

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

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1.	Introduction . . . . .	3
1.1.	Motivation and Goals . . . . .	3
1.1.1.	Single Stack IPv6 Operation . . . . .	4
1.1.2.	Stateless Operation . . . . .	4
1.1.3.	IPv4 Address Conservation . . . . .	4
1.1.4.	No Loss of End User's IPv4 Source Address . . . . .	5
1.1.5.	Compatible with Standard IPv6 Implementations . . . . .	5
1.1.6.	No Architectural Dependency on IPv4 . . . . .	6
2.	Terminology . . . . .	6
3.	Architectural Overview . . . . .	7
3.1.	DNS Configuration . . . . .	9
3.2.	Packet Flow . . . . .	9
4.	Deployment Guidelines . . . . .	11
4.1.	Application Support for NAT . . . . .	12
4.2.	Application Support for IPv6 . . . . .	12
4.3.	Application Communication Pattern . . . . .	12
4.4.	Choice of Translation Prefix . . . . .	13
4.5.	Routing Considerations . . . . .	13
4.6.	Location of the SIIT-DC Gateways . . . . .	14
4.7.	Migration from Dual Stack . . . . .	15
4.8.	Packet Size and Fragmentation Considerations . . . . .	15
4.8.1.	IPv4/IPv6 Header Size Difference . . . . .	16
4.8.2.	IPv6 Atomic Fragments . . . . .	16
4.8.3.	Minimum Path MTU Difference Between IPv4 and IPv6 . . . . .	16
5.	Implementation Requirements . . . . .	18
5.1.	Compliance with RFC6145 and RFC6052 . . . . .	18
5.2.	Static Address Mapping Function . . . . .	18
5.3.	Support for Increasing the IPv6 Path MTU . . . . .	19
5.4.	Loop Prevention Mechanism . . . . .	19
6.	Acknowledgements . . . . .	20
7.	Requirements Language . . . . .	20
8.	IANA Considerations . . . . .	20
9.	Security Considerations . . . . .	20
9.1.	Mistaking the Translation Prefix for a Trusted Network . . . . .	20
9.2.	Packets Looping Through the SIIT-DC Function . . . . .	20
10.	References . . . . .	21
10.1.	Normative References . . . . .	21
10.2.	Informative References . . . . .	21
	Appendix A. Complete SIIT-DC topology example . . . . .	23
	Appendix B. Comparison to Other Deployment Approaches . . . . .	26
B.1.	IPv4-only . . . . .	26
B.2.	IPv4-only + NAT44 . . . . .	26
B.3.	IPv4-only + NAT64 . . . . .	28

B.4. Dual Stack . . . . .	29
Author's Address . . . . .	30

## 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:

- o Promote the deployment of native IPv6 services (cf. [RFC6540]).
- o Provide IPv4 service availability for legacy users with no loss of performance or functionality.
- o To ensure that that the legacy users' IPv4 addresses remain available to the servers and applications.
- o To conserve and maximise the utilisation of IPv4 addresses.
- o To avoid introducing more complexity than absolutely necessary, especially on the servers and applications.
- o 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 invisible 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 communicate with this address directly without invoking SIIT-DC. IPv4-only clients also communicate 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





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

```

+-----+
| IP Version:           4 |
| Source Address:      203.0.113.50 |
| Destination Address: 192.0.2.1 |
| Protocol:           TCP |
+-----+
| TCP SYN [...] |
+-----+

```

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

```

+-----+
| IP Version:           6 |
| Source Address:      2001:db8:46::203.0.113.50 |
| Destination Address: 2001:db8:12:34::1 |
| Next Header:        TCP |
+-----+
| TCP SYN [...] |
+-----+

```

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

```

+-----+
| IP Version:           6
| Source Address:      2001:db8:12:34::1
| Destination Address: 2001:db8:46::203.0.113.50
| Next Header:         TCP
+-----+
| TCP SYN+ACK [...]
+-----+

```

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

```

+-----+
| IP Version:           4
| Source Address:      192.0.2.2
| Destination Address: 203.0.113.50
| Protocol:            TCP
+-----+
| TCP SYN+ACK [...]
+-----+

```

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 it 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 between 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 turn 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:

- o 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.
- o 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 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:

- o 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.
- o 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:

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

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

- o 192.0.2.1 => 2001:db8:12:34::1
- o 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] Anderson, 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:

- o 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.
- o 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.
- o 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].
- o 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





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:

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

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

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

## B.2. IPv4-only + NAT44

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 NAT44 [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 + NAT44 operation compared to SIIT-DC include:

- o Compatible with IPv4-only devices.

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

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

- o Does not provide any form of IPv6 availability.
- o Requires network devices that track all flow state, which may create a performance bottleneck and be an easy target for Denial of Service attacks.
- o Limits routing flexibility (prevents closest exit routing), as outbound traffic must pass across the same NAPT44 device that handled the inbound traffic.
- o Limited potential for horizontal scaling, as packets cannot be load-balanced across multiple NAT devices.
- o Depending on whether or not the NAPT44 device rewrites source addresses in order to attract the return traffic to itself:
- o
  - \* 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:

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

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

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

- o 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:

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

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

- o Does not alleviate IPv4 address scarcity (assuming NAPT44 isn't used).
- o 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.
- o 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.

- o 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](mailto:tore@redpill-linpro.com)

IPv6 Operations  
Internet-Draft  
Intended status: Standards Track  
Expires: March 19, 2015

T. Anderson  
Redpill Linpro  
September 15, 2014

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

Abstract

This document describes an extension of the SIIT-DC [I-D.anderson-v6ops-siit-dc] architecture, which allows applications that are incompatible with IPv6, SIIT-DC and/or Network Address Translation in general to operate correctly in an SIIT-DC environment. This is accomplished by introducing a new component called a SIIT-DC Host Agent, which reverses the translations made by an SIIT-DC Gateway. The application is thus provided with seemingly native IPv4 connectivity.

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

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 March 19, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. SIIT-DC Host Agent Specification . . . . .	4
4. Architectural Overview . . . . .	4
5. Deployment Considerations . . . . .	6
5.1. IPv6 Path MTU . . . . .	6
5.2. IPv4 MTU . . . . .	6
6. Acknowledgements . . . . .	6
7. Requirements Language . . . . .	6
8. IANA Considerations . . . . .	7
9. Security Considerations . . . . .	7
9.1. Address Spoofing . . . . .	7
10. References . . . . .	7
10.1. Normative References . . . . .	7
10.2. Informative References . . . . .	8
Author's Address . . . . .	8

## 1. Introduction

SIIT-DC [I-D.anderson-v6ops-siit-dc] describes an architecture where IPv4-only users can access IPv6-only services through a stateless translation gateway. However, this only works for applications that are compatible with Network Address Translation (NAT), due to the fact that the SIIT-DC Gateway will rewrite the addresses in the IP header as part of the translation process. SIIT-DC will also fail to work correctly for applications that make use of legacy IPv4-only socket calls.

This document remedies this problem by defining an extension to SIIT-DC. Translations performed by the SIIT-DC Gateway will also be done in reverse by an SIIT-DC Host Agent running on the server. The resulting IPv4 packets are then passed to the application. This way, the application will be able to use legacy IPv4-only socket calls and /or include references to its own IPv4 address in the application payload, while maintaining correct operation.

The approach is heavily inspired by and very similar to 464XLAT [RFC6877]. The SIIT-DC Host Agent described in this document is almost identical to the CLAT component in 464XLAT, except for the



fact that it will be located on a server, rather than on the customer-side node. Furthermore, an SIIT-DC Host Agent uses statically configured public IP addresses, whereas a 464XLAT CLAT uses a dynamic IPv6 address and a private IPv4 address. The SIIT-DC Gateway described in [I-D.anderson-v6ops-siit-dc] is used instead of the PLAT described by 464XLAT.

## 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.

**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 communicate 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 performing [I-D.anderson-v6ops-siit-dc] translation on packets passing through it. See Section 3.

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

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

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

The SIIT-DC Host Agent runs on the servers hosting application which do not work correctly with the SIIT-DC architecture as specified by [I-D.anderson-v6ops-siit-dc]. Its task is the performing the exact same packet translation as the SIIT-DC Gateway, only in reverse. It therefore shares the same implementation requirements as the SIIT-DC Gateway defined in Section 4 of [I-D.anderson-v6ops-siit-dc], with one exception: The SIIT-DC Host Agent is not required to support configuring an arbitrary number of Static Address Mappings, but it must support at least one.

The SIIT-DC Host Agent must be configured with a Static Address Mapping that corresponds exactly with the same mapping found on the SIIT-DC Gateway. The IPv4 address of the Static Address Mapping (i.e., the IPv4 Service Address) must be configured on a virtual network interface which applications running on the server can bind to, and the server is expected to install a default IPv4 route pointing to this virtual IPv4 interface. The IPv6 address of the Static Address Mapping must be a secondary address that is routed to the server by the IPv6 network. The server must forward all packets it receives destined for this IPv6 address to the SIIT-DC Host Agent.

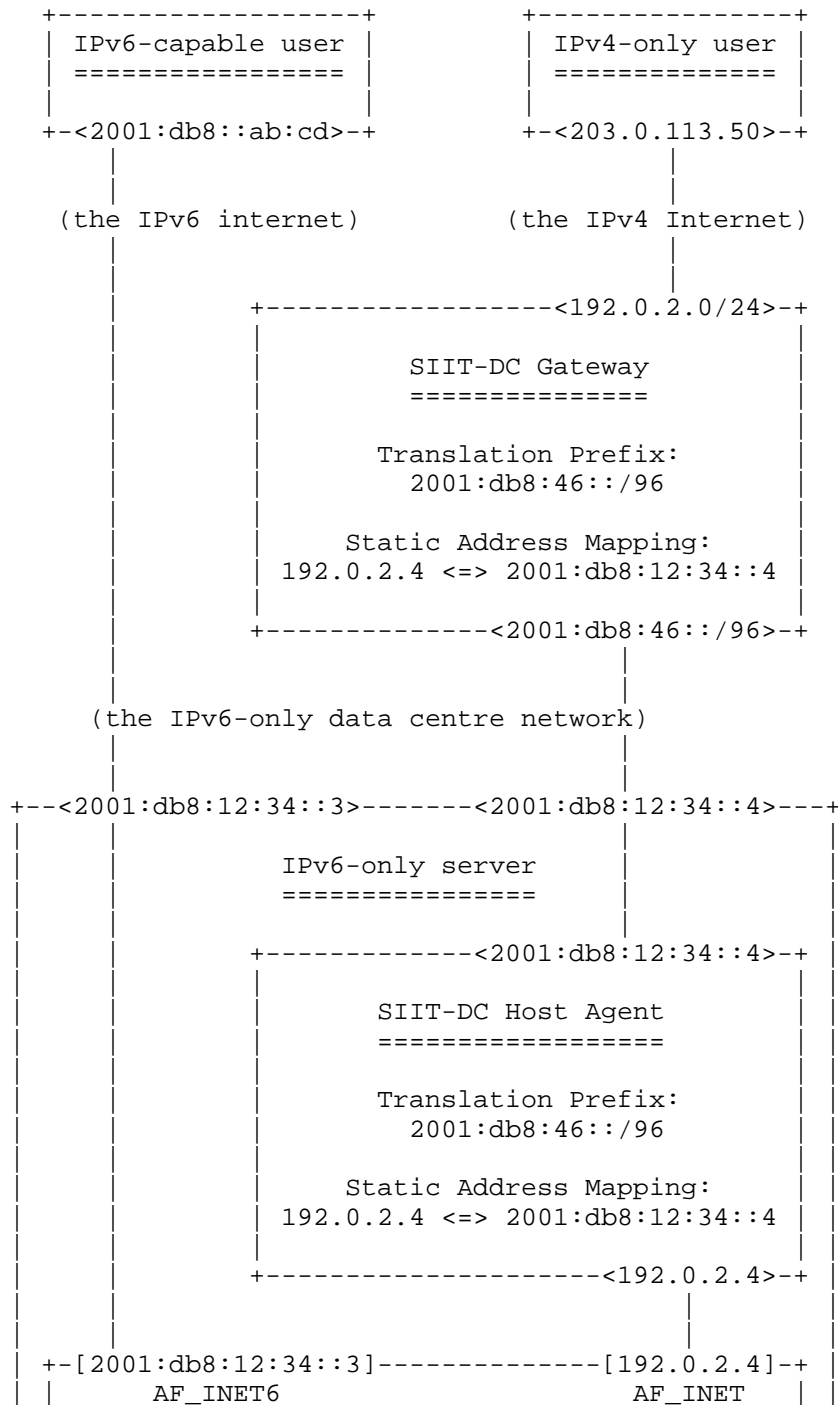
### 4. Architectural Overview

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

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

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

SIIT-DC Host Agent Architecture



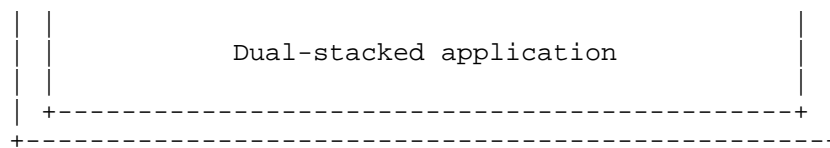


Figure 1

## 5. Deployment Considerations

### 5.1. IPv6 Path MTU

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

### 5.2. IPv4 MTU

In order to avoid fragmentation, it is RECOMMENDED that the virtual IPv4 interface is configured with an MTU value identical to the configured IPv6 Path MTU - 20. This ensures that the application may do its part in avoiding IP-level fragmentation from occurring, e.g., by segmenting/fragmenting outbound packets at the application layer, and advertising the maximum size its peer may use for inbound packets (e.g., through the use of the TCP MSS option).

## 6. Acknowledgements

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

The author would like also to thank the following individuals for their contributions, suggestions, corrections, and criticisms: Fred Baker, Tobias Brox, [YOUR NAME GOES HERE].

## 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

This section discusses security considerations specific to the use of a SIIT-DC Host Agent. See the Security Considerations in I-D .anderson-v6ops-siit-dc [I-D.anderson-v6ops-siit-dc] for additional security considerations applicable to the SIIT-DC architecture in general.

### 9.1. Address Spoofing

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

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

## 10. References

### 10.1. Normative References

- [I-D.anderson-v6ops-siit-dc]  
Anderson, T., "SIIT-DC: Stateless IP/ICMP Translation in IPv6 Data Centre Environments", draft-anderson-v6ops-siit-dc-00 (work in progress), September 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

## 10.2. Informative References

- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, April 2013.

## Author's Address

Tore Anderson  
Redpill Linpro  
Vitaminveien 1A  
0485 Oslo  
NORWAY

Phone: +47 959 31 212  
Email: [tore@redpill-linpro.com](mailto:tore@redpill-linpro.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 30, 2015

G. Chen  
H. Deng  
China Mobile  
October 27, 2014

IPv6 Considerations for Network Function Virtualization (NFV)  
draft-chen-v6ops-nfv-ipv6-00

Abstract

NFV adoption is gaining significant momentum, driven largely by the need to improve service agility and reduce operational cost. IPv6 is a fundamental feature should be enabled. This memo describes the layered NFV components and typical implementations. The IPv6 considerations have been elaborated to each component in order to consolidate IPv6 demands across entire NFV system.

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 30, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Overview on IPv6 Considerations in the NFV Architecture . . .	3
3. IPv6 Considerations on VIM . . . . .	4
4. IPv6 Considerations on Vitrual Network . . . . .	5
5. IPv6 considerations on Virtualisation Layer . . . . .	6
5.1. IPv6-enable Libvirt . . . . .	7
5.2. IPv6-enable KVM . . . . .	7
5.3. IPv6-enable Linux . . . . .	7
6. IPv6 Considerations on Network Hardware . . . . .	7
7. IPv6 Considerations on VNF . . . . .	8
8. IANA Considerations . . . . .	8
9. Security Considerations . . . . .	8
10. References . . . . .	8
10.1. Normative References . . . . .	8
10.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

Network Virtualization Function (NFV) is a new trend for the telcom industry revolution. It leverages IT infrastructure to take over the telcom functions. Driven largely by the need to improve service agility and reduce operational cost, virtualization has been adopted in the NFV architecture. Server, storage, and network resources are abstracted from their physical functions, e.g. processor, memory, I/O controllers, disks, network and storage switches, etc, into pools of functionality which can be managed functionally regardless of their implementation or location. In other words, all servers, storage, and network devices can be aggregated into independent pools of resources to be used as needed, regardless of the actual implementation of those resources.

Depending on the virtualization, NFV system gains good scalability. However, this expansion also can't survive on the exhausting IPv4 address space. IPv6 is definitely the only way out to this pressing needs, because the larger IP address space makes it easier to manage large cloud infrastructures. The memo intends to enumerate IPv6 considerations regarding to the different components in the NFV architecture. It's expected early adopters could reconsider the way they design NFV cloud network so as to get more scalable and manageable infrastructure.



2. Overview on IPv6 Considerations in the NFV Architecture

European Telecommunications Standards Institute (ETSI) has defined the NFV architecture framework [GS\_NFV\_002]. NFV system has been structured from three main working domain in the high-level framework as shown in the Figure 1.

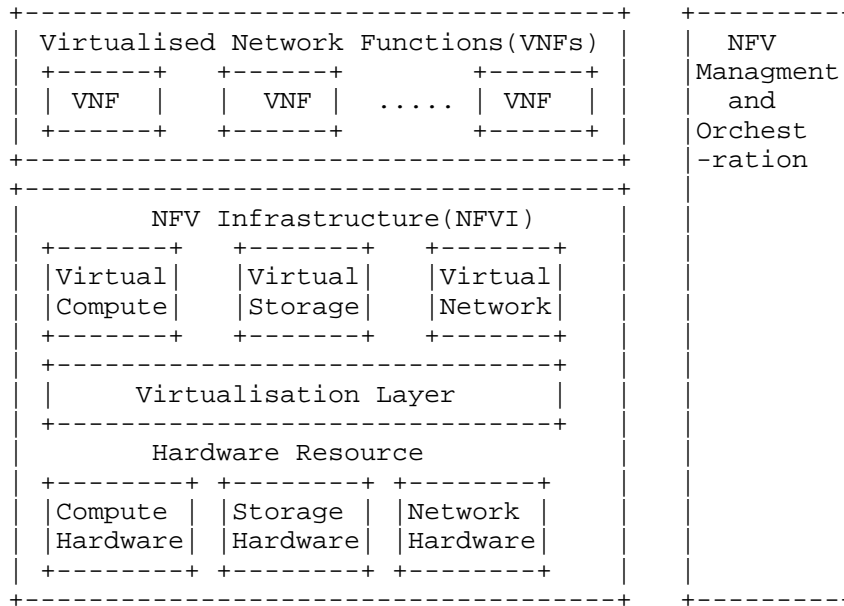


Figure 1: High-Level NFV Framework

We try to document the effort made to enable IPv6 across all components as illustrated in the overall NFV architecture. The Figure 2 lists components, which should take specific considerations to perform IPv6 functions. For each component, a typical implementation has been exemplified. Those implementations are leveraged towards the realization of IPv6-capable NFV system.

NFV Components	Implementations Instance
VI Management	Openstack
Virtual Network	OpenDayLight, OpenVSwitch
Virtualisation Layer	KVM, Libvirt, Linux Kernel
Network Hardware	DPDK
VNF	OpenEPC

Figure 2: IPv6 Relevant NFV Components

### 3. IPv6 Considerations on VIM

Virtualised Infrastructure Management (VIM) comprises the functionalities that are used to control and manage the interaction of a VNF with computing, storage and network resources under its authority, as well as their virtualisation. We can clearly see OpenStack gaining more and more traction. Openstack is composed by several core projects, e.g., Compute (Nova), Network (Neutron), Image (Glance), Object Storage (Swift) and Block Storage (Cinder) and etc. The major concerns of IPv6 capability should be implemented into the Neutron project. Neutron could offers sophisticated networking functionality to coordinate network resources. Numerous IPv6 features could be merged into Neutron.

In general, Neutron is responsible for all topologies work in a multi-tenant environment. IPv6 enable Neutron is able to allow IPv6 address static configuration and auto assignment. The internal IPv6 communications between Virtual Machines (VMs) and external IPv6 interconnection via Neutron and external router/border gateway should be supported. The following considerations facilitate the IPv6 communications goals:

- o Address Management: several IPv6 configuration modes such as SLAAC [RFC4862], DHCPv6 Stateless [RFC3736] and DHCPv6 Stateful [RFC3315] are recommended to be supported. It includes the ability for a user to create a port on a IPv6 subnet and assign a specific IPv6 address or multiple IPv6 addresses to the port and have it taken out the DHCP address pool. Prefix delegation is also expected to be used to automatically configure neutron routers with prefixes so that IPv6 prefixes are obtained and renumbering can be done automatically.

- o External IPv6 Interconnections: IPv6 subnet could be routed via Layer 3 (L3) agent to an external IPv6 network. Both VLAN and overlay (e.g. GRE, VXLAN) subnet attached to VMs can be used to support multiple L3 agents for a given external network to support scaling. Neutron scheduler could be used to assign virtual routers to the L3 agents. Openstack takes the concept of floating IP to allow internal servers to be accessed from external networks. That is the normal cases in IPv4. Given the large address space that IPv6 offers, the floating IP may be unnecessary. End-to-end native IPv6 is more desirable than any of the transition solutions.
- o Floating IP: Floating IP is used in Openstack to make internal servers to be accessible from external Internet. Floating IP support for IPv6 Addresses could be used for internal IPv6 connecting to external IPv6.
- o Security Group: security group is set to interrogate and/or disallow IP flows. Full support for IPv6 TCP/UDP/ICMP in IPv6 security groups are necessary in a IPv6 environment.
- o User Interface and Command Line (CLI): it's important for users to manipulate networks with IPv6 features. During the network, subnet, router creation, it should have the option to allow user to specify the type of address management they would like. This includes the supports via Neutron API (Restful and CLI) as well as via Openstack UI (i.e., Horizon). It's also essential to enable that feature to be able to specify Floating IPs via Neutron API (restful and CLI) and control and manage all IPv6 security group capabilities via Neutron/Nova API (restful and CLI) .

#### 4. IPv6 Considerations on Virtual Network

Virtual Networks is used to isolate resources and network overlays. It could be orchestrated by Openstack Neutron to link network resources to be able to better address the requirements of rich multi-tenant environments. In order to make system more scaleable, Neutron adopts a plug-in model for various 3rd party components to provide the networking service. New technologies (e.g., software-defined networking (SDN)) are emerging to increase the flexibility and agility of the network, decoupling the control from the forwarding plane to make it easier to provision, automate and orchestrate network services. The OpenDaylight provides a plugin and a corresponding agent to enable integration with Neutron. IPv6 demands should also be considered in OpenDaylight softwares including a pluggable controller, interfaces and applications.

The target of IPv6-enable OpenDaylight is to make the overlay and underlay networks in the cloud architecture both being developed with IPv6. The OpenDaylight project, like OpenFlow, is a good initiative to accelerate the IPv6 transition. OpenFlow v1.3 could dynamically learn the Layer 3 IPv6 hosts. This can be facilitated by supporting the IPv6 Neighbor Discovery Protocol (NDP) or supporting DHCPv6. In this case, the OpenDaylight controller should has the ability to perform matching on IPv6 packets and pushes down a flow-table entry to each of the edge devices enabling the forwarding of these packets up to the controller or application to process.

Each of the underlay devices would need to support the optional IPv6 features of OpenFlow and support the required combinations of match/action on the IPv6 header. This also includes the ability to support masking of address fields. Open vSwitch (OVS) is a typical effort to enable the IPv6 process with the overwhelming superiority, such as flexible controller in user-space and fast datapath in kernel. A IPv6-enable vSwitch should be able to support IPv6 flows via OpenFlow. The flow could be identified by the combination of any IPv6 features, such as IPv6 ND target, IPv6 source address or IPv6 destination address. The implementation of OVS would the dedicated IPv6 module to enable IPv6 forwarding.

## 5. IPv6 considerations on Virtualisation Layer

The virtualisation layer abstracts the hardware resources and decouples the VNF software from the underlying hardware. It enables the software that implements the VNF to use the underlying virtualised infrastructure. Typically, this type of functionality is provided for computing and storage resources in the form of hypervisors. In order to facilitate the management of different kinds of hypervisors, libvirt virtualization API is created to provide management tool for managing platform virtualization. The Figure 3 elaborates the relations of different components in virtualisation layer. The sub-section will describe the detailed consideration for each one.

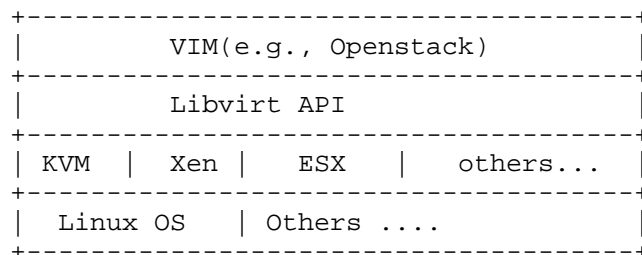


Figure 3: Virtualisation Layer Components

### 5.1. IPv6-enable Libvirt

Libvirt could provide a common and stable layer sufficient to securely manage VNF instances. libvirt provides all APIs needed to do the management, such as provision, create, modify, monitor, control, migrate and stop the instances. IPv6 network configurations can be enabled by the libvirt networking APIs, which is formulated by network XML format. Libvirt define network profile from different elements including general metadata, connectivity and addressing. To enable IPv6, each attributes should be configured properly. For the IPv6 addressing, Libvirt could take SLAAC as default and optionally enable DHCP services. Libvirt could configure static routes for IPv6 forwarding, but lack of supports for dynamic routing protocol.

### 5.2. IPv6-enable KVM

KVM should provide same operations corresponding to Libvirt. It may be straight forward to enable IPv6 on KVM guests by configure the host machine and interfaces with IPv6 address. The necessary firewall rules could be also added to ip6tables on the host machine. NDIS driver in KVM also should be able to handle the IPv6 packages.

### 5.3. IPv6-enable Linux

Linux system should have to enable the IPv6 support in the kernel. Some interface configuration file should add IPv6 address information and restart the networking. Other consideration is the MTU setting. The MTU size of the NIC on Linux defaults to 1500 bytes. It may be good to support Jumbo frames in the cloud infrastructure. Large MTU size not only gives you better network performance, but also provides you with workaround for software issues. It has been observed that many IPv6 packages may exceed 1500-bytes. Therefore, it's very important to enable jumbo frames to avoid the corruption.

## 6. IPv6 Considerations on Network Hardware

Network hardware is capable of high-performance packet processing. There are optimized data plane solutions for the IP package processing. The Intel Data Plane Development Kit (DPDK) is a set of optimized software libraries and drivers, that enable high-performance data plane on network elements. The IPv6 demands to DPDK are targeted to support IPv6 forwarding, including IPv6 fragmentation reassembly. For the fast path, it would support IPv6 exact match flow classification.

## 7. IPv6 Considerations on VNF

The traditional mobile node functions would gradually be migrated to Virtual Network Function (VNF). Examples of VNF are 3GPP Evolved Packet Core (EPC) network elements, e.g., Mobility Management Entity (MME), Serving Gateway (SGW), Packet Data Network Gateway (PGW). VNF may remodel the network node functions into the different instances. For examples, the IPv6 relevant functions of SGW/PGW include PDN signaling processing, IPv6 data-plane filtering, classification, forwarding and IPv6 Charging control. Those IPv6 processing should also be supported in the new-built VNF instances.

## 8. IANA Considerations

This document makes no request of IANA.

## 9. Security Considerations

TBD

## 10. References

### 10.1. Normative References

[GS\_NFV\_002]

European Telecommunications Standards Institute, ETSI.,  
"Network Functions Virtualisation (NFV); Architectural  
Framework", March 2009.

### 10.2. Informative References

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C.,  
and M. Carney, "Dynamic Host Configuration Protocol for  
IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol  
(DHCP) Service for IPv6", RFC 3736, April 2004.

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

Authors' Addresses

Gang Chen  
China Mobile  
53A,Xibianmennei Ave.,  
Xuanwu District,  
Beijing 100053  
China

Email: phdgang@gmail.com

Hui Deng  
China Mobile  
53A,Xibianmennei Ave.,  
Xuanwu District,  
Beijing 100053  
China

Email: denghui@chinamobile.com

IPv6 Operations Working Group (v6ops)  
Internet-Draft  
Intended status: Informational  
Expires: September 9, 2015

F. Gont  
SI6 Networks / UTN-FRH  
J. Linkova  
Google  
T. Chown  
University of Southampton  
W. Liu  
Huawei Technologies  
March 8, 2015

Observations on IPv6 EH Filtering in the Real World  
draft-gont-v6ops-ipv6-ehs-in-real-world-02

Abstract

This document presents real-world data regarding the extent to which packets with IPv6 extension headers are filtered in the Internet (as measured in August 2014), and where in the network such filtering occurs. The aforementioned results serve as a problem statement that is expected to trigger operational advice on the filtering of IPv6 packets carrying IPv6 Extension Headers, so that the situation improves over time. This document also explains how the aforementioned results were obtained, such that the corresponding measurements can be reproduced by other members of the community.

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 September 9, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.



This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Support of IPv6 Extension Headers in the Internet . . . . .	3
3. IANA Considerations . . . . .	6
4. Security Considerations . . . . .	6
5. Acknowledgements . . . . .	7
6. References . . . . .	7
6.1. Normative References . . . . .	7
6.2. Informative References . . . . .	8
Appendix A. Reproducing Our Experiment . . . . .	9
A.1. Obtaining the List of Domain Names . . . . .	9
A.2. Obtaining AAAA Resource Records . . . . .	9
A.3. Filtering the IPv6 Address Datasets . . . . .	10
A.4. Performing Measurements with Each IPv6 Address Dataset . . . . .	10
A.5. Obtaining Statistics from our Measurements . . . . .	11
Appendix B. Measurements Caveats . . . . .	13
B.1. Isolating the Dropping Node . . . . .	13
B.2. Obtaining the Responsible Organization for the Packet Drops . . . . .	14
Appendix C. Troubleshooting Packet Drops due to IPv6 Extension Headers . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

IPv6 Extension Headers (EHs) allow for the extension of the IPv6 protocol, and provide support for core functionality such as IPv6 fragmentation. While packets employing IPv6 Extension Headers have been suspected to be dropped in some IPv6 deployments, there was not much concrete data on the topic. Some preliminary measurements have been presented in [PMTUD-Blackholes], [Gont-IEPG88] and [Gont-Chown-IEPG89], whereas [Linkova-Gont-IEPG90] presents more comprehensive results on which this document is based.

This document presents real-world data regarding the extent to which IPv6 Extension Headers are filtered in the Internet, as measured in August 2014 (pending operational advice in this area).

## 2. Support of IPv6 Extension Headers in the Internet

This section summarizes the results obtained when measuring the support of IPv6 Extension Headers on the path towards different types of public IPv6 servers. Two sources were employed for the list of public IPv6 servers: the "World IPv6 Launch Day" site (<http://www.worldipv6launch.org/>) and Alexa's top 1M web sites (<http://www.alexa.com>). For each list of domain names, the following datasets were obtained:

- o Web servers (AAAA records of the aforementioned list)
- o Mail servers (MX -> AAAA of such list)
- o Name servers (NS -> AAAA of such list)

IPv6 addresses other than global unicast addresses and duplicate addresses were eliminated from each of those lists prior to obtaining the results included in this document. Additionally, addresses that were found to be unreachable were discarded from the dataset (please see Appendix B for further details).

For each of the aforementioned address sets, three different types of probes were performed:

- o IPv6 packets with a Destination Options header of 8 bytes
- o IPv6 packets resulting in two IPv6 fragments of 512 bytes each (approximately)
- o IPv6 packets with a Hop-by-Hop Options header of 8 bytes

In the case of packets with Destination Options Header and Hop-by-Hop Options header, the desired EH size was achieved by means of PadN options [RFC2460]. The upper-layer protocol of the probe packets was, in all cases, TCP [RFC0793] segments with the Destination Port set to the service port [IANA-PORT-NUMBERS] of the corresponding dataset. For example, the probe packets for all the measurements involving web servers were TCP segments with the destination port set to 80.

Besides obtaining the packet drop rate when employing the aforementioned IPv6 extension headers, we tried to identify whether the Autonomous System (AS) dropping the packets was the same as the Autonomous System of the destination/target address. This is of particular interest since it essentially reveals whether the packet drops are under the control of the intended destination of the packets. Packets dropped by the destination AS are less of a

concern, since the device dropping the packets is under the control of the same organization as that to which the packets are destined (hence, it is probably easier to update the filtering policy if deemed necessary). On the other hand, packets dropped by transit ASes are more of a concern, since they affect the deployability and usability of IPv6 extension headers (including IPv6 fragmentation) by a third-party (the destination AS). In any case, we note that it is impossible to tell whether, in those cases where IPv6 packets with extension headers get dropped, the packet drops are the result of an explicit and intended policy, or the result of improper device configuration defaults, buggy devices, etc. Thus, packet drops that occur at the destination AS might still prove to be problematic.

Since there is some ambiguity when identifying the autonomous system to which a specific router belongs, our measurements result in a percentage *\*range\** (see Appendix B.2). In the following tables, the values shown within parentheses represent the estimated range of possibility that when a packet is dropped, the packet drop occurs in an AS other than the destination AS.

Dataset	DO8	HBH8	FH512
Webservers	11.88% (17.60%-20.80%)	40.70% (31.43%-40.00%)	30.51% (5.08%-6.78%)
Mailservers	17.07% (6.35%-26.98%)	48.86% (40.50%-65.42%)	39.17% (2.91%-12.73%)
Nameservers	15.37% (14.29%-33.46%)	43.25% (42.49%-72.07%)	38.55% (3.90%-13.96%)

Table 1: WIPv6LD dataset: Packet drop rate for different destination types, and estimated percentage of dropped packets that were deemed to be dropped in a different AS (lower, in parentheses)

NOTE: As an example, we note that the cell describing the support of IPv6 packets with DO8 for webservers (containing the value "11.88% (17.60%-20.80%)") should be read as: "when sending IPv6 packets with DO8 to public webservers, 11.88% of such packets get dropped. Among those packets that get dropped, between 17.60%-20.80% of them get dropped at an AS other than the destination AS".

EH Type	Webservers	Mailservers	Nameservers
DO8	11.88% (17.60%-20.80%)	17.07% (6.35%-26.98%)	15.37% (14.29%-33.46%)
HBH8	40.70% (31.43%-40.00%)	48.86% (40.50%-65.42%)	43.25% (42.49%-72.07%)
FH512	30.51% (5.08%-6.78%)	39.17% (2.91%-12.73%)	38.55% (3.90%-13.96%)

Table 2: WIPv6LD dataset: Packet drop rate for different EH types, and estimated percentage of dropped packets that were deemed to be dropped in a different AS (lower, in parentheses)

NOTE: This table contains the same information as Table 1, but makes it easier to obtain the drop rates for each EH type. Each cell should be read in exactly the same way as each cell in Table 1.

Dataset	DO8	HBH8	FH512
Webservers	10.91% (46.52%-53.23%)	39.03% (36.90%-46.35%)	28.26% (53.64%-61.43%)
Mailservers	11.54% (2.41%-21.08%)	45.45% (41.27%-61.13%)	35.68% (3.15%-10.92%)
Nameservers	21.33% (10.27%-56.80%)	54.12% (50.64%-81.00%)	55.23% (5.66%-32.23%)

Table 3: Alexa's top 1M sites dataset: Packet drop rate for different destination types, and estimated percentage of dropped packets that were deemed to be dropped in a different AS (lower, in parentheses)

EH Type	Webservers	Mailservers	Nameservers
DO8	10.91% (46.52%-53.23%)	11.54% (2.41%-21.08%)	21.33% (10.27%-56.80%)
HBH8	39.03% (36.90%-46.35%)	45.45% (41.27%-61.13%)	54.12% (50.64%-81.00%)
FH512	28.26% (53.64%-61.43%)	35.68% (3.15%-10.92%)	55.23% (5.66%-32.23%)

Table 4: Alexa's top 1M sites dataset: Packet drop rate for different EH types, and estimated percentage of dropped packets that were deemed to be dropped in a different AS (lower, in parentheses)

NOTE: This table contains the same information as Table 3, but makes it easier to obtain the drop rates for each EH type. Each cell should be read in exactly the same way as each cell in Table 3.

There are a number of observations to be made based on the results presented above. Firstly, while it has been generally assumed that it is IPv6 fragments that are dropped by operators, our results indicate that it is IPv6 extension headers in general that result in packet drops. Secondly, our results indicate that a significant percentage of such packet drops occur in transit Autonomous Systems; that is, the packet drops are not under the control of the same organization as the final destination.

### 3. IANA Considerations

There are no IANA registries within this document. The RFC-Editor can remove this section before publication of this document as an RFC.

### 4. Security Considerations

This document presents real-world data regarding the extent to which IPv6 packets employing extension headers are filtered in the Internet. As such, this document does not introduce any new security issues.

## 5. Acknowledgements

The authors would like to thank (in alphabetical order) Mark Andrews, Fred Baker, Brian Carpenter and Tatuya Jinmei for providing valuable comments on earlier versions of this document. Additionally, the authors would like to thank participants of the v6ops and opsec working groups for their valuable input on the topics discussed in this document.

The authors would like to thank Fred Baker for his guidance in improving this document.

Fernando Gont would like to thank Jan Zorz and Go6 Lab <<http://go6lab.si/>> for providing access to systems and networks that were employed to produce some of the measurement results presented in this document. Additionally, he would like to thank SixXS <<https://www.sixxs.net>> for providing IPv6 connectivity.

## 6. References

### 6.1. Normative References

- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6946] Gont, F., "Processing of IPv6 "Atomic" Fragments", RFC 6946, May 2013.

## 6.2. Informative References

## [Gont-Chown-IEPG89]

Gont, F. and T. Chown, "A Small Update on the Use of IPv6 Extension Headers", IEPG 89. London, UK. March 2, 2014, <<http://www.iepg.org/2014-03-02-ietf89/fgont-iepg-ietf89-eh-update.pdf>>.

## [Gont-IEPG88]

Gont, F., "Fragmentation and Extension header Support in the IPv6 Internet", IEPG 88. Vancouver, BC, Canada. November 13, 2013, <<http://www.iepg.org/2013-11-ietf88/fgont-iepg-ietf88-ipv6-frag-and-eh.pdf>>.

## [IANA-PORT-NUMBERS]

IANA, "Service Name and Transport Protocol Port Number Registry", <<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt>>.

## [IPv6-Toolkit]

"SI6 Networks' IPv6 Toolkit", <<http://www.si6networks.com/tools/ipv6toolkit>>.

## [Linkova-Gont-IEPG90]

Linkova, J. and F. Gont, "IPv6 Extension Headers in the Real World v2.0", IEPG 90. Toronto, ON, Canada. July 20, 2014, <<http://www.iepg.org/2014-07-20-ietf90/iepg-ietf90-ipv6-ehs-in-the-real-world-v2.0.pdf>>.

## [PMTUD-Blackholes]

De Boer, M. and J. Bosma, "Discovering Path MTU black holes on the Internet using RIPE Atlas", July 2012, <<http://www.nlnetlabs.nl/downloads/publications/pmtu-black-holes-msc-thesis.pdf>>.

[RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, July 2010.

[RFC6980] Gont, F., "Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery", RFC 6980, August 2013.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, December 2013.

[RFC7113] Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", RFC 7113, February 2014.

[RFC7123] Gont, F. and W. Liu, "Security Implications of IPv6 on IPv4 Networks", RFC 7123, February 2014.

[blackhole6] blackhole6, , "blackhole6 tool manual page", <<http://www.si6networks.com/tools/ipv6toolkit>>, 2014.

[path6] path6, , "path6 tool manual page", <<http://www.si6networks.com/tools/ipv6toolkit>>, 2014.

## Appendix A. Reproducing Our Experiment

This section describes, step by step, how to reproduce the experiment with which we obtained the results presented in this document. Each subsection represents one step in the experiment. The tools employed for the experiment are traditional UNIX-like tools (such as gunzip), and the SI6 Networks' IPv6 Toolkit [IPv6-Toolkit].

### A.1. Obtaining the List of Domain Names

The primary data source employed was Alexa's Top 1M web sites, available at: <<http://s3.amazonaws.com/alexa-static/top-1m.csv.zip>>. The file is a zipped file containing the list of the most popular web sites, in CSV format. The aforementioned file can be extracted with "gunzip < top-1m.csv.zip > top-1m.csv".

A list of domain names (i.e., other data stripped) can be obtained with the following command of [IPv6-Toolkit]: "cat top-1m.csv | script6 get-alexa-domains > top-1m.txt". This command will create a "top-1m.txt" file, containing one domain name per line.

NOTE: The domain names corresponding to the WIPv6LD dataset is available at: <<http://www.si6networks.com/datasets/wipv6day-domains.txt>>. Since the corresponding file is a text file containing one domain name per line, the steps produced in this subsection need not be performed. The WIPv6LD data set should be processed in the same way as the Alexa Dataset, starting from Appendix A.2.

### A.2. Obtaining AAAA Resource Records

The file obtained in the previous subsection contains a list of domain names that correspond to web sites. The AAAA records for such domains can be obtained with:

```
$ cat top-1m.txt | script6 get-aaaa > top-1m-web-aaaa.txt
```



The AAAA records corresponding to the mailservers of each of the aforementioned domain names can be obtained with:

```
$ cat top-lm.txt | script6 get-mx | script6 get-aaaa > top-lm-mail-aaaa.txt
```

The AAAA records corresponding to the nameservers of each of the aforementioned domain names can be obtained with:

```
$ cat top-lm.txt | script6 get-ns | script6 get-aaaa > top-lm-dns-aaaa.txt
```

### A.3. Filtering the IPv6 Address Datasets

The lists of IPv6 addresses obtained in the previous step could possibly contain undesired addresses (i.e., non-global unicast addresses) and/or duplicate addresses. In order to remove both undesired and duplicate addresses each of the three files from the previous section should be filtered accordingly:

```
$ cat top-lm-web-aaaa.txt | addr6 -i -q -B multicast -B unspec -k global > top-lm-web-aaaa-unique.txt
```

```
$ cat top-lm-mail-aaaa.txt | addr6 -i -q -B multicast -B unspec -k global > top-lm-mail-aaaa-unique.txt
```

```
$ cat top-lm-dns-aaaa.txt | addr6 -i -q -B multicast -B unspec -k global > top-lm-dns-aaaa-unique.txt
```

### A.4. Performing Measurements with Each IPv6 Address Dataset

#### A.4.1. Measurements with web servers

In order to measure DO8 with the list of webservers:

```
# cat top-lm-web-aaaa-unique.txt | script6 trace6 do8 tcp 80 > > top-lm-web-aaaa-do8-m.txt
```

In order to measure HBH8 with the list of webservers:

```
# cat top-lm-web-aaaa-unique.txt | script6 trace6 hbh8 tcp 80 > > top-lm-web-aaaa-hbh8--m.txt
```

In order to measure FH512 with the list of webservers:

```
# cat top-lm-web-aaaa-unique.txt | script6 trace6 fh512 tcp 80 > > top-lm-web-aaaa-fh512-m.txt
```

#### A.4.2. Measurements with mail servers

In order to measure DO8 with the list of mailservers:

```
# cat top-lm-mail-aaaa-unique.txt | script6 trace6 do8 tcp 25 > top-  
lm-mail-aaaa-do8-m.txt
```

In order to measure HBH8 with the list of webservers:

```
# cat top-lm-mail-aaaa-unique.txt | script6 trace6 hbh8 tcp 25 > top-  
lm-mail-aaaa-hbh8-m.txt
```

In order to measure FH512 with the list of webservers:

```
# cat top-lm-mail-aaaa-unique.txt | script6 trace6 fh512 tcp 25 >  
top-lm-mail-aaaa-fh512-m.txt
```

#### A.4.3. Measurements with DNS servers

In order to measure DO8 with the list of nameservers:

```
# cat top-lm-dns-aaaa-unique.txt | script6 trace6 do8 tcp 53 > top-  
lm-dns-aaaa-do8-m.txt
```

In order to measure HBH8 with the list of webservers:

```
# cat top-lm-dns-aaaa-unique.txt | script6 trace6 hbh8 tcp 53 > top-  
lm-dns-aaaa-hbh8-m.txt
```

In order to measure FH512 with the list of webservers:

```
# cat top-lm-dns-aaaa-unique.txt | script6 trace6 fh512 tcp 53 > top-  
lm-dns-aaaa-fh512-m.txt
```

### A.5. Obtaining Statistics from our Measurements

#### A.5.1. Statistics for Web Servers

In order to compute the statistics corresponding to our measurements of DO8 with the list of webservers:

```
$ cat top-lm-web-aaaa-do8-m.txt | script6 get-trace6-stats > top-lm-  
web-aaaa-do8-stats.txt
```

In order to compute the statistics corresponding to our measurements of HBH8 with the list of webservers:

```
$ cat top-1m-web-aaaa-hbh8-m.txt | script6 get-trace6-stats > top-1m-  
web-aaaa-hbh8-stats.txt
```

In order to compute the statistics corresponding to our measurements of FH512 with the list of webservers:

```
$ cat top-1m-web-aaaa-fh512-m.txt | script6 get-trace6-stats > top-  
1m-web-aaaa-fh512-stats.txt
```

#### A.5.2. Statistics for Mail Servers

In order to compute the statistics corresponding to our measurements of DO8 with the list of mailservers:

```
$ cat top-1m-mail-aaaa-do8-m.txt | script6 get-trace6-stats > top-1m-  
mail-aaaa-do8-stats.txt
```

In order to compute the statistics corresponding to our measurements of HBH8 with the list of mailservers:

```
$ cat top-1m-mail-aaaa-hbh8-m.txt | script6 get-trace6-stats > top-  
1m-mail-aaaa-hbh8-stats.txt
```

In order to compute the statistics corresponding to our measurements of FH512 with the list of mailservers:

```
$ cat top-1m-mail-aaaa-fh512-m.txt | script6 get-trace6-stats > top-  
1m-mail-aaaa-fh512-stats.txt
```

#### A.5.3. Statistics for Name Servers

In order to compute the statistics corresponding to our measurements of DO8 with the list of nameservers:

```
$ cat top-1m-dns-aaaa-do8-m.txt | script6 get-trace6-stats > top-1m-  
dns-aaaa-do8-stats.txt
```

In order to compute the statistics corresponding to our measurements of HBH8 with the list of mailservers:

```
$ cat top-1m-dns-aaaa-hbh8-m.txt | script6 get-trace6-stats > top-1m-  
dns-aaaa-hbh8-stats.txt
```

In order to compute the statistics corresponding to our measurements of FH512 with the list of mailservers:

```
$ cat top-1m-dns-aaaa-fh512-m.txt | script6 get-trace6-stats > top-  
1m-dns-aaaa-fh512-stats.txt
```

## Appendix B. Measurements Caveats

A number of issues have needed some consideration when producing the results presented in this document. These same issues should be considered when troubleshooting connectivity problems resulting from the use of IPv6 Extension headers.

### B.1. Isolating the Dropping Node

Let us assume that we find that IPv6 packets with EHs are being dropped on their way to the destination system 2001:db8:d::1, and that the output of running traceroute towards such destination is:

1. 2001:db8:1:1000::1
2. 2001:db8:2:4000::1
3. 2001:db8:3:4000::1
4. 2001:db8:3:1000::1
5. 2001:db8:4:4000::1
6. 2001:db8:4:1000::1
7. 2001:db8:5:5000::1
8. 2001:db8:5:6000::1
9. 2001:db8:d::1

Additionally, let us assume that the output of EH-enabled traceroute to the same destination is:

1. 2001:db8:1:1000::1
2. 2001:db8:2:4000::1
3. 2001:db8:3:4000::1
4. 2001:db8:3:1000::1
5. 2001:db8:4:4000::1

For the sake of brevity, let us refer to the last-responding node in the EH-enabled traceroute ("2001:db8:4:4000::1" in this case) as "M". Assuming both packets in both traceroutes employ the same path, we'll refer to "the node following the last responding node in the EH-enabled traceroute" ("2001:db8:4:1000::1" in our case), as "M+1", etc.

Based on traceroute information above, which node is the one actually dropping the EH-enabled packets will depend on whether the dropping node filters packets before making the forwarding decision, or after making the forwarding decision. If the former, the dropping node will be M+1. If the latter, the dropping node will be "M".

Throughout this document (and our measurements), we assume that those nodes filtering packets that carry IPv6 EHs apply their filtering policy, and only then, if necessary, forward the packets. Thus, in

our example above the last responding node to the EH-enabled traceroute ("M") is "2001:db8:4:4000::1", and therefore we assume the dropping node to be "2001:db8:4:1000::1" ("M+1").

Additionally, we note that when isolating the dropping node we assume that both the EH-enabled and the EH-free traceroutes result in the same paths. However, this might not be the case.

## B.2. Obtaining the Responsible Organization for the Packet Drops

In order to identify the organization operating the dropping node, one would be tempted to lookup the ASN corresponding to the dropping node. However, assuming that M and M+1 are two peering routers, any of these two organizations could be providing the address space employed for such peering. Or, in the case of an Internet eXchange Point (IXP), the address space could correspond to the IXP AS, rather than to any of the participating ASes. Thus, the organization operating the dropping node (M+1) could be the AS for M+1, but it might as well be the AS for M+2. Only when the ASN for M+1 is the same as the ASN for M+2 we have certainty about who the responsible organization for the packet drops is (see slides 21-23 of [Linkova-Gont-IEPG90]).

In the measurement results presented in Section 2, the aforementioned ambiguity results in "percentage ranges" (rather than a specific ratio): the lowest percentage value means that, when in doubt, we assume the packet drops occur in the same AS as the destination; on the other hand, the highest percentage value means that, when in doubt, we assume the packet drops occur at different AS than the destination AS.

We note that the aforementioned ambiguity should also be considered when troubleshooting and reporting IPv6 packet drops, since identifying the organization responsible for the packet drops might prove to be a non-trivial task.

Finally, we note that a specific organization might be operating more than one Autonomous System. However, our measurements assume that different Autonomous System Numbers imply different organizations.

## Appendix C. Troubleshooting Packet Drops due to IPv6 Extension Headers

Isolating IPv6 blackholes essentially involves performing IPv6 traceroute for a destination system with and without IPv6 extension headers. The (EH-free) traceroute would provide the full working path towards a destination, while the EH-enabled traceroute would provide the address of the last-responding node for EH-enabled packets (say, "M"). In principle, one could isolate the dropping

node by looking-up "M" in the EH-free traceroute, with the dropping node being "M+1" (see Appendix B.1 for caveats).

At the time of this writing, most traceroute implementations do not support IPv6 extension headers. However, the path6 tool [path6] of [IPv6-Toolkit] provides such support. Additionally, the blackhole6 tool [blackhole6] automates the troubleshooting process and can readily provide information such as: dropping node's IPv6 address, dropping node's Autonomous System, etc.

#### Authors' Addresses

Fernando Gont  
SI6 Networks / UTN-FRH  
Evaristo Carriego 2644  
Haedo, Provincia de Buenos Aires 1706  
Argentina

Phone: +54 11 4650 8472  
Email: fgont@si6networks.com  
URI: <http://www.si6networks.com>

J. Linkova  
Google  
1600 Amphitheatre Parkway  
Mountain View, CA 94043  
USA

Email: [furry@google.com](mailto:furry@google.com)

Tim Chown  
University of Southampton  
Highfield  
Southampton, Hampshire SO17 1BJ  
United Kingdom

Email: [tjc@ecs.soton.ac.uk](mailto:tjc@ecs.soton.ac.uk)

Will(Shucheng) Liu  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen 518129  
P.R. China

Email: [liushucheng@huawei.com](mailto:liushucheng@huawei.com)

v6ops WG  
Internet-Draft  
Obsoletes: 3068, 6732 (if approved)  
Intended status: Best Current Practice  
Expires: August 1, 2015

O. Troan  
Cisco  
B. Carpenter, Ed.  
Univ. of Auckland  
January 28, 2015

Deprecating Anycast Prefix for 6to4 Relay Routers  
draft-ietf-v6ops-6to4-to-historic-11.txt

Abstract

Experience with the "Connection of IPv6 Domains via IPv4 Clouds (6to4)" IPv6 transition mechanism defined in RFC 3056 has shown that when used in its anycast mode, the mechanism is unsuitable for widespread deployment and use in the Internet. This document therefore requests that RFC 3068, "An Anycast Prefix for 6to4 Relay Routers", be made obsolete and moved to historic status. It also obsoletes RFC 6732 "6to4 Provider Managed Tunnels". It recommends that future products should not support 6to4 anycast and that existing deployments should be reviewed. This complements the guidelines in RFC 6343.

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 August 1, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Related Work . . . . .	3
2. Conventions . . . . .	3
3. 6to4 operational problems . . . . .	3
4. Deprecation . . . . .	4
5. Implementation Recommendations . . . . .	5
6. Operational Recommendations . . . . .	5
7. IANA Considerations . . . . .	6
8. Security Considerations . . . . .	6
9. Acknowledgements . . . . .	6
10. References . . . . .	7
10.1. Normative References . . . . .	7
10.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

The original form of the 6to4 transition mechanism [RFC3056] relies on unicast addressing. However, its extension specified in "An Anycast Prefix for 6to4 Relay Routers" [RFC3068] has been shown to have severe practical problems when used in the Internet. This document requests that RFC 3068 and RFC 6732 be moved to Historic status as defined in section 4.2.4 of [RFC2026]. It complements the deployment guidelines in [RFC6343].

6to4 was designed to help transition the Internet from IPv4 to IPv6. It has been a good mechanism for experimenting with IPv6, but because of the high failure rates seen with anycast 6to4 [HUSTON], end users may end up disabling IPv6 on hosts as a result, and in the past some content providers were reluctant to make content available over IPv6 for this reason.

[RFC6343] analyses the known operational issues in detail and describes a set of suggestions to improve 6to4 reliability, given the widespread presence of hosts and customer premises equipment that support it. The advice to disable 6to4 by default has been widely adopted in recent operating systems, and the failure modes have been widely hidden from users by many browsers adopting the "Happy Eyeballs" approach [RFC6555].



Nevertheless, a measurable amount of 6to4 traffic is still observed by IPv6 content providers. The remaining successful users of anycast 6to4 are likely to be on hosts using the obsolete policy table [RFC3484], which prefers 6to4 above IPv4, and running without Happy Eyeballs. Furthermore, they must have a route to an operational anycast relay and they must be accessing an IPv6 host that has a route to an operational return relay.

However, experience shows that operational failures caused by anycast 6to4 have continued, despite the advice in RFC 6343 being available.

### 1.1. Related Work

IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) [RFC5969] explicitly builds on the 6to4 mechanism, using a service provider prefix instead of 2002::/16. However, the deployment model is based on service provider support, such that 6rd avoids the problems observed with anycast 6to4.

The framework for 6to4 Provider Managed Tunnels [RFC6732] is intended to help a service provider manage 6to4 anycast tunnels. This framework only exists because of the problems observed with anycast 6to4.

## 2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

The word "deprecate" and its derivatives are used only in their generic sense of "criticize or express disapproval" and do not have any specific normative meaning. A deprecated function might exist in the Internet for many years to allow backwards compatibility.

## 3. 6to4 operational problems

6to4 is a mechanism designed to allow isolated IPv6 islands to reach each other using IPv6 over IPv4 automatic tunneling. To reach the native IPv6 Internet the mechanism uses relay routers both in the forward and reverse direction. The mechanism is supported in many IPv6 implementations. With the increased deployment of IPv6, the mechanism has been shown to have a number of shortcomings.

In the forward direction a 6to4 node will send IPv4 encapsulated IPv6 traffic to a 6to4 relay, that is connected both to the 6to4 cloud and to native IPv6. In the reverse direction a 2002::/16 route is

injected into the native IPv6 routing domain to attract traffic from native IPv6 nodes to a 6to4 relay router. It is expected that traffic will use different relays in the forward and reverse direction.

One model of 6to4 deployment, described in section 5.2 of RFC 3056, suggests that a 6to4 router should have a set of managed connections (via BGP connections) to a set of 6to4 relay routers. While this makes the forward path more controlled, it does not guarantee a functional reverse path. In any case this model has the same operational burden as manually configured tunnels and has seen no deployment in the public Internet.

RFC 3068 adds an extension that allows the use of a well known IPv4 anycast address to reach the nearest 6to4 relay in the forward direction. However, this anycast mechanism has a number of operational issues and problems, which are described in detail in Section 3 of [RFC6343]. This document is intended to deprecate the anycast mechanism.

Peer-to-peer usage of the 6to4 mechanism exists in the Internet, likely unknown to many operators. This usage is harmless to third parties and is not dependent on the anycast 6to4 mechanism that this document deprecates.

#### 4. Deprecation

This document formally deprecates the anycast 6to4 transition mechanism defined in [RFC3068] and the associated anycast IPv4 address 192.88.99.1. It is no longer considered to be a useful service of last resort.

The prefix 192.88.99.0/24 MUST NOT be reassigned for other use except by a future IETF standards action.

The basic unicast 6to4 mechanism defined in [RFC3056] and the associated 6to4 IPv6 prefix 2002::/16 are not deprecated. The default address selection rules specified in [RFC6724] are not modified.

In the absence of 6to4 anycast, 6to4 Provider Managed Tunnels [RFC6732] will no longer be necessary, so they are also deprecated by this document.

Incidental references to 6to4 should be reviewed and possibly removed from other IETF documents if and when they are updated. These documents include RFC3162, RFC3178, RFC3790, RFC4191, RFC4213,

RFC4389, RFC4779, RFC4852, RFC4891, RFC4903, RFC5157, RFC5245, RFC5375, RFC5971, RFC6071 and RFC6890.

## 5. Implementation Recommendations

It is NOT RECOMMENDED to include the anycast 6to4 transition mechanism in new implementations. If included in any implementations, the anycast 6to4 mechanism MUST be disabled by default.

In host implementations, unicast 6to4 MUST also be disabled by default. All hosts using 6to4 MUST support the IPv6 address selection policy described in [RFC6724].

In router implementations, 6to4 MUST be disabled by default. In particular, enabling IPv6 forwarding on a device MUST NOT automatically enable 6to4.

## 6. Operational Recommendations

This document does not imply a recommendation for the generalized filtering of traffic or routes for 6to4 or even anycast 6to4. It simply recommends against further deployment of the anycast 6to4 mechanism, calls for current 6to4 deployments to evaluate the efficacy of continued use of the anycast 6to4 mechanism, and makes recommendations intended to prevent any use of 6to4 from hampering broader deployment and use of native IPv6 on the Internet as a whole.

Networks SHOULD NOT filter out packets whose source address is 192.88.99.1, because this is normal 6to4 traffic from a 6to4 return relay somewhere in the Internet. This includes ensuring that traffic from a local 6to4 return relay with a source address of 192.88.99.1 is allowed through anti-spoofing filters such as those described in [RFC2827] and [RFC3704] or through Unicast Reverse-Path-Forwarding (uRPF) checks [RFC5635].

The guidelines in Section 4 of [RFC6343] remain valid for those who choose to continue operating Anycast 6to4 despite its deprecation.

Current operators of an anycast 6to4 relay with the IPv4 address 192.88.99.1 SHOULD review the information in [RFC6343] and the present document, and then consider carefully whether the anycast relay can be discontinued as traffic diminishes. Internet service providers that do not operate an anycast relay but do provide their customers with a route to 192.88.99.1 SHOULD verify that it does in fact lead to an operational anycast relay, as discussed in Section 4.2.1 of [RFC6343]. Furthermore, Internet service providers and other network providers MUST NOT originate a route to

192.88.99.1, unless they actively operate and monitor an anycast 6to4 relay service as detailed in Section 4.2.1 of [RFC6343].

Operators of a 6to4 return relay responding to the IPv6 prefix 2002::/16 SHOULD review the information in [RFC6343] and the present document, and then consider carefully whether the return relay can be discontinued as traffic diminishes. To avoid confusion, note that nothing in the design of 6to4 assumes or requires that return packets are handled by the same relay as outbound packets. As discussed in Section 4.5 of RFC 6343, content providers might choose to continue operating a return relay for the benefit of their own residual 6to4 clients. Internet service providers SHOULD announce the IPv6 prefix 2002::/16 to their own customers if and only if it leads to a correctly operating return relay as described in RFC 6343. IPv6-only service providers, including those operating a NAT64 service [RFC6146], are advised that their own customers need a route to such a relay in case a residual 6to4 user served by a different service provider attempts to communicate with them.

Operators of 6to4 Provider Managed Tunnels [RFC6732] SHOULD carefully consider when this service can be discontinued as traffic diminishes.

## 7. IANA Considerations

The document creating the IANA IPv4 Special-Purpose Address Registry [RFC6890] included the 6to4 relay anycast prefix (192.88.99.0/24) as Table 10. Instead, IANA is requested to mark the 192.88.99.0/24 prefix originally defined by [RFC3068] as "Deprecated (6to4 Relay Anycast)", pointing to the present document. Redefinition of this prefix for any usage requires justification via an IETF Standards Action [RFC5226].

## 8. Security Considerations

There are no new security considerations pertaining to this document. General security issues with tunnels are listed in [RFC6169] and more specifically to 6to4 in [RFC3964] and [RFC6324].

## 9. Acknowledgements

The authors would like to acknowledge Tore Anderson, Mark Andrews, Dmitry Anipko, Jack Bates, Cameron Byrne, Ben Campbell, Lorenzo Colitti, Gert Doering, Nick Hilliard, Philip Homburg, Ray Hunter, Joel Jaeggli, Victor Kuarsingh, Kurt Erik Lindqvist, Jason Livingood, Jeroen Massar, Keith Moore, Tom Petch, Daniel Roesen, Mark Townsley and James Woodyatt for their contributions and discussions on this topic.

Special thanks go to Fred Baker, David Farmer, Wes George, and Geoff Huston for their significant contributions.

Many thanks to Gunter Van de Velde for documenting the harm caused by non-managed tunnels and stimulating the creation of this document.

## 10. References

### 10.1. Normative References

- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [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.
- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.
- [RFC6890] Cotton, M., Vegoda, L., Bonica, R., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, April 2013.

## 10.2. Informative References

- [HUSTON] Huston, , "Flailing IPv6", December 2010, <<http://www.potaroo.net/ispcol/2010-12/6to4fail.html>>.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.
- [RFC3964] Savola, P. and C. Patel, "Security Considerations for 6to4", RFC 3964, December 2004.
- [RFC5635] Kumari, W. and D. McPherson, "Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)", RFC 5635, August 2009.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.
- [RFC6169] Krishnan, S., Thaler, D., and J. Hoagland, "Security Concerns with IP Tunneling", RFC 6169, April 2011.
- [RFC6324] Nakibly, G. and F. Templin, "Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", RFC 6324, August 2011.
- [RFC6343] Carpenter, B., "Advisory Guidelines for 6to4 Deployment", RFC 6343, August 2011.
- [RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.
- [RFC6732] Kuarsingh, V., Lee, Y., and O. Vautrin, "6to4 Provider Managed Tunnels", RFC 6732, September 2012.

## Authors' Addresses

Ole Troan  
Cisco  
Oslo  
Norway

Email: [ot@cisco.com](mailto:ot@cisco.com)

Brian Carpenter (editor)  
Department of Computer Science  
University of Auckland  
PB 92019  
Auckland 1142  
New Zealand

Email: [brian.e.carpenter@gmail.com](mailto:brian.e.carpenter@gmail.com)

V6OPS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: May 17, 2017

P. Matthews  
Nokia  
V. Kuarsingh  
Cisco  
November 13, 2016

Routing-Related Design Choices for IPv6 Networks  
draft-ietf-v6ops-design-choices-12

Abstract

This document presents advice on certain routing-related design choices that arise when designing IPv6 networks (both dual-stack and IPv6-only). The intended audience is someone designing an IPv6 network who is knowledgeable about best current practices around IPv4 network design, and wishes to learn the corresponding practices for IPv6.

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 May 17, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of



the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Design Choices . . . . .	3
2.1. Addresses . . . . .	3
2.1.1. Where to Use Addresses . . . . .	4
2.1.2. Which Addresses to Use . . . . .	6
2.2. Interfaces . . . . .	7
2.2.1. Mix IPv4 and IPv6 on the Same Layer-3 Interface? . . . . .	7
2.3. Static Routes . . . . .	8
2.3.1. Link-Local Next-Hop in a Static Route? . . . . .	8
2.4. IGPs . . . . .	9
2.4.1. IGP Choice . . . . .	9
2.4.2. IS-IS Topology Mode . . . . .	12
2.4.3. RIP / RIPng . . . . .	13
2.5. BGP . . . . .	14
2.5.1. Which Transport for Which Routes? . . . . .	14
2.5.1.1. BGP Sessions for Unlabeled Routes . . . . .	16
2.5.1.2. BGP sessions for Labeled or VPN Routes . . . . .	17
2.5.2. eBGP Endpoints: Global or Link-Local Addresses? . . . . .	18
3. General Observations . . . . .	19
3.1. Use of Link-Local Addresses . . . . .	19
3.2. Separation of IPv4 and IPv6 . . . . .	20
4. IANA Considerations . . . . .	20
5. Security Considerations . . . . .	20
6. Acknowledgements . . . . .	21
7. Informative References . . . . .	21
Authors' Addresses . . . . .	25

## 1. Introduction

This document discusses routing-related design choices that arise when designing an IPv6-only or dual-stack network. The focus is on choices that do not come up when designing an IPv4-only network. The document presents each choice and the alternatives, and then discusses the pros and cons of the alternatives in detail. Where consensus currently exists around the best practice, this is documented; otherwise the document simply summarizes the current state of the discussion. Thus this document serves to both document the reasoning behind best current practices for IPv6, and to allow a designer to make an informed choice where no such consensus exists.

The design choices presented apply to both Service Provider and Enterprise network environments. Where choices have selection criteria which differ between the Service Provider and the Enterprise

environment, this is noted. The designer is encouraged to ensure that they familiarize themselves with any of the discussed technologies to ensure the best selection is made for their environment.

This document does not present advice on strategies for adding IPv6 to a network, nor does it discuss transition in these areas, see [RFC6180] for general advice, [RFC6782] for wireline service providers, [RFC6342] for mobile network providers, [RFC5963] for exchange point operators, [RFC6883] for content providers, and both [RFC4852] and [RFC7381] for enterprises. Nor does this document discuss the particulars of creating an IPv6 addressing plan; for advice in this area, see [RFC5375] or [v6-addressing-plan]. The document focuses on unicast routing design only and does not cover multicast or the issues involved in running MPLS over IPv6 transport

Section 2 presents and discusses a number of design choices. Section 3 discusses some general themes that run through these choices.

## 2. Design Choices

Each subsection below presents a design choice and discusses the pros and cons of the various options. If there is consensus in the industry for a particular option, then the consensus position is noted.

### 2.1. Addresses

This section discusses the choice of addresses for router loopbacks and links between routers. It does not cover the choice of addresses for end hosts.

In IPv6, an interface is always assigned a Link-Local Address (LLA) [RFC4291]. The link-local address can only be used for communicating with devices that are on-link, so often one or more additional addresses are assigned which are able to communicate off-link. This additional address or addresses can be one of three types:

- o Provider-Independent Global Unicast Address (PI GUA): IPv6 address allocated by a regional address registry [RFC4291]
- o Provider-Aggregatable Global Unicast Address (PA GUA): IPv6 Address allocated by your upstream service provider
- o Unique Local Address (ULA): IPv6 address locally assigned [RFC4193]

This document uses the term "multi-hop address" to collectively refer to these three types of addresses.

PI GUAs are, for many situations, the most flexible of these choices. Their main disadvantages are that a regional address registry will only allocate them to organizations that meet certain qualifications, and one must pay an annual fee. These disadvantages mean that many smaller organization may not qualify or be willing to pay for these addresses.

PA GUAs have the advantage that they are usually provided at no extra charge when you contract with an upstream provider. However, they have the disadvantage that, when switching upstream providers, one must give back the old addresses and get new addresses from the new provider ("renumbering"). Though IPv6 has mechanisms to make renumbering easier than IPv4, these techniques are not generally applicable to routers and renumbering is still fairly hard [RFC5887] [RFC6879] [RFC7010]. PA GUAs also have the disadvantage that it is not easy to have multiple upstream providers ("multi-homing") if they are used (see "Ingress Filtering Problem" in [RFC5220]).

ULAs have the advantage that they are extremely easy to obtain and cost nothing. However, they have the disadvantage that they cannot be routed on the Internet, so must be used only within a limited scope. In many situations, this is not a problem, but in certain situations this can be problematic. Though there is currently no document that describes these situations, many of them are similar to those described in [RFC6752]. See also [I-D.ietf-v6ops-ula-usage-recommendations].

Not discussed in this document is the possibility of using the technology described in [RFC6296] to work around some of the limitations of PA GUAs and ULAs.

#### 2.1.1.1. Where to Use Addresses

As mentioned above, all interfaces in IPv6 always have a link-local address. This section addresses the question of when and where to assign multi-hop addresses in addition to the LLA. We consider four options:

- a. Use only link-local addresses on all router interfaces.
- b. Assign multi-hop addresses to all link interfaces on each router, and use only a link-local address on the loopback interfaces.
- c. Assign multi-hop addresses to the loopback interface on each router, and use only a link-local address on all link interfaces.

- d. Assign multi-hop addresses to both link and loopback interfaces on each router.

Option (a) means that the router cannot be reached (ping, management, etc.) from farther than one-hop away. The authors are not aware of anyone using this option.

Option (b) means that the loopback interfaces are effectively useless, since link-local addresses cannot be used for the purposes that loopback interfaces are usually used for. So option (b) degenerates into option (d).

Thus the real choice comes down to option (c) vs. option (d).

Option (c) has two advantages over option (d). The first advantage is ease of configuration. In a network with a large number of links, the operator can just assign one multi-hop address to each router and then enable the IGP, without going through the tedious process of assigning and tracking the addresses on each link. The second advantage is security. Since packets with link-local addresses cannot be should not be routed, it is very difficult to attack the associated nodes from an off-link device. This implies less effort around maintaining security ACLs.

Countering these advantages are various disadvantages to option (c) compared with option (d):

- o It is not possible to ping a link-local-only interface from a device that is not directly attached to the link. Thus, to troubleshoot, one must typically log into a device that is directly attached to the device in question, and execute the ping from there.
- o A traceroute passing over the link-local-only interface will return the loopback address of the router, rather than the address of the interface itself.
- o In cases of parallel point to point links it is difficult to determine which of the parallel links was taken when attempting to troubleshoot unless one sends packets directly between the two attached link-locals on the specific interfaces. Since many network problems behave differently for traffic to/from a router than for traffic through the router(s) in question, this can pose a significant hurdle to some troubleshooting scenarios.
- o On some routers, by default the link-layer address of the interface is derived from the MAC address assigned to interface. When this is done, swapping out the interface hardware (e.g.

interface card) will cause the link-layer address to change. In some cases (peering config, ACLs, etc) this may require additional changes. However, many devices allow the link-layer address of an interface to be explicitly configured, which avoids this issue. This problem should fade away over time as more and more routers select interface identifiers according to the rules in [RFC7217].

- o The practice of naming router interfaces using DNS names is difficult and not recommended when using link-locals only. More generally, it is not recommended to put link-local addresses into DNS; see [RFC4472].
- o It is often not possible to identify the interface or link (in a database, email, etc) by giving just its address without also specifying the link in some manner.

It should be noted that it is quite possible for the same link-local address to be assigned to multiple interfaces. This can happen because the MAC address is duplicated (due to manufacturing process defaults or the use of virtualization), because a device deliberately re-uses automatically-assigned link-local addresses on different links, or because an operator manually assigns the same easy-to-type link-local address to multiple interfaces. All these are allowed in IPv6 as long as the addresses are used on different links.

For more discussion on the pros and cons, see [RFC7404]. See also [RFC5375] for IPv6 unicast address assignment considerations.

Today, most operators use option (d).

#### 2.1.2. Which Addresses to Use

Having considered above whether or not to use a "multi-hop address", we now consider which of the addresses to use.

When selecting between these three "multi-hop address" types, one needs to consider exactly how they will be used. An important consideration is how Internet traffic is carried across the core of the network. There are two main options: (1) the classic approach where Internet traffic is carried as unlabeled traffic hop-by-hop across the network, and (2) the more recent approach where Internet traffic is carried inside an MPLS LSP (typically as part of a L3 VPN).

Under the classic approach:

- o PI GUAs are a very reasonable choice, if they are available.

- o PA GUAs suffer from the "must renumber" and "difficult to multi-home" problems mentioned above.
- o ULAs suffer from the "may be problematic" issues described above.

Under the MPLS approach:

- o PA GUAs are a reasonable choice, if they are available.
- o PA GUAs suffer from the "must renumber" problem, but the "difficult to multi-home" problem does not apply.
- o ULAs are a reasonable choice, since (unlike in the classic approach) these addresses are not visible to the Internet, so the problematic cases do not occur.

## 2.2. Interfaces

### 2.2.1. Mix IPv4 and IPv6 on the Same Layer-3 Interface?

If a network is going to carry both IPv4 and IPv6 traffic, as many networks do today, then a question arises: Should an operator mix IPv4 and IPv6 traffic or keep them separated? More specifically, should the design:

- a. Mix IPv4 and IPv6 traffic on the same layer-3 interface, OR
- b. Separate IPv4 and IPv6 by using separate interfaces (e.g., two physical links or two VLANs on the same link)?

Option (a) implies a single layer-3 interface at each end of the connection with both IPv4 and IPv6 addresses; while option (b) implies two layer-3 interfaces at each end, one for IPv4 addresses and one with IPv6 addresses.

The advantages of option (a) include:

- o Requires only half as many layer 3 interfaces as option (b), thus providing better scaling;
- o May require fewer physical ports, thus saving money and simplifying operations;
- o Can make the QoS implementation much easier (for example, rate-limiting the combined IPv4 and IPv6 traffic to or from a customer);

- o Works well in practice, as any increase in IPv6 traffic is usually counter-balanced by a corresponding decrease in IPv4 traffic to or from the same host (ignoring the common pattern of an overall increase in Internet usage);
- o And is generally conceptually simpler.

For these reasons, there is a relatively strong consensus in the operator community that option (a) is the preferred way to go. Most networks today use option (a) wherever possible.

However, there can be times when option (b) is the pragmatic choice. Most commonly, option (b) is used to work around limitations in network equipment. One big example is the generally poor level of support today for individual statistics on IPv4 traffic vs IPv6 traffic when option (a) is used. Other, device-specific, limitations exist as well. It is expected that these limitations will go away as support for IPv6 matures, making option (b) less and less attractive until the day that IPv4 is finally turned off.

## 2.3. Static Routes

### 2.3.1. Link-Local Next-Hop in a Static Route?

For the most part, the use of static routes in IPv6 parallels their use in IPv4. There is, however, one exception, which revolves around the choice of next-hop address in the static route. Specifically, should an operator:

- a. Use the far-end's link-local address as the next-hop address, OR
- b. Use the far-end's GUA/ULA address as the next-hop address?

Recall that the IPv6 specs for OSPF [RFC5340] and ISIS [RFC5308] dictate that they always use link-locals for next-hop addresses. For static routes, [RFC4861] section 8 says:

A router MUST be able to determine the link-local address for each of its neighboring routers in order to ensure that the target address in a Redirect message identifies the neighbor router by its link-local address. For static routing, this requirement implies that the next-hop router's address should be specified using the link-local address of the router.

This implies that using a GUA or ULA as the next hop will prevent a router from sending Redirect messages for packets that "hit" this static route. All this argues for using a link-local as the next-hop address in a static route.

However, there are two cases where using a link-local address as the next-hop clearly does not work. One is when the static route is an indirect (or multi-hop) static route. The second is when the static route is redistributed into another routing protocol. In these cases, the above text from RFC 4861 notwithstanding, either a GUA or ULA must be used.

Furthermore, many network operators are concerned about the dependency of the default link-local address on an underlying MAC address, as described in the previous section.

Today most operators use GUAs as next-hop addresses.

## 2.4. IGPs

### 2.4.1. IGP Choice

One of the main decisions for a network operator looking to deploy IPv6 is the choice of IGP (Interior Gateway Protocol) within the network. The main options are OSPF, IS-IS and EIGRP. RIPng is another option, but very few networks run RIP in the core these days, so it is covered in a separate section below.

OSPF [RFC2328] [RFC5340] and IS-IS [RFC5120][RFC5120] are both standardized link-state protocols. Both protocols are widely supported by vendors, and both are widely deployed. By contrast, EIGRP [RFC7868] is a Cisco proprietary distance-vector protocol. EIGRP is rarely deployed in service-provider networks, but is quite common in enterprise networks, which is why it is discussed here.

It is out of scope for this document to describe all the differences between the three protocols; the interested reader can find books and websites that go into the differences in quite a bit of detail. Rather, this document simply highlights a few differences that can be important to consider when designing IPv6 or dual-stack networks.

Versions: There are two versions of OSPF: OSPFv2 and OSPFv3. The two versions share many concepts, are configured in a similar manner and seem very similar to most casual users, but have very different packet formats and other "under the hood" differences. The most important difference is that OSPFv2 will only route IPv4, while OSPFv3 will route both IPv4 and IPv6 (see [RFC5838]). OSPFv2 was by far the most widely deployed version of OSPF when this document was published. By contrast, both IS-IS and EIGRP have just a single version, which can route both IPv4 and IPv6.

Transport. IS-IS runs over layer 2 (e.g. Ethernet). This means that the functioning of IS-IS has no dependencies on the IP layer: if



there is a problem at the IP layer (e.g. bad addresses), two routers can still exchange IS-IS packets. By contrast, OSPF and EIGRP both run over the IP layer. This means that the IP layer must be configured and working OSPF or EIGRP packets to be exchanged between routers. For EIGRP, the dependency on the IP layer is simple: EIGRP for IPv4 runs over IPv4, while EIGRP for IPv6 runs over IPv6. For OSPF, the story is more complex: OSPFv2 runs over IPv4, but OSPFv3 can run over either IPv4 or IPv6. Thus it is possible to route both IPv4 and IPv6 with OSPFv3 running over IPv6 or with OSPFv3 running over IPv4. This means that there are number of choices for how to run OSPF in a dual-stack network:

- o Use OSPFv2 for routing IPv4 , and OSPFv3 running over IPv6 for routing IPv6, OR
- o Use OSPFv3 running over IPv6 for routing both IPv4 and IPv6, OR
- o Use OSPFv3 running over IPv4 for routing both IPv4 and IPv6.

Summarization and MPLS: For most casual users, the three protocols are fairly similar in what they can do, with two glaring exceptions: summarization and MPLS. For summarization, both OSPF and IS-IS have the concept of summarization between areas, but the two area concepts are quite different, and an area design that works for one protocol will usually not work for the other. EIGRP has no area concept, but has the ability to summarize at any router. Thus a large network will typically have a very different OSPF, IS-IS and EIGRP designs, which is important to keep in mind if you are planning on using one protocol to route IPv4 and a different protocol for IPv6. The other difference is that OSPF and IS-IS both support RSVP-TE, a widely-used MPLS signaling protocol, while EIGRP does not: this is due to OSPF and IS-IS both being link-state protocols while EIGRP is a distance-vector protocol.

The table below sets out possible combinations of protocols to route both IPv4 and IPv6, and makes some observations on each combination. Here "EIGRP-v4" means "EIGRP for IPv4" and similarly for "EIGRP-v6". For OSPFv3, it is possible to run it over either IPv4 or IPv6; this is not indicated in the table.

IGP for IPv4	IGP for IPv6	Protocol separation	Similar configuration possible	Multiple Known Deployments
OSPFv2	OSPFv3	YES	YES	YES (8)
OSPFv2	IS-IS	YES	-	YES (3)
OSPFv2	EIGRP-v6	YES	-	-
OSPFv3	OSPFv3	NO	YES	-
OSPFv3	IS-IS	YES	-	-
OSPFv3	EIGRP-v6	YES	-	-
IS-IS	OSPFv3	YES	-	YES (2)
IS-IS	IS-IS	-	YES	YES (12)
IS-IS	EIGRP-v6	YES	-	-
EIGRP-v4	OSPFv3	YES	-	? (1)
EIGRP-v4	IS-IS	YES	-	-
EIGRP-v4	EIGRP-v6	-	YES	? (2)

In the column "Multiple Known Deployments", a YES indicates that a significant number of production networks run this combination, with the number of such networks indicated in parentheses following, while a "?" indicates that the authors are only aware of one or two small networks that run this combination. Data for this column was gathered from an informal poll of operators on a number of mailing lists. This poll was not intended to be a thorough scientific study of IGP choices, but to provide a snapshot of known operator choices at the time of writing (Mid-2015) for successful production dual stack network deployments. There were twenty six (26) network implementations represented by 17 respondents. Some respondents provided information on more than one network or network deployment. Due to privacy considerations, the networks' represented and respondents are not listed in this document.

A number of combinations are marked as offering "Protocol separation". These options use a different IGP protocol for IPv4 vs IPv6. With these options, a problem with routing IPv6 is unlikely to affect IPv4 or visa-versa. Some operator may consider this as a benefit when first introducing dual stack capabilities or for ongoing technical reasons.

Three combinations are marked "Similar configuration possible". This means it is possible (but not required) to use very similar IGP configuration for IPv4 and IPv6: for example, the same area boundaries, area numbering, link costing, etc. If you are happy with your IPv4 IGP design, then this will likely be a consideration. By contrast, the options that use, for example, IS-IS for one IP version and OSPF for the other version will require considerably different configuration, and will also require the operations staff to become familiar with the difference between the two protocols.

It should be noted that a number of ISPs have run OSPF as their IPv4 IGP for quite a few years, but have selected IS-IS as their IPv6 IGP. However, there are very few (none?) that have made the reverse choice. This is, in part, because routers generally support more nodes in an IS-IS area than in the corresponding OSPF area, and because IS-IS is seen as more secure because it runs at layer 2.

#### 2.4.2. IS-IS Topology Mode

When IS-IS is used to route both IPv4 and IPv6, then there is an additional choice of whether to run IS-IS in single-topology or multi-topology mode.

With single-topology mode (also known as Native mode) [RFC5308]:

- o IS-IS keeps a single link-state database for both IPv4 and IPv6.
- o There is a single set of link costs which apply to both IPv4 and IPv6.
- o All links in the network must support both IPv4 and IPv6, as the calculation of routes does not take this into account. If some links do not support IPv6 (or IPv4), then packets may get routed across links where support is lacking and get dropped. This can cause problems if some network devices do not support IPv6 (or IPv4).
- o It is also important to keep the previous point in mind when adding or removing support for either IPv4 or IPv6.

With multi-topology mode [RFC5120]:

- o IS-IS keeps two link-state databases, one for IPv4 and one for IPv6.
- o IPv4 and IPv6 can have separate link metrics. Note that most implementations today require separate link metrics: a number of operators have rudely discovered that they have forgotten to configure the IPv6 metric until sometime after deploying IPv6 in multi-topology mode!
- o Some links can be IPv4-only, some IPv6-only, and some dual-stack. Routes to IPv4 and IPv6 addresses are computed separately and may take different paths even if the addresses are located on the same remote device.
- o The previous point may help when adding or removing support for either IPv4 or IPv6.

In the informal poll of operators, out of 12 production networks that ran IS-IS for both IPv4 and IPv6, 6 used single topology mode, 4 used multi-topology mode, and 2 did not specify. One motivation often cited by then operators for using Single Topology mode was because some device did not support multi-topology mode.

When asked, many people feel multi-topology mode is superior to single-topology mode because it provides greater flexibility at minimal extra cost. Never-the-less, as shown by the poll results, a number of operators have used single-topology mode successfully.

Note that this issue does not come up with OSPF, since there is nothing that corresponds to IS-IS single-topology mode with OSPF.

#### 2.4.3. RIP / RIPng

A protocol option not described in the table above is RIP for IPv4 and RIPng for IPv6 [RFC2080]. These are distance vector protocols that are almost universally considered to be inferior to OSPF, IS-IS, or EIGRP for general use.

However, there is one specialized use where RIP/RIPng is still considered to be appropriate: in star topology networks where a single core device has lots and lots of links to edge devices and each edge device has only a single path back to the core. In such networks, the single path means that the limitations of RIP/RIPng are mostly not relevant and the very light-weight nature of RIP/RIPng gives it an advantage over the other protocols mentioned above. One concrete example of this scenario is the use of RIP/RIPng between cable modems and the CMTS.

## 2.5. BGP

### 2.5.1. Which Transport for Which Routes?

BGP these days is multi-protocol. It can carry routes of many different types, or more precisely, many different AFI/SAFI combinations. It can also carry routes when the BGP session, or more accurately the underlying TCP connection, runs over either IPv4 or IPv6 (here referred to as either "IPv4 transport" or "IPv6 transport"). Given this flexibility, one of the biggest questions when deploying BGP in a dual-stack network is the question of which route types should be carried over sessions using IPv4 transport and which should be carried over sessions using IPv6 transport.

This section discusses this question for the three most-commonly-used SAFI values: unlabeled (SAFI 1), labeled (SAFI 4) and VPN (SAFI 128). Though we do not explicitly discuss other SAFI values, many of the comments here can be applied to the other values.

Consider the following table:

Route Family	Transport	Comments
Unlabeled IPv4	IPv4	Works well
Unlabeled IPv4	IPv6	Next-hop
Unlabeled IPv6	IPv4	Next-hop
Unlabeled IPv6	IPv6	Works well
Labeled IPv4	IPv4	Works well
Labeled IPv4	IPv6	Next-hop
Labeled IPv6	IPv4	(6PE) Works well
Labeled IPv6	IPv6	Next-hop or MPLS over IPv6
VPN IPv4	IPv4	Works well
VPN IPv4	IPv6	Next-hop
VPN IPv6	IPv4	(6VPE) Works well
VPN IPv6	IPv6	Next-hop or MPLS over IPv6

The first column in this table lists various route families, where "unlabeled" means SAFI 1, "labeled" means the routes carry an MPLS label (SAFI 4, see [RFC3107]), and "VPN" means the routes are normally associated with a layer-3 VPN (SAFI 128, see [RFC4364]). The second column lists the protocol used to transport the BGP session, frequently specified by giving either an IPv4 or IPv6 address in the "neighbor" statement.

The third column comments on the combination in the first two columns:

- o For combinations marked "Works well", these combinations are standardized, widely supported and widely deployed.

- o For combinations marked "Next-hop", these combinations are not standardized and are less-widely supported. These combinations all have the "next-hop mismatch" problem: the transported route needs a next-hop address from the other address family than the transport address (for example, an IPv4 route needs an IPv4 next-hop, even when transported over IPv6). Some vendors have implemented ways to solve this problem for specific combinations, but for combinations marked "next-hop", these solutions have not been standardized (cf. 6PE and 6VPE, where the solution has been standardized).
- o For combinations marked as "Next-hop or MPLS over IPv6", these combinations either require a non-standard solution to the next-hop problem, or require MPLS over IPv6. At the time of writing, MPLS over IPv6 is not widely supported or deployed.

Also, it is important to note that changing the set of address families being carried over a BGP session requires the BGP session to be reset (unless something like [I-D.ietf-idr-dynamic-cap] or [I-D.ietf-idr-bgp-multisession] is in use). This is generally more of an issue with eBGP sessions than iBGP sessions: for iBGP sessions it is common practice for a router to have two iBGP sessions, one to each member of a route reflector pair, so one can change the set of address families on first one of the sessions and then the other.

The following subsections discuss specific combinations in more detail.

#### 2.5.1.1. BGP Sessions for Unlabeled Routes

Unlabeled routes are commonly carried on eBGP sessions, as well as on iBGP sessions in networks where Internet traffic is carried unlabeled across the network.

In these scenarios, there are three reasonable choices:

- a. Carry unlabeled IPv4 and IPv6 routes over IPv4, OR
- b. Carry unlabeled IPv4 and IPv6 routes over IPv6, OR
- c. Carry unlabeled IPv4 routes over IPv4, and unlabeled IPv6 routes over IPv6

Options (a) and (b) have the advantage that one BGP session is required between pairs of routers. However, option (c) is widely considered to be the best choice. There are several reasons for this :

- o It gives a clean separation between IPv4 and IPv6. This can be especially useful when first deploying IPv6 and troubleshooting resulting problems.
- o This avoids the next-hop problem described above.
- o The status of the routes follows the status of the underlying transport. If, for example, the IPv6 data path between the two BGP speakers fails, then the IPv6 session between the two speakers will fail and the IPv6 routes will be withdrawn, which will allow the traffic to be re-routed elsewhere. By contrast, if the IPv6 routes were transported over IPv4, then the failure of the IPv6 data path might leave a working IPv4 data path, so the BGP session would remain up and the IPv6 routes would not be withdrawn, and thus the IPv6 traffic would be sent into a black hole.
- o It avoids resetting the BGP session when adding IPv6 to an existing session, or when removing IPv4 from an existing session.

Rarely, there are situations where option (c) is not practical. In those cases today, most operators use option (a), carrying both route types over a single BGP session.

#### 2.5.1.2. BGP sessions for Labeled or VPN Routes

When carrying labeled or VPN routes, the only widely-supported solution at time of writing is to carry both route types over IPv4. This may change in as MPLS over IPv6 becomes more widely implemented.

There are two options when carrying both over IPv4:

- a. Carry all routes over a single BGP session, OR
- b. Carry the routes over multiple BGP sessions (e.g. one for VPN IPv4 routes and one for VPN IPv6 routes)

Using a single session is usually simplest for an iBGP session going to a route reflector handling both route families. Using a single session here usually means that the BGP session will reset when changing the set of address families, but as noted above, this is usually not a problem when redundant route reflectors are involved.

In eBGP situations, two sessions are usually more appropriate.  
[JUSTIFICATION?]



### 2.5.2. eBGP Endpoints: Global or Link-Local Addresses?

When running eBGP over IPv6, there are two options for the addresses to use at each end of the eBGP session (or more properly, the underlying TCP session):

- a. Use link-local addresses for the eBGP session, OR
- b. Use global addresses for the eBGP session.

Note that the choice here is the addresses to use for the eBGP sessions, and not whether the link itself has global (or unique-local) addresses. In particular, it is quite possible for the eBGP session to use link-local addresses even when the link has global addresses.

The big attraction for option (a) is security: an eBGP session using link-local addresses is extremely difficult to attack from a device that is off-link. This provides very strong protection against TCP RST and similar attacks. Though there are other ways to get an equivalent level of security (e.g. GTSM [RFC5082], MD5 [RFC5925], or ACLs), these other ways require additional configuration which can be forgotten or potentially mis-configured.

However, there are a number of small disadvantages to using link-local addresses:

- o Using link-local addresses only works for single-hop eBGP sessions; it does not work for multi-hop sessions.
- o One must use "next-hop self" at both endpoints, otherwise re-advertising routes learned via eBGP into iBGP will not work. (Some products enable "next-hop self" in this situation automatically).
- o Operators and their tools are used to referring to eBGP sessions by address only, something that is not possible with link-local addresses.
- o If one is configuring parallel eBGP sessions for IPv4 and IPv6 routes, then using link-local addresses for the IPv6 session introduces extra operational differences between the two sessions which could otherwise be avoided.
- o On some products, an eBGP session using a link-local address is more complex to configure than a session that uses a global address.

- o If hardware or other issues cause one to move the cable to a different local interface, then reconfiguration is required at both ends: at the local end because the interface has changed (and with link-local addresses, the interface must always be specified along with the address), and at the remote end because the link-local address has likely changed. (Contrast this with using global addresses, where less re-configuration is required at the local end, and no reconfiguration is required at the remote end).
- o Finally, a strict application of [RFC2545] forbids running eBGP between link-local addresses, as [RFC2545] requires the BGP next-hop field to contain at least a global address.

For these reasons, most operators today choose to have their eBGP sessions use global addresses.

### 3. General Observations

There are two themes that run through many of the design choices in this document. This section presents some general discussion on these two themes.

#### 3.1. Use of Link-Local Addresses

The proper use of link-local addresses is a common theme in the IPv6 network design choices. Link-layer addresses are, of course, always present in an IPv6 network, but current network design practice mostly ignores them, despite efforts such as [RFC7404].

There are three main reasons for this current practice:

- o Network operators are concerned about the volatility of link-local addresses based on MAC addresses, despite the fact that this concern can be overcome by manually-configuring link-local addresses;
- o It is very difficult to impossible to ping a link-local address from a device that is not on the same subnet. This is a troubleshooting disadvantage, though it can also be viewed as a security advantage.
- o Most operators are currently running networks that carry both IPv4 and IPv6 traffic, and wish to harmonize their IPv4 and IPv6 design and operational practices where possible.

### 3.2. Separation of IPv4 and IPv6

Currently, most operators are running or planning to run networks that carry both IPv4 and IPv6 traffic. Hence the question: To what degree should IPv4 and IPv6 be kept separate? As can be seen above, this breaks into two sub-questions: To what degree should IPv4 and IPv6 traffic be kept separate, and to what degree should IPv4 and IPv6 routing information be kept separate?

The general consensus around the first question is that IPv4 and IPv6 traffic should generally be mixed together. This recommendation is driven by the operational simplicity of mixing the traffic, plus the general observation that the service being offered to the end user is Internet connectivity and most users do not know or care about the differences between IPv4 and IPv6. Thus it is very desirable to mix IPv4 and IPv6 on the same link to the end user. On other links, separation is possible but more operationally complex, though it does occasionally allow the operator to work around limitations on network devices. The situation here is roughly comparable to IP and MPLS traffic: many networks mix the two traffic types on the same links without issues.

By contrast, there is more of an argument for carrying IPv6 routing information over IPv6 transport, while leaving IPv4 routing information on IPv4 transport. By doing this, one gets fate-sharing between the control and data plane for each IP protocol version: if the data plane fails for some reason, then often the control plane will too.

### 4. IANA Considerations

This document makes no requests of IANA.

### 5. Security Considerations

This document introduces no new security considerations that are not already documented elsewhere.

The following is a brief list of pointers to documents related to the topics covered above that the reader may wish to review for security considerations.

For general IPv6 security, [RFC4942] provides guidance on security considerations around IPv6 transition and coexistence.

For OSPFv3, the base protocol specification [RFC5340] has a short security considerations section which notes that the fundamental

mechanism for protecting OSPFv3 from attacks is the mechanism described in [RFC4552].

For IS-IS, [RFC5308] notes that IS-IS for IPv6 raises no new security considerations over IS-IS for IPv4 over those documented in [ISO10589] and [RFC5304].

For BGP, [RFC2545] notes that BGP for IPv6 raises no new security considerations over those present in BGP for IPv4. However, there has been much discussion of BGP security recently, and the interested reader is referred to the documents of the IETF's SIDR working group.

## 6. Acknowledgements

Many, many people in the V6OPS working group provided comments and suggestions that made their way into this document. A partial list includes: Rajiv Asati, Fred Baker, Michael Behringer, Marc Blanchet, Ron Bonica, Randy Bush, Cameron Byrne, Brian Carpenter, KK Chittimaneni, Tim Chown, Lorenzo Colitti, Gert Doering, Francis Dupont, Bill Fenner, Kedar K Gaonkar, Chris Grundemann, Steinar Haug, Ray Hunter, Joel Jaeggli, Victor Kuarsingh, Jen Linkova, Ivan Pepelnjak, Alexandru Petrescu, Rob Shakir, Mark Smith, Jean-Francois Tremblay, Dave Thaler, Tina Tsou, Eric Vyncke, Dan York, and Xuxiaohu.

The authors would also like to thank Pradeep Jain and Alastair Johnson for helpful comments on a very preliminary version of this document.

## 7. Informative References

- [I-D.ietf-idr-bgp-multisession]  
Scudder, J., Appanna, C., and I. Varlashkin, "Multisession BGP", draft-ietf-idr-bgp-multisession-07 (work in progress), September 2012.
- [I-D.ietf-idr-dynamic-cap]  
Ramachandra, S. and E. Chen, "Dynamic Capability for BGP-4", draft-ietf-idr-dynamic-cap-14 (work in progress), December 2011.
- [I-D.ietf-v6ops-ula-usage-recommendations]  
Liu, B. and S. Jiang, "Considerations For Using Unique Local Addresses", draft-ietf-v6ops-ula-usage-recommendations-05 (work in progress), May 2015.

- [ISO10589] International Standards Organization, "Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", International Standard 10589:2002, Nov 2002.
- [RFC2080] Malkin, G. and R. Minnear, "RIPng for IPv6", RFC 2080, DOI 10.17487/RFC2080, January 1997, <<http://www.rfc-editor.org/info/rfc2080>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<http://www.rfc-editor.org/info/rfc2328>>.
- [RFC2545] Marques, P. and F. Dupont, "Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing", RFC 2545, DOI 10.17487/RFC2545, March 1999, <<http://www.rfc-editor.org/info/rfc2545>>.
- [RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <<http://www.rfc-editor.org/info/rfc3107>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<http://www.rfc-editor.org/info/rfc4193>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<http://www.rfc-editor.org/info/rfc4291>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<http://www.rfc-editor.org/info/rfc4364>>.
- [RFC4472] Durand, A., Ihren, J., and P. Savola, "Operational Considerations and Issues with IPv6 DNS", RFC 4472, DOI 10.17487/RFC4472, April 2006, <<http://www.rfc-editor.org/info/rfc4472>>.
- [RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <<http://www.rfc-editor.org/info/rfc4552>>.

- [RFC4852] Bound, J., Pouffary, Y., Klynsmas, S., Chown, T., and D. Green, "IPv6 Enterprise Network Analysis - IP Layer 3 Focus", RFC 4852, DOI 10.17487/RFC4852, April 2007, <<http://www.rfc-editor.org/info/rfc4852>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<http://www.rfc-editor.org/info/rfc4861>>.
- [RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/Co-existence Security Considerations", RFC 4942, DOI 10.17487/RFC4942, September 2007, <<http://www.rfc-editor.org/info/rfc4942>>.
- [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007, <<http://www.rfc-editor.org/info/rfc5082>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-IS)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<http://www.rfc-editor.org/info/rfc5120>>.
- [RFC5220] Matsumoto, A., Fujisaki, T., Hiromi, R., and K. Kanayama, "Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules", RFC 5220, DOI 10.17487/RFC5220, July 2008, <<http://www.rfc-editor.org/info/rfc5220>>.
- [RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <<http://www.rfc-editor.org/info/rfc5304>>.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <<http://www.rfc-editor.org/info/rfc5308>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<http://www.rfc-editor.org/info/rfc5340>>.
- [RFC5375] Van de Velde, G., Popoviciu, C., Chown, T., Bonness, O., and C. Hahn, "IPv6 Unicast Address Assignment Considerations", RFC 5375, DOI 10.17487/RFC5375, December 2008, <<http://www.rfc-editor.org/info/rfc5375>>.

- [RFC5838] Lindem, A., Ed., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, DOI 10.17487/RFC5838, April 2010, <<http://www.rfc-editor.org/info/rfc5838>>.
- [RFC5887] Carpenter, B., Atkinson, R., and H. Flinck, "Renumbering Still Needs Work", RFC 5887, DOI 10.17487/RFC5887, May 2010, <<http://www.rfc-editor.org/info/rfc5887>>.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<http://www.rfc-editor.org/info/rfc5925>>.
- [RFC5963] Gagliano, R., "IPv6 Deployment in Internet Exchange Points (IXPs)", RFC 5963, DOI 10.17487/RFC5963, August 2010, <<http://www.rfc-editor.org/info/rfc5963>>.
- [RFC6180] Arkko, J. and F. Baker, "Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment", RFC 6180, DOI 10.17487/RFC6180, May 2011, <<http://www.rfc-editor.org/info/rfc6180>>.
- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, DOI 10.17487/RFC6296, June 2011, <<http://www.rfc-editor.org/info/rfc6296>>.
- [RFC6342] Koodli, R., "Mobile Networks Considerations for IPv6 Deployment", RFC 6342, DOI 10.17487/RFC6342, August 2011, <<http://www.rfc-editor.org/info/rfc6342>>.
- [RFC6752] Kirkham, A., "Issues with Private IP Addressing in the Internet", RFC 6752, DOI 10.17487/RFC6752, September 2012, <<http://www.rfc-editor.org/info/rfc6752>>.
- [RFC6782] Kuarsingh, V., Ed. and L. Howard, "Wireline Incremental IPv6", RFC 6782, DOI 10.17487/RFC6782, November 2012, <<http://www.rfc-editor.org/info/rfc6782>>.
- [RFC6879] Jiang, S., Liu, B., and B. Carpenter, "IPv6 Enterprise Network Renumbering Scenarios, Considerations, and Methods", RFC 6879, DOI 10.17487/RFC6879, February 2013, <<http://www.rfc-editor.org/info/rfc6879>>.
- [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>>.

- [RFC7010] Liu, B., Jiang, S., Carpenter, B., Venaas, S., and W. George, "IPv6 Site Renumbering Gap Analysis", RFC 7010, DOI 10.17487/RFC7010, September 2013, <<http://www.rfc-editor.org/info/rfc7010>>.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <<http://www.rfc-editor.org/info/rfc7217>>.
- [RFC7381] Chittimaneni, K., Chown, T., Howard, L., Kuarsingh, V., Pouffary, Y., and E. Vyncke, "Enterprise IPv6 Deployment Guidelines", RFC 7381, DOI 10.17487/RFC7381, October 2014, <<http://www.rfc-editor.org/info/rfc7381>>.
- [RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", RFC 7404, DOI 10.17487/RFC7404, November 2014, <<http://www.rfc-editor.org/info/rfc7404>>.
- [RFC7868] Savage, D., Ng, J., Moore, S., Slice, D., Paluch, P., and R. White, "Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP)", RFC 7868, DOI 10.17487/RFC7868, May 2016, <<http://www.rfc-editor.org/info/rfc7868>>.
- [v6-addressing-plan] SurfNet, "Preparing an IPv6 Address Plan", 2013, <<http://www.ripe.net/lir-services/training/material/IPv6-for-LIRs-Training-Course/Preparing-an-IPv6-Addressing-Plan.pdf>>.

## Authors' Addresses

Philip Matthews  
Nokia  
600 March Road  
Ottawa, Ontario K2K 2E6  
Canada

Phone: +1 613-784-3139  
Email: [philip\\_matthews@magma.ca](mailto:philip_matthews@magma.ca)



Victor Kuarsingh  
Cisco  
88 Queens Quay  
Toronto, ON M5J0B8  
Canada

Email: [victor@jvknet.com](mailto:victor@jvknet.com)

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: November 3, 2015

B. Liu  
S. Jiang  
Huawei Technologies  
May 2, 2015

Considerations For Using Unique Local Addresses  
draft-ietf-v6ops-ula-usage-recommendations-05

Abstract

This document provides considerations for using IPv6 Unique Local Addresses (ULAs). It identifies cases where ULA addresses are helpful as well as potential problems that their use could introduce, based on an analysis of different ULA usage scenarios.

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 November 3, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1.	Introduction . . . . .	2
2.	Requirements Language . . . . .	3
3.	Analysis of ULA Features . . . . .	3
3.1.	Automatically Generated . . . . .	3
3.2.	Globally Unique . . . . .	3
3.3.	Independent Address Space . . . . .	3
3.4.	Well Known Prefix . . . . .	4
3.5.	Stable or Temporary Prefix . . . . .	4
4.	Analysis and Operational Considerations of Scenarios Using ULAs . . . . .	4
4.1.	Isolated Networks . . . . .	4
4.2.	Connected Networks . . . . .	5
4.2.1.	ULA-Only Deployment . . . . .	5
4.2.2.	ULAs along with PA Addresses . . . . .	7
4.3.	IPv4 Co-existence Considerations . . . . .	9
5.	General Considerations For Using ULAs . . . . .	10
5.1.	Do Not Treat ULA Equal to RFC1918 . . . . .	10
5.2.	Using ULAs in a Limited Scope . . . . .	10
6.	ULA Usages Considered Helpful . . . . .	10
6.1.	Used in Isolated Networks . . . . .	11
6.2.	ULA along with PA . . . . .	11
6.3.	Some Specific Use Cases . . . . .	11
6.3.1.	Special Routing . . . . .	11
6.3.2.	Used as NAT64 Prefix . . . . .	11
6.3.3.	Used as Identifier . . . . .	12
7.	Security Considerations . . . . .	13
8.	IANA Considerations . . . . .	13
9.	Acknowledgements . . . . .	13
10.	References . . . . .	13
10.1.	Normative References . . . . .	13
10.2.	Informative References . . . . .	14
	Authors' Addresses . . . . .	16

## 1. Introduction

Unique Local Addresses (ULAs) are defined in [RFC4193] as provider-independent prefixes that can be used locally, for example, on isolated networks, internal networks, or VPNs. Although ULAs may be treated like addresses of global scope by applications, normally they are not used on the public Internet. ULAs are a possible alternative to site-local addresses (deprecated in [RFC3879]) in some situations, but there are differences between the two address types.

The use of ULAs in various types of networks has been confusing to network operators. This document aims to clarify the advantages and disadvantages of ULAs and how they can be most appropriately used.

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings, and are not to be interpreted as [RFC2119] key words.

## 3. Analysis of ULA Features

### 3.1. Automatically Generated

ULA prefixes can be automatically generated using the algorithms described in [RFC4193]. This feature allows automatic prefix allocation. Thus one can get a network working immediately without applying for prefix(es) from an RIR/LIR (Regional Internet Registry/Local Internet Registry).

### 3.2. Globally Unique

ULAs are intended to have an extremely low probability of collision. Since multiple networks in which the hosts have been assigned with ULAs may occasionally be merged into one network, this uniqueness is necessary. The randomization of 40 bits in a ULA prefix is considered sufficient enough to ensure a high degree of uniqueness (refer to [RFC4193] Section 3.2.3 for details) and simplifies merging of networks by avoiding the need to renumber overlapping IP address space. Such overlapping was a major drawback to the deployment of private [RFC1918] addresses in IPv4.

Note that, as described in [RFC4864], applications may treat ULAs in practice like global-scope addresses, but address selection algorithms may need to distinguish between ULAs and Global-scope Unicast Addresses (GUAs) to ensure bidirectional communications. As a further note, the default address selection policy table in [RFC6724]) responds to this requirement.

### 3.3. Independent Address Space

ULAs provide internal address independence in IPv6 since they can be used for internal communications even without Internet connectivity. They need no registration, so they can support on-demand usage and do not carry any RIR/LIR burden of documentation or fees.

### 3.4. Well Known Prefix

The prefixes of ULAs are well known thus they are easily identified and filtered.

This feature is convenient for management of security policies and troubleshooting. For example, network administrators can segregate packets containing data which must stay in the internal network by assigning ULAs to internal servers. Externally-destined data can be sent to the Internet or telecommunication network by a separate function, through an appropriate gateway/firewall.

### 3.5. Stable or Temporary Prefix

A ULA prefix can be generated once, at installation time or factory reset, and then possibly never be changed. Alternatively, it can be regenerated regularly, depending on deployment requirements.

## 4. Analysis and Operational Considerations of Scenarios Using ULAs

### 4.1. Isolated Networks

IP is used ubiquitously. Some networks like industrial control bus (e.g. [RS-485], [SCADA], or even non-networked digital interfaces like [MIL-STD-1397] have begun to use IP. In these kinds of networks, the system may lack the ability to communicate with the public networks.

As another example, there may be some networks in which the equipment has the technical capability to connect to the Internet, but is prohibited by administration or just temporarily not connected. These networks may include separate financial networks, lab networks, machine-to-machine (e.g. vehicle networks), sensor networks, or even normal LANs, and can include very large numbers of addresses.

Serious disadvantages and impact on applications due to the use of ambiguous address space have been well documented in [RFC1918]. However, ULA is a straightforward way to assign the IP addresses in the kinds of networks just described, with minimal administrative cost or burden. Also, ULAs fit in multiple subnet scenarios, in which each subnet has its own ULA prefix. For example, when we assign vehicles with ULA addresses, it is then possible to separate in-vehicle embedded networks into different subnets depending on real-time requirements, device types, services and more.

However, each isolated network has the possibility to be connected in the future. Administrators need to consider the following before deciding whether to use ULAs:

- o If the network eventually connects to another isolated or private network, the potential for address collision arises. However, if the ULAs were generated in the standard way, this will not be a big problem.
- o If the network eventually connects to the global Internet, then the operator will need to add a new global prefix and ensure that the address selection policy is properly set up on all interfaces.

If these further considerations are unacceptable for some reason, then the administrator needs to be careful about using ULAs in currently isolated networks.

Operational considerations:

- o Prefix generation: Randomly generated according to the algorithms defined in [RFC4193] or manually assigned. Normally, automatic generation of the prefixes is recommended, following [RFC4193]. If there are some specific reasons that call for manual assignment, administrators have to plan the prefixes carefully to avoid collision.
- o Prefix announcement: In some cases, networks may need to announce prefixes to each other. For example, in vehicle networks with infrastructure-less settings such as Vehicle-to-Vehicle (V2V) communication, prior knowledge of the respective prefixes is unlikely. Hence, a prefix announcement mechanism is needed to enable inter-vehicle communications based on IP. As one possibility, such announcements could rely on extensions to the Router Advertisement message of the Neighbor Discovery Protocol (e.g., [I-D.petrescu-autoconf-ra-based-routing] and [I-D.jhlee-mext-mnpp]).

## 4.2. Connected Networks

### 4.2.1. ULA-Only Deployment

In some situations, hosts and interior interfaces are assigned ULAs and not GUAs, but the network needs to communicate with the outside. Two models can be considered:

- o Using Network Prefix Translation

Network Prefix Translation (NPTv6) [RFC6296] is an experimental specification that provides a stateless one-to-one mapping between internal addresses and external addresses. The specification considers translating ULA prefixes into GUA prefixes as an use case. Although NPTv6 works differently from

traditional stateful NAT/NAPT (which is discouraged in [RFC5902]), it introduces similar additional complexity to applications, which may cause applications to break.

Thus this document does not recommend the use of ULA+NPTv6. Rather, this document considers ULA+PA (Provider Aggregated) as a better approach to connect to the global network when ULAs are expected to be retained. The use of ULA+PA is discussed in detail in Section 4.2.2 below.

- o Using Application-Layer Proxies

The proxies terminate the network-layer connectivity of the hosts and associate separate internal and external connections.

In some environments (e.g., information security sensitive enterprise or government), central control is exercised by allowing the endpoints to connect to the Internet only through a proxy. With IPv4, using private address space with proxies is an effective and common practice for this purpose, and it is natural to pick ULA as its counterpart in IPv6.

Benefits of using ULAs in this scenario:

- o Allowing minimal management burden on address assignment for some specific environments.

Drawbacks:

- o The serious disadvantages and impact on applications imposed by NATs have been well documented in [RFC2993] and [RFC3027]. Although NPTv6 is a mechanism that has fewer architectural problems than a traditional stateful Network Address Translator in an IPv6 environment [RFC6296], it still breaks end-to-end transparency and hence in general is not recommended by the IETF.

Operational considerations:

- o Firewall deployment: [RFC6296] points out that an NPTv6 translator does not have the same security properties as a traditional NAT44, and hence needs be supplemented with a firewall if security at the boundary is an issue. The operator has to decide where to locate the firewall.
  - If the firewall is located outside the NPTv6 translator, then filtering is based on the translated GUA prefixes, and when the internal ULA prefixes are renumbered, the filtering rules do not need to be changed. However, when the GUA prefixes of the

NPTv6 are renumbered, the filtering rules need to be updated accordingly.).

- If the firewall is located inside the NPTv6 translator, the filtering is then based on the ULA prefixes, and the rules need to be updated correspondingly. There is no need to update when the NPTv6 GUA prefixes are renumbered.

#### 4.2.2. ULAs along with PA Addresses

Two classes of network might need to use ULA with PA (Provider Aggregated) addresses:

- o Home network. Home networks are normally assigned with one or more globally routed PA prefixes to connect to the uplink of an ISP. In addition, they may need internal routed networking even when the ISP link is down. Then ULA is a proper tool to fit the requirement. [RFC7084] requires the CPE to support ULA. Note: ULAs provide more benefit for multiple-segment home networks; for home networks containing only one segment, link-local addresses are better alternatives.
- o Enterprise network. An enterprise network is usually a managed network with one or more PA prefixes or with a PI prefix, all of which are globally routed. The ULA can be used to improve internal connectivity and make it more resilient, or to isolate certain functions like OAM for servers.

Benefits of Using ULAs in this scenario:

- o Separated local communication plane: for either home networks or enterprise networks, the main purpose of using ULAs along with PA addresses is to provide a logically local routing plane separated from the global routing plane. The benefit is to ensure stable and specific local communication regardless of the ISP uplink failure. This benefit is especially meaningful for the home network or for private OAM function in an enterprise.
- o Renumbering: in some special cases such as renumbering, enterprise administrators may want to avoid the need to renumber their internal-only, private nodes when they have to renumber the PA addresses of the rest of the network because they are changing ISPs, because the ISP has restructured its address allocations, or for some other reason. In these situations, ULA is an effective tool for addressing internal-only nodes. Even public nodes can benefit from ULA for renumbering, on their internal interfaces. When renumbering, as [RFC4192] suggests, old prefixes continue to be valid until the new prefix(es) is(are) stable. In the process



of adding new prefix(es) and deprecating old prefix(es), it is not easy to keep local communication disentangled from global routing plane change. If we use ULAs for local communication, the separated local routing plane can isolate the effects of global routing change.

Drawbacks:

- o Operational Complexity: there are some arguments that in practice the use of ULA+PA creates additional operational complexity. This is not a ULA-specific problem; the multiple-addresses-per-interface is an important feature of IPv6 protocol. Nevertheless, running multiple prefixes needs more operational consideration than running a single one.

Operational considerations:

- o Default Routing: connectivity may be broken if ULAs are used as default route. When using RIO (Route Information Option) in [RFC4191], specific routes can be added without a default route, thus avoiding bad user experience due to timeouts on ICMPv6 redirects. This behavior was well documented in [RFC7084] as rule ULA-5 "An IPv6 CE router MUST NOT advertise itself as a default router with a Router Lifetime greater than zero whenever all of its configured and delegated prefixes are ULA prefixes." and along with rule L-3 "An IPv6 CE router MUST advertise itself as a router for the delegated prefix(es) (and ULA prefix if configured to provide ULA addressing) using the "Route Information Option" specified in Section 2.3 of [RFC4191]. This advertisement is independent of having or not having IPv6 connectivity on the WAN interface.". However, it needs to be noticed that current OSes don't all support [RFC4191].
- o SLAAC/DHCPv6 co-existing: Since SLAAC and DHCPv6 might be enabled in one network simultaneously; the administrators need to carefully plan how to assign ULA and PA prefixes in accordance with the two mechanisms. The administrators need to know the current issue of the SLAAC/DHCPv6 interaction (please refer to [I-D.ietf-v6ops-dhcpv6-slaac-problem] for details).
- o Address selection: As mentioned in [RFC5220], there is a possibility that the longest matching rule will not be able to choose the correct address between ULAs and global unicast addresses for correct intra-site and extra-site communication. [RFC6724] claims that a site-specific policy entry can be used to cause ULAs within a site to be preferred over global addresses.

- o DNS relevant: if administrators choose not to do reverse DNS delegation inside of their local control of ULA prefixes, a significant amount of information about the ULA population may leak to the outside world. Because reverse queries will be made and naturally routed to the global reverse tree, so external parties will be exposed to the existence of a population of ULA addresses. [ULA-IN-WILD] provides more detailed situations on this issue. Administrators may need a split DNS to separate the queries from internal and external for ULA entries and GUA entries.

#### 4.3. IPv4 Co-existence Considerations

Generally, this document does not consider IPv4 to be in scope. But regarding ULA, there is a special case needs to be recognized, which is described in Section 3.2.2 of [RFC5220]. When an enterprise has IPv4 Internet connectivity but does not yet have IPv6 Internet connectivity, and the enterprise wants to provide site-local IPv6 connectivity, a ULA is the best choice for site-local IPv6 connectivity. Each employee host will have both an IPv4 global or private address and a ULA. Here, when this host tries to connect to an outside node that has registered both A and AAAA records in the DNS, the host will choose AAAA as the destination address and the ULA for the source address according to the IPv6 preference of the default policy table defined in the old address selection standard [RFC3484]. This will clearly result in a connection failure. The new address selection standard [RFC6724] has corrected this behavior by preferring IPv4 than ULAs in the default policy table. However, there are still lots of hosts using the old standard [RFC3484], thus this could be an issue in real networks.

Happy Eyeballs [RFC6555] solves this connection failure problem, but unwanted timeouts will obviously lower the user experience. One possible approach to eliminating the timeouts is to deprecate the IPv6 default route and simply configure a scoped route on hosts (in the context of this document, only configure the ULA prefix routes). Another alternative is to configure IPv4 preference on the hosts, and not include DNS A records but only AAAA records for the internal nodes in the internal DNS server. Then outside nodes have both A and AAAA records and can be connected through IPv4 as default and internal nodes can always connect through IPv6. But since IPv6 preference is default, changing the default in all nodes is not suitable at scale.

## 5. General Considerations For Using ULAs

### 5.1. Do Not Treat ULA Equal to RFC1918

ULA and [RFC1918] are similar in some aspects. The most obvious one is as described in Section 3.1.3 that ULA provides an internal address independence capability in IPv6 that is similar to how [RFC1918] is commonly used. ULA allows administrators to configure the internal network of each platform the same way it is configured in IPv4. Many organizations have security policies and architectures based around the local-only routing of [RFC1918] addresses and those policies may directly map to ULA [RFC4864].

But this does not mean that ULA is equal to an IPv6 version of [RFC1918] deployment. [RFC1918] usually combines with NAT/NAPT for global connectivity. But it is not necessary to combine ULAs with any kind of NAT. Operators can use ULA for local communications along with global addresses for global communications (see Section 4.2.2). This is a big advantage brought by default support of multiple-addresses-per-interface feature in IPv6. (People may still have a requirement for NAT with ULA, this is discussed in Section 4.2.1. But people also need to keep in mind that ULA is not intentionally designed for this kind of use case.)

Another important difference is the ability to merge two ULA networks without renumbering (because of the uniqueness), which is a big advantage over [RFC1918].

### 5.2. Using ULAs in a Limited Scope

A ULA is by definition a prefix that is never advertised outside a given domain, and is used within that domain by agreement of those networked by the domain.

So when using ULAs in a network, the administrators need to clearly set the scope of the ULAs and configure ACLs on relevant border routers to block them out of the scope. And if internal DNS is enabled, the administrators might also need to use internal-only DNS names for ULAs and might need to split the DNS so that the internal DNS server includes records that are not presented in the external DNS server.

## 6. ULA Usages Considered Helpful

### 6.1. Used in Isolated Networks

As analyzed in Section 4.1, ULA is very suitable for isolated networks. Especially when there are subnets in the isolated network, ULA is a reasonable choice.

### 6.2. ULA along with PA

As described in Section 4.2.2, using ULAs along with PA addresses to provide a logically separated local plane can benefit OAM functions and renumbering.

### 6.3. Some Specific Use Cases

Along with the general scenarios, this section provides some specific use cases that could benefit from using ULA.

#### 6.3.1. Special Routing

For various reasons the administrators may want to have private routing be controlled and separated from other routing. For example, in the business-to-business case described in [I-D.baker-v6ops-b2b-private-routing], two companies might want to use direct connectivity that only connects stated machines, such as a silicon foundry with client engineers that use it. A ULA provides a simple way to assign prefixes that would be used in accordance with an agreement between the parties.

#### 6.3.2. Used as NAT64 Prefix

The NAT64 PREF64 is just a group of local fake addresses for the DNS64 to point traffic to a NAT64. Using a ULA prefix as the PREF64 easily ensures that only local systems can use the translation resources of the NAT64 system since the ULA is not intended to be globally routable. The ULA helps clearly identify traffic that is locally contained and destined to a NAT64. Using ULA for PREF64 is deployed and it is an operational model.

But there is an issue needs to be noted. The NAT64 standard [RFC6146] specifies that the PREF64 should align with [RFC6052], in which the IPv4-Embedded IPv6 Address format was specified. If we pick a /48 for NAT64, it happens to be a standard 48/ part of ULA (7bit ULA well-known prefix+ 1 "L" bit + 40bit Global ID). Then the 40bit of ULA is not violated by being filled with part of the 32bit IPv4 address. This is important, because the 40bit assures the uniqueness of ULA. If the prefix is shorter than /48, the 40bit would be violated, and this could cause conformance issues. But it is considered that the most common use case will be a /96 PREF64, or

even /64 will be used. So it seems this issue is not common in current practice.

It is most common that ULA PREF64 will be deployed on a single internal network, where the clients and the NAT64 share a common internal network. ULA will not be effective as PREF64 when the access network must use an Internet transit to receive the translation service of a NAT64 since the ULA will not route across the Internet.

According to the default address selection table specified in [RFC6724], the host would always prefer IPv4 over ULA. This could be a problem in NAT64-CGN scenario as analyzed in Section 8 of [RFC7269]. So administrators need to add additional site-specific address selection rules to the default table to steer traffic flows going through NAT64-CGN. However, updating the default policy tables in all hosts involves significant management cost. This may be possible in an enterprise (using a group policy object, or other configuration mechanisms), but it is not suitable at scale for home networks.

#### 6.3.3. Used as Identifier

ULAs could be self-generated and easily grabbed from the standard IPv6 stack. And ULAs don't need to be changed as the GUA prefixes do. So they are very suitable to be used as identifiers by the up layer applications. And since ULA is not intended to be globally routed, it is not harmful to the routing system.

Such kind of benefit has been utilized in real implementations. For example, in [RFC6281], the protocol BTMM (Back To My Mac) needs to assign a topology-independent identifier to each client host according to the following considerations:

- o TCP connections between two end hosts wish to survive in network changes.
- o Sometimes one needs a constant identifier to be associated with a key so that the Security Association can survive the location changes.

It needs to be noticed again that in theory ULA has the possibility of collision. However, the probability is desirably small enough and can be ignored in most cases when ULAs are used as identifiers.

## 7. Security Considerations

Security considerations regarding ULAs, in general, please refer to the ULA specification [RFC4193]. Also refer to [RFC4864], which shows how ULAs help with local network protection.

As mentioned in Section 4.2.2, when using NPTv6, the administrators need to know where the firewall is located to set proper filtering rules.

Also as mentioned in Section 4.2.2, if administrators choose not to do reverse DNS delegation inside their local control of ULA prefixes, a significant amount of information about the ULA population may leak to the outside world.

## 8. IANA Considerations

This memo has no actions for IANA.

## 9. Acknowledgements

Many valuable comments were received in the IETF v6ops WG mail list, especially from Cameron Byrne, Fred Baker, Brian Carpenter, Lee Howard, Victor Kuarsingh, Alexandru Petrescu, Mikael Abrahamsson, Tim Chown, Jen Linkova, Christopher Palmer Jong-Hyouk Lee, Mark Andrews, Lorenzo Colitti, Ted Lemon, Joel Jaeggli, David Farmer, Doug Barton, Owen DeLong, Gert Doering, Bill Jouris, Bill Cervený, Dave Thaler, Nick Hilliard, Jan Zorz, Randy Bush, Anders Brandt, , Sofiane Imadali and Wesley George.

Some test of using ULA in the lab was done by our research partner BNRC-BUPT (Broad Network Research Centre in Beijing University of Posts and Telecommunications). Thanks for the work of Prof. Xiangyang Gong and student Dengjia Xu.

Tom Taylor did a language review and revision through the whole document. The authors appreciate a lot for his help.

This document was produced using the xml2rfc tool [RFC2629] (initially prepared using 2-Word-v2.0.template.dot.).

## 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.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

## 10.2. Informative References

[I-D.baker-v6ops-b2b-private-routing]  
Baker, F., "Business to Business Private Routing", draft-baker-v6ops-b2b-private-routing-00 (work in progress), July 2007.

[I-D.ietf-v6ops-dhcpv6-slaac-problem]  
Liu, B., Jiang, S., Bonica, R., Gong, X., and W. Wang, "DHCPv6/SLAAC Address Configuration Interaction Problem Statement", draft-ietf-v6ops-dhcpv6-slaac-problem-03 (work in progress), October 2014.

[I-D.jhlee-mext-mnpp]  
Tsukada, M., Ernst, T., and J. Lee, "Mobile Network Prefix Provisioning", draft-jhlee-mext-mnpp-00 (work in progress), October 2009.

[I-D.petrescu-autoconf-ra-based-routing]  
Petrescu, A., Janneteau, C., Demailly, N., and S. Imadali, "Router Advertisements for Routing between Moving Networks", draft-petrescu-autoconf-ra-based-routing-05 (work in progress), July 2014.

[MIL-STD-1397]  
"Military Standard, Input/Output Interfaces, Standard Digital Data, Navy Systems (MIL-STD-1397B), 3 March 1989".

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

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

[RFC3027] Holdrege, M. and P. Srisuresh, "Protocol Complications with the IP Network Address Translator", RFC 3027, January 2001.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

- [RFC3879] Huitema, C. and B. Carpenter, "Deprecating Site Local Addresses", RFC 3879, September 2004.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.
- [RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPv6 Network without a Flag Day", RFC 4192, September 2005.
- [RFC4864] Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6", RFC 4864, May 2007.
- [RFC5220] Matsumoto, A., Fujisaki, T., Hiromi, R., and K. Kanayama, "Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules", RFC 5220, July 2008.
- [RFC5902] Thaler, D., Zhang, L., and G. Lebovitz, "IAB Thoughts on IPv6 Network Address Translation", RFC 5902, July 2010.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [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.
- [RFC6281] Cheshire, S., Zhu, Z., Wakikawa, R., and L. Zhang, "Understanding Apple's Back to My Mac (BTMM) Service", RFC 6281, June 2011.
- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.
- [RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.
- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.
- [RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, November 2013.



- [RFC7269] Chen, G., Cao, Z., Xie, C., and D. Binet, "NAT64 Deployment Options and Experience", RFC 7269, June 2014.
- [RS-485] "Electronic Industries Association (1983). Electrical Characteristics of Generators and Receivers for Use in Balanced Multipoint Systems. EIA Standard RS-485."
- [SCADA] "Boyer, Stuart A. (2010). SCADA Supervisory Control and Data Acquisition. USA: ISA - International Society of Automation."
- [ULA-IN-WILD] "G. Michaelson, "conference.apnic.net/data/36/apnic-36-ula\_1377495768.pdf"."

## Authors' Addresses

Bing Liu  
Huawei Technologies  
Q14, Huawei Campus, No.156 Beiqing Road  
Hai-Dian District, Beijing, 100095  
P.R. China

Email: leo.liubing@huawei.com

Sheng Jiang  
Huawei Technologies  
Q14, Huawei Campus, No.156 Beiqing Road  
Hai-Dian District, Beijing, 100095  
P.R. China

Email: jiangsheng@huawei.com

INTERNET-DRAFT  
Intended Status: Informational  
Expires: June 4, 2015

M.Nakatani  
JPCERT/CC  
Y.Kitaguchi  
Kanazawa University  
K.Nagami  
M.Kosugi  
R.Hiromi  
INTEC Inc.  
December 1, 2014

Introducing IPv6 vulnerability test program in Japan  
draft-jpcert-ipv6vulnerability-check-02

Abstract

Japan Computer Emergency Response Team Coordination Center, known as JPCERT/CC have been researching about vulnerability in use of IPv6. JPCERT/CC provided the information toward vendors in Japan. They also verified the occurring those security incidents with several products.

In 2013, JPCERT/CC called for vendors to participate their IPv6 security program. JPCERT/CC collects the results of equipments and open to the public for an user reference of procurement.

In this document we describe about the program to share the experiment of activity.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- 1 Introduction . . . . . 3
  - 1.1 Requirements Language . . . . . 3
- 2 Terminology . . . . . 3
- 3 IPv6 Vulnerability Test Program . . . . . 4
  - 3.1 Test Concept and requirement . . . . . 4
  - 3.2 Test Items and its Criteria . . . . . 4
  - 3.3 Providing Test Tools and Manual . . . . . 6
  - 3.4 Handling results . . . . . 6
- 4 Conclusion . . . . . 7
- 5 Security Considerations . . . . . 8
- 6 IANA Considerations . . . . . 8
- 7 Acknowledgements . . . . . 8
- 8 References . . . . . 9
  - 8.1 Normative References . . . . . 9
  - 8.2 Informative References . . . . . 14
- Appendix A: IPv6 vulnerability reference RFCs and i-Ds . . . . . 15
- Authors' Addresses . . . . . 19

## 1 Introduction

JPCERT/CC started "The IPv6 Security Test" in Japan in 2013. The target equipments are routers and to verify their ability for the protection of vulnerabilities which are pointed out in RFC or Internet-Drafts. JPCERT/CC focuses exclusively on the possible attacks coming from the Internet. Providing test materials(tool and document), JPCERT/CC collects the results from vendors and published IPv6 Security Test respondent product List. This list is keeping to be up to date. In this document we describe about the program to share this experimental activity.

### 1.1 Requirements Language

Take careful note: Unlike other IETF documents, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are not used as described in RFC 2119 [RFC2119]. This document uses these keywords not strictly for the purpose of interoperability, but rather for the purpose of establishing industry-common baseline functionality. As such, the document points to several other specifications (preferable in RFC or stable form) to provide additional guidance to implementers regarding any protocol implementation required to produce a successful CE router that interoperates successfully with a particular subset of currently deploying and planned common IPv6 access networks.

## 2 Terminology

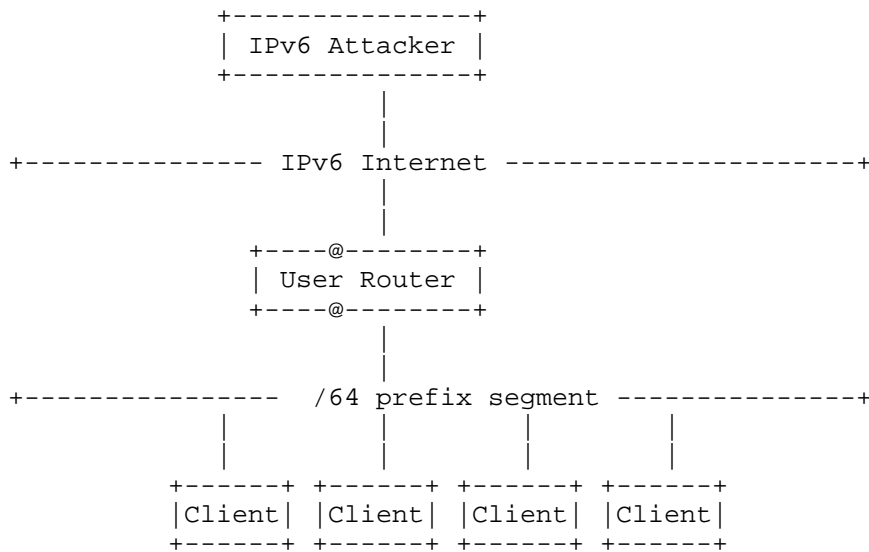
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

### 3 IPv6 Vulnerability Test Program

#### 3.1 Test Concept and requirement

This test program is focused on exclusively on the inbound attacks which possibly caused at WAN port(then through LAN port). JPCERT/CC narrowed down 15 items out of 80[Appendix.A]. Fig.1 shows basic network topology. In this test. Basically test packets sent to both LAN and WAN then confirm the robustness.

Figure.1 Basic Network Topology



#### 3.2 Test Items and its Criteria

Here is 15 test items.

- [01] Disabling type 0 routing header processing
- [02] Protection for a DoS attack on the router by hop-by-hop option header
- [03] Protection for unexpected jumbo packet by extra large payload option
- [04] Corresponding completely overwrite packet information by unauthorized fragment header(overlap-first-zero fragmentation)
- [05] Corresponding completely overwrite packet information by unauthorized fragment header(overlap-last-zero fragmentation)
- [06] Corresponding partially overwrite packet information by

- unauthorized fragment header(overlap-first-hop fragmentation)
- [07] Corresponding partially overwrite packet information by unauthorized fragment header(overlap-last-hop fragmentation)
- [08] Detection of a DoS attack by tiny fragment header
- [09] Protection for tiny fragment of a DoS attack with a large amount of using the small fragment header
- [10] Protection for a DoS attack by transmitting the first fragmented packet only
- [11] Protection for a DoS attack by single fragmented packet using atomic fragment
- [12] Protection for a DoS attack by single fragmented packet with a large amount of atomic fragments
- [13] Protection for an attack from the off-path attacker by fragment ID prediction
- [14] Protection for a DoS attack to the router using the neighbor discovery service
- [15] Protection for a DoS attack by sending a large number of broken packets to the router

Table.1 Type of Attack and Criteria for the evaluation

No.	Type of Attack	Criteria
01	DoS Attack packet filtering evasion	comply the DoS resistance policy(*) discard packet or error reply
02	DoS Attack	comply the DoS resistance policy(*)
03	DoS Attack	comply the DoS resistance policy(*)
04	packet filtering evasion	discard packet or error reply
05	packet filtering evasion	discard packet or error reply
06	packet filtering evasion	discard packet or error reply
07	packet filtering evasion	discard packet or error reply
08	DoS Attack	comply the DoS resistance policy(*)
09	DoS Attack	comply the DoS resistance policy(*)
10	DoS Attack	comply the DoS resistance policy(*)
11	DoS Attack	comply the DoS resistance policy(*)

12	DoS Attack	comply the DoS resistance policy(*)	
13	DoS Attack	comply the DoS resistance policy(*)	
14	DoS Attack	comply the DoS resistance policy(*)	
15	DoS Attack	comply the DoS resistance policy(*)	

(\*) the DoS resistance policy

Router that "PASSED" this test has ability with all the result in the below.

1. do not reboot
2. do not hung-up  
(slow-down will be acceptable)
3. return to the original condition after DoS attack stopped  
(to see the condition of the router, ping to the router from a connected node)

### 3.3 Providing Test Tools and Manual

JPCERT/CC provides a testing tool to an applicant developer due to execute these tests at same procedure and methodology. Prior to the open up this test program JPCERT/CC examined test cases itself and test tool with open source software then combined some software into a distribution tool.

Current test tool includes these software ; - THC IPv6 Toolkit 2.3  
THC IPv6 Toolkit 2.3 - SI6 Networks IPv6 ToolKit v1.4.1 - nmap 6.40 - WireShark Version 1.2.15 - minicom

slight modification was made to the software to fix for the test cases.

JPCERT/CC also provides a technical guide and an manual. The technical guide is can be downloaded from their Web page[WEB] for the general test guide to public.

### 3.4 Handling results

JPCERT/CC asks for the result of the test from associate participants. Results are listed and released in the JPCERT/CC's web site[WEB] under an agreement. JPCERT/CC updates the list continually when they gets new information.

#### 4 Conclusion

IPv6 is in the way of universal deployment. In Japan, an organization named JPCERT/CC started to provide a IPv6 related security evaluation program. After one year of the activity, JPCERT/CC also publish the result of test. End users of small and mid-sized companies or SIers can refer the list for an procurement even if they have lack of knowledge about IPv6 and its security consideration. For the vendors, they can develop IPv6 secure appraisal product that suited for targeted companies in base line.

The benefit of this activity is;

- (1) developer and JPCERT/CC  
JPCERT/CC is able to informed possible threats to vendors proactively. Vendors are able to create more safer products in advance. This scheme changes incident-first to information-first approach.
- (2) customer  
Especially for a small and mid-sized companies, they are going to start to adopt IPv6 easier if they don't have much knowledge.

Currently JPCERT/CC defined 15 items for the test case. Beyond controversy they will review and enhance the test program from time to time.



5 Security Considerations

Possible security threats are same as what pointed out in original protocols and technologies referred in this document.

6 IANA Considerations

This document has no actions for IANA.

7 Acknowledgements

Thanks for the following vendors/organizations with the contribution of this activity.

IPv6 Promotion Council, Brocade Communications Systems Inc., NEC Platforms, Ltd., Furukawa Electric Co., Ltd., Hitachi Metals, Ltd, CENTURY SYSTEMS Co., Ltd and Codenomicon.

## 8 References

### 8.1 Normative References

- [RFC1858] G. Ziemba, D. Reed, and P. Traina, "Security Considerations for IP Fragment Filtering", RFC 1858, October 1995.
- [RFC1883] S. Deering, and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification (Obsoleted by RFC 2460)", RFC 1883, December 1995.
- [RFC2460] S. Deering, and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2529] B. Carpenter and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, March 1999.
- [RFC2661] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.
- [RFC2671] P. Vixie, "Extension Mechanisms for DNS (EDNS0)", RFC 2661, August 1999.
- [RFC2675] D. Borman, S. Deering and R. Hinden "IPv6 Jumbograms", RFC 2675, August 1999.
- [RFC2694] P. Srisuresh, G. Tsirtsis, P. Akkiraju and A. Heffernan, "DNS extensions to Network Address Translators (DNS\_ALG)", RFC 2694, September 1999.
- [RFC2710] S. Deering, W. Fenner and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.
- [RFC2766] G. Tsirtsis and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", RFC 2766, February 2000.
- [RFC3056] B. Carpenter and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [RFC3068] C. Huitema, "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.
- [RFC3089] H. Kitamura, "A SOCKS-based IPv6/IPv4 Gateway Mechanism", RFC 3089, April 2001.
- [RFC3128] I. Miller, "Protection Against a Variant of the Tiny

Fragment Attack", RFC 3128, June 2001.

- [RFC3142] J. Hagino and K. Yamamoto, "An IPv6-to-IPv4 Transport Relay Translator", RFC 3142, June 2001.
- [RFC3493] R. Gilligan, S. Thomson, J. Bound, J. McCann and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.
- [RFC3756] P. Nikander, J. Kempf and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.
- [RFC3775] Johnson, D., Perkins, C. and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.
- [RFC3810] R. Vida and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC3879] C. Huitema and B. Carpenter, "Deprecating Site Local Addresses", RFC 3879, September 2004.
- [RFC3964] P. Savola and C. Patel, "Security Considerations for 6to4", RFC 3964, December 2004.
- [RFC3971] J. Arkko, J. Kempf, B. Zill and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC3972] T. Aura, "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.
- [RFC3973] A. Adams, J. Nicholas and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, January 2005.
- [RFC4191] R. Draves and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.
- [RFC4193] R. Hinden and B. Haberman, "Unique Local IPv6 Addresses", RFC 4193, October 2005.
- [RFC4225] P. Nikander, J. Arkko, T. Aura, G. Montenegro and E. Nordmark, "Mobile IP Version 6 Route Optimization Security Design Background", RFC 4225, December 2005.
- [RFC4291] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

- [RFC4380] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.
- [RFC4795] B. Aboba, D. Thaler and L. Esibov, "Link-Local Multicast Name Resolution (LLMNR)", RFC 4795, January 2007.
- [RFC4861] T. Narten, E. Nordmark, W. Simpson and H. Soliman, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] S. Thomson, T. Narten and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.
- [RFC4942] E. Davies, S. Krishnan and P. Savola, "IPv6 Transition/Co-existence Security Considerations", RFC 4942, September 2007.
- [RFC4943] S. Roy, A. Durand and J. Paugh, "IPv6 Neighbor Discovery On-Link Assumption Considered Harmful", RFC 4943, September 2007.
- [RFC4966] C. Aoun and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", RFC 4966, July 2007.
- [RFC5095] C. Malamud, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, May 2005.
- [RFC5110] P. Savola, "Overview of the Internet Multicast Routing Architecture", RFC 5110, January 2008.
- [RFC5157] T. Chown, "IPv6 Implication for Network Scanning", RFC 5157, March 2008.
- [RFC5214] Templin, F., Gleeson T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.
- [RFC5227] S. Cheshire, "IPv4 Address Conflict Detection", RFC 5227, July 2008.
- [RFC5294] P. Savola and J. Lingard, "Host Threats to Protocol Independent Multicast (PIM)", RFC 5294, August 2008.

- [RFC5572] M. Blanchet and F. Parent, "IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP)", RFC 5572, February 2010.
- [RFC5722] S. Krishnan, "Handling of Overlapping IPv6 Fragments", RFC 5722, December 2009.
- [RFC5927] F. Gont, "ICMP Attacks against TCP", RFC 5927, July 2010.
- [RFC5952] S. Kawamura and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.
- [RFC5969] W. Townsley and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.
- [RFC5991] D. Thaler, S. Krishnan and J. Hoagland, "Teredo Security Updates", RFC 5991, September 2010.
- [RFC6052] C. Bao, C. Huitema, M. Bagnulo, M. Boucadair and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [RFC6104] T. Chown and S. Venaas, "Rogue IPv6 Router Advertisement Problem Statement", RFC 6104, February 2011.
- [RFC6105] E. Levy-Abegnoli, G. Van de Velde, C. Popoviciu and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, February 2011.
- [RFC6106] J. Jeong, S. Park, L. Beloeil and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.
- [RFC6144] F. Baker, X. Li, C. Bao and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.
- [RFC6145] X. Li, C. Bao and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6146] M. Bagnulo, P. Matthews and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6147] M. Bagnulo, A. Sullivan, P. Matthews and I. Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.
- [RFC6169] S. Krishnan, D. Thaler and J. Hoagland, "Security Concerns

with IP Tunneling", RFC 6169, April 2011.

- [RFC6275] C. Perkins, D. Johnson and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.
- [RFC6296] M. Wasserman and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.
- [RFC6324] G. Nakibly and F. Templin, "Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", RFC 6324, August 2011.
- [RFC6437] S. Amante, B. Carpenter, S. Jiang and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.
- [RFC6564] S. Krishnan, J. Woodyatt, E. Kline, J. Hoagland and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, April 2012.
- [RFC6583] I. Gashinsky, J. Jaeggli and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, March 2012.
- [RFC6586] J. Arkko and A. Keranan, "Experiences from an IPv6-Only Network", RFC 6586, April 2012.
- [ID-dns-discovery] D. Thaler and J. Hagino, "IPv6 Stateless DNS Discovery ", draft-ietf-ipngwg-dns-discovery-03, November 2001. (expired)
- [ID-ipv6-hopbyhop] S. Krishnan, "The case against Hop-by-Hop options", draft-krishnan-ipv6-hopbyhop-05, October 2010. (expired)
- [RFC6762] S. Cheshire and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.
- [RFC6763] S. Cheshire and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.
- [ID-ipv6-smurf-amplifier] F. Gont, "Security Implications of IPv6 options of Type 10xxxxxx", draft-gont-6man-ipv6-smurf-amplifier-02, March 2013. (expired)
- [ID-tiny-fragments-issues] V. Manral, "Tiny Fragments in IPv6", draft-manral-6man-tiny-fragments-issues-00, February 2012. (expired)
- [ID-predictable-fragment-id] F. Gont, "Security Implications of

Predictable Fragment Identification Values", draft-ietf-6man-predictable-fragment-id-01, April 2014.

[ID-flowlabel-security] F. Gont, "Security Assessment of the IPv6 Flow Label", draft-gont-6man-flowlabel-security-03, March 2012. (expired)

[RFC6889] R. Renno, T. Saxena, M. Boucadair and S. Sivakumar, "Analysis of Stateful 64 Translation", RFC 6889, April 2013.

[ID-dnsop-respsize] P. Vixie and A. Kato, "DNS Referral Response Size Issues", draft-ietf-dnsop-respsize-15, February 2014.

[RFC7112] F. Gont and V. Manral, "Implications of Oversized IPv6 Header Chains", RFC 7112, January 2014.

[ID-slaac-dns-config-issues] F. Gont and P. Simerda, "Current issues with DNS Configuration Options for SLAAC", draft-gont-6man-slaac-dns-config-issues-00, June 2012. (expired)

[ID-dhcpv6-shield] F. Gont, W. Liu and G. Van de Velde, "DHCPv6-Shield: Protecting Against Rogue DHCPv6 Servers", draft-ietf-opsec-dhcpv6-shield-04, July 2014.

[RFC7113] F. Gont, "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", RFC 7113, February 2014.

[RFC6946] F. Gont, "Processing of IPv6 "atomic" fragments", RFC 6946, May 2013.

[RFC6980] F. Gont, "Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery", RFC 6980, August 2013.

[RFC7123] F. Gont and W. Liu, "Security Implications of IPv6 on IPv4 Networks", RFC 7123, February 2014.

[ID-ipv6-host-scanning] F. Gont and T. Chown, "Network Reconnaissance in IPv6 Networks", draft-ietf-opsec-ipv6-host-scanning-04, June 2014.

## 8.2 Informative References

[WEB] JPCERT/CC, IPv6 Security Test Appraisal List, September 2014, <[https://www.jpCERT.or.jp/research/ipv6product\\_list.html](https://www.jpCERT.or.jp/research/ipv6product_list.html)>.

## Appendix A: IPv6 vulnerability reference RFCs and i-Ds

Here is possible threats list and related RFC and internet-drafts.

## 1. Basic Header/Extension Header definition

- 1-1 Access filtering policy evasion using by Type 0 Routing Header, RFC4942;RFC5095;RFC5871
- 1-2 DoS attack caused by Type 0 Routing Header, RFC4942;RFC5095;RFC5871
- 1-3 DoS attack caused by Hop by Hop Option Header, RFC4942
- 1-4 Handling problem and resource management problem of jumbogram, RFC4942
- 1-5 Packet overwrite by unauthorized fragment header, RFC4942;RFC5722
- 1-6 DoS attack caused by tiny fragmented packets, RFC7112
- 1-7 Abuse by receiving a lot of first fragment packets
- 1-8 DoS attack caused by atomic fragment header, RFC6946
- 1-9 DoS attack caused by prediction of fragment identification values, draft-ietf-6man-predictable-fragment-id-01
- 1-10 Distinctiveness on firewall implementation for packet reassembly, RFC4942;RFC7112;RFC5722
- 1-11 Implementation problems in processing extension header chain; RFC4942;RFC7112;RFC5722
- 1-12 Implementation problems in Unknown Headers/Destination Options, RFC4942;RFC6564
- 1-13 Abuse using by Pad1 and PadN Options in Hop-by-Hop and Destination option headers, RFC4942
- 1-14 DoS attack using by old specification of Flow Label, RFC3697;RFC6437
- 1-15 Covert Channel using by Flow Label, RFC6437;draft-gont-6man-flowlabel-security-03
- 1-16 Information Leaking by Flow Label, RFC6437;draft-gont-6man-flowlabel-security-03

## 2. NDP (link layer address resolution)

- 2-1 Neighbor Solicitation/Advertisement Spoofing, RFC3756;RFC6980
- 2-2 Neighbor Unreachability Detection (NUD) failure, RFC3756;RFC6980



- 2-3 Duplicate Address Detection DoS Attack,  
RFC3756;RFC6980;draft-ietf-6man-enhanced-dad-06
- 2-4 Neighbor Discovery DoS Attack,  
RFC3756;RFC4942
- 2-5 Abuse on Neighbor cache table,  
RFC3756;RFC4942
- 3. NDP (address auto-configuration)
  - 3-1 Juggled default route,  
RFC3756;RFC6104;RFC6105;RFC7113
  - 3-2 Juggled prefixes,  
RFC3756;RFC6104;RFC6105;RFC7113
  - 3-3 Juggled DNS server information,  
RFC3756;RFC6104;RFC6105;RFC6106;draft-gont-6man-slaac-dns-  
config-issues-00
  - 3-4 Sniffing caused by following old specification of on-link  
assumption,  
RFC3756;RFC4943;RFC6104;RFC6105;RFC6583;RFC7113
  - 3-5 Parameter Spoofing,  
RFC3756;RFC6104;RFC6105;RFC7113
  - 3-6 DoS attack caused by Router Advertisement,  
RFC3756;RFC6104;RFC6105;RFC7113
  - 3-7 Filtering Policy Evasion by fragment packets  
RFC7113;RFC5722
- 4. ICMPv6
  - 4-1 Spoofed Redirect Message,  
RFC3756;draft-gont-opsec-ipv6-nd-shield-00;RFC6980
  - 4-2 DoS attack to Upper-layer protocol by crafted ICMPv6 error  
messages,  
RFC4942;RFC5927
  - 4-3 Covert conversation through the payload of ICMPv6 error  
messages,  
RFC4942
  - 4-4 DoS attack by unprocessable packets to router,  
RFC4942;RFC5927
- 5. IP Address definition
  - 5-1 Anycast Traffic Identification,  
RFC4942;RFC4291
  - 5-2 Site Local Address as well-known DNS server addresses,  
draft-ietf-ipngwg-dns-discovery-03;RFC6586
  - 5-3 Malicious use of IPv6 addressing scheme,  
RFC4942;RFC5157;draft-ietf-opsec-ipv6-host-scanning-04
  - 5-4 Dynamic DNS and secure updates,

- RFC4942;RFC4472
- 5-5 Complexity on plural address operating by IPv4-mapped address,  
RFC4942
- 5-6 Filtering policy evasion using by IPv4-mapped address  
RFC4942
- 5-7 Firewalls cannot perform deep packet inspection and filtering  
with IPsec,  
RFC4942
- 5-8 IPv6 tunnels break IPv4 network security policy,  
RFC4942
- 6. Multicast
  - 6-1 DoS attack by hijacked multicast router,  
RFC3810
  - 6-2 DoS attack by forged Report message in MLD,  
RFC3810;RFC2710
  - 6-3 Extra processing on the network equipment by forged Done  
messages in MLD,  
RFC3810;RFC2710
  - 6-4 DoS attack over multicast network with ICMPv6 error messages,  
RFC4942
  - 6-5 Abuse in multicast distribution tree on PIM-DM with  
temporary addresses,  
RFC3973
  - 6-6 Denial-of-Service Attack on the Link,  
RFC5294
- 7. Mobile IPv6
  - 7-1 Attacks against Binding Update Protocols,  
RFC4225
  - 7-2 Filtering Policy evasion due to not support type 2 routing  
header,  
RFC4225;RFC6275
- 8. Tunneling
  - 8-1 Filtering Policy evasion occurred in IPv6 transition/coexistence  
technologies on "IPv4-only" networks,  
RFC4942;RFC6169;RFC7123
  - 8-2 Source Routing after the Tunnel Client combined with old  
specification of Routing Header 0,  
RFC6169;RFC5095;RFC7123
  - 8-3 Attacks by malicious use of NDP may go to 6to4 Router/6to4  
Relay Router/6rd Border Router,  
RFC3964;RFC4942;RFC5969;RFC7123
  - 8-4 Attack toward IPv6 clients from IPv4 network via

- 6to4 Router/6to4 Relay Router,  
RFC3964;RFC6169RFC5969;RFC7123
- 8-5 Attack toward 6to4 clients from IPv4 network via  
6to4 Router/6to4 Relay Router,  
RFC3964;RFC6169RFC5969;RFC7123
- 8-6 IPv4 broadcast attack via 6to4 Router/6to4 Relay Router,  
RFC3964;RFC6169RFC5969;RFC7123
- 8-7 Sniffing at 6to4 Router/6to4 Relay Router,  
RFC3964;RFC6169;RFC5969;RFC7123
- 8-8 Routing Loop Attack Using IPv6 Automatic Tunnels,  
RFC6324
- 8-9 Filtering bypass by Teredo,  
RFC6169;RFC7123
- 8-10 Port exposure with Teredo,  
RFC6169;RFC5991;RFC7123
- 8-11 Teredo Tunnel Address Concerns,  
RFC6119
- 8-12 Sniffing at Teredo Router/Teredo Relay Router,  
RFC3964;RFC6169;RFC5969;RFC7123

## 9. Translation

- 9-1 Address Spoofing used by IPv4-embedded IPv6 address,  
RFC6052;RFC6145;RFC6889
- 9-2 Concerns of using DNS64,  
RFC6147;RFC6889

## 10. DNS

- 10-1 Dual stack operation bring overloading to name servers,  
RFC4472;RFC4942;draft-ietf-dnsop-respsize-15
- 10-2 Operational difficulty of reverse zones and concerns,  
RFC4472;RFC4942
- 10-3 Rogue DHCPv6 Servers,  
draft-ietf-opsec-dhcpv6-shield-04

## 11. Other Operational concerns

- 11-1 Network segment violation by leakage of NDP in VLAN networks
- 11-2 RFC5952 text representation compliance for safer operation,  
RFC5952
- 11-3 Dual stack nodes in IPv4 only network without supervision

## Authors' Addresses

Masayuki Nakatani  
Japan Computer Emergency Response Team Coordination Center  
3-17, Kanda Nishiki-cho, Chiyoda-ku, Tokyo,  
Japan

EMail: ww-info@jpcert.or.jp

Yoshiaki Kitaguchi  
Kanazawa University  
Kakuma-machi, Kanazawa, Ishikawa,  
Japan

EMail: kitaguchi@imc.kanazawa-u.ac.jp

Kenichi Nagami  
INTEC Inc.  
1-3-3, Shinsuna, Koto-ku, Tokyo,  
Japan

EMail: nagami@inetcore.com

Masataka Kosugi  
INTEC Inc.  
626-1, Kyoda, Takaoka-City, Toyama,  
Japan

EMail: kosugi\_masataka@intec.co.jp

Ruri Hiromi  
INTEC Inc.  
1-1-25, Shin Urashima-cho, Kanagawa-ku, Yokohama,  
Japan

EMail: hiromi@inetcore.com

V6OPS  
Internet-Draft  
Intended status: Informational  
Expires: April 30, 2015

B. Liu  
Huawei Technologies  
R. Bonica  
Juniper Networks  
T. Yang  
China Mobile  
October 27, 2014

DHCPv6/SLAAC Interaction Operational Guidance  
draft-liu-v6ops-dhcpv6-slaac-guidance-03

Abstract

The IPv6 Neighbor Discovery (ND) Protocol [RFC4861] specifies an ICMPv6 Router Advertisement (RA) message. The RA message contains three flags that indicate which address autoconfiguration mechanisms are available to on-link hosts. These are the M, O and A flags. The M, O and A flags are all advisory, not prescriptive.

In [I-D.ietf-v6ops-dhcpv6-slaac-problem], test results show that in several cases the M, O and A flags elicit divergent host behaviors, which might cause some operational problems. This document aims to provide some operational guidance to eliminate the impact caused by divergent host behaviors as much as possible.

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 30, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Operational Guidance . . . . .	3
2.1. Always Turn RAs On . . . . .	3
2.2. Guidance for DHCPv6/SLAAC Provisioning Scenarios . . . . .	3
2.2.1. DHCPv6-only . . . . .	3
2.2.2. SLAAC-only . . . . .	4
2.2.3. DHCPv6/SLAAC Co-existence . . . . .	4
2.3. Guidance for Renumbering . . . . .	5
2.3.1. Adding a New Address from another Address Configuration Mechanisms . . . . .	5
2.3.2. Switching one Address Configuration Mechanism to another . . . . .	6
3. Security Considerations . . . . .	6
4. IANA Considerations . . . . .	6
5. Acknowledgements . . . . .	6
6. References . . . . .	7
6.1. Normative References . . . . .	7
6.2. Informative References . . . . .	7
Authors' Addresses . . . . .	7

## 1. Introduction

The IPv6 Neighbor Discovery (ND) Protocol [RFC4861] specifies an ICMPv6 Router Advertisement (RA) message. The RA message contains three flags that indicate which address autoconfiguration mechanisms are available to on-link hosts. These are the M, O and A flags. The M, O and A flags are all advisory, not prescriptive.

In [I-D.ietf-v6ops-dhcpv6-slaac-problem], test results show that in several cases the M, O and A flags elicit divergent host behaviors, which might cause some operational problems. This document aims to provide some operational guidance to eliminate the impact caused by divergent host behaviors as much as possible.

This document does not intent to cover the topic of selection between RA and DHCPv6 [RFC3315] for the overlapped functions. There always

are arguments about what should be done through RA options or through DHCPv6 options. For this general issue, draft [I-D.yourtchenko-ra-dhcpv6-comparison] could be referred.

## 2. Operational Guidance

### 2.1. Always Turn RAs On

Currently, turning RAs on is actually a basic requirement for running IPv6 networks since only RAs could advertise default route(s) for the end nodes. And if the nodes want to communicate with each other on the same link via DHCPv6-configured addresses, they also need to be advertised with L flag set in RAs. So for current networks, an IPv6 network could NOT run without RAs, unless the network only demands a communication via link-local addresses.

### 2.2. Guidance for DHCPv6/SLAAC Provisioning Scenarios

#### 2.2.1. DHCPv6-only

In IPv4, there is only one method (DHCPv4) for automatically configuring the hosts. Many network operations/mechanisms, especially in enterprise networks, are built around this central-managed model. So it is reasonable for people who are accustomed to DHCPv4-only deployment still prefer DHCPv6-only in IPv6 networks. Besides, some networks just prefer central management of all IP addressing. These networks may want to assign addresses only via DHCPv6.

This can be accomplished by sending RAs that indicate DHCPv6 is available (M=1), installing DHCPv6 servers or DHCPv6 relays on all links, and setting A=0 in the Prefix Information Options of all prefixes in the RAs. (Instead of forcing the A flag off, simply not including any PIO in RAs could also make the same effect). But before doing this, the administrators need to be sure that every node in their intended management scope supports DHCPv6.

Note that RAs are still necessary in order for hosts to be able to use these addresses. This is for two reasons:

- o If there is no RA, some hosts will not attempt to obtain address configuration via DHCPv6 at all.
- o DHCPv6 can assign addresses but not routing. Routing can be implemented on hosts by means of accepting and implementing information from RA messages containing default-route, Prefix Information Option with O=1, or Route Information Option, or by configuring manual routing. Without routing, IPv6 addresses won't

be used for communication outside the host. Thus, for example, if there is no RA and no static routing, then addresses assigned by DHCPv6 cannot be used even for communication between hosts on the same link.

Also note that unlike SLAAC [RFC4862], DHCPv6 is not a strict requirement for IPv6 hosts [RFC6434], and some nodes do not support DHCPv6. Thus, this model can only be used if all the hosts that need IPv6 connectivity support DHCPv6.

#### 2.2.2. SLAAC-only

In contrast with DHCPv6-only, some scenarios might be suitable for SLAAC-only which allows minimal administration burden and node capability requirement.

The administrators MUST turn the A flag on, and MUST turn M flag off. Note that some platforms (e.g. Windows 8) might still initiate DHCPv6 session regardless of M flag off. But since there is no DHCPv6 service available, the only problem is that there would be some unnecessary traffic.

#### 2.2.3. DHCPv6/SLAAC Co-existence

##### - Scenarios of DHCPv6/SLAAC Co-existence

- \* For provisioning redundancy: If the administrators want all nodes at least could configure a global scope address, then they could turn A flag and M flag both on in case some nodes only support one of the mechanisms. For example, some hosts might only support SLAAC; while some hosts might only support DHCPv6 due to manual/mistaken configurations.
- \* For different provisioning: the two address configuration mechanisms might provide two addresses for the nodes respectively. For example, SLAAC-configured address is for basic connectivity and another address configured by DHCPv6 is for a specific service.

##### - Cautions

- \* Notice that enabling both DHCPv6 and SLAAC would cause one host to configure more IPv6 addresses. Typically, there would be one more DHCPv6-configured address than SLAAC-only configuration; and two more addresses based on SLAAC and privacy extension than DHCPv6-only configuration. Too many addresses might cause ND cache overflow problem in some



situations (please refer to Section 3.4 of [I-D.liu-v6ops-running-multiple-prefixes] for details).

- \* For provisioning redundancy scenario, there is a concern that SLAAC/DHCPv6 addresses based on the same prefix might cause some applications confusing. [Open Question] Call for real experiences on this issues.
- \* Besides address configuration, DNS can also be configured both by SLAAC and DHCPv6. If the DNS information in RAs and DHCPv6 are different, the host might confuse. So in terms of operation, the operators should make sure DNS configuration in RAs and DHCPv6 are the same.

### 2.3. Guidance for Renumbering

This document only considers the renumbering cases where DHCPv6/SLAAC interaction is involved. These renumbering operations need the A/M flags transition which might cause unpredictable host behaviors. Two renumbering cases are discussed as the following.

#### 2.3.1. Adding a New Address from another Address Configuration Mechanisms

- o Adding a DHCPv6 Address for a SLAAC-configured Host

As discussed in Section 2.2.3, some operating systems that having configured SLAAC addresses would NOT care about the newly added DHCPv6 provision unless the current SLAAC address lifetime is expired. In theory, one possible way is to stop advertising RAs and wait the SLAAC addresses expired (this makes the hosts return to the initial stage), then advertise RAs again with the M flag set, so that the host would configure SLAAC and DHCPv6 addresses simultaneously. However, there would be some outage period during this operation, which might be unacceptable for many situations. Thus, It is better for the administrators to carefully plan the network provisioning so that to make SLAAC and DHCPv6 available simultaneously (through RA with M=1) at the initial stage rather than configuring one and then configuring another.

- o Adding a SLAAC Address for a DHCPv6-configured Host

As tested in [I-D.ietf-v6ops-dhcpv6-slaac-problem]), current mainstream operating systems all support this renumbering operation. The only thing need to care about is to make sure the M flag is on in the RAs, since some operating systems would immediately release the DHCPv6 addresses if M flag is off.

### 2.3.2. Switching one Address Configuration Mechanism to another

#### o DHCPv6 to SLAAC

This operation is supported by all the tested operating systems in [I-D.ietf-v6ops-dhcpv6-slaac-problem]. However, the behaviors are different. As said above, if A flag is on while M flag is off, a flash switching renumbering would happen on some operating systems. So while turning the A flag on, it is recommended to retain the M flag on and stop the DHCPv6 server to response the renew messages so that the DHCPv6 addresses could be released when the lifetimes expired.

#### o SLAAC to DHCPv6

This operation is also supported by all the tested operating systems. And the behaviors are the same since no operating systems would immediatly release the SLAAC addresses when A flag is off. However, for safe operation, while turning the M flag on, it is also recommended to retain the A flag on and stop advertising RAs so that the SLAAC addresses could be released when the lifetimes expired.

### 3. Security Considerations

No more security considerations than the Neighbor Discovery protocol [RFC4861].

### 4. IANA Considerations

This draft does not request any IANA action.

### 5. Acknowledgements

Valuable comments were received from Sheng Jiang and Brian E Carpenter to initiate the draft. Some texts in Section 2.2.1 were based on Lorenzo Colitti and Mikael Abrahamsson's proposal. There were also comments from Erik Nordmark, Ralph Droms, John Brzozowski, Andrew Yourtchenko and Wesley George to improve the draft. The authors would like to thank all the above contributors.

This document was produced using the xml2rfc tool [RFC2629]. (This document was initially prepared using 2-Word-v2.0.template.dot. )

## 6. References

### 6.1. Normative References

- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, December 2011.

### 6.2. Informative References

- [I-D.ietf-v6ops-dhcpv6-slaac-problem]  
Liu, B., Jiang, S., Bonica, R., Gong, X., and W. Wang, "DHCPv6/SLAAC Address Configuration Interaction Problem Statement", draft-ietf-v6ops-dhcpv6-slaac-problem-02 (work in progress), October 2014.
- [I-D.liu-v6ops-running-multiple-prefixes]  
Liu, B., Jiang, S., and Y. Bo, "Considerations for Running Multiple IPv6 Prefixes", draft-liu-v6ops-running-multiple-prefixes-02 (work in progress), October 2014.
- [I-D.yourtchenko-ra-dhcpv6-comparison]  
Yourtchenko, A., "A comparison between the DHCPv6 and RA based host configuration", draft-yourtchenko-ra-dhcpv6-comparison-00 (work in progress), November 2013.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

### Authors' Addresses

Bing Liu  
Huawei Technologies  
Q14, Huawei Campus, No.156 Beiqing Road  
Hai-Dian District, Beijing, 100095  
P.R. China

Email: leo.liubing@huawei.com

Ron Bonica  
Juniper Networks  
Sterling, Virginia  
20164  
USA

Email: rbonica@juniper.net

Tianle Yang  
China Mobile  
32, Xuanwumenxi Ave.  
Xicheng District, Beijing 100053  
P.R. China

Email: yangtianle@chinamobile.com

V6OPS  
Internet-Draft  
Intended status: Informational  
Expires: September 26, 2015

B. Liu  
S. Jiang  
Y. Bo  
Huawei Technologies  
March 25, 2015

Multiple IPv6 Prefixes: Background and Considerations  
draft-liu-v6ops-running-multiple-prefixes-03

Abstract

This document describes several typical multiple prefixes use cases, and discusses that running multiple IPv6 prefixes/addresses in one network/host should be common practice that administrators need to adapt.

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 September 26, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Multiple Prefixes Use cases . . . . .	3
2.1. Multiple Prefixes with Different Scopes . . . . .	3
2.2. Multihoming based on Multiple PA Prefixes . . . . .	3
2.3. Multiple Prefix Co-existing during Network Renumbering . . . . .	4
2.4. Service Prefixes . . . . .	4
3. Operational Availability and Considerations . . . . .	4
3.1. Multiple prefix provisioning . . . . .	4
3.2. Address Selection . . . . .	5
3.3. Exit-router selection . . . . .	5
4. Security Considerations . . . . .	6
5. IANA Considerations . . . . .	6
6. Acknowledgements . . . . .	6
7. References . . . . .	6
7.1. Normative References . . . . .	6
7.2. Informative References . . . . .	7
Authors' Addresses . . . . .	8

## 1. Introduction

In IPv6 networks, there are deployment scenarios in which multiple prefixes coexists simultaneously in one network. Several typical use cases are:

- Multiple Prefixes with Different Scopes (described in Section 2.1)
- IPv6 multihoming based on multiple PA prefixes (described in Section 2.2)
- Make-before-break renumbering (described in Section 2.3)
- An IPv6 network with multiple services, each of which has a distinct prefix (described in Section 2.4) .

To support the multiple prefixes running mode, there have been some technologies developed. This document discusses these technologies of different aspects, which could allow and smoothen the multiple prefix operation.

Note that, although MIF (Multiple InterFaces) [RFC6418] architecture also involves multiple IPv6 prefixes, it mainly targets different interfaces which attach to different networks respectively. This document discusses the multiple IPv6 prefixes running in the same network.

## 2. Multiple Prefixes Use cases

### 2.1. Multiple Prefixes with Different Scopes

IPv6 contains link-local addresses, global addresses and unique local addresses, which by definition are global but normally are site-scope by practice.

As specified in [RFC4291], all interfaces are required to have at least one Link-Local unicast address. This is the basic case of running multiple prefixes. However, this does not require operations from the network administrators since it is automatically processed.

Besides Link-Local addresses, the Unique Local Addresses (ULAs, [RFC4193]) might also be used for the internal communication within a site network. In many deployment, the ULA is used along with PA (Provider Aggregated) addresses, which connect to the public network. The benefit of such combination is to provide separate local communication from the globally communication so that the local communication would not be impacted when ISP uplink fail or prefix(es) be renumbered. It is especially beneficial for the home network and private OAM plane or internal-only nodes in an enterprise.

### 2.2. Multihoming based on Multiple PA Prefixes

When a network is multihomed, the multiple upstream network providers would assign prefixes respectively. If a network does not acquire a PI (Provider Independent) address space, multihoming will result coexistent multiple PA prefixes. In such network, a single host have multiple PA IPv6 addresses that associated with different prefixes.

This scenario rarely exists in IPv4 networks, since IPv4 only allows single address per interface. But it is quite practical in IPv6. This new feature of IPv6 allows the SMEs (Small/Medium Enterprises) to multihome without the burden of running PI address space or running IPv6 NAT. Furthermore, multiple PA spaces do not have the potential global routing system scalable issue as the PI does [RFC4894].

However, multihoming with multiple PA prefixes has some operational issues which mainly include address selection, next-hop selection, and exit-router selection. For detailed discussion, please refer to [RFC7157]. [Editor's note: more discussion to be filled.]

### 2.3. Multiple Prefix Co-existing during Network Renumbering

[RFC4192] describes a procedure that can be used to renumber a network from one prefix to another smoothly through a "make-before-break" transition. In the transition period, both the old and new prefixes are available; the usage of multiple prefixes provides the smooth transition and avoids the session outage issue in most of renumbering operations.

### 2.4. Service Prefixes

An IPv6 network may simultaneously provide multiple services, such as IPTV, Internet access, VPN, etc. Each of these services should have a distinct prefix. The network may apply different policy based on the distinguished prefixes. This deployment would simplify the management and processing on network devices, such as forwarding routers, access authentication devices, account devices, border filter, etc. The ISPs would provide one subscriber multiple addresses/prefixes to access different services. This deployment would particularly benefit for traffic recognition and management.

## 3. Operational Availability and Considerations

This section discusses some technologies of different aspects, which could allow and smooth the multiple prefix operation.

### 3.1. Multiple prefix provisioning

#### o Multiple Prefixes from Different Provisioning Domains

In [I-D.ietf-mif-mpvd-arch], provisioning domain is defined as consistent set of network configuration information. Classically, the entire set available on a single interface is provided by a single source, such as network administrator, and can therefore be treated as a single provisioning domain.

But in modern IPv6 networks, multihoming or service prefixes may result in provisioning information from more than one provisioning domains being presented on a single link. In these scenarios, current technologies lack support of distinguishing information from multiple provisioning domains, thus the host would not be able to associate configuration information with provisioning domains.

However, there are several techniques under developing in MIF WG to solve the problems, we could expect them to be standardized in the near future.



- o Co-existing DHCPv6/SLAAC

Both SLAAC [RFC4862] and DHCPv6-PD [RFC3633] could assign IPv6 prefixes. DHCPv6-PD is normally run between routers and routers or routers and DHCPv6 [RFC3315] servers; while SLAAC is normally run between routers and downstream hosts. The two protocols could collaborate sufficiently to cover the whole network's prefix provisioning.

If operate properly, SLAAC and DHCPv6 could also co-exist for IPv6 addresses provisioning based on different prefixes. They need to carefully deal with the interaction between the two protocols. It is mostly regarding to the M flag in Neighbor Discovery [RFC4861] messages.

### 3.2. Address Selection

In order to support multiple addresses well, IPv6 introduced address selection mechanism which utilize a address selection policy table to calculate a proper source address for a given destination address. Of course, destination addresses selection is also defined. [RFC6724] described the rationale and algorithms in detail, and also defines a default address selection policy table for operating systems.

Note that, the [RFC6724] is a replacement of the old [RFC3484] specification to improve some behaviors (e.g. to prefer IPv4 over ULA for outside connectivity). Currently, so far there haven't been many operating systems supporting the new standard, but we could expect that the new standard would be available in all new released operating systems and becomes the mainstream in the near future.

### 3.3. Exit-router selection

In multiple PA multihoming networks, if the ISPs enable ingress filtering at the edge (BCP38, [RFC2827]), then there comes the exit router selection issues that outgoing packets are routed to the appropriate border router and ISP link. Normally, a packet sourced from an address assigned by ISP X should not be sent via ISP Y, otherwise it would be filtered by ISP Y.

In the past, the administrators have to either communicate with the ISP for not filtering the prefixes or manually configure routing policies within the network to make sure the traffics are forwarded to the right upstream link, based on source prefixes. Now, there are some source-based routing technologies under development and standardization. We could expect these solutions available soon.

#### 4. Security Considerations

This document does not introduce any new mechanisms or protocols technologies and as such does not introduce any new security threads.

Nevertheless, relevant important security considerations are worth to be iterated here:

- o [RFC7157] gives the security considerations for multi-prefix based multihoming.
- o Address selection relevant security considerations are described in [RFC6724].
- o ND cache exhaustion caused by multiple addresses per host in a big L2 network is described in Section 3.2. It is possibility that malicious users intentionally configure massive addresses on host to make the gateway ND cache exhausted. So administrators always need to consider mitigation operations for potential ND cache DoS attack which is documented as [RFC6583].

#### 5. IANA Considerations

This draft does not request any IANA action.

#### 6. Acknowledgements

Valuable inputs of the texts/ideas were from Ole Troan.

Useful comments were received from Brian Carpenter, Victor Kuarsingh, Lorenzo Colliti, Mikael Abrahamsson, Fred Baker, Lee Howard and Roberta Maglione.

This document was produced using the xml2rfc tool [RFC2629].  
(initiallly prepared using 2-Word-v2.0.template.dot. )

#### 7. References

##### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

## 7.2. Informative References

- [I-D.ietf-mif-mpvd-arch] Anipko, D., "Multiple Provisioning Domain Architecture", draft-ietf-mif-mpvd-arch-11 (work in progress), March 2015.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.
- [RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPv6 Network without a Flag Day", RFC 4192, September 2005.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4894] Hoffman, P., "Use of Hash Algorithms in Internet Key Exchange (IKE) and IPsec", RFC 4894, May 2007.
- [RFC6418] Blanchet, M. and P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement", RFC 6418, November 2011.
- [RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, March 2012.

- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown,  
"Default Address Selection for Internet Protocol Version 6  
(IPv6)", RFC 6724, September 2012.
- [RFC6879] Jiang, S., Liu, B., and B. Carpenter, "IPv6 Enterprise  
Network Renumbering Scenarios, Considerations, and  
Methods", RFC 6879, February 2013.
- [RFC7157] Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D.  
Wing, "IPv6 Multihoming without Network Address  
Translation", RFC 7157, March 2014.

Authors' Addresses

Bing Liu  
Huawei Technologies  
Q14, Huawei Campus, No.156 Beiqing Road  
Hai-Dian District, Beijing, 100095  
P.R. China

Email: leo.liubing@huawei.com

Sheng Jiang  
Huawei Technologies  
Q14, Huawei Campus, No.156 Beiqing Road  
Hai-Dian District, Beijing, 100095  
P.R. China

Email: jiangsheng@huawei.com

Bo Yang  
Huawei Technologies  
Q21, Huawei Campus, No.156 Beiqing Road  
Hai-Dian District, Beijing, 100095  
P.R. China

Email: boyang.bo@huawei.com

IPv6 Operations Working Group (v6ops)  
Internet-Draft  
Intended status: Experimental  
Expires: April 30, 2015

O. Nakamura  
Keio Univ./WIDE Project  
H. Hazezama  
NAIST / WIDE Project  
Y. Ueno  
Keio Univ./WIDE Project  
A. Kato  
Keio Univ. / WIDE Project  
October 27, 2014

A Special Purpose TLD to resolve IPv4 Address Literal on DNS64/NAT64  
environments  
draft-osamu-v6ops-ipv4-literal-in-url-02

#### Abstract

In an IPv6-only environment with DNS64/NAT64 based translation service, there is no way to get access a URL whose domain name part includes an IPv4 address literal. This memo proposes a special purpose TLD so that the IPv4 address literal is accessible from such a DNS64/NAT64 environments.

#### 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 30, 2015.

#### Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## 1. Introduction and Overview

When a host in an IPv6 only environment (an IPv6-only host) has to access an IPv4-only destination, a translator-based approach is a powerful tool. The translator-based approach is usually composed of a DNS64 server [RFC6147] and a stateful NAT64 translator [RFC6146]. The DNS64 server responds with a AAAA record of an IPv4 embedded IPv6 address with a certain IPv6 prefix assigned to the NAT64 translator, for example, the well known NAT64 prefix (64:ff9b::) or a global IPv6 prefix. The IPv6-only host sends an IPv6 packet, which is translated by the NAT64 box to an IPv4 packet. In this memo, an IPv4 embedded IPv6 address with a NAT64 prefix is described as ``Pref64::/n address''. The translation of responded IPv4 packet back into an IPv6 packet is also performed in the NAT64 translator.

The NAT64 with DNS64 approach works well for most destinations. But it does not work well when the DNS response packet resulted NXDOMAIN or SERVFAIL to the AAAA query, partly described in [RFC4074]. Resolutions of this case are out of scope of this memo.

It is legitimate to embed an IPv4 address literal in an URL such as follows:

```
http://192.0.2.10/index.html
```

In the environment described above, the destination is not accessible from an IPv6-only host. This problem has already been reported in [RFC6586] and others.

The reason why the destination specified by above notation cannot be

accessible is that no DNS lookup is performed, and no DNS64 service is able to tell a Pref64::

This memo proposes a special-purpose TLD and defines behaviors of resolvers and of the authoritative servers to treat the special-purpose TLD. This memo also considers implementation strategy of .TLD and side effects of .TLD usages to the current communications on the Internet. The special-purpose TLD is denoted as .TLD which will be replaced with an actual TLD allocated by IANA.

The concept of .TLD is simple: All IPv4 address literal notations are rewritten to ``<ipv4-address-literal>.TLD'' on a host. As ``<ipv4-address-literal>.TLD'' is seemed to be a regular FQDN, ``<ipv4-address-literal>.TLD'' lets DNS64 servers resolve IPv4 address literal as a regular FQDN and translate the A record of ``<ipv4-address-literal>.TLD'' to a corresponding Pref64::

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

## 2. Scope of this memo

This memo focuses only on smooth migration to an IPv6-only environment with the DNS64/NAT64 solution. Therefore, this memo focuses on only ``IPv4 address literal'' problem mentioned in [RFC6586].

The ``IPv6 address literal'' is out of scope of this memo, because an URL including IPv6 address literal can be accessible in IPv6-only networks and in dual stack networks. The solutions to keep IPv4-only hosts or IPv4-only applications in IPv6 only environment are out of scope on this memo.

## 3. A special-purpose TLD for IPv4 Address Literal

When the part of IPv4 address literal is written to form a pseudo FQDN and the pseudo FQDN is resolved as an IPv4 address, a DNS64 server can return a AAAA record with the specified IPv4 address that is mapped to an appropriate NAT64 prefix.

Once a AAAA record is obtained, the IPv6-only host can send IPv6 packets to the destination. IPv6 packets will be translated back via NAT64 translator in exactly the same as a regular IPv4-only destination.

### 3.1. .TLD Authoritative DNS server behavior

The authoritative DNS server of .TLD SHOULD be operated only for a special purpose.

1. If a DNS query asks ``<ipv4-address-literal>.TLD``, .TLD authoritative server MUST return ``<ipv4-address-literal>`` as the A record of ``<ipv4-address-literal>.TLD``.
2. Otherwise, .TLD authoritative server MUST return NXDOMAIN.

### 3.2. DNS64 behaviors

When a DNS64 receives a query of <ipv4-address-literal>.TLD, it SHOULD issue a DNS query to one of the .TLD authoritative servers. The response from .TLD authoritative server will be either an A record of the issued <ipv4-address-literal> or NXDOMAIN. If the response contains an A record, the DNS64 MUST translate the IPv4 address in the A record to the AAAA record by Pref64:: address according to [RFC6147].

Taking into account of scalability, the DNS64 WOULD cache the AAAA record of <ipv4-address-literal>.TLD in a certain interval. As one of possible ways to get more scalability, the DNS64 CLOUD have the function of .TLD authoritative server.

### 3.3. Client behaviors

#### 3.3.1. Case 1: manual type-writing

When a client (human) wants to access an IPv4 only server by IPv4 address literal in a DNS64/NAT64 network, he / she manually attaches .TLD to the IPv4 address of the IPv4 only server. When the network has DNS64/NAT64 function, the AAAA record, that is Pref64:: address of the issued <ipv4-address-literal> , will be return.

The client COULD attach .TLD to the IPv4 address of the IPv4 only server in an IPv4 only network or a dual stack network. When the network situation is IPv4 only or dual stack, the A record of the issued <ipv4-address-literal>.TLD will be returned.

If the client uses FQDN or IPv6 address literal, he / she MUST NOT attach .TLD.



### 3.3.2. Case 2: device or application

A client (device or application), that has a name resolution function, SHOULD attach .TLD when the input value of getaddrinfo is an IPv4 address literal. For example, <ipv4-address-literal> SHOULD be rewritten to <ipv4-address-literal>.TLD. If the input value of getaddrinfo is not IPv4 address literal, the client MUST NOT attach .TLD.

Of course, the client CAN take self-synthesizing of mapped address mentioned in [RFC7050], or MAY combine .TLD method and [RFC7050] self-synthesizing method.

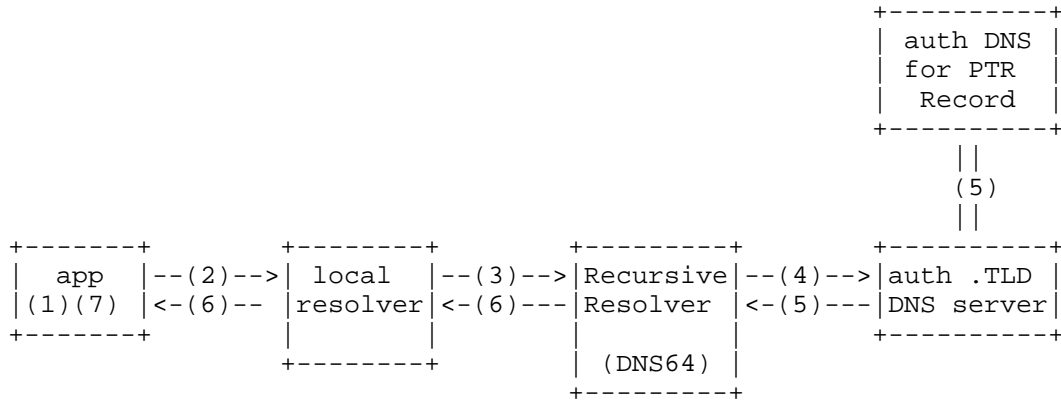
Some access authentication may not allow any external accesses until access authentication procedure is finished, and may use an IPv4 address literal on the redirected authentication web page. Taking into account such corner case, client WOULD check the reachability to the external network initially.

NOTE: migrating from IPv4 to IPv6, access authentication SHOULD avoid to use IPv4 address literal and SHOULD use FQDN for dual stack client or IPv6 only client.

### 3.4. DNS query flow

Figure 1 shows a DNS query flow on the .TLD.

1. An application on a client creates <ipv4-address-literal>.TLD.
2. The application inputs the query of AAAA or ANY about <ipv4-address-literal>.TLD. to its local resolver.
3. The local resolver forwards the query to a recursive resolver that would be a DNS64 server in DNS64/NAT64 environment.
4. The recursive resolver sends a recursive query of <ipv4-address-literal>.TLD.
5. .TLD authoritative server creates the A record of the issued <ipv4-address-literal>.TLD, and MAY check PTR record of the issued <ipv4-address-literal>. Then, .TLD authoritative server returns the DNS response to the recursive resolver.
6. When the recursive resolver has DNS64 function, it creates the AAAA record according to [RFC6147] and replies the AAAA record to the local resolver on the client. If the recursive resolver does not have DNS64 function, the recursive resolver returns the A record responded from .TLD authoritative server.
7. The application on the client gets the appropriate IP address (IPv4 address or Pref64::/n address), then creates an appropriate socket.



DNS Query Flow on .TLD

Figure 1

This solution would not require the modification of common shared libraries on any Operating Systems. The DNS implementations, SHOULD support .TLD. As the query flow mentioned above, .TLD authoritative server SHOULD be placed. The modification of NAT64 or DHCP are not required in this method.

### 3.5. Use cases

#### 3.5.1. Use case 1: manual type-writing

For example, consider living on an IPv6-only network with DNS64/NAT64, and receiving a message like ``please download a file foo.doc from a ftp server 192.0.2.10''. Usually, you may estimate the NAT64 prefix and calculate Pref64::/n address through [RFC7050] or [RFC7051]. Under the proposed mechanism on this memo, you can just type as follow;

```
% ftp 192.0.2.10.TLD
```

The packet would be transferred along with [RFC6384].

#### 3.5.2. Use case 2: browser plug-in

An IPv4 address literal is often used in URL for the lazy DNS operation, a temporary HTTP server or a hidden (private) server. Taking into account user convenience, a browser plug-in can be developed that it converts the <ipv4-address-literal> on the hostname

part of an URL to <ipv4-address-literal>.TLD. It may be suggested to turn this function on when the host is on IPv6-only network, however, it may not be easy to detect the situation of the network (IPv4 only, dual stack or DNS64/NAT64 environment). A sample of Google Chrome plug-in is attached in Appendix B

### 3.6. Recommendation

For usability in manual type-writing, the .TLD SHOULD be as short as possible, and SHOULD express the special purpose in the name space. ``.v4`` is recommended as a candidate of .TLD, because of the simplicity and the expression of IPv4.

## 4. Considerations

### 4.1. Attached the special-purpose TLD to a regular FQDN

Conceptually, the special-purpose TLD would be attached to only IPv4 address literals, however, the special-purpose TLD may be attached to a regular FQDN notation like ``foo.bar.com.TLD``. Such misuses SHOULD be avoided.

### 4.2. An embedded IP address literal in the content part of URL

In some case, <ipv4-address-literal> may be embedded into the content part of a URL, however, it may be difficult for users or browser plug-ins to recognize unambiguously that a string like <ipv4-address-literal> surely means some IPv4 address. From the point of view of IPv6 migration, embedded IP address literal in the content part of an URL MUST be avoided.

### 4.3. Prevention the leak of the special-purpose TLD

When .TLD is actually employed in the operation, .TLD may leak to the public DNS infrastructure including root DNS servers as seen in ``.local``. Therefore, once consensus is obtained, the relevant TLD SHOULD be delegated to a set of DNS servers.

Two possible DNS operation methods can be considered. One is to delegate the TLD to AS112 servers [as112-servers]. When one of the AS112 servers received a query with .TLD, it returns with NXDOMAIN.

The other possible DNS operation is to deploy a set of special purpose DNS servers which accept queries with .TLD and synthesize an A record corresponding to the IPv4 address in the QNAME when it is a legitimate IPv4 address. Otherwise, NXDOMAIN MUST be returned.

#### 4.4. Possibility to break connections with Apache VirtualHost concept

Changing the URL (swapping the DNS name or adding in a Pref64) frequently breaks the connections since the application is aware of the name it expects, and connecting correctly to the correct IP address is not sufficient, the name must also be the same in many cases.

For example, many websites use the Apache VirtualHost concept. When a web site that changes contents along with accessed IP address family like `http://www.kame.net/` or `http://dual.tlund.se/`, and if some client accesses such web site by `<ipv4-address-literal>.TLD` instead of FQDN, the VirtualHost may not work as intended.

Therefore, such web site, that uses the Apache VirtualHost concept, SHOULD NOT use `<ipv4-address-literal>` in URL and SHOULD use appropriate FQDN.

#### 4.5. Inaffinity with HTTP/HTTPS Cookie

This solution may not work with HTTP/HTTPS cookie. We should also consider the HTTP security considerations for the cases where someone puts one of the names into a URL. For example, consider `http://192.0.2.10.TLD/` to an origin that sets a cookie on the domain `"*.10.TLD"`.

There are likely already plenty of ways to do the same thing out there, so this may not be a major issue.

#### 4.6. TLD alternatives

In Section 3.6, we propose `.v4` as the TLD, and comparisons with other candidates are discussed as follows.

##### 4.6.1. `.v4.arpa`

```v4.arpa``` may be a candidate of `.TLD` that does not require new TLD, however, it may be confused with [RFC7050] ```ipv4only.arpa```, and the length (8 characters) of ```v4.arpa``` is bit longer than the length (3 characters) of ```v4``` for type-writing usages.

##### 4.6.2. `.host`

```.host``` has already been assigned as one of the new gTLDs, and not considered a candidate here unless the authority of `.host` offers 256 (or 356 -- see discussion in Section 4.6.3) delegations to this purpose.

#### 4.6.3. TLD less delegation

When it is feasible to "delegate" 256 TLDs (from ".0" through ".255") or 366 TLDs (".00", ".000", and others are added) for this particular purpose, it is possible to implement the functionality described in this memo without assigning a particular .TLD. It contributes 256 (or 356) extra TLDs in the Root zone.

It is known that DNS queries with such TLDs have been observed, and this delegation may interfere with undocumented usage of such TLDs.

If such 256 (or 366) delegations is suitable, bogus such queries to the root servers will be redirected to the DNS server described in Section 5.

#### 4.7. Usages of IPv6 address literal

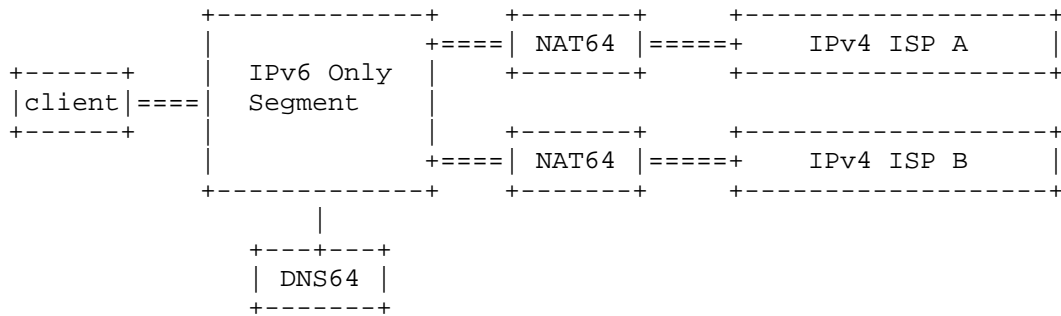
The special-purpose TLD may be applied to IPv6 address cases in same ways, however, such notation is not required in dual stack / IPv6-only environment, generally.

#### 4.8. RFC7050 ipv4only.arpa

[RFC7050] defines a method to estimate a NAT64 prefix by querying Well-Known IPv4-only Name ``ipv4only.arpa``. [RFC7050] does not cover several situations. .TLD method is aimed to solve such situations as follows:

##### 4.8.1. Multiple NAT64 prefixes for load balancing

One of situations is multihoming, illustrated in Figure 2. In this situation, the NAT64 prefix estimated by [RFC7050] method may be different from the one that the operator intends.

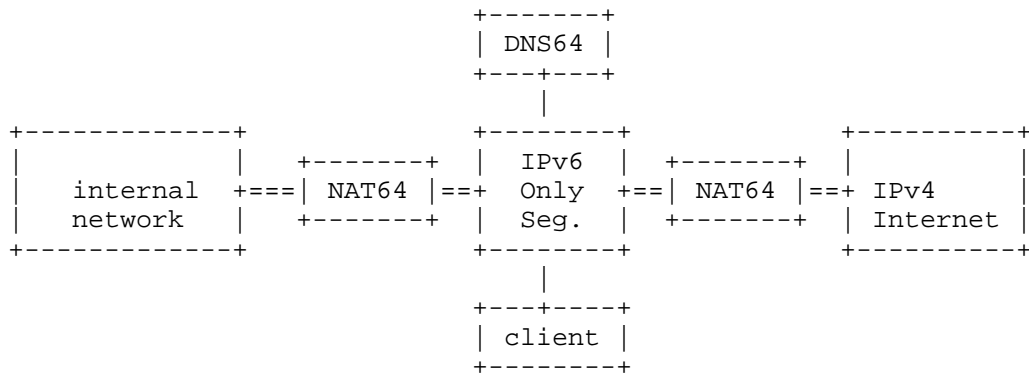


Situation A : multiple NAT64 prefixes for optimizing routes on multihoming

Figure 2

4.8.2. Multiple NAT64 prefixes for external / internal IPv4 only networks

Another situation is where multiple NAT64 prefixes are operated for accessing the external IPv4 Internet and an internal private IPv4 only network from an internal IPv6 only network. Figure 3 draws this situation. In this situation, the NAT64 prefix estimated by [RFC7050] method could not be reached to the internal IPv4 only network.



Situation B : multiple NAT64 prefixes for internal / external

Figure 3

#### 4.8.3. Difficulty of conversion from octet expression to hex expression by human type-writing

As the initial motivation of this memo, IPv4 address literal is often used for a personal / private server that is not registered in DNS record because of lazy operation, temporal usage, or the intention to hide from DNS query scans. ``ipv4only.arpa`` solution can be available to synthesize the Pref64::/n address for the private server, however, the owner of the private server has to convert the octet expression of the IPv4 address on his/her private server to the hex expression by manual. Usually, conversion from octet expression to hex expression by manual is difficult or tiresome operation.

### 5. Implementation Strategy

It is suggested to implement the .TLD rewriting as in the following order:

1. Define .TLD  
Once the community agrees to accept the rewriting scheme described in this memo, it must fix the .TLD to be used. The .TLD WOULD require the update of [RFC6761].
2. .TLD delegation  
DNS queries with .TLD can leak to the DNS of the global Internet, it is highly suggested to delegate .TLD to a set of authoritative DNS servers as discussed in Section 4.3.
3. DNS64 modification  
DNS64 implementation is suggested to modify to respond corresponding AAAA record to a query with .TLD. This process can be done in parallel to the step 2 above.
4. Start using .TLD rewriting  
After, at least the step 2 is completed, the TLD rewriting may be used in manually described in Section 3.5.1 or automatically by browser plugins described in Section 3.5.2. While further discussions and observation is required, the use of an URL in IPv4 literal embedded might be discouraged. Instead, the use of .TLD notation as a legitimate URL might be encouraged even in the server side.

### 6. Security Considerations

The recommendation contains security considerations related to DNS. The special purpose DNS servers of this memo only treats the IPv4 address literal with .TLD. Therefore, the special DNS MAY use self-signed / authorized key for DNS responses.

When a client is to access an URL with IPv4 literal address embedded,

it triggers a DNS query, and the query may be sent over the Internet to the nearest authoritative .TLD DNS server. It may break the confidentiality against the DNS service.

TBD

## 7. IANA Considerations

This memo calls for ``.v4`` as the special-purpose TLD to the IANA registry.

## 8. Acknowledgments

Authors thank to WIDE Project members for their active discussion, implementations, and evaluations. Especially, we thank to Atsushi ONOE for the revision of this solution, Hirochika ASAI for the contribution of the prototype implementation of the special purpose authoritative DNS, and Hirotaka NAKAJIMA for the contribution of the Google chrome plug-in. We also thank to Yoshiaki KITAGUCHI, Yu-ya KAWAKAMI and others who evaluated our proof of concept special purpose DNS (.v4.wide.ad.jp) and the Google Chrome plugin-in at JANOG34 DNS64/NAT64 experiment networks. Teeme Savolainen, Cameron Byrne, Dan Wing, Erik Nygren gave us various considerations on the actual operation of .TLD.

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4074] Morishita, Y. and T. Jinmei, "Common Misbehavior Against DNS Queries for IPv6 Addresses", RFC 4074, May 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.
- [RFC6384] van Beijnum, I., "An FTP Application Layer Gateway (ALG)



for IPv6-to-IPv4 Translation", RFC 6384, October 2011.

- [RFC6586] Arkko, J. and A. Keranen, "Experiences from an IPv6-Only Network", RFC 6586, April 2012.
- [RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, February 2013.
- [RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, November 2013.
- [RFC7051] Korhonen, J. and T. Savolainen, "Analysis of Solution Proposals for Hosts to Learn NAT64 Prefix", RFC 7051, November 2013.

## 9.2. Informative References

- [as112-servers] AS112 Project, "AS112 Project", October 2009, <<https://www.as112.net/>>.

## Appendix A. A Test Server of the special TLD

We run a prototype implementation of the special-purpose DNS server in the WIDE backbone (AS 2500). We use ``.v4.wide.ad.jp`` as .TLD.

## Appendix B. Sample extension for Google Chrome

We developed a sample plug-in code for Google Chrome ``IPv4 Address Literal Appender`` that automatically converts <ipv4-address-literal> in URL to <ipv4-address-literal>.TLD. The .TLD can be customized in the option. The ``IPv4 Address Literal Appender`` is freely available in Google Chrome Web Store, and also in github <https://github.com/nunnun/nat64-v4-literal-extension>.

```
var wr = chrome.webRequest;

var v4Suffix = ".TLD";
var ipAddrRegex = /^(\\d|[01]?\\d\\d|2[0-4]\\d|25[0-5])\\. (\\d|[01]?\\d\\d|2[0-4]\\d|25[0-5])\\. (\\d|[01]?\\d\\d|2[0-4]\\d|25[0-5])\\. (\\d|[01]?\\d\\d|2[0-4]\\d|25[0-5])$/;

function onBeforeRequest(details) {
  var tmpuri = new URI(details.url);
  var tmphost = tmpuri.host();
  var finalUri = '';
  tmphost.replace(ipAddrRegex, function(str, p1, p2, p3, p4, offset, s) {
    finalUri=tmpuri.host(p1+"."+p2+"."+p3+"."+p4+v4Suffix).toString();
  });
  if('' != finalUri) {
    console.log(finalUri);
    return {redirectUrl: finalUri};
  }
};

wr.onBeforeRequest.addListener(onBeforeRequest, {urls: ["https://*/**",
"http://*/**", "ftp://*/**"]}, ["blocking"]);
```

#### Authors' Addresses

Osamu Nakamura  
Keio Univ./WIDE Project  
5322 Endo  
Fujisawa, Kanagawa 252-0882  
JP

Phone: +81 466 49 1100  
Email: osamu@wide.ad.jp

Hiroaki Hazeyama  
NAIST / WIDE Project  
8916-5 Takayama  
Ikoma, Nara 630-0192  
JP

Phone: +81 743 72 5111  
Email: hiroa-ha@is.naist.jp

Yukito Ueno  
Keio Univ./WIDE Project  
5322 Endo  
Fujisawa, Kanagawa 252-0882  
JP

Phone: +81 466 49 1100  
Email: eden@sfc.wide.ad.jp

Akira Kato  
Keio Univ. / WIDE Project  
Graduate School of Media Design, 4-1-1 Hiyoshi  
Kohoku, Yokohama 223-8526  
JP

Phone: +81 45 564 2490  
Email: kato@wide.ad.jp



Sunset4 Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 30, 2015

L. Song  
Beijing Internet Institute  
P. Vixie  
Farsight Security, Inc.  
D. Ma  
ZDNS  
October 27, 2014

Considerations on IPv6-only DNS Development  
draft-song-sunset4-ipv6only-dns-00

Abstract

Deployment of IPv6-only networks are impacted by assumptions of IPv4-only or dual-stack transition scenarios. For example, these assumptions are in the operations of DNS. This memo is problem statement and hopes to eventually propose a mitigation technique.

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 30, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Revisit to current situation . . . . .	3
3.1. DNS Referral Response Size limitation . . . . .	3
3.2. Additional section in IPv4/IPv6 Environments . . . . .	4
3.3. DNS proxy . . . . .	5
4. Mitigation approach . . . . .	6
5. Security Considerations . . . . .	6
6. IANA Considerations . . . . .	6
7. Acknowledgements . . . . .	6
8. References . . . . .	6
8.1. Normative References . . . . .	6
8.2. URIs . . . . .	7
Authors' Addresses . . . . .	7

## 1. Introduction

It's commonly believed that the dual-stack model is the best practice for IPv6 transition in which IPv4 and IPv6 function can work in parallel without mutual interference. Based on this model, IP stacks and applications are expected to be converted into IPv6 smoothly when IPv4 address pool run out. The dual-stack approach gives IPv4/IPv6 capability on end system, network devices, DNS and application servers, but, as a side effect, brings additional problems, such as IPv4 fallback [RFC6555] or even IPv4/IPv6 competition. This issue makes the dual stack model more complicated to deploy and manage, and overall network less reliable.

To accelerate the transition to a fully connected IPv6 network, IPv6-only experiments [RFC6586] and IETF standards [RFC6333], [RFC7040] are documented. Some techniques verify IPv6 capability and support the IPv6-only deployment. In IPv6-only environments, DNS resolvers or modules are provisioned only with IPv6 address. It is mainly due to three aspects:

- 1) To save more free IPv4 addresses in deploying new DNS resolvers;
- 2) To reduce the cost and risk of management in dual stack environment;
- 3) To follow the inherent requirement in the IPv6 transition scenarios, such as DS-Lite [RFC6333];

It's worthwhile to mention that the tunnel technology provides an approach that allow IPv6-only network deployment become independent from the rest of the world which makes the IPv6-only strategy much popular. In the IPv6-only network, the ISPs only provision IPv6 address to the end system, network and DNS element via DHCPv6. However, IPv6-only resolver will face an Internet which are partly running in IPv4 only environment and partly in dual-stack, yet with IPv4-preferred paradigm. As a result, the DNS element in IPv6-only environment is suggested to be forwarding requests by relying on the upstream dual-stack DNS recursive server section 5.5 [1] in [RFC6333]. However, using the DNS proxy mechanism is a compromise in IPv6 transition context, which still has implicit limitations [RFC5625].

This memo revisits the behavior and implicit inertia of DNS in existing architecture which may hinder the IPv6-only DNS development.

## 2. Terminology

A: A resource record type used to specify an IPv4 address [RFC1034]

AAAA: A resource record type used to specify an IPv6 address [RFC3596]

EDNS0: Version 0 of Extension mechanisms for DNS [RFC6891]

DNSSEC: DNS Security Extensions [RFC4033]

MTU: Maximum Transmission Unit, the maximum size for a datagram to be forwarded on an interface without needing fragmentation [RFC0791], [RFC2460]

Additional Section: Section in DNS query/response carrying RRs which may be helpful in using the RRs in the other sections [RFC1034]. Note that in this memo the data in additional section is the A/AAAA information of NS server, particular for root zone.

## 3. Revisit to current situation

### 3.1. DNS Referral Response Size limitation

Due to the required minimum IP reassembly limit for IPv4, the original DNS standard [RFC1034][RFC1035] limited the UDP message size to 512 octets. It became an historical and practical hard DNS protocol limit, even after EDNS0 [RFC6891] was introduced to mitigate this issue[draft-ietf-dnsop-respsize-15]. This limit presents a problem for zones wishing to (1) add more authority servers or (2) advertise the

IPv6 addresses of newly updated dual-stack NS name servers, or (3) use DNSSEC.

In the context of this memo, the limitation may be relaxed due to the larger base MTU of IPv6 (1280 octets) which is the default for IPv6-only networks.

### 3.2. Additional section in IPv4/IPv6 Environments

Given there is hard limitation in the DNS referral response size, the implementations preferably decide to keep as much data as possible in the UDP responses no matter it is "critical" or "courtesy" Appendix B.2 in [RFC4472] . It is a typical case in priming exchange between recursive resolver and root server. When a name server resolver bootstrap, it performs the NS lookup for root zone. In the response packet from root server, the additional section is supposed to contain all the A & AAAA records of NS domain name. Ultimately, when all 13 root name servers are assigned IPv6 addresses, the priming response will increase in size to 800 bytes.

There are different strategies for root server operators to choose which RRset (A or AAAA) should be in the additional data if not all of the glue information can be included. Note that in dual-stack environment, IPv4 glue and IPv6 glue of same zone are actually competing for the room of DNS UDP packets. For example, some of DNS root servers prefer to return as many IPv4 glue records as possible. In that case only 2 out 10 IPv6 glues are included as shown below, irrespective of IPv4 or IPv6 DNS transport.

;; ADDITIONAL SECTION:

```
a.root-servers.net. 518400 IN A 198.41.0.4
b.root-servers.net. 518400 IN A 192.228.79.201
c.root-servers.net. 518400 IN A 192.33.4.12
d.root-servers.net. 518400 IN A 199.7.91.13
e.root-servers.net. 518400 IN A 192.203.230.10
f.root-servers.net. 518400 IN A 192.5.5.241
g.root-servers.net. 518400 IN A 192.112.36.4
h.root-servers.net. 518400 IN A 128.63.2.53
i.root-servers.net. 518400 IN A 192.36.148.17
```



```
j.root-servers.net. 518400 IN A 192.58.128.30
k.root-servers.net. 518400 IN A 193.0.14.129
l.root-servers.net. 518400 IN A 199.7.83.42
m.root-servers.net. 518400 IN A 202.12.27.33
a.root-servers.net. 518400 IN AAAA 2001:503:ba3e::2:30
b.root-servers.net. 518400 IN AAAA 2001:500:84::b
```

In the context of IPv6-only deployments, these glue records are much less optimal. They are based on IPv4 or dual-stack assumptions, where IPv4 is still dominant. It may negatively impact the IPv6 services in IPv6-only deployments.

If the glue set sent in the response is correlated with the IP version of the DNS transport, then the answer, in most cases, will be more optimal. There are two reasons why it is not adopted as an optimization. One is that it breaks the model of independence of DNS transport and resource records section 1.2 [2] in [RFC4472]. Another is that it will bring unpredictable risk to the performance and stability of current root server system.

### 3.3. DNS proxy

In IPv6-only networking, DNS proxy approach is recommended for IPv6-only DNS element. On one hand, it avoids the difficulty to perform all DNS resolution over IPv6 transport, given that still many networks on Internet are only on IPv4. On another hand, it loses the opportunity to perform a full recursive resolver function via IPv6, at least in Root and TLD level which are mostly IPv6 enabled.

In additional, as described in the beginning of [RFC5625], the DNS proxy function is not an optimal solution to serve the IPv6-only resolver requirement. Large packets caused by priming request or DNSSEC validation packets will be blocked due to the proxy implementation. It is suggested that: "To ensure full DNS protocol interoperability it is preferred that client stub resolvers should communicate directly with full-feature, upstream recursive resolvers wherever possible."

As more and more NS servers updated to IPv6 transport and reachable over the IPv6 Internet, the direct IPv6 resolution will be preferable in IPv6-only resolver. But regarding the long-tail feature of IPv6 adoption in NS servers, certain back-forward compatible mechanism

should be designed, which indeed make an incentive model for IPv6 adoption over IPv4 as well.

#### 4. Mitigation approach

TBD

#### 5. Security Considerations

TBD

#### 6. IANA Considerations

TBD

#### 7. Acknowledgements

TBD

#### 8. References

##### 8.1. Normative References

[I-D.ietf-dnsop-respsize]

Vixie, P., Kato, A., and J. Abley, "DNS Referral Response Size Issues", draft-ietf-dnsop-respsize-15 (work in progress), February 2014.

[I-D.lee-dnsop-scalingroot]

Lee, X., Vixie, P., and Z. Yan, "How to scale the DNS root system?", draft-lee-dnsop-scalingroot-00 (work in progress), July 2014.

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

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

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

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.
- [RFC4472] Durand, A., Ihren, J., and P. Savola, "Operational Considerations and Issues with IPv6 DNS", RFC 4472, April 2006.
- [RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, August 2009.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.
- [RFC6586] Arkko, J. and A. Keranen, "Experiences from an IPv6-Only Network", RFC 6586, April 2012.
- [RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, April 2013.
- [RFC7040] Cui, Y., Wu, J., Wu, P., Vautrin, O., and Y. Lee, "Public IPv4-over-IPv6 Access Network", RFC 7040, November 2013.

## 8.2. URIs

- [1] <http://tools.ietf.org/html/rfc6333#section-5.5>
- [2] <http://tools.ietf.org/html/rfc4472#section-1.2>

## Authors' Addresses

Linjian Song  
Beijing Internet Institute  
2508 Room, 25th Floor, Tower A, Time Fortune  
Beijing 100028  
P. R. China

Email: [songlinjian@gmail.com](mailto:songlinjian@gmail.com)

Paul Vixie  
Farsight Security, Inc.  
155 Bovet Road, #476  
San Mateo, CA 94402  
USA

Phone: +1 650 489 7919  
Email: vixie@farsightsecurity.com

Di Ma  
ZDNS  
Beijing  
P. R. China

Email: madi@zdns.cn

DHC Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: March 19, 2015

W. Wang  
L. Zhang  
X. Que  
BUPT University  
L. Li  
Tsinghua University  
Y. Wang  
BUPT University  
September 15, 2014

Discovery of the IPv6 Prefix in 464XLAT  
draft-wang-v6ops-xlat-prefix-discovery-00

Abstract

The 464XLAT[RFC6877] provides a solution with limited IPv4 connectivity across an IPv6-only network. In the architecture, the CLAT must discover the PLAT-side translation IPv6 prefix. This document defines a mechanism for CLAT to learn the IPv6 prefix used for protocol translation on an access network.

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 March 19, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Requirements Language . . . . .	2
3. Solution Overview . . . . .	3
4. Client-Server Interaction . . . . .	3
5. DHCPv6 Options . . . . .	4
5.1. PLAT IPv6 PREFIX Option . . . . .	4
6. Security Considerations . . . . .	5
7. IANA Considerations . . . . .	5
8. References . . . . .	5
8.1. Normative References . . . . .	5
8.2. Informative References . . . . .	5
Authors' Addresses . . . . .	5

## 1. Introduction

464XLAT describes an IPv4-over-IPv6 solution as one of the techniques for IPv4 service extension and encouragement of IPv6 deployment. The 464XLAT architecture uses IPv4/IPv6 translation standardized in [RFC6145] and [RFC6146]. It encourages the IPv6 transition by making IPv4 service reachable across IPv6-only networks and providing IPv6 and IPv4 connectivity to single-stack IPv4 or IPv6 servers and peers.

Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis [RFC7050] describes a method for detecting the presence of DNS64 and for learning the IPv6 prefix used for protocol translation on an access network. But it is difficult and depends on DNS64.

This document defines a mechanism for CLAT to learn the IPv6 prefix used for protocol translation on an access network. One new DHCPv6 option is defined to inform the CLAT of the IPv6 prefix used for IPv6 address synthesis.

## 2. 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].

### 3. Solution Overview

In the 464XLAT architecture, the CLAT must discover the PLAT-side translation IPv6 prefix used as a destination of the PLAT. The CLAT will use this prefix as the destination of all translation packets that require stateful translation to the IPv4 Internet.

The CLAT implements `OPTION_V6_PLATPREFIX`, which is a DHCPv6 option containing the IPv6 prefix used as a destination of the PLAT. The client includes this option within the ORO option in its DHCPv6 request, indicates its support for the IPv6 prefix to the DHCP server.

`OPTION_V6_PLATPREFIX` is also implemented by the server to identify the client which support IPv6 prefix. With this option, the server informs the client of the IPv6 prefix used as a destination of the PLAT.

### 4. Client-Server Interaction

The following diagram shows the client/server message flow and how the DHCPv6 option `OPTION_V6_PLATPREFIX` is used. In each step, the relevant DHCPv6 message is given above the arrow and the `OPTION_V6_PLATPREFIX` below the arrow.

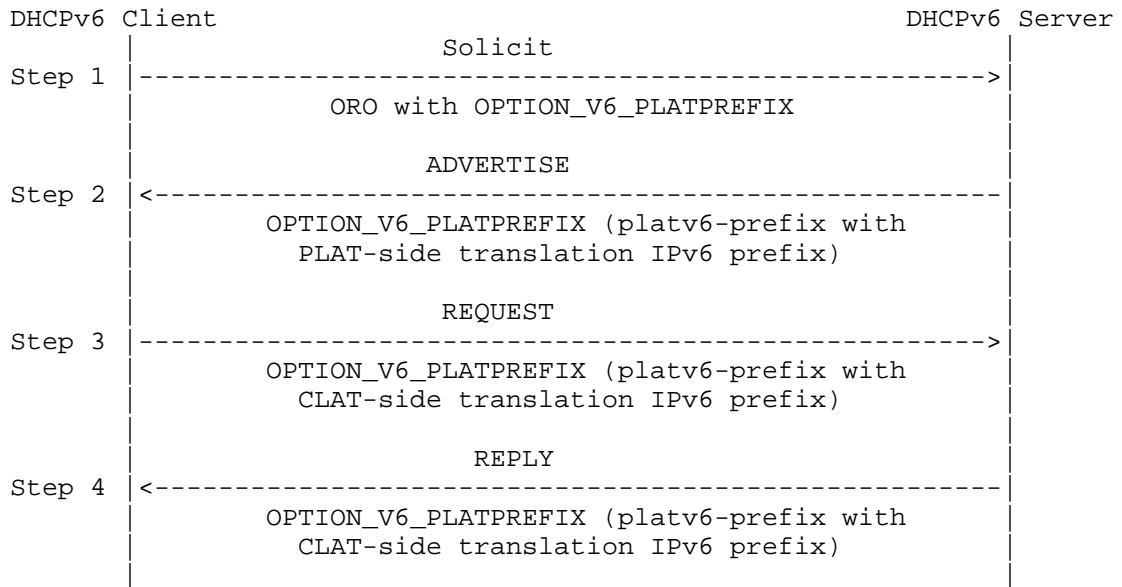


Figure 1: Server/Client Interaction Procedure

The DHCPv6 Server and Client MAY implement the `OPTION_V6_PLATPREFIX`. A Client that intends to dynamically discover the PLAT-side translation IPv6 prefix SHOULD include the code of `OPTION_V6_PLATPREFIX` in the ORO when it sends a Solicit message.

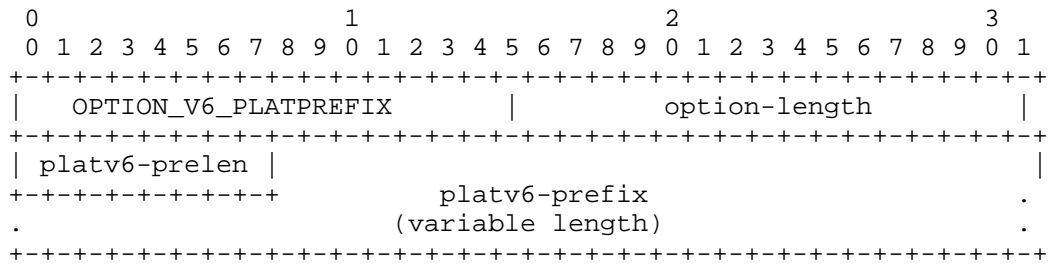
When a DHCPv6 server replies with a `ADVERTISE` message, it SHOULD include the `platv6-prefix` with PLAT-side transition IPv6 prefix. The `OPTION_V6_PLATPREFIX` is used by the server to inform the client of the PLAT-side transition IPv6 prefix.

When the client sends a `REQUEST` message, it SHOULD include the `platv6-prefix` with CLAT-side translation IPv6 prefix. The `OPTION_V6_PLATPREFIX` is used by the client to inform the server of the transition IPv6 prefix.

## 5. DHCPv6 Options

### 5.1. PLAT IPv6 PREFIX Option





- o option-code: OPTION\_V6\_PLATPREFIX (TBA1)
- o option-length: 1 + length of platv6-prefix, specified in bytes.
- o platv6-prelen: 8-bit field expressing the bit mask length of the IPv6 prefix specified in platv6-prefix.
- o platv6-prefix: The IPv6 prefix that the server uses to inform the client of the IPv6 prefix used for IPv6 address synthesis.

6. Security Considerations

TBA

7. IANA Considerations

This document defines one new DHCPv6 option, the OPTION\_V6\_PLATPREFIX option in Section 4.

8. References

8.1. Normative References

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

8.2. Informative References

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

Authors' Addresses

Wendong Wang  
BUPT University  
Beijing University of Posts and Telecommunications (BUPT)  
Beijing 100876  
P.R.China

Phone: +86-10-6228-1175  
Email: wdwang@bupt.edu.cn

Lanshan Zhang  
BUPT University  
Beijing University of Posts and Telecommunications (BUPT)  
Beijing 100876  
P.R.China

Phone: +86-13146885878  
Email: zls326@sina.com

Xirong Que  
BUPT University  
Beijing University of Posts and Telecommunications (BUPT)  
Beijing 100876  
P.R.China

Phone: +86-10-6228-3411  
Email: rongqx@bupt.edu.cn

Lishan Li  
Tsinghua University  
Beijing 100084  
P.R.China

Phone: +86-15201441862  
Email: lilishan9248@126.com

Yuqi Wang  
BUPT University  
Beijing University of Posts and Telecommunications (BUPT)  
Beijing 100876  
P.R.China

Email: wyqbupt@163.com