34::1) and vice versa. When the SIIT-DC Gateway translates an IPv4 packet to IPv6, any IPv4 Service Address found in the original IPv4 header will be replaced with the corresponding IPv6 Service Address in the resulting IPv6 header, and vice versa when translating an IPv6 packet to IPv4.

2001:db8:46::/96 is the Translation Prefix into which the entire IPv4 address space is mapped. It is used for translation of the end user's IPv4 address to IPv6 and vice versa according to the algorithm defined in Section 2.2 of RFC6052 [RFC6052]. This algorithmic mapping has a lower precedence than the configured Static Address Mappings.

The SIIT-DC Gateway itself can be either a separate device or a logical function in another multi-purpose device, for example an IP router. Any number of SIIT-DC Gateways may exist simultaneously in an operators infrastructure, as long as they all have the same translation prefix and list of Static Mappings configured.

3.1. DNS Configuration

The IPv6 Service Address of should be registered in DNS using an AAAA record, while its corresponding IPv4 Service Address should be registered using an A record. This results in the following DNS records:

DNS Configuration for a SIIT-DC enabled service

```
app.domain.tld.    IN AAAA  2001:db8:12:34::1
app.domain.tld.    IN A     192.0.2.1
```

Figure 2

3.2. Packet Flow

In this example, "IPv4-only user" initiates a request to the application running on the IPv6-only server. He starts by looking up the IN A record of "app.domain.tld" in DNS, and attempts to connect to this address on the service by transmitting the following IPv4 packet destined for the IPv4 Service Address:

Stage 1: Client -> Server, IPv4

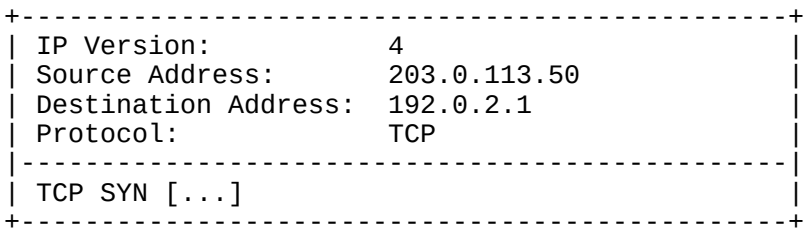

Figure 3

This packet is then routed over the Internet to the (nearest) SIIT-DC Gateway, which translates it into the following IPv6 packet and forward it into the IPv6 network:

Stage 2: Client -> Server request, IPv4

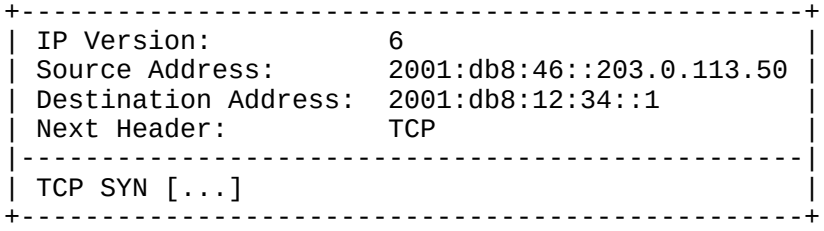

Figure 4

The destination address field was translated to the IPv6 Service Address according to the configured Static Address Mapping, while the source address was field translated according to the [RFC6052] mapping using the Translation Prefix (because it did not match any Static Address Mapping). The rest of the IP header was translated according to [RFC6145]. The Layer 4 payload is copied verbatim, with the exception of the TCP checksum being recalculated.

Note that the IPv6 address 2001:db8:46::203.0.113.50 may also be expressed as 2001:db8:46::cb00:7132, cf. Section 2.2 of RFC2373 [RFC2373].

Next, the application receives receives this IPv6 packet and responds to it like it would with any other IPv6 packet:

Stage 3: Server -> Client response, IPv6

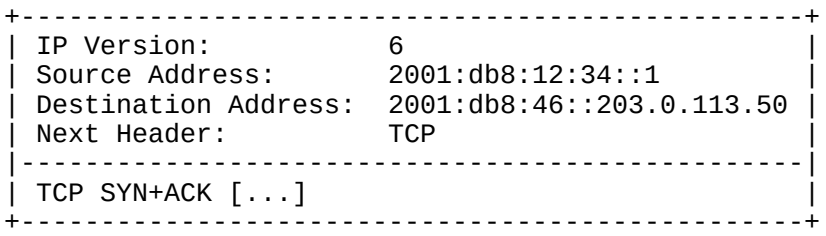

Figure 5

The response packet is routed to the (nearest) SIIT-DC Gateway's IPv6 interface, which will translate it back to IPv4 as follows:

Stage 4: Server -> Client response, IPv4

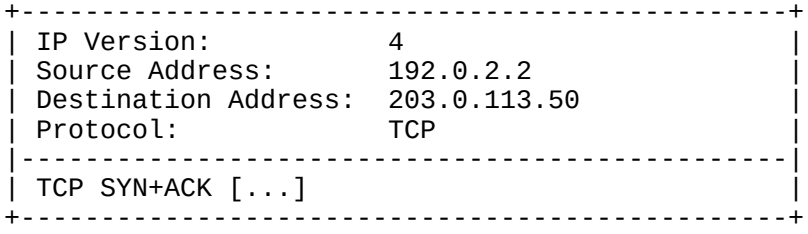

Figure 6

This time, the source address matched the Static Address Mapping and was translated accordingly, while the destination address did not, and was therefore translated according to [RFC6052] by having the Translation Prefix stripped. The rest of the packet was translated according to [RFC6145].

The resulting IPv4 packet is transmitted back to the end user over the IPv4 Internet. Subsequent packets in the flow will follow the exact same translation pattern. They may or may not cross the same translators as earlier packets in the same flow.

The end user's IPv4 stack has no idea that it is communicating with an IPv6 server, nor does the server's IPv6 stack have any idea that is is communicating with an IPv4 client. To them, it's just plain IPv4 or IPv6, respectively. However, the applications running on the server may optionally be updated to recognise and strip the Translation Prefix, so that the end user's IPv4 address may be used for logging, Geo-Location, abuse handling, and so forth.

4. Deployment Guidelines

In this section, we list recommendations and guidelines for operators who would like to deploy a SIIT-DC service in their data centre network.

4.1. Application Support for NAT

Not all application protocols are able to operate in a network environment where rewriting of IP addresses occur. An operator should therefore carefully evaluate the applications he would like to make available for IPv4 users through SIIT-DC, to ensure they do not fall in this category. In general, if an application layer protocol works correctly through standard NAT44 (see [RFC3235]), it will most likely work correctly through SIIT-DC as well.

Higher-level protocols that embed IP addresses as part of their payload are especially problematic, as noted in [RFC2663], [RFC2993], and [RFC3022]. Such protocols will most likely not work through any form of address translation, including SIIT-DC. One well-known example of such a protocol is FTP [RFC0959].

The SIIT-DC architecture may be extended with a Host Agent that reverses the translation performed by the SIIT-DC Gateway before passing the packets to the application software. This allows the problematic application protocols described above to work correctly in an SIIT-DC environment as well. See [I-D.anderson-v6ops-siit-dc-2xlat] for a description of this extension.

4.2. Application Support for IPv6

SIIT-DC requires that the application software supports IPv6 networking, and that it has no dependency on IPv4 networking. If this is not the case, the approach described in [I-D.anderson-v6ops-siit-dc-2xlat] may be used, as it provides the application with seemingly native IPv4 connectivity. This allows IPv4-only applications to work correctly in an otherwise IPv6-only environment.

4.3. Application Communication Pattern

SIIT-DC is ideally suited for applications where IPv4-only nodes on the Internet initiate traffic towards the IPv6-only services, which in turn are only passively listening for inbound traffic and responding as necessary. One well-known example of such a protocol is HTTP [RFC2616]. This is due to the fact that in this case, an IPv4 user looks exactly like an ordinary IPv6 user from the host and application’s point of view, and requires no special treatment.

It is possible to combine SIIT-DC with DNS64 [RFC6147] in order to allow an IPv6-only application to initiate communication with IPv4-only nodes through an SIIT-DC Gateway. However, in this case, care must be taken so that all outgoing communication is sourced from the IPv6 Service Address that has a Static Mapping configured on the SIIT-DC Gateway. If another unmapped address is used, the SIIT-DC Gateway will discard the packet.

An alternative approach to the above would be to make use of an SIIT-DC Host Agent as described in [I-D.anderson-v6ops-siit-dc-2xlat]. This provides the application with seemingly native IPv4 connectivity, which it may use for both inbound and outbound communication without requiring the application to select a specific source address for its outbound communications.

4.4. Choice of Translation Prefix

Either a Network-Specific Prefix (NSP) from the provider's own IPv6 address space or the IANA-allocated Well-Known Prefix 64:ff9b::/96 (WKP) may be used. From a technical point of view, both should work equally well, however as only a single WKP exists, if a provider would like to deploy more than one instance of SIIT-DC in his network, or Stateful NAT64 [RFC6146], an NSP must be used anyway for all but one of those deployments.

Furthermore, the WKP cannot be used in inter-domain routing. By using an NSP, a provider will have the possibility to provide SIIT-DC service to other operators across Autonomous System borders.

For these reasons, this document recommends that an NSP is used. Section 3.3 of [RFC6052] discusses the choice of translation prefix in more detail.

The Translation Prefix may use any of the lengths described in Section 2.2 of RFC6052 [RFC6052], but /96 has two distinct advantages over the others. First, converting it to IPv4 can be done in a single operation by simply stripping off the first 96 bits; second, it allows for IPv4 addresses to be embedded directly into the text representation of an IPv6 address using the familiar dotted quad notation, e.g., "2001:db8::198.51.100.10" (cf. Section 2.4 of RFC6052 [RFC6052])), instead of being converted to hexadecimal notation. This makes it easier to write IPv6 ACLs and similar that match translated endpoints in the IPv4 Internet. Use of a /96 prefix length is therefore recommended.

4.5. Routing Considerations

The prefixes that constitute the IPv4 Service Address Pool and the IPv6 Translation Prefix may be routed to the SIIT-DC Gateway(s) as any other IPv4 or IPv6 route in the provider's network.

If more than one SIIT-DC Gateway is being deployed, it is recommended that a dynamic routing protocol (such as BGP, IS-IS, or OSPF) is being used to advertise the routes within the provider's network. This will ensure that the traffic that is to be translated will reach the closest SIIT-DC Gateway, reducing or eliminating sub-optimal traffic patterns, as well as provide high availability - if one SIIT-DC Gateway fails, the dynamic routing protocol will automatically redirect the traffic to the next-best translator.

4.6. Location of the SIIT-DC Gateways

The goal of SIIT-DC is to facilitate a true IPv6-only application and network architecture, with the sole exception being the IPv4 interfaces of the SIIT-DC Gateways and the network infrastructure required to connect them to the IPv4 Internet. Therefore, the SIIT-DC Gateways should be located somewhere beween the IPv4 Internet and the application delivery stack. This should be understood to include all servers, load balancers, firewalls, intrusion detection systems, and similar devices that are processing traffic to a greater extent than merely forwarding it.

It is optimal to place the SIIT-DC Gateways as close as possible to the direct path between the servers and the end users. If the closest translator is located a long way from the optimal path, all packets in both directions must make a detour. This would increase the RTT between the server and the end user by by two times the extra latency incurred by the detour, as well as cause unnecessary load on the network links on the detour path.

Where possible, it is beneficial to implement the SIIT-DC Gateways as a logical function within the routers would have handled the traffic anyway, had the topology been dual stacked. This way, the translation service would not need to be assigned separate networks ports (which might become saturated and impact the service quality), nor would it require extra rack space and energy. Some particularly good choices of the location could be within a data centre's access routers, or within the provider's border routers. When every single application in the data centre or the provider's network eventually runs on single-stack IPv6, there would no need to run IPv4 on the inside of the SIIT-DC Gateway. This reduces complexity, and allows the operator to reclaim IPv4 addresses from the network infrastructure that may instead be used as IPv4 Service Address Pools.

Finally, another possibility is that the data centre operator outsources the SIIT-DC service to another entity, for example his upstream ISP. Doing so allows the data centre operator to build a true IPv6-only infrastructure. However, in this case, care must be taken to ensure that the path between the data centre and the SIIT-DC operator has a stable and known MTU, and that the SIIT-DC Gateways are not too far away from the data centre (otherwise, translated traffic could incur a latency penalty).

4.7. Migration from Dual Stack

While this document discusses the use of IPv6-only servers and applications, there is no technical requirement that the servers are IPv4 free. SIIT-DC works equally well for dual stacked servers, which makes migration easy - after setting up the translation function, the DNS A record for the service is updated to point to the IPv4 address that will be translated to IPv6, the previously used IPv4 service address may continue to be assigned to the server. This makes roll-back to dual stack easy, as it is only a matter of changing the DNS record back to what it was before.

For high-volume services migrating to SIIT-DC from dual stack, DNS Round Robin may be used to gradually migrate the service’s IPv4 traffic from its native IPv4 address(es) to the translated IPv4 Service Address(s).

4.8. Packet Size and Fragmentation Considerations

There are some key differences between IPv4 and IPv6 relating to packet sizes and fragmentation that one should consider when deploying SIIT-DC. They result in a few problematic corner cases, which can be dealt with in a few different ways. The following subsections will discuss these in detail, and provide operational guidance.

In particular, the operator may find that relying on fragmentation in the IPv6 domain is undesired or even operationally impossible [I-D.taylor-v6ops-fragdrop]. For this reason, the recommendations in this section seeks to minimise the use of IPv6 fragmentation.

Unless otherwise stated, the following subsections assume that the MTU in both the IPv4 and IPv6 domains is 1500 bytes.

4.8.1. IPv4/IPv6 Header Size Difference

The IPv6 header is up to 20 bytes larger than the IPv4 header. This means that a full-size 1500 bytes large IPv4 packet cannot be translated to IPv6 without being fragmented, otherwise it would likely have resulted in a 1520 bytes large IPv6 packet.

If the transport protocol used is TCP, this is generally not a problem, as the IPv6 server will advertise a TCP MSS of 1440 bytes. This causes the client to never send larger packets than what can be translated to a single full-size IPv6 packet, eliminating any need for fragmentation.

For other transport protocols, full-size IPv4 packets with the DF flag cleared will need to be fragmented by the SIIT-DC Gateway. The only way to avoid this is to increase the Path MTU between the SIIT-DC Gateway and the servers to 1520 bytes. Note that the servers' MTU SHOULD NOT be increased accordingly, as that would cause them to undergo Path MTU Discovery for most native IPv6 destinations. However, the servers would need to be able to accept and process incoming packets larger than their own MTU. If the server's IPv6 implementation allows the MTU to be set differently for specific destinations, it could be increased to 1520 for destinations within the Translation Prefix specifically.

4.8.2. IPv6 Atomic Fragments

In keeping with the fifth paragraph of Section 4 of RFC6145 [RFC6145], an SIIT-DC Gateway will by default add an IPv6 Fragmentation header to the resulting IPv6 packet when translating an IPv4 packet with the Don't Fragment flag set to 0.

This happens even though the resulting IPv6 packet isn't actually fragmented into several pieces, resulting in an IPv6 Atomic Fragment [RFC6946]. These Atomic Fragments are generally not useful in a data centre environment, and it is therefore recommended that this behaviour is disabled in the SIIT-DC Gateways. To this end, Section 4 of RFC6145 [RFC6145] notes that the "translator MAY provide a configuration function that allows the translator not to include the Fragment Header for the non-fragmented IPv6 packets".

Note that [I-D.gont-6man-deprecate-atomfrag-generation] seeks to update [RFC6145], making the functionality described above as the standard and only mode of operation.

4.8.3. Minimum Path MTU Difference Between IPv4 and IPv6

Section 5 of RFC2460 [RFC2460] specifies that the minimum IPv6 link MTU is 1280 bytes. Therefore, an IPv6 node can reasonably assume that if it transmits an IPv6 packet that is 1280 bytes or smaller, it is guaranteed to reach its destination without requiring fragmentation or invoking the Path MTU Discovery algorithm [RFC1981]. However, this assumption fails if the destination is an IPv4 node reached through a protocol translator such as an SIIT-DC Gateway, as the minimum IPv4 link MTU is 68 bytes. See Section 3.2 of RFC791 [RFC0791].

Section 5.1 of RFC6145 [RFC6145] specifies that an SIIT-DC Gateway should set the IPv4 Don't Fragment flag to 1 when it translates an unfragmented IPv6 packet to IPv4. This means that when the path to the destination IPv4 node contains an IPv4 link with an MTU smaller than 1260 bytes (which corresponds to an IPv6 MTU smaller than 1280 bytes, cf. Section 4.8.1), the Path MTU Discovery algorithm will be invoked, even if the original IPv6 packet was only 1280 bytes large. This happens as a result of the IPv4 router connecting to the IPv4 link with the small MTU returning an ICMPv4 Need To Fragment error with an MTU value smaller than 1260, which in turns is translated by the SIIT-DC Gateway to an ICMPv6 Packet Too Big error with an MTU value smaller than 1280 which is then transmitted to the origin IPv6 node.

When an IPv6 node receives an ICMPv6 Packet Too Big error indicating an MTU value smaller than 1280, the last paragraph of Section 5 of RFC2460 [RFC2460] gives it two choices on how to proceed:

- It may reduce its Path MTU value to the value indicated in the Packet Too Big, i.e., limit the size of subsequent packets transmitted to that destination to the indicated value. This approach causes no problems for the SIIT-DC function, as it simply allows Path MTU Discovery to work transparently across the SIIT-DC Gateway.

- It may reduce its Path MTU value to exactly 1280, and in addition include a Fragmentation header in subsequent packets sent to that destination. In other words, the IPv6 node will start emitting Atomic Fragments. The Fragmentation header signals to the the SIIT-DC Gateway that the Don't Fragment flag should be set to 0 in the resulting IPv4 packet, and it also provides the Identification value.

If the use of the IPv6 Fragmentation header is problematic, and the operator has IPv6 nodes that implement the second option above, the operator should consider enabling the functionality described as the "second approach" in Section 6 of RFC6145 [RFC6145]. This functionality changes the SIIT-DC Gateway's behaviour as follows:

- When translating ICMPv4 Need To Fragment to ICMPv6 Packet Too Big, the resulting packet will never contain an MTU value lower than 1280. This prevents the IPv6 nodes from generating Atomic Fragments.

- When translating IPv6 packets smaller than or equal to 1280 bytes, the Don't Fragment flag in the resulting IPv4 packet will be set to 0. This ensures that in the eventuality that the path contains an IPv4 link with an MTU smaller than 1260, the IPv4 router connected to that link will have the responsibility to fragment the packet before forwarding it towards its destination.

In summary, this approach could be seen as prompting the IPv4 protocol itself to provide the "link-specific fragmentation and reassembly at a layer below IPv6" required for links that "cannot convey a 1280-octet packet in one piece", to paraphrase Section 5 of RFC2460 [RFC2460]. Note that [I-D.gont-6man-deprecate-atomfrag-generation] seeks to update [RFC6145], making the approach described above as the standard and only mode of operation.

5. Implementation Requirements

This normative section specifies the SIIT-DC protocol that is implemented by an SIIT-DC Gateway. Because SIIT-DC builds on and closely resembles SIIT [RFC6145], this section should be read as a set of additions and changes that are applied to an implementation already compliant to SIIT [RFC6145]. Each of the following subsections discuss how the requirement relates to with any corresponding requirements in SIIT [RFC6145].

5.1. Compliance with RFC6145 and RFC6052

Unless otherwise stated in the following sections, an SIIT-DC implementation MUST comply fully with [RFC6145]. It must also implement the algorithmic address mapping defined in [RFC6052].

5.2. Static Address Mapping Function

The implementation MUST allow the operator to configure an arbitrary number of Static Address Mappings which override the default [RFC6052] algorithm. It SHOULD be possible to specify a single bi-directional mapping that will be used in both the IPv4=>IPv6 and IPv6=>IPv4 directions, but it MAY additionally (or alternatively) support unidirectional mappings.

An example of such a bidirectional Static Address Mapping would be:

- 192.0.2.1 <=> 2001:db8:12:34::1

To accomplish the same using unidirectional mappings, the following two mappings must instead be configured:

- 192.0.2.1 => 2001:db8:12:34::1

- 2001:db8:12:34::1 => 192.0.2.1

In both cases, if the SIIT-DC Gateway receives an IPv6 packet that has the value 2001:db8:12:34::1 in either the source or destination field of the IPv6 header, it MUST rewrite this field to 192 0.2.1 when translating to IPv4. Similarly, if the SIIT-DC Gateway receives an IPv4 packet that has the value 192.0.2.1 as the either the source or destination field of the IPv4 header, it MUST rewrite this field to 2001:db8:12:34::1 when translating to IPv6. For all IPv4 or IPv6 source or destination field values for which there are no matching Static Address Mapping, [RFC6052] compliant mapping MUST be used instead.

Relation to [RFC6145]: The Static Address Mapping is a novel feature feature that is not discussed in [RFC6145]. It conflicts with [RFC6145]’s requirement that all addresses must be translated according to the [RFC6052] algorithm.

5.3. Support for Increasing the IPv6 Path MTU

The SIIT-DC Gateway MUST provide a configuration function for the network administrator to adjust the threshold of the minimum IPv6 MTU to a value that reflects the real value of the minimum IPv6 MTU in the network (greater than 1280 bytes). This will help reduce the chance of including the Fragment Header in the resulting IPv6 packets.

Relation to [RFC6145]: This strengthens the corresponding "MAY" requirement located in Section 4 of RFC6145 [RFC6145] to a "MUST".

5.4. Loop Prevention Mechanism

As noted in Section 9.2, there is a potential for packets looping through the SIIT-DC function if it receives an IPv4 packet for which there is no Static Address Mapping. It is therefore RECOMMENDED that the implementation has a mechanism that automatically prevents this behaviour. One way this could be accomplished would be to discard any IPv4 packets that would be translated into an IPv6 packet that would be routed straight back into the SIIT-DC function.

If such a mechanism isn't provided, the implementation MUST provide a way to manually filter or null-route the destination addresses that would otherwise cause loops.

Relation to [RFC6145]: This security consideration applies only when an SIIT-DC Gateway translates a packet in "pure" SIIT [RFC6145] mode (i.e., when both address fields are translated according to [RFC6052]). This consideration is in other words not specific to SIIT-DC, it is inherited from [RFC6145]. In spite of this, [RFC6145] does not describe this consideration or any methods of prevention. The requirements in this section is therefore novel to SIIT-DC, even though they apply equally to [RFC6145].

6. Acknowledgements

The author would like to thank the following individuals for their contributions, suggestions, corrections, and criticisms: Fred Baker, Cameron Byrne, Brian E Carpenter, Ross Chandler, Dagfinn Ilmari Mannsaaker, Lars Olafsen, Stig Sandbeck Mathisen, Knut A. Syed.

7. Requirements Language

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].

8. IANA Considerations

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

9. Security Considerations

9.1. Mistaking the Translation Prefix for a Trusted Network

If a Network-Specific Prefix from the provider's own address space is chosen for the translation prefix, as is recommended, care must be taken if the translation service is used in front of services that have application-level ACLs that distinguish between the operator's own networks and the Internet at large, as the translated IPv4 end users on the Internet will appear to be located within the provider's own IPv6 address space. It is therefore important that the translation prefix is treated the same as the Internet at large, rather than as a trusted network.

9.2. Packets Looping Through the SIIT-DC Function

If the SIIT-DC Gateway receives an IPv4 packet destined to an address for which there is no Static Address Mapping, its destination address will be rewritten according to [RFC6052], making the resulting IPv6 packet have a destination address within the translation prefix, which is likely routed to back to the SIIT-DC function. This will cause the packet to loop until its Time To Live / Hop Limit reaches zero, potentially creating a Denial Of Service vulnerability.

To avoid this, it should be ensured that packets sent to IPv4 destinations addresses for which there are no Static Address Mappings, or whose resulting IPv6 address does not have a more-specific route to the IPv6 network, are immediately discarded.

10. References

10.1. 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.

10.2. Informative References

[I-D.anderson-v6ops-siit-dc-2xlat] tore, t., "SIIT-DC: Dual Translation Mode", draft-anderson-v6ops-siit-dc-2xlat-00 (work in progress), September 2014.

[I-D.gont-6man-deprecate-atomfrag-generation] Gont, F., Will, W., and t. tore, "Deprecating the Generation of IPv6 Atomic Fragments", draft-gont-6man-deprecate-atomfrag-generation-01 (work in progress), August 2014.

[I-D.taylor-v6ops-fragdrop] Jaeggli, J., Colitti, L., Kumari, W., Vyncke, E., Kaeo, M., and T. Taylor, "Why Operators Filter Fragments and What It Implies", draft-taylor-v6ops-fragdrop-02 (work in progress), December 2013.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.

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

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC2373] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, November 2000.

[RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, November 2000.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC3235] Senie, D., "Network Address Translator (NAT)-Friendly Application Design Guidelines", RFC 3235, January 2002.

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

[RFC4217] Ford-Hutchinson, P., "Securing FTP with TLS", RFC 4217, October 2005.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

[RFC6540] George, W., Donley, C., Liljenstolpe, C., and L. Howard, "IPv6 Support Required for All IP-Capable Nodes", BCP 177, RFC 6540, April 2012.

[RFC6946] Gont, F., "Processing of IPv6 "Atomic" Fragments", RFC 6946, May 2013.

[RFC7269] Chen, G., Cao, Z., Xie, C., and D. Binet, "NAT64 Deployment Options and Experience", RFC 7269, June 2014.

Appendix A. Complete SIIT-DC topology example

This figure shows a more complete SIIT-DC topology, in order to better demonstrate the beneficial properties it has. In particular, it tries to highlight the following:

- Stateless operation: Any number of SIIT-DC Gateways may be deployed side-by side, or indeed anywhere in the IPv6 network, as any standard routing mechanism may be used to direct traffic to them (shown here with BGP on the IPv4 side and ECMP on the IPv6 side). This in turn leads to high availability, should one of the SIIT-DC Gateways fail or become unavailable, those standard routing mechanisms will ensure that traffic is automatically redirect one of the remaining SIIT-DC Gateways.

- IPv4 address conservation: Even though the to customers in the example have several hundred servers, most of them are not used for externally available services, and thus do not require an IPv4 address. The network between the servers and the SIIT-DC Gateways require no IPv4 addresses, either. Furthermore, the IPv4 addresses that are used do not have to be assigned to customers in the form of aggregated blocks or prefixes; which makes it easy to achieve 100% effective utilisation of the IPv4 service address pools.

- Application support: The translation-friendly applications HTTP and SMTP will work through SIIT-DC without requiring any special customisation. Furthermore, translation-unfriendly applications such as FTP will also work if an host agent in present, cf. [I-D.anderson-v6ops-siit-dc-2xlat].

- Native IPv6 as the foundation: Every server, application, and network component has access to native and untranslated IPv6

connectivity to each other and to the Internet. Traffic through the SIIT-DC Gateways will diminish over time as IPv6 is deployed throughout the Internet. Eventually they may be shut down entirely, which causes no disruption to the application stacks' ability to deliver their services over native IPv6.

Example data centre topology using SIIT-DC

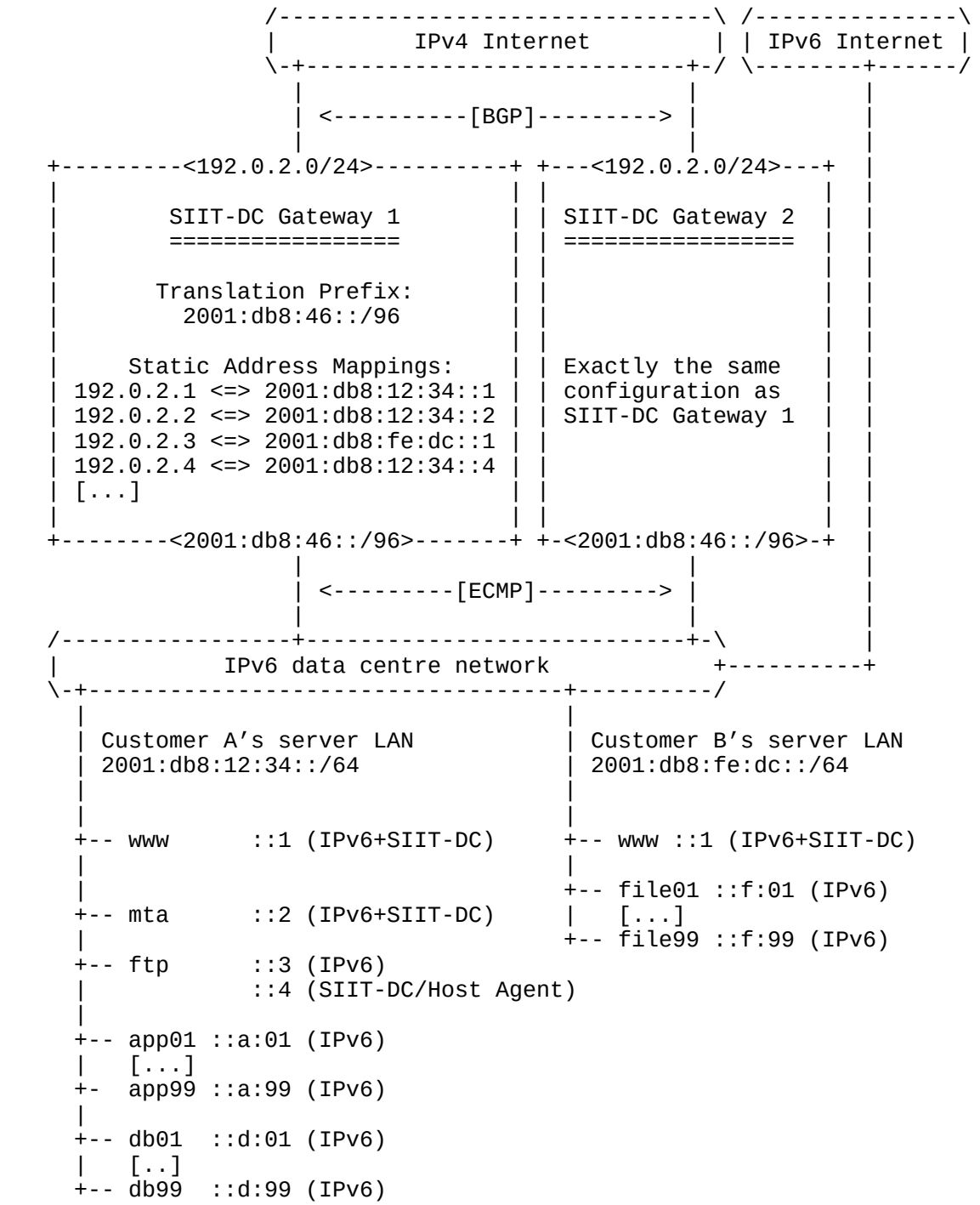
IPv4 Internet
IPv6 Internet
<----------[BGP]---------->
<192.0.2.0/24>
<192.0.2.0/24>
SIIT-DC Gateway 1
SIIT-DC Gateway 2
Translation Prefix:
2001:db8:46::/96
Static Address Mappings:
192.0.2.1 <=> 2001:db8:12:34::1
192.0.2.2 <=> 2001:db8:12:34::2
192.0.2.3 <=> 2001:db8:fe:dc::1
192.0.2.4 <=> 2001:db8:12:34::4
[...]
Exactly the same configuration as SIIT-DC Gateway 1
<2001:db8:46::/96>
<2001:db8:46::/96>
<---------[ECMP]--------->
IPv6 data centre network
Customer A's server LAN
2001:db8:12:34::/64
Customer B's server LAN
2001:db8:fe:dc::/64
www ::1 (IPv6+SIIT-DC)
mta ::2 (IPv6+SIIT-DC)
ftp ::3 (IPv6)
::4 (SIIT-DC/Host Agent)
app01 ::a:01 (IPv6)
[...]
app99 ::a:99 (IPv6)
db01 ::d:01 (IPv6)
[..]
db99 ::d:99 (IPv6)
www ::1 (IPv6+SIIT-DC)
file01 ::f:01 (IPv6)
[...]
file99 ::f:99 (IPv6)

Figure 7

Appendix B. Comparison to Other Deployment Approaches

There are a number of alternative deployment strategies a data centre operator may follow. They each have different properties and helps solve a different set of challenges. This section aims to compare the SIIT-DC approach with each of the most common ones, by highlighting the benefits and disadvantages of each.

B.1. IPv4-only

At the time of writing, IPv4-only operation remains the status quo for most operators. As such, it is well understood and supported. An operator can reasonably expect everything to work correctly in an IPv4-only environment.

Benefits of IPv4-only operation compared to SIIT-DC include:

- No translation occurs, the end-to-end principle is intact.
- Compatible with all common application protocols.
- Compatible with IPv4-only devices.
- Compatible with IPv4-only application software, without requiring a host agent.

Disadvantages of IPv4-only operation compared to SIIT-DC include:

- Does not provide any form of IPv6 connectivity.
- Does not alleviate IPv4 address scarcity.

B.2. IPv4-only + NAPT44

An operator who would otherwise chose a traditional IPv4-only approach, but cannot due to having insufficient public IPv4 addresses available, could chose to deploy using a combination of private IPv4 addresses [RFC1918] and NAPT44 [RFC3022] devices which will translate between a smaller number of public IPv4 addresses and the private addresses assigned to the servers that provide public services to the Internet.

Benefits of IPv4-only + NAPT44 operation compared to SIIT-DC include:

- Compatible with IPv4-only devices.

- Compatible with IPv4-only application software, without requiring a host agent.

Disadvantages of IPv4-only + NAPT44 operation compared to SIIT-DC include:

- Does not provide any form of IPv6 availability.
- Requires network devices that track all flow state, which may create a performance bottleneck and be an easy target for Denial of Service attacks.
- Limits routing flexibility (prevents closest exit routing), as outbound traffic must pass across the same NAPT44 device that handled the inbound traffic.
- Limited potential for horizontal scaling, as packets cannot be load-balanced across multiple NAT devices.
- Depending on whether or not the NAPT44 device rewrites source addresses in order to attract the return traffic to itself:
-
    * Obscures the true source address of the user from the server/ application, preventing it from e.g. performing geo-location lookups, or:
    * Requires an IPv4 default route to be pointed to the NAPT44 device, also attracting native traffic that does not need to undergo translation.

In addition, application compatibility is a consideration with both NAPT44 and SIIT-DC, but the exact nature depends from application to application, so it is hard to objectively quantify if there is a clear advantage to either approach here. Some translation-unfriendly application protocols may work without host modifications through the use of Application Layer Gateway support in the NAPT44 device (e.g., FTP [RFC0959]), or in the SIIT-DC architecture when a host agent is being used [I-D.anderson-v6ops-siit-dc-2xlat]. Other application protocols might not work with NAPT44 at all, but will work in the SIIT-DC if a host agent is being used (e.g., FTP/TLS [RFC4217]).

In summary, the most accurate statement would be to say that an NAPT44 architecture is more compatible with translation-unfriendly protocols than plain SIIT-DC, while SIIT-DC is more compatible than NAPT44 if a host agent is used.

For a more complete discussion of potential issues with running NAPT44, see Architectural Implications of NAT [RFC2993].

B.3. IPv4-only + NAT64

An operator who would otherwise chose a traditional IPv4-only approach, but would in addition like to provide service availability for IPv6 end users, could use Stateful NAT64 [RFC6146] to accomplish this. In a sense, this would be the mirror image of an SIIT-DC architecture: The infrastructure and servers remains single-stacked, while connectivity to the other IP stack is provided through a translation system. Further information about operating Stateful NAT64 is found in [RFC7269].

Note that Stateful NAT64 can be deployed with or without NAPT44. With the exception that IPv6 service availability is being provided, the discussion in the previous two sections fully applies to an IPv4-only environment that includes NAT64.

Benefits of IPv4-only + NAT64 operation compared to SIIT-DC include:

- Compatible with IPv4-only devices.
- Compatible with IPv4-only application software, without requiring a host agent.

Disadvantages of IPv4-only + NAT64 operation compared to SIIT-DC include:

- Does not alleviate IPv4 address scarcity (assuming NAPT44 isn’t used).
- Requires network devices that track all flow state, which may create a performance bottleneck and be an easy target for Denial of Service attacks.
- Limits routing flexibility (prevents closest exit routing), as outbound traffic must pass across the same NAT64 device that handled the inbound traffic.
- Limited potential for horizontal scaling, as packets cannot be load-balanced across multiple NAT devices.
- Obscures the true source address of the user from the server/ application, preventing it from e.g. performing geo-location lookups.

- The traffic levels on the Stateful NAT64 routers will increase over time, in lockstep with the increased deployment of IPv6 in the Internet. For this reason, Section 3.2 of RFC7269 [RFC7269] notes that the use of Stateful NAT64 in a data centre environment "is only reasonable at an early stage". With SIIT-DC, the inverse is true; the traffic levels on the SIIT-DC Gateways will decrease over time, as end users will prefer to use native IPv6 once it is available to them.

B.4. Dual Stack

Dual Stack [RFC4213] could be used both with or without NAPT44 to handle IPv4. In general, the benefits and disadvantages are equal to the corresponding IPv4-only option, except for the fact that Dual Stack does provides IPv6 connectivity. Therefore, his section only lists the benefits and disadvantages which are unique to a Dual Stack environment.

Benefits of Dual Stack operation compared to SIIT-DC include:

- No translation occurring, the end-to-end principle is intact (assuming NAPT44 isn’t used).
- Compatible with all common application protocols (assuming NAPT44 isn’t used).
- Compatible with IPv4-only devices.
- Compatible with IPv4-only application software, without requiring a host agent.

Disadvantages of Dual Stack operation compared to SIIT-DC include:

- Does not alleviate IPv4 address scarcity (assuming NAPT44 isn’t used).
- Increases the complexity of the infrastructure, as many things must done twice (once for IPv4 and once for IPv6). Examples of things that must be duplicated in this manner under Dual Stack include: Firewall rules/ACLs, IGP topology, monitoring, troubleshooting.
- Encourages software developers, systems administrators, etc. to build architectures that cannot operate correctly without IPv4. This in turn makes it difficult to make use of Dual Stack as a short term transitional stage, rather than a near-permanent end state.

- Increases the amount of things that can encounter failures, and increases the time required to locate and fix such failures. This reduces reliability.

Author's Address

Tore Anderson
Redpill Linpro
Vitaminveien 1A
0485 Oslo
NORWAY

Phone: +47 959 31 212
Email: tore@redpill-linpro.com