Internet Engineering Task Force (IETF)
Request for Comments: 7755
Category: Informational
ISSN: 2070-1721

T. Anderson
Redpill Linpro

February 2016

SIIT-DC: Stateless IP/ICMP Translation for IPv6 Data Center Environments

Abstract

This document describes the use of the Stateless IP/ICMP Translation Algorithm (SIIT) in an IPv6 Internet Data Center (IDC). In this deployment model, traffic from legacy IPv4-only clients on the Internet is translated to IPv6 upon reaching the IDC operator’s network infrastructure. From that point on, it may be treated the same as traffic from native IPv6 end users. The IPv6 endpoints may be numbered using arbitrary (non-IPv4-translatable) IPv6 addresses. This facilitates a single-stack IPv6-only network infrastructure, as well as efficient utilization of public IPv4 addresses.

The primary audience is IDC operators who are deploying IPv6, running out of available IPv4 addresses, and/or feeling that dual stack causes undesirable operational complexity.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Historically, dual stack [RFC4213] [RFC6883] has been the recommended way to transition from a legacy IPv4-only environment to one capable of serving IPv6 users. However, for IDC operators, 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 little return on investment for as long as large parts of the public Internet remains predominantly IPv4 only. Furthermore, the dual-stack approach does not in any way help with the depletion of the IPv4 address space, which at the time of writing is a pressing concern in most parts of the world.

Therefore, some IDC operators may instead prefer an approach in which they only need to operate one protocol in the data center as they prepare for the future. Stateless IP/ICMP Translation for IPv6 Data Center Environments (SIIT-DC) is one such approach. Its design goals include:

- Promote the deployment of native IPv6 services (cf. [RFC6540]).

- Provide IPv4 service availability for legacy users with no loss of performance or functionality.

- Ensure that the legacy users' IPv4 addresses remain visible to the nodes and applications located in the IPv6 network.

- Conserve and maximize the utilization of the operator's public IPv4 addresses.

- Avoid introducing more complexity than absolutely necessary, especially on the nodes and applications.

- Easy to scale and deploy in a fault-tolerant manner.

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

1.1. Single-Stack IPv6 Operation

SIIT-DC allows IDC operators to build their infrastructure and 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. This promotes universal IPv6 deployment for the IDC operator's services and applications.

SIIT-DC requires no special support or change from the underlying IPv6 infrastructure; it is compatible with all standard IPv6 networks. Traffic between IPv6-enabled end users and IPv6-enabled services will always be transported native end to end; SIIT-DC does not intercept or handle native IPv6 traffic at all.

When the day comes to discontinue all support for IPv4, no change needs to be made to the overall architecture -- it’s only a matter of shutting off the SIIT-DC Border Relays (BRs). Operators who deploy native IPv6 along with SIIT-DC will thus avoid requiring any future migration or deployment projects relating to IPv6 deployment and/or IPv4 sunsetting.

1.2. Stateless Operation

Unlike other solutions that provide either dual-stack availability to single-stack services (e.g., Stateful Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers (NAT64) [RFC6146] and Layer 4/7 proxies) or conservation of IPv4 addresses (e.g., IPv4 address translation (NAPT44) [RFC3022]), SIIT-DC does not maintain any state associated with individual connections or flows. In this sense, it operates exactly like a regular 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 BRs to easily attain "line-rate" performance, but it also allows for per-packet load balancing between multiple BRs using Equal-Cost Multipath Routing [RFC2991]. Asymmetric routing is also acceptable, which makes it easy to avoid suboptimal traffic patterns; the prefixes involved may be anycasted from all the BRs 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 a BR should fail, its traffic can be rerouted onto another BR using a standard IP routing protocol. This does not impact existing flows any more than what any other IP rerouting event would.

1.3. IPv4 Address Conservation

In most parts of the world, it is difficult or even impossible to obtain generously sized IPv4 delegations from the Internet Numbers Registry System [RFC7020]. The resulting scarcity in turn impacts individual end users and operators, whom 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 where no suitable block for sale can be located, and/or turn out to be prohibitively expensive. In spite of this, an IDC operator will find that providing IPv4 service remains essential, as a large share of the Internet end users still do not have IPv6 connectivity.

A key goal of SIIT-DC is to help reduce a data center operator's IPv4 address requirement to the absolute minimum by allowing the operator to remove them entirely from nodes and applications 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 communicating with 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 exists no requirement that IPv4 Service Addresses are to be 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 as an IPv4 Service Address.

1.4. Clients' IPv4 Source Addresses Visible to Applications

SIIT-DC uses the [RFC6052] algorithm to map the entire end-user's IPv4 source address into a predefined IPv6 translation prefix. This ensures that there is no loss of information; the end-user's IPv4 source address remains available to the application located in the IPv6 network, allowing it to perform tasks like geolocation, logging, abuse handling, and so forth.

1.5. Compatible with Standard IPv4 and IPv6 Stacks

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

2. Terminology

This document makes use of the following terms:

SIIT-DC Border Relay (BR):
A device or a logical function that performs stateless protocol translation between IPv4 and IPv6. It MUST do so in accordance with [RFC6145] and [RFC7757].

SIIT-DC Edge Relay (ER):
A device or logical function that provides "native" IPv4 connectivity to IPv4-only devices or application software. It is very similar in function to a BR but is typically located close to the IPv4-only component(s) it is supporting rather than on the IDC's outer network border. The ER is an optional component of SIIT-DC. It is discussed in more detail in [RFC7756].

IPv4 Service Address:
An IPv4 address representing a node or service located in an IPv6 network. It is coupled with an IPv6 Service Address using an Explicit Address Mapping (EAM). Packets sent to this address are translated to IPv6 by the BR, and possibly back to IPv4 by an ER, before reaching the node or service.

IPv4 Service Address Pool:
One or more IPv4 prefixes routed to the BR's IPv4 interface. IPv4 Service Addresses are allocated from this pool. This does not necessarily have to be a "pool" per se, as it could also be one or more host routes (whose prefix lengths are equal to /32). The 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:
An IPv6 address assigned to an application, node, or service either directly or indirectly (through an ER). It is coupled with an IPv4 Service Address using an EAM. IPv4-only clients communicate with the IPv6 Service Address through SIIT-DC.

Explicit Address Mapping (EAM):
A bidirectional coupling between an IPv4 Service Address and an IPv6 Service Address configured in a BR or ER. When translating between IPv4 and IPv6, the BR/ER changes the address fields in the translated packet's IP header according to any matching EAM. The EAM algorithm is specified in [RFC7757].

Translation Prefix:
An IPv6 prefix into which the entire IPv4 address space is mapped, according to the algorithm in [RFC6052]. The translation prefix is routed to the BR's IPv6 interface. When translating between IPv4 and IPv6, a BR/ER will insert/remove the translation prefix into/from the address fields in the translated packet's IP header, unless an EAM exists for the IP address that is being translated.

IPv4-Translatable IPv6 Addresses:
As defined in Section 1.3 of [RFC6052].

IDC:
Short for "Internet Data Center"; a data center whose main purpose is to deliver services to the public Internet. SIIT-DC is primarily targeted at being deployed in an IDC. An IDC is typically operated by an Internet Content Provider or a Managed Services Provider.

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

XLAT:
Short for "Translation". Used in figures to indicate where a BR/ER uses SIIT [RFC6145] 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. Architectural Overview

This section describes the basic SIIT-DC architecture.

```
              IPv6-capable user      IPv4-only user
              <2001:db8::ab:cd>      <203.0.113.50>
                |                          |
             (the IPv6 Internet)   (the IPv4 Internet)
                |                          |
                |  +-[BR]---------<192.0.2.0/24>--------------+
                |  |                                          |
                |  | EAM #1:      192.0.2.1,2001:db8:12:34::1 |
                |  | EAM #2..#n:  [...]                       |
                |  | XLAT Prefix: 2001:db8:46::/96            |
                |  |                                          |
                |  +-------------<2001:db8:46::/96>-----------+
                |                        |
               (the IPv6-only data center network)
                           |
            +--<2001:db8:12:34::1>--[v6-only server]-+
            |     |                                  |
            | +-[2001:db8:12:34::1]--[v6-only app]-+ |
            | |   AF_INET6 socket                  | |
            | +------------------------------------+ |
            +----------------------------------------+
```

Figure 1: SIIT-DC Architecture

In Figure 1, 192.0.2.0/24 is the IPv4 Service Address Pool. Individual IPv4 Service Addresses are assigned from this prefix, and traffic destined for it is routed to the BR’s IPv4-facing network interface. 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 BR’s IPv4-facing network interface.

When translating packets between IPv4 and IPv6, the BR uses EAM #1 to replace any occurrence of the IPv4 Service Address (192.0.2.1) with its corresponding IPv6 Service Address (2001:db8:12:34::1). Addresses that do not match any EAM configured in the BR are translated by inserting or removing the translation prefix (2001:db8:46::/96); cf. Section 2.2 of [RFC6052].

The BR can be deployed as a separate device or as a logical function in another multipurpose device, such as an IP router. Any number of BRs may exist simultaneously in the IDC’s network infrastructure, as long as they are all configured with the same translation prefix and an identical EAM Table.

The IPv6 Service Address should be registered in DNS using an "IN AAAA" record, while its corresponding IPv4 Service Address should be registered using an "IN A" record. This ensures that IPv6-capable clients access the application/service directly using native IPv6 end to end, while IP4-only clients will access it through SIIT-DC.

3.1. Packet Flow

In this example, the "IPv4-only user" from Figure 1 initiates a connection to the application running on the IPv6-only server. After first having looked up the "IN A" record in DNS, the user starts by transmitting a TCP SYN packet to the IPv4 Service Address. This IPv4 packet is routed to the BR and is there translated to IPv6 as follows:

```
+--[IPv4]----------+     +--[IPv6]-----------------------+
| SRC 203.0.113.50 |     | SRC 2001:db8:46::203.0.113.50 |
| DST 192.0.2.1    | --> | DST 2001:db8:12:34::1         |
| TCP SYN [..]     |     | TCP SYN [..]                  |
+------------------+     +-------------------------------+
```

Figure 2: IPv4-to-IPv6 Translation

The resulting IPv6 packet is routed to the IPv6-only server, which processes and responds to it as if it had been a native IPv6 packet all along. The server’s IPv6 response packet is then routed back to the BR, where it is translated back to IPv4 as follows:

```
+--[IPv6]-----------------------+      +--[IPv4]----------+
| SRC 2001:db8:12:34::1         |      | SRC 192.0.2.1    |
| DST 2001:db8:46::203.0.113.50 | -->  | DST 203.0.113.50 |
| TCP SYN/ACK [..]              |      | TCP SYN/ACK [..] |
+-------------------------------+      +------------------+
```

Figure 3: IPv6-to-IPv4 Translation

It is important to note that neither the IPv4 client nor the IPv6 server/application need any special support to participate in SIIT-DC. However, the application may optionally be taught to extract the embedded IPv4 source address from incoming IPv6 packets with source addresses within the translation prefix. This will allow it to perform IPv4-specific tasks such as geolocation, logging, abuse handling, and so on.

4. Deployment Considerations and Guidelines

4.1. Application/Device Support for IPv6

SIIT-DC as described in this document requires that the application (and/or the node the application is located on) supports IPv6 networking and that it has no dependency on local IPv4 network connectivity.

SIIT-DC can, however, support legacy IPv4-dependent applications and nodes through the introduction of an ER. The ER provides the legacy application or node with seemingly native IPv4 Internet connectivity, so that it may operate correctly in an otherwise IPv6-only network environment. This approach is described in more detail in [RFC7756].

4.2. Application Support for NAT

The operator should carefully examine whether or not the application protocols he would like to use SIIT-DC with are able to operate in a network environment where rewriting of IP addresses occurs. 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 particularly problematic [RFC2663] [RFC2993] [RFC3022]. One well-known example of such a protocol is FTP [RFC959]. Such protocols can be made to work with SIIT-DC through the introduction of an ER, which provides end-to-end IPv4 address transparency by reversing the translations performed by the BR before passing the packets to the NAT-incompatible application. This approach is described in more detail in [RFC7756].

4.3. Application Communication Pattern

SIIT-DC is best suited for traditional client/server applications where IPv4-only clients on the Internet initiate traffic towards an IPv6-only service, which in turn is passively listening for inbound traffic and responding as necessary. In this case, an IPv4 client looks exactly like a native IPv6 client from the IPv6 service’s point of view and thus does not require any special treatment. One particularly common application protocol that follows this client/server communication pattern, and thus is ideally suited for use with SIIT-DC, is HTTP [RFC7230].

It is also possible to combine SIIT-DC with DNS64 [RFC6147] in order to allow an IPv6-only application to initiate communication with IPv4-only nodes through SIIT-DC. However, in this case, care must be taken so that all outgoing communication is sourced from an IPv6 Service Address that is found in an EAM configured in the BR. If another address is used, the BR will most likely be unable to translate it to IPv4, causing the packet to be discarded. This could be prevented by altering the Default Address Selection Policy Table [RFC6724] on the IPv6 node.

An alternative approach to the above would be to place an ER in front of the application in question, as described in [RFC7756]. This provides the application with seemingly native IPv4 connectivity, which it may use freely for bidirectional communication with the IPv4 Internet. An application or node located behind an ER does not need to worry about selecting a specific source address, as it will only have valid options available.

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 (WKP) 64:ff9b::/96 may be used. From a technical point of view, both work equally well. However, only a single WKP exists, so if a provider would like to deploy more than one instance of SIIT-DC in his network, or another translation technology such as Stateful NAT64 [RFC6146], the operator will be forced to use an NSP for all but one of those deployments.

Another consideration is that the WKP cannot be used in inter-domain routing. By using an NSP instead, SIIT-DC will support a deployment where the BR and the IPv6 Service Address are located in different Autonomous Systems.

The translation prefix may use any of the lengths described in Section 2.2 of [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]), instead of being converted to hexadecimal notation. This makes it easier to write literal IPv6 addresses (e.g., in ACLs) that correspond to translated endpoints in the IPv4 Internet.

For the reasons discussed above, this document recommends that an NSP with a prefix length of /96 be used. Section 3.3 of [RFC6052] discusses the choice of the translation prefix in more detail.

4.5. Routing Considerations

The prefixes that constitute the IPv4 Service Address Pool and the IPv6 translation prefix may be routed to the BRs like any other IPv4 or IPv6 route in the provider's network. If more than one BR is being deployed, it is recommended that a routing protocol (IGP) be 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 BR, reducing or eliminating suboptimal traffic patterns as well as providing high availability: should one BR fail, the IGP will automatically redirect the traffic to the closest alternate BR.

4.6. Location of the SIIT-DC Border Relays

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 BRs and the network infrastructure required to connect the BRs to the IPv4 Internet. Therefore, the BRs must be located somewhere between the IPv4 Internet and the application delivery stack, which includes 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 BRs as close as possible to the direct path between the location of the IPv6 Service Address and the end users. If the closest BR was located a long way from the direct path, all packets in both directions must make a detour in order to traverse the BR. This would increase the RTT between the service and the end user 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 BRs as a logical function within the routers that also handle the native IPv6 traffic between the IPv6 Service Address and the IPv6 Internet. This way, an SIIT-DC deployment does not require separate networks ports (which might become saturated and impact the service quality) nor will it require extra rack space and energy. Some particularly good choices for the location could be within the IDC's access routers or within the Autonomous System's border routers.

Finally, another possibility is that the IDC operator outsources the SIIT-DC service to another entity, for example, his upstream ISP. Doing so allows the IDC operator to build a true IPv6-only infrastructure.

4.7. Migration from Dual Stack

While this document mainly discusses the use of IPv6-only nodes and applications, it is important to note that SIIT-DC is fully compatible with dual-stack infrastructures, including dual-stack nodes and applications.

Thus, migrating a dual-stacked service to an IPv6-only one where SIIT-DC provides the IPv4 Internet connectivity is easy. The operator would start out by designating the service's current native IPv6 address as the IPv6 Service Address and assigning it a corresponding IPv4 Service Address. At this point, the service will respond on both its old (native) IPv4 address and the SIIT-DC IPv4 Service Address. The operator may now move traffic from the former to the latter by changing the service's "IN A" DNS record. Once all IPv4 traffic has been successfully moved to SIIT-DC, the old IPv4 address may be reclaimed.

4.8. Translation of ICMPv6 Errors to IPv4

In response to an IPv4 packet subsequently translated to IPv6 by the BR, an IPv6 router in the IDC network may need to transmit an ICMPv6 error back to the origin IPv4 node. By default, such an ICMPv6 error will most likely be discarded by the BR, unless the source address of the ICMPv6 error happens to be an IPv4-translatable IPv6 address or covered by an EAM.

To facilitate reliable delivery of such ICMPv6 errors, an SIIT-DC operator SHOULD implement the recommendations in [RFC6791] in the BRs.

4.9. MTU and Fragmentation

There are some key differences between IPv4 and IPv6 relating to packet sizes and fragmentation that one MUST 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 [FRAGMENTS]. For this reason, the recommendations in this section seek to minimize the use of IPv6 fragmentation.

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

4.9.1. 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; the IPv6 node will advertise a TCP Maximum Segment Size (MSS) of 1440 bytes during the initial TCP handshake. This causes the IPv4 clients 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 Don’t Fragment (DF) flag cleared will need to be fragmented by the BR. This may be avoided by increasing the Path MTU between the BR and the IPv6 nodes to 1520 bytes or greater. If this is done, the MTU on the IPv6 nodes themselves SHOULD NOT be increased accordingly, as doing so would cause them to undergo Path MTU Discovery for all destinations on the IPv6 Internet. The nodes MUST, however, be able to accept and process incoming packets larger than their own MTU. If the nodes’ IPv6 implementation allows the initial Path MTU to be set differently for specific destinations, it MAY be increased to 1520 for destinations within the translation prefix specifically.

4.9.2. IPv6 Atomic Fragments

In keeping with the fifth paragraph of Section 4 of [RFC6145], a stateless translator like a BR will by default add an IPv6 Fragmentation header to the resulting IPv6 packet when translating an IPv4 packet with the DF 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 an IDC environment, and it is therefore recommended that this behavior be disabled in the BRs. To this end, Section 4 of [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 work is currently in progress (in [RFC6145bis]) to deprecate IPv6 Atomic Fragments. As a result, a BR that conforms to that document is required to behave as recommended above.

In IPv6, the Identification value is located inside the Fragmentation header. That means that if the generation of IPv6 Atomic Fragments is disabled, the IPv4 Identification value will be lost during translation to IPv6. This could potentially confuse some diagnostic tools.

4.9.3. Minimum Path MTU Difference between IPv4 and IPv6

Section 5 of [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 might prove false if the destination is an IPv4 node reached through a protocol translator such as a BR, as the minimum IPv4 link MTU is 68 bytes. See Section 3.2 of [RFC791].

Section 5.1 of [RFC6145] specifies that a stateless translator should set the IPv4 Don't Fragment flag to 1 when it translates a non-fragmented 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 4.9.1), 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 turn is translated by the BR 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, it is not allowed to reduce its Path MTU estimation to the indicated value. It must instead include a Fragmentation header in subsequent packets sent on that path [RFC1981]. In other words, the IPv6 node will start emitting Atomic Fragments. The Fragmentation header signals to the BR 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, the operator should consider enabling the functionality described as the "second approach" in Section 6 of [RFC6145]. This functionality changes the BR's behavior 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].

Note that work is currently in progress (in [RFC6145bis]) to deprecate IPv6 Atomic Fragments. As a result, a BR that conforms to that document is required to behave as suggested above.

4.10. IPv4-Translatable IPv6 Service Addresses

SIIT-DC is designed so that the IPv6 Service Addresses are not required to be IPv4-translatable IPv6 addresses. Section 2 of [RFC7757] discusses why it is desirable to avoid requiring the use of IPv4-translatable IPv6 addresses.

It is, however, quite possible to deploy SIIT-DC in combination with IPv4-translatable IPv6 Service Addresses. The primary benefits in doing so are:

- The operator is not required to provision EAMs for IPv4-translatable IPv6 Service Addresses onto the BR/ERs.

- [RFC6145] translation can be performed in a checksum-neutral manner; cf. Section 4.1 of [RFC6052].

The trade-off is that the IPv4-translatable IPv6 Service Addresses must be configured on the IPv6 nodes, and the applications must be set up to use them -- likely in addition to their primary (non-IPv4-translatable) IPv6 addresses. The IPv4-translatable IPv6 Service Addresses must also be routed from the BR through the IDC's IPv6 network infrastructure to the nodes on which they are assigned. This essentially requires the entire IPv6 infrastructure to be made aware of and handle translated IPv4 traffic as a special case, which significantly increases complexity. As previously described in Section 1.1, avoiding such drawbacks is a design goal of SIIT-DC. The use of IPv4-translatable IPv6 Service Addresses is therefore discouraged.

5. Security Considerations

5.1. 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 recommended in Section 4.4, 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 traffic from translated IPv4 end users on the Internet might appear to be originating from the provider's own network. It is therefore important that the translation prefix be treated the same as the Internet at large rather than as a trusted network.

In order to alleviate this problem, the operator may opt to use a translation prefix that is distinct from and not a subset of the IPv6 prefixes used elsewhere in the network infrastructure.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <http://www.rfc-editor.org/info/rfc6052>.

[RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, DOI 10.17487/RFC6145, April 2011, <http://www.rfc-editor.org/info/rfc6145>.

[RFC6791] Li, X., Bao, C., Wing, D., Vaithianathan, R., and G. Huston, "Stateless Source Address Mapping for ICMPv6 Packets", RFC 6791, DOI 10.17487/RFC6791, November 2012, <http://www.rfc-editor.org/info/rfc6791>.

[RFC7757] Anderson, T. and A. Leiva, "Explicit Address Mappings for Stateless IP/ICMP Translation", RFC 7757, DOI 10.17487/RFC7757, February 2016, <http://www.rfc-editor.org/info/rfc7757>.

6.2. Informative References

[FRAGMENTS] Jaeggli, J., Colitti, L., Kumari, W., Vyncke, E., Kaeo, M., and T. Taylor, "Why Operators Filter Fragments and What It Implies", Work in Progress, draft-taylor-v6ops-fragdrop-02, December 2013.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <http://www.rfc-editor.org/info/rfc791>.

[RFC959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, DOI 10.17487/RFC0959, October 1985, <http://www.rfc-editor.org/info/rfc959>.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, DOI 10.17487/RFC1981, August 1996, <http://www.rfc-editor.org/info/rfc1981>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, DOI 10.17487/RFC2663, August 1999, <http://www.rfc-editor.org/info/rfc2663>.

[RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, DOI 10.17487/RFC2991, November 2000, <http://www.rfc-editor.org/info/rfc2991>.

[RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, DOI 10.17487/RFC2993, November 2000, <http://www.rfc-editor.org/info/rfc2993>.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <http://www.rfc-editor.org/info/rfc3022>.

[RFC3235] Senie, D., "Network Address Translator (NAT)-Friendly Application Design Guidelines", RFC 3235, DOI 10.17487/RFC3235, January 2002, <http://www.rfc-editor.org/info/rfc3235>.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, DOI 10.17487/RFC4213, October 2005, <http://www.rfc-editor.org/info/rfc4213>.

[RFC6145bis] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm (rfc6145bis)", Work in Progress, draft-bao-v6ops-rfc6145bis-05, January 2016.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <http://www.rfc-editor.org/info/rfc6146>.

[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, DOI 10.17487/RFC6147, April 2011, <http://www.rfc-editor.org/info/rfc6147>.

[RFC6540] George, W., Donley, C., Liljenstolpe, C., and L. Howard, "IPv6 Support Required for All IP-Capable Nodes", BCP 177, RFC 6540, DOI 10.17487/RFC6540, April 2012, <http://www.rfc-editor.org/info/rfc6540>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <http://www.rfc-editor.org/info/rfc6724>.

[RFC6883] Carpenter, B. and S. Jiang, "IPv6 Guidance for Internet Content Providers and Application Service Providers", RFC 6883, DOI 10.17487/RFC6883, March 2013, <http://www.rfc-editor.org/info/rfc6883>.

[RFC6946] Gont, F., "Processing of IPv6 "Atomic" Fragments", RFC 6946, DOI 10.17487/RFC6946, May 2013, <http://www.rfc-editor.org/info/rfc6946>.

[RFC7020] Housley, R., Curran, J., Huston, G., and D. Conrad, "The Internet Numbers Registry System", RFC 7020, DOI 10.17487/RFC7020, August 2013, <http://www.rfc-editor.org/info/rfc7020>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <http://www.rfc-editor.org/info/rfc7230>.

[RFC7756] Anderson, T. and S. Steffann, "Stateless IP/ICMP Translation for IPv6 Internet Data Center Environments (SIIT-DC): Dual Translation Mode", RFC 7756, DOI 10.17487/RFC7756, February 2016, <http://www.rfc-editor.org/info/rfc7756>.

Appendix A. Complete SIIT-DC IDC Topology Example

Figure 4 attempts to "tie it all together" and show a more complete SIIT-DC topology, in order to better demonstrate its advantageous properties discussed in Section 1. These are discussed in more detail below.

```
                  /--------------------------------\ /---------------\
                  |          IPv4 Internet         | | IPv6 Internet |
                  \-+----------------------------+-/ \--------+------/
                    |                            |            |
                    | <----------[BGP]---------> |          [BGP]
                    |                            |            |
     +-------<192.0.2.0/24>---------+ +---<192.0.2.0/24>---+  |
     |                        BR #1 | | BR #2              |  |
     | EAM Table:                   | |                    |  |
     | ==========                   | |                    |  |
     | 192.0.2.1,2001:db8:12:34::1  | |                    |  |
     | 192.0.2.2,2001:db8:12:34::2  | |  Exactly the same  |  |
     | 192.0.2.3,2001:db8:fe:dc::1  | |  configuration as  |  |
     | 192.0.2.4,2001:db8:12:34::4  | |  BR #1             |  |
     | 192.0.2.5,2001:db8:fe:dc::e  | |                    |  |
     |                              | |                    |  |
     | XLAT Prefix 2001:db8:46::/96 | |                    |  |
     |                              | |                    |  |
     +--------<2001:db8:46::/96>----+ +-<2001:db8:46::/96>-+  |
                       |                      |               |
                       | <------[ECMP]------> |               |
                       |                      |               |
     /-----------------+----------------------+--\            |
     |         IPv6 IDC network w/ OSPFv3        +------------/
     \-+--------------------------------+--------/
       |                                |
       | Tenant A’s server LAN          | Tenant B’s server LAN
       | 2001:db8:12:34::/64            | 2001:db8:fe:dc::/64
       |                                |
       +-- www      ::1 (IPv6+SIIT-DC)  +-- www-lb ::1 (IPv6+SIIT-DC)
       |                                |
       +-- mta      ::2 (IPv6+SIIT-DC)  +-- web ::80:01 (IPv6 only)
       |                                |   [...]
       +-- ftp      ::3 (IPv6)          +-- web ::80:99 (IPv6 only)
       |            ::4 (IPv4, via ER)  |
       |                                |         +----+
       +-- app01 ::a:01 (IPv6 only)     \---- ::e | ER | --\
       |   [...]                                  +----+   |
       +-  app99 ::a:99 (IPv6 only)                        |
       |                                  ftp 192.0.2.5 ---/
       +-- db01  ::d:01 (IPv6 only)
       |   [..]
       \-- db99  ::d:99 (IPv6 only)
```

Figure 4: Example SIIT-DC IDC Topology

Single-Stack IPv6 Operation:

As discussed in Section 1.1, SIIT-DC facilitates an IPv6-only IDC network infrastructure. The only places where IPv4 is absolutely required are between the BRs and the IPv4 Internet and between any ERs and the IPv4-only applications or devices they are serving (illustrated here as the two tenants' FTP servers). The figure also illustrates how SIIT-DC does not interfere with native IPv6; when there is no longer a need to support IPv4 clients, the BRs may be decommissioned without causing any impact to native IPv6 traffic.

Stateless Operation:

As discussed in Section 1.2, SIIT-DC operates in a stateless fashion. In the illustration, both BRs are simultaneously advertising (i.e., anycasting) the IPv4 Service Address Pool and the IPv6 translation prefix, so incoming traffic from the IPv4 Internet may arrive at either of the BRs, while outgoing IPv6 traffic destined for IPv4 endpoints are load balanced between them using Equal-Cost Multipath Routing. No continuous state synchronization between the two BRs occurs. Should one of the BRs fail, the BGP and OSPF protocols will ensure that traffic converges on the remaining BR. Existing sessions will not be disrupted beyond any disruption caused by the BGP/OSPF convergence process itself.

IPv4 Address Conservation:

As discussed in Section 1.3, SIIT-DC conserves the IDC operator's IPv4 address space. Even though the two customers in the example above have several hundred servers, the majority of the servers are not used for running services made available directly from the Internet and therefore do not need to consume IPv4 addresses. The IDC network infrastructure consumes no IPv4 addresses, either. Finally, the IPv4 addresses that are assigned to the SIIT-DC function as IPv4 Service Address Pools may be assigned with 100% efficiency, one address at a time; there is no requirement to assign multiple addresses to a single customer in a contiguous block.

Application Support:

As discussed in Section 1.5, as long as the application protocol is translation friendly (illustrated here with HTTP and SMTP), it will work with SIIT-DC without requiring any special adaptation. Furthermore, translation-unfriendly applications (illustrated here with FTP) will also work when located behind an ER [RFC7756]. Tenant A's FTP server illustrates how an ER may be located in the networking stack of a node, while Tenant B's FTP server illustrates how the ER may be deployed as a network service. The latter approach enables SIIT-DC to support IPv4-only nodes/devices.

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, Tobias Gondrom, Christer Holmberg, Dagfinn Ilmari Mannsaaker, Lars Olafsen, Stig Sandbeck Mathisen, Knut A. Syed, Qin Wu, and Andrew Yourtchenko.

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