v6ops
Internet-Draft
Intended status: Informational
Expires: April 13, 2019

C. Byrne
T-Mobile USA
J. Palet Martinez
The IPv6 Company

October 10, 2018

IPv6-Ready DNS/DNSSSEC Infrastructure
draft-bp-v6ops-ipv6-ready-dns-dnssec-00

Abstract

This document defines the timing for implementing a worldwide IPv6-Ready DNS and DNSSEC infrastructure, in order to facilitate the global IPv6-only deployment.

A key issue for this, is the need for a global support of DNSSEC and DNS64, which in some scenarios do not work well together. This document states that any DNSSEC signed resources records should include a native IPv6 resource record as the most complete and expedient path to solve any deployment conflict with DNS64 and DNSSEC

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 https://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 13, 2019.

Table of Contents

1. Introduction

One of the main issues to ensure the best path for the IPv4 to IPv6 transition and the support of an IPv6-only Internet, is to ensure that all the services remain accessible by means of DNS.

One of the alternatives is the use of NAT64 ([RFC6146]) and DNS64 ([RFC6147]), sometimes by means 464XLAT ([RFC6877]), which will help to ensure that, when a network or part of it, becomes IPv6-only, still can have access to IPv4-only resources.

DNS64 ([RFC6147]) is a widely deployed technology allowing hundreds of millions of IPv6-only hosts/networks to reach IPv4-only resources. DNSSEC is a technology used to validate the authenticity of information in the DNS, however, as DNS64 ([RFC6147]) modifies DNS answers and DNSSEC is designed to detect such modifications, DNS64 ([RFC6147]) can break DNSSEC in some circumstances.

Furthermore, the deployment of those transition mechanisms means that the cost of the transition is on the back of the service provider, because the investment required in the devices that take care of that transition services and the support of the helpdesks to resolve issues. So in the end, all that cost is indirectly charged to the end-user, which is unfair.

It seems obvious that should not be that way, and the end-goal is a situation where we get rid-off IPv4-only services, and meanwhile, the

cost borne by the IPv4 laggards operating those services.

This document provides the steps to be able to tackle that situation and advance with the global IPv6 deployment in a fair way.

The document also states that the most complete and expedient path to avoid any negative interactions is, for the DNSSEC signed resources, to always include IPv6 AAAA resources records. As stated in [RFC6540], IPv6 [RFC8200] is not optional and failing to support IPv6 may result in failure to communicate on the Internet, especially when DNSSEC signed IPv4-only resources are present.

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. The Conflict Between DNS64 and DNSSEC

DNS64 ([RFC6147]) is a key part of widely deployed IPv6-only transition mechanism such as 464XLAT ([RFC6877]) and Happy Eyeballs version 2 ([RFC8305]). Currently, hundreds of millions of hosts rely on DNS64 ([RFC6147]) for access to the Internet. A core function of DNS64 ([RFC6147]) is generating an inauthentic AAAA DNS record when an authentic AAAA DNS record for a host is not available from the authoritative nameserver. DNSSEC’s fundamental feature is detecting and denying inauthentic DNS resource records. While DNS64 ([RFC6147]) outlines may work in harmony with DNSSEC, the preconditions may not always exist for harmony to be achieved.

4. Resolving the DNS64 and DNSSEC Conflict by Requiring AAAA

DNS64 ([RFC6147]) and DNSSEC are both important components of the current and future Internet. The limitation for how these protocols interact is unlikely to changes. Deploying DNSSEC and IPv6 are both commonly achievable for a typical Internet system operator using their own systems or using a third-party service. The resolution to the DNS64 ([RFC6147]) and DNSSEC conflict is to simply deploy both, IPv6 and DNSSEC in tandem.

Deploying DNSSEC signed IPv4 resources records without matching IPv6 records is a risk and not recommend.

Ultimately, this guidance is simply restating [RFC6540], that IPv6 is mandatory for all Internet systems.

5. Ensuring a smooth IPv4-IPv6 transition by Requiring AAAA

Similarly, to what is stated in the precedent section for DNS64 ([RFC6147]) and DNSSEC, a smoother and less painful transition from IPv4 to IPv6, and the succesful deployment of an IPv6-only Internet, can be facilitated by requiring AAAA resource records at every DNS instance.

6. Definition of IPv6-Ready DNS/DNSSEC Infrastructure

In the context of this document, and others that may be generated as a consequence of it, "IPv6-Ready DNS/DNSSEC Infrastructure" means that a DNS/DNSSEC server (root, TLD, authoritative NS, others) is fully accessible and operational if queried either from a remote dual-stack network or an IPv6-only network.

In general, that means having AAAA RRs in addition to A RRs, ensuring that PMTUD works correctly and fragmentation is correctly handled.

In case DNSSEC is implemented with IPv4, it MUST support also IPv6-only operation according the above considerations.

7. Implementation timing

Towards the implementation of the worldwide IPv6-Ready DNS/DNSSEC infrastructure, considering that there are no excuses for a DNS operator to support IPv6, the following deadlines are defined counting since the date this document becomes an RFC:

1. All the root and TLDs MUST be IPv6-Ready in 6 months.

2. All the DNSSEC signed zones MUST be IPv6-Ready in 6 months.

3. All the authoritative NS MUST be IPv6-Ready in 12 months.

4. The remaining RRs in other DNS servers, MUST be IPv6-Ready in 18 months.

Probing mechanisms to verify that the relevant AAAA are fully operational MUST be setup by IANA. If there is a failure at the deadline in complying with those requirements, the relevant NS, MUST be temporarily suspended until there is a subsequent successful verification.

8. Security Considerations

DNSSEC is a good security practice. Providing AAAA DNSSEC signed records wherever a DNSSEC signed A record is used ensures the most effective use of DNSSEC.

9. IANA Considerations

IANA and ICANN are instructed by means of this document, to take the relevant measures for ensuring the steps towards the above indicated implementation timing.

It is suggested that frequent warnings are provided to the relevant stakeholders, in advance to each of the deadlines.

10. Acknowledgements

The author would like to acknowledge the inputs of ... TBD.

11. Normative References

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

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

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <https://www.rfc-editor.org/info/rfc6147>.

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

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <https://www.rfc-editor.org/info/rfc6877>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

Authors' Addresses

Cameron Byrne
T-Mobile USA
Bellevue, WA
United States of America

Email: Cameron.Byrne@T-Mobile.com

Jordi Palet Martinez
The IPv6 Company
Molino de la Navata, 75
La Navata - Galapagar, Madrid 28420
Spain

Email: jordi.palet@theipv6company.com
URI: http://www.theipv6company.com/

v6ops
Internet-Draft
Intended status: Informational
Expires: January 12, 2020

J. Palet Martinez
The IPv6 Company

July 11, 2019

Additional NAT64/464XLAT Deployment Guidelines in Operator and Enterprise Networks
draft-ietf-v6ops-nat64-deployment-08

Abstract

This document describes how NAT64 (including 464XLAT) can be deployed in an IPv6 network, whether cellular ISP, broadband ISP, or enterprise, and possible optimizations. The document also discusses issues to be considered when having IPv6-only connectivity, regarding: a) DNS64, b) applications or devices that use literal IPv4 addresses or non-IPv6 compliant APIs, and c) IPv4-only hosts or applications.

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 https://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 January 12, 2020.

Table of Contents

1. Introduction

Stateful NAT64 ([RFC6146]) describes a stateful IPv6 to IPv4 translation mechanism, which allows IPv6-only hosts to communicate with IPv4-only servers using unicast UDP, TCP, or ICMP, by means of IPv4 public addresses sharing, among multiple IPv6-only hosts. Unless otherwise stated, references in the rest of this document to NAT64 (function) should be interpreted as to Stateful NAT64.

The translation of the packet headers is done using the IP/ICMP translation algorithm defined in [RFC7915] and algorithmically translating the IPv4 addresses to IPv6 addresses and vice versa, following [RFC6052].

DNS64 ([RFC6147]) is in charge of the synthesis of AAAA records from the A records, so only works for applications making use of DNS. It was designed to avoid changes in both, the IPv6-only hosts and the IPv4-only server, so they can use a NAT64 function. As discussed in Section 5.5 of [RFC6147], a security-aware and validating host has to perform the DNS64 function locally.

However, the use of NAT64 and/or DNS64 present three drawbacks:

a. Because DNS64 ([RFC6147]) modifies DNS answers, and DNSSEC is designed to detect such modifications, DNS64 ([RFC6147]) may potentially break DNSSEC, depending on a number of factors, such as the location of the DNS64 function (at a DNS server or validator, at the end host, ...), how it has been configured, if the end-hosts is validating, etc.

b. Because the need of using DNS64 ([RFC6147]) or an alternative "host/application built-in" mechanism for address synthesis, there may be an issue for NAT64 ([RFC6146]), as it doesn't work when IPv4 literal addresses or non-IPv6 compliant APIs are being used.

c. NAT64 alone, was not designed to provide a solution for IPv4-only hosts or applications located within a network which are

connected to a service provider IPv6-only access, as it was designed for a very specific scenario ([RFC6144], Section 2.1).

Above drawbacks may be true if part of, an enterprise network, is connected to other parts of the same network or third-party networks by means of IPv6-only connectivity. This is just an example which may apply to many other similar cases. All them are deployment specific.

According to that, across this document, the use of "operator", "operator network", "service provider", and similar ones, are interchangeable with equivalent cases of enterprise networks (and similar ones). This may be also the case for "managed end-user networks".

Note that if all the hosts in a network were performing the address synthesis, as described in Section 7.2 of [RFC6147], some of the drawbacks may vanish. However, it is unrealistic today to expect that, considering the high number of devices and applications that aren't yet IPv6-enabled. So, in this document, this will be considered only for specific scenarios that can guarantee it.

An analysis of stateful IPv4/IPv6 mechanisms is provided in [RFC6889].

This document looks into different possible NAT64 ([RFC6146]) deployment scenarios, including IPv4-IPv6-IPv4 (464 for short) and similar ones, which were not documented in [RFC6144], such as 464XLAT ([RFC6877]), in operator (broadband and cellular) and enterprise networks, and provides guidelines to avoid operational issues.

Towards that, this document first looks into the possible NAT64 deployment scenarios (split in "known to work" and "known to work under special conditions"), providing a quick and generic comparison table among them. Then the document describes the issues that an operator need to understand on different matters that will allow to define what is the best approach/scenario for each specific network case. A summary provides some recommendations and decision points. A section with clarifications on the usage of this document for enterprise networks, is also provided. Finally, an annex provides an example of a broadband deployment using 464XLAT and another annex provides hints for a CLAT implementation.

[RFC7269] already provides information about NAT64 deployment options and experiences. Both, this document and [RFC7269] are complementary; they are looking into different deployment considerations and furthermore, this document is considering the updated deployment experience and newer standards.

The target deployment scenarios in this document may be covered as well by other IPv4-as-a-Service (IPv4aaS) transition mechanisms. Note that this is true only for the case of broadband networks, as in the case of cellular networks the only supported solution is the use of NAT64/464XLAT. So, it is out of scope of this document to provide a comparison among the different IPv4aaS transition mechanisms, which is being analyzed already in [I-D.lmhp-v6ops-transition-comparison].

Consequently, this document should not be understood as a guide for an operator or enterprise to decide which IPv4aaS is the best one for its own network. Instead it should be used as a tool for understanding all the implications, including relevant documents (or even specific parts of them), for the deployment of NAT64/464XLAT and facilitate the decision process regarding specific deployment details.

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. NAT64 Deployment Scenarios

Section 7 of DNS64 ([RFC6147]), provides three scenarios, depending on the location of the DNS64 function. However, since the publication of that document, other deployment scenarios and NAT64 use cases need to be considered in actual networks, despite some of them were specifically ruled out by the original NAT64/DNS64 work.

Consequently, the perspective in this document is to broaden those scenarios, including a few new ones. However, in order to be able to reduce the number of possible cases, we work under the assumption that typically, the service provider wants to make sure that all the customers have a service without failures. This means considering the following assumptions for the worst possible case:

a. There are hosts that will be validating DNSSEC.

b. IPv4 literal addresses and non-IPv6 compliant APIs are being used.

c. There are IPv4-only hosts or applications beyond the IPv6-only link (e.g., tethering in cellular networks).

The document uses a common set of possible "participant entities":

1. An IPv6-only access network (IPv6).

2. An IPv4-only remote network/server/service (IPv4).

3. A NAT64 function (NAT64) in the service provider.

4. A DNS64 function (DNS64) in the service provider.

5. An external service provider offering the NAT64 function and/or the DNS64 function (extNAT64/extDNS64).

6. 464XLAT customer side translator (CLAT).

Note that the nomenclature used in parenthesis is the one that, for short, will be used in the figures. Note also that for simplicity, the boxes in the figures don't mean they are actually a single device; they just represent one or more functions as located in that part of the network (i.e. a single box with NAT64 and DNS64 functions can actually be several devices, not just one).

The possible scenarios are split in two general categories:

1. Known to work.

2. Known to work under special conditions.

3.1. Known to Work

The scenarios in this category are known to work, as there are well-known existing deployments from different operators using them. Each one may have different pros and cons, and in some cases the trade-offs, maybe acceptable for some operators.

3.1.1. Service Provider NAT64 with DNS64

In this scenario (Figure 1), the service provider offers both, the NAT64 and the DNS64 functions.

This is the most common scenario as originally considered by the designers of NAT64 ([RFC6146]) and DNS64 ([RFC6147]), however also may have the implications related the DNSSEC.

This scenario also may fail to solve the issue of IPv4 literal addresses or non-IPv6 compliant APIs, as well as the issue of IPv4-only hosts or applications behind the IPv6-only access network.

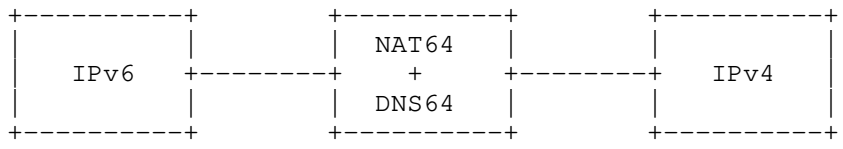

Figure 1: NAT64 with DNS64

A similar scenario (Figure 2) will be if the service provider offers only the DNS64 function, and the NAT64 function is provided by an outsourcing agreement with an external provider. All the considerations in the previous paragraphs of this section, are the same for this sub-case.

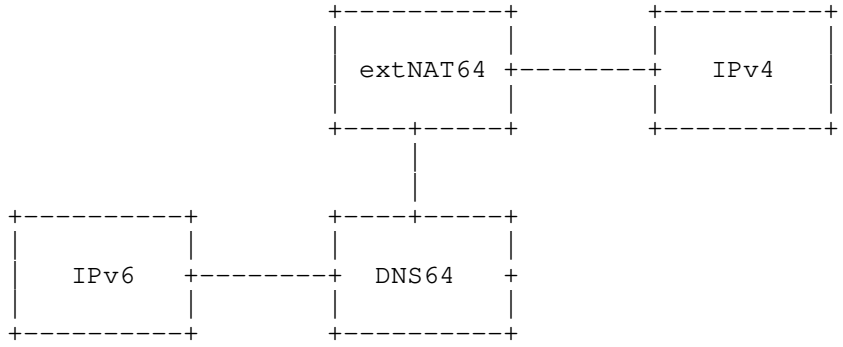

Figure 2: NAT64 in external service provider

This is equivalent to the scenario (Figure 3) where the outsourcing agreement with the external provider is to provide both the NAT64 and DNS64 functions. Once more, all the considerations in the previous paragraphs of this section are the same for this sub-case.

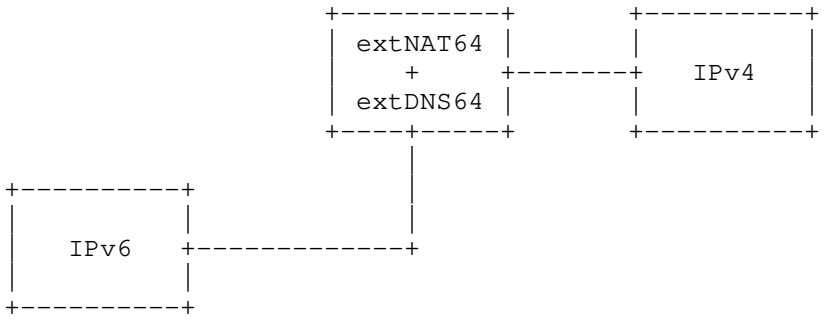

Figure 3: NAT64 and DNS64 in external provider

One additional equivalent scenario (Figure 4) will be if the service provider offers the NAT64 function only, and the DNS64 function is from an external provider with or without a specific agreement among them. This is a scenario already common today, as several "global" service providers provide free DNS/DNS64 services and users often configure manually their DNS. This will only work if both the NAT64 and the DNS64 functions are using the WKP (Well-Known Prefix) or the same NSP (Network-Specific Prefix). All the considerations in the previous paragraphs of this section, are the same for this sub-case.

Of course, if the external DNS64 function is agreed with the service provider, then we are in the same case as in the previous ones already depicted in this scenario.

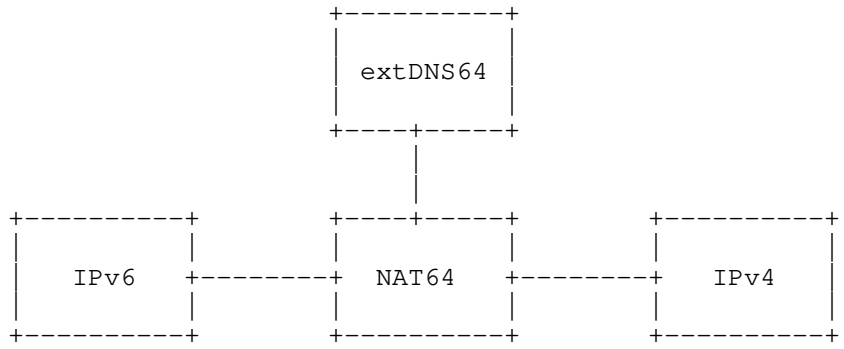

Figure 4: NAT64; DNS64 by external provider

3.1.2. Service Provider Offering 464XLAT, with DNS64

464XLAT ([RFC6877]) describes an architecture that provides IPv4 connectivity across a network, or part of it, when it is only natively transporting IPv6. [RFC7849] already suggest the need to support the CLAT function in order to ensure the IPv4 service continuity in IPv6-only cellular deployments.

In order to do that, 464XLAT ([RFC6877]) relies on the combination of existing protocols:

1. The customer-side translator (CLAT) is a stateless IPv4 to IPv6 translator (NAT46) ([RFC7915]) implemented in the end-user device or CE (Customer Edge Router), located at the "customer edge" of the network.

2. The provider-side translator (PLAT) is a stateful NAT64 ([RFC6146]), implemented typically at in the operator network.

3. Optionally, DNS64 ([RFC6147]), may allow an optimization: a single translation at the NAT64, instead of two translations (NAT46+NAT64), when the application at the end-user device supports IPv6 DNS (uses AAAA Resource Records).

Note that even if in the 464XLAT ([RFC6877]) terminology, the provider-side translator is referred as PLAT, for simplicity and uniformity, across this document is always referred as NAT64 (function).

In this scenario (Figure 5) the service provider deploys 464XLAT with a DNS64 function.

As a consequence, the DNSSEC issues remain, unless the host is doing the address synthesis.

464XLAT ([RFC6877]) is a very simple approach to cope with the major NAT64+DNS64 drawback: Not working with applications or devices that use literal IPv4 addresses or non-IPv6 compliant APIs.

464XLAT ([RFC6877]) has been used initially mainly in IPv6-only cellular networks. By supporting a CLAT function, the end-user device applications can access IPv4-only end-networks/applications, despite those applications or devices use literal IPv4 addresses or non-IPv6 compliant APIs.

In addition to that, in the same example of the cellular network above, if the User Equipment (UE) provides tethering, other devices behind it will be presented with a traditional NAT44, in addition to the native IPv6 support, so clearly it allows IPv4-only hosts behind the IPv6-only access network.

Furthermore, as discussed in [RFC6877], 464XLAT can be used in broadband IPv6 network architectures, by implementing the CLAT function at the CE.

The support of this scenario in a network, offers two additional advantages:

- DNS load optimization: A CLAT should implement a DNS proxy (as per [RFC5625]), so that only IPv6 native queries and only for AAAA records are sent to the DNS64 server. Otherwise doubling the number of queries may impact the DNS infrastructure.

- Connection establishment delay optimization: If the UE/CE implementation is detecting the presence of a DNS64 function, it may issue only the AAAA query, instead of both the AAAA and A queries.

In order to understand all the communication possibilities, let's assume the following representation of two dual-stack peers:

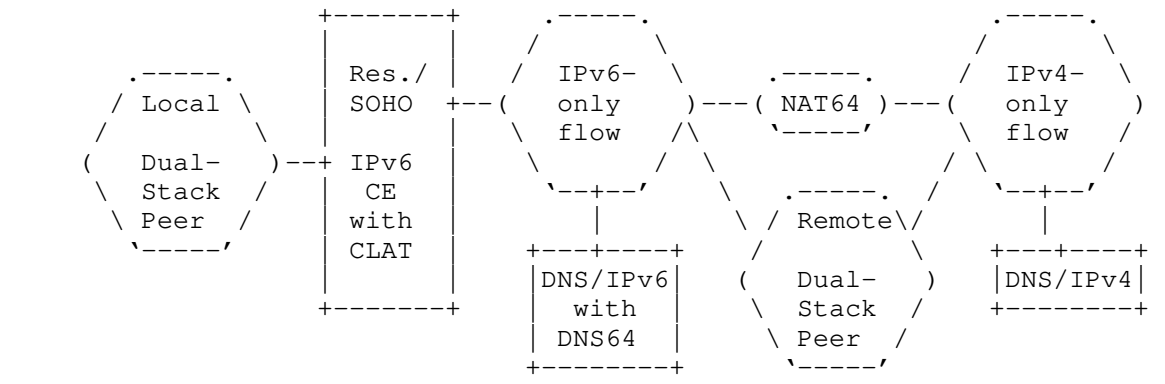

Figure A: Representation of 464XLAT among two peers with DNS64

The possible communication paths, among the IPv4/IPv6 stacks of both peers, in this case, are:

a. Local-IPv6 to Remote-IPv6: Regular DNS and native IPv6 among peers.

b. Local-IPv6 to Remote-IPv4: DNS64 and NAT64 translation.

c. Local-IPv4 to Remote-IPv6: Not possible unless the CLAT implements EAM (Explicit Address Mappings) as indicated by Section 4.9. In principle, it is not expected that services are deployed in Internet using IPv6-only, unless there is certainty that peers will also be IPv6-capable.

d. Local-IPv4 to Remote-IPv4: DNS64, CLAT and NAT64 translations.

e. Local-IPv4 to Remote-dual-stack using EAM optimization: If the CLAT implements EAM as indicated by Section 4.9, instead of using the path d. above, NAT64 translation is avoided and the flow will use IPv6 from the CLAT to the destination.

The rest of the figures in this section show different choices for placing the different elements.

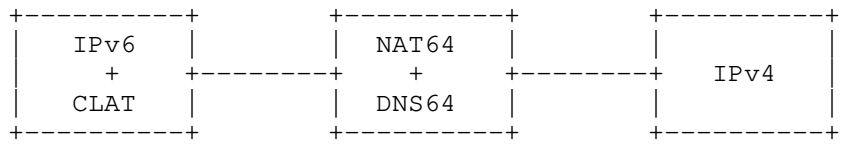

Figure 5: 464XLAT with DNS64

A similar scenario (Figure 6) will be if the service provider offers only the DNS64 function, and the NAT64 function is provided by an outsourcing agreement with an external provider. All the considerations in the previous paragraphs of this section are the same for this sub-case.

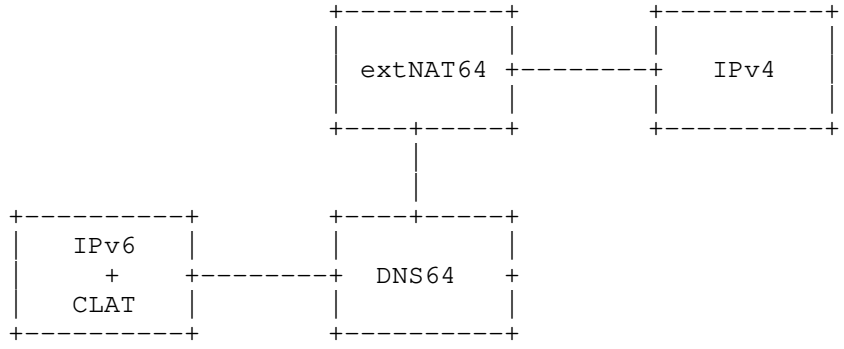

Figure 6: 464XLAT with DNS64; NAT64 in external provider

As well, is equivalent to the scenario (Figure 7) where the outsourcing agreement with the external provider is to provide both the NAT64 and DNS64 functions. Once more, all the considerations in the previous paragraphs of this section are the same for this sub-case.

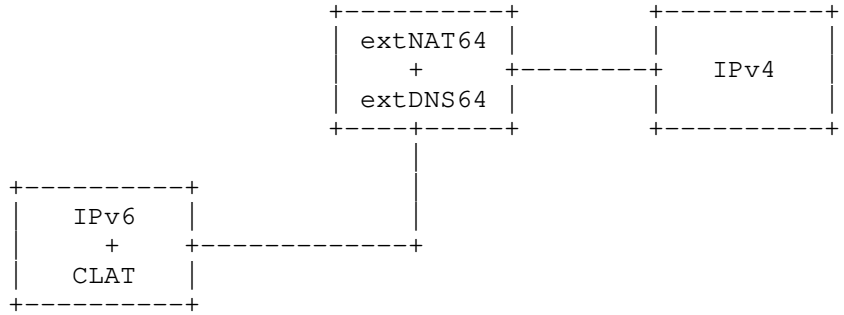

Figure 7: 464XLAT with DNS64; NAT64 and DNS64 in external provider

3.1.3. Service Provider Offering 464XLAT, without DNS64

The major advantage of this scenario (Figure 8), using 464XLAT without DNS64, is that the service provider ensures that DNSSEC is never broken, even in case the user modifies the DNS configuration. Nevertheless, some CLAT implementations or applications may impose an extra delay, which is induced by the dual A/AAAA queries (and wait for both responses), unless Happy Eyeballs v2 ([RFC8305]) is also present.

A possible variation of this scenario is the case when DNS64 is used only for the discovery of the NAT64 prefix. The rest of the document is not considering it as a different scenario, because once the prefix has been discovered, the DNS64 function is not used, so it behaves as if the DNS64 synthesis function is not present.

In this scenario, as in the previous one, there are no issues related to IPv4-only hosts (or IPv4-only applications) behind the IPv6-only access network, neither related to the usage of IPv4 literals or non-IPv6 compliant APIs.

The support of this scenario in a network, offers one advantage:

- DNS load optimization: A CLAT should implement a DNS proxy (as per [RFC5625]), so that only IPv6 native queries are sent to the DNS64 server. Otherwise doubling the number of queries may impact the DNS infrastructure.

As indicated earlier, the connection establishment delay optimization is achieved only in the case of devices, Operating Systems, or applications that use Happy Eyeballs v2 ([RFC8305]), which is very common.

Let’s assume the representation of two dual-stack peers as in the previous case:

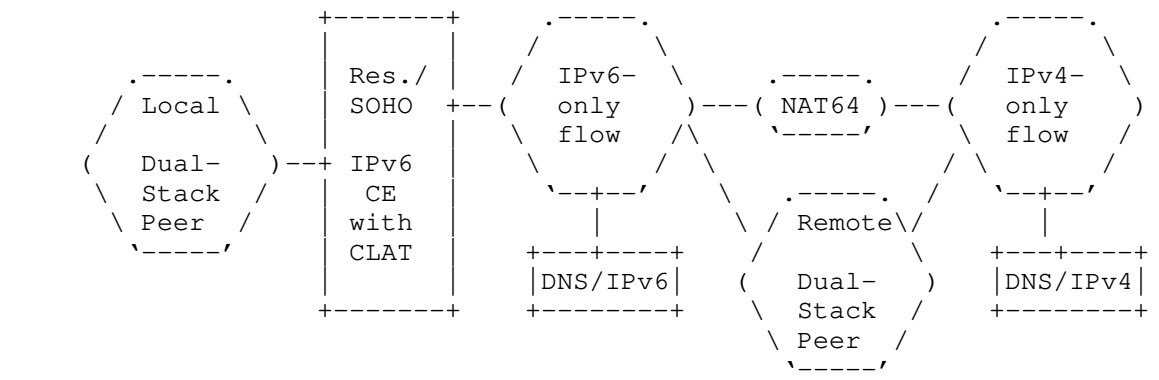

Figure B: Representation of 464XLAT among two peers without DNS64

The possible communication paths, among the IPv4/IPv6 stacks of both peers, in this case, are:

a. Local-IPv6 to Remote-IPv6: Regular DNS and native IPv6 among peers.

b. Local-IPv6 to Remote-IPv4: Regular DNS, CLAT and NAT64 translations.

c. Local-IPv4 to Remote-IPv6: Not possible unless the CLAT implements EAM as indicated by Section 4.9. In principle, it is not expected that services are deployed in Internet using IPv6-only, unless there is certainty that peers will also be IPv6-capable.

d. Local-IPv4 to Remote-IPv4: Regular DNS, CLAT and NAT64 translations.

e. Local-IPv4 to Remote-dual-stack using EAM optimization: If the CLAT implements EAM as indicated by Section 4.9, instead of using the path d. above, NAT64 translation is avoided and the flow will use IPv6 from the CLAT to the destination.

It needs to be noticed that this scenario works while the local hosts/applications are dual-stack (which is the current situation), because the connectivity from a local-IPv6 to a remote-IPv4 is not possible without an AAAA synthesis. This aspect is important only when in the LANs behind the CLAT there are IPv6-only hosts and they need to communicate with remote IPv4-only hosts. However, it doesn't look a sensible approach from an Operating System or application vendor perspective, to provide IPv6-only support unless, similarly to case c above, there is certainty of peers supporting IPv6 as well. A

solution approach to this is also presented in [I-D.palet-v6ops-464xlat-opt-cdn-caches].

The following figures show different choices for placing the different elements.

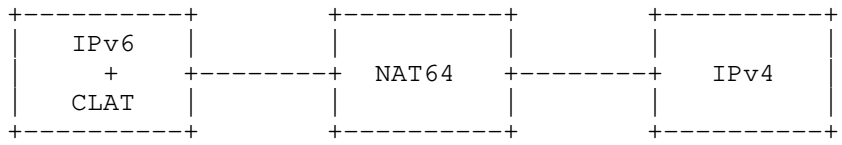

Figure 8: 464XLAT without DNS64

This is equivalent to the scenario (Figure 9) where there is an outsourcing agreement with an external provider for the NAT64 function. All the considerations in the previous paragraphs of this section are the same for this sub-case.

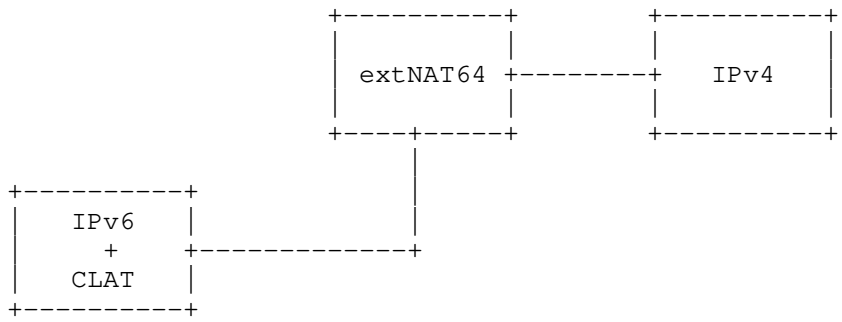

Figure 9: 464XLAT without DNS64; NAT64 in external provider

3.2. Known to Work Under Special Conditions

The scenarios in this category are known to not work unless significant effort is devoted to solve the issues, or are intended to solve problems across "closed" networks, instead of as a general Internet access usage. In addition to the different pros, cons and trade-offs, which may be acceptable for some operators, they have implementation difficulties, as they are beyond the original expectations of the NAT64/DNS64 original intent.

3.2.1. Service Provider NAT64 without DNS64

In this scenario (Figure 10), the service provider offers a NAT64 function, however there is no DNS64 function support at all.

As a consequence, an IPv6 host in the IPv6-only access network, will not be able to detect the presence of DNS64 by means of [RFC7050], neither to learn the IPv6 prefix to be used for the NAT64 function.

This can be sorted out as indicated in Section 4.1.1.

However, despite that, because the lack of the DNS64 function, the IPv6 host will not be able to obtain AAAA synthesized records, so the NAT64 function becomes useless.

An exception to this "useless" scenario will be manually configure mappings between the A records of each of the IPv4-only remote hosts and the corresponding AAAA records, with the WKP (Well-Known Prefix) or NSP (Network-Specific Prefix) used by the service provider NAT64 function, as if they were synthesized by a DNS64 function.

This mapping could be done by several means, typically at the authoritative DNS server, or at the service provider resolvers by means of DNS RPZ (Response Policy Zones, [I-D.vixie-dns-rpz]) or equivalent functionality. DNS RPZ, may have implications in DNSSEC, if the zone is signed. Also, if the service provider is using an NSP, having the mapping at the authoritative server, may create troubles to other parties trying to use different NSP or the WKP, unless multiple DNS "views" (split-DNS) is also being used at the authoritative servers.

Generally, the mappings alternative, will only make sense if a few set of IPv4-only remote hosts need to be accessed by a single network (or a small number of them), which support IPv6-only in the access. This will require some kind of mutual agreement for using this procedure, so it doesn't care if they become a trouble for other parties across Internet ("closed services").

In any case, this scenario doesn't solve the issue of IPv4 literal addresses or non-IPv6 compliant APIs, neither it solves the problem of IPv4-only hosts within that IPv6-only access network.

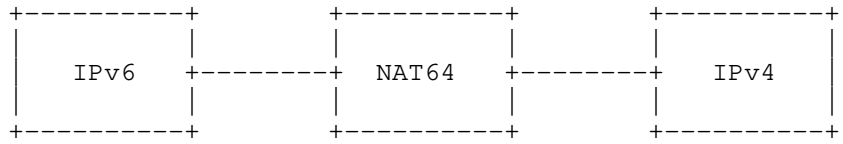

Figure 10: NAT64 without DNS64

3.2.2. Service Provider NAT64; DNS64 in the IPv6 hosts

In this scenario (Figure 11), the service provider offers the NAT64 function, but not the DNS64 function. However, the IPv6 hosts have a built-in DNS64 function.

This may become common if the DNS64 function is implemented in all the IPv6 hosts/stacks. However, commonly this is not the actual situation, even if it may happen in the medium-term. At this way, the DNSSEC validation is performed on the A record, and then the host can use the DNS64 function so to be able to use the NAT64 function, without any DNSSEC issues.

This scenario fails to solve the issue of IPv4 literal addresses or non-IPv6 compliant APIs, unless the IPv6 hosts also supports Happy Eyeballs v2 ([RFC8305], Section 7.1), which may solve that issue.

However, this scenario still fails to solve the problem of IPv4-only hosts or applications behind the IPv6-only access network.

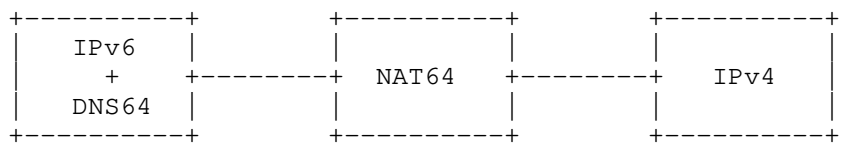

Figure 11: NAT64; DNS64 in IPv6 hosts

3.2.3. Service Provider NAT64; DNS64 in the IPv4-only remote network

In this scenario (Figure 12), the service provider offers the NAT64 function only. The remote IPv4-only network offers the DNS64 function.

This is not common, and looks like doesn't make too much sense that a remote network, not deploying IPv6, is providing a DNS64 function. As in the case of the scenario depicted in Section 3.2.1, it will only work if both sides are using the WKP or the same NSP, so the same considerations apply. It can be also tuned to behave as in Section 3.1.1

This scenario still fails to solve the issue of IPv4 literal addresses or non-IPv6 compliant APIs.

This scenario also fails to solve the problem of IPv4-only hosts or applications behind the IPv6-only access network.

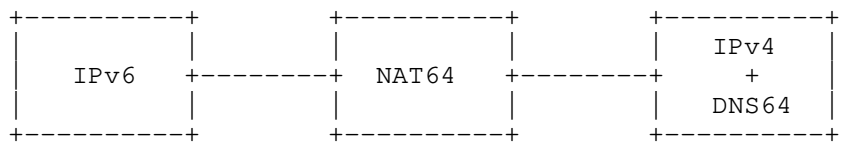

Figure 12: NAT64; DNS64 in the IPv4-only

3.3. Comparing the Scenarios

This section compares the different scenarios, including the possible variations (each one represented in the precedent sections by a different figure), looking at the following criteria:

a. DNSSEC: Are there hosts validating DNSSEC?

b. Literal/APIs: Are there applications using IPv4 literals or non-IPv6 compliant APIs?

c. IPv4-only: Are there hosts or applications using IPv4-only?

d. Foreign DNS: Is the scenario surviving if the user, Operating System, applications or devices change the DNS?

e. DNS load opt. (DNS load optimization): Are there extra queries that may impact DNS infrastructure?

f. Connect. opt. (Connection establishment delay optimization): Is the UE/CE issuing only the AAAA query or also an A query and waiting for both responses?

In the next table, the columns represent each of the scenarios from the previous sections, by the figure number. The possible values are:

- "-" Scenario "bad" for that criteria.
- "+" Scenario "good" for that criteria.
- "*" Scenario "bad" for that criteria, however it is typically resolved, with the support of Happy Eyeballs v2 ([RFC8305]).

In some cases, "countermeasures", alternative or special configurations, may be available for the criteria designated as "bad". So, this comparison is considering a generic case, as a quick comparison guide. In some cases, a "bad" criterion is not necessarily a negative aspect, all it depends on the specific needs/ characteristics of the network where the deployment will take place.

For instance, in a network which has only IPv6-only hosts and apps using only DNS and IPv6-compliant APIs, there is no impact using only NAT64 and DNS64, but if the hosts may validate DNSSEC, that item is still relevant.

Item / Figure	1	2	3	4	5	6	7	8	9	10	11	12
DNSSEC	-	-	-	-	-	-	-	+	+	+	+	+
Literal/APIs	-	-	-	-	+	+	+	+	+	-	-	-
IPv4-only	-	-	-	-	+	+	+	+	+	-	-	-
Foreign DNS	-	-	-	-	+	+	+	+	+	-	+	-
DNS load opt.	+	+	+	+	+	+	+	+	+	+	+	+
Connect. opt.	+	+	+	+	+	+	+	*	*	+	+	+

Figure 13: Scenario Comparison

As a general conclusion, we should note that, if the network must support applications using any of the following:

- IPv4 literals
- non-IPv6-compliant APIs
- IPv4-only hosts or applications

Then, only the scenarios with 464XLAT, a CLAT function, or equivalent built-in local address synthesis features, will provide a valid solution. Further to that, those scenarios will also keep working if the DNS configuration is modified. Clearly also, depending on if DNS64 is used or not, DNSSEC may be broken for those hosts doing DNSSEC validation.

All the scenarios are good in terms of DNS load optimization, and in the case of 464XLAT it may provide an extra degree of optimization. Finally, all them are also good in terms of connection establishment delay optimization. However, in the case of 464XLAT without DNS64, it requires the usage of Happy Eyeballs v2. This is not an issue, as commonly it is available in actual Operating Systems.

4. Issues to be Considered

This section reviews the different issues that an operator needs to consider towards a NAT64/464XLAT deployment, as they may bring to specific decision points about how to approach that deployment.

4.1. DNSSEC Considerations and Possible Approaches

As indicated in Section 8 of [RFC6147] (DNS64, Security Considerations), because DNS64 modifies DNS answers and DNSSEC is designed to detect such modifications, DNS64 may break DNSSEC.

If a device connected to an IPv6-only access network, queries for a domain name in a signed zone, by means of a recursive name server that supports DNS64, and the result is a synthesized AAAA record, and the recursive name server is configured to perform DNSSEC validation and has a valid chain of trust to the zone in question, it will cryptographically validate the negative response from the authoritative name server. This is the expected DNS64 behavior: The recursive name server actually "lies" to the client device. However, in most of the cases, the client will not notice it, because generally, they don't perform validation themselves and instead, rely on the recursive name servers.

A validating DNS64 resolver in fact, increase the confidence on the synthetic AAAA, as it has validated that a non-synthetic AAAA for sure, doesn't exists. However, if the client device is NAT64-oblivious (most common case) and performs DNSSEC validation on the AAAA record, it will fail as it is a synthesized record.

The best possible scenario from DNSSEC point of view, is when the client requests the DNS64 server to perform the DNSSEC validation (by setting the DO bit to 1 and the CD bit to 0). In this case, the DNS64 server validates the data, thus tampering may only happen inside the DNS64 server (which is considered as a trusted part, thus its likelihood is low) or between the DNS64 server and the client. All other parts of the system (including transmission and caching) are protected by DNSSEC ([Threat-DNS64]).

Similarly, if the client querying the recursive name server is another name server configured to use it as a forwarder, and is performing DNSSEC validation, it will also fail on any synthesized AAAA record.

All those considerations are extensively covered in Sections 3, 5.5 and 6.2 of [RFC6147].

A solution to avoid DNSSEC issues, will be that all the signed zones

also provide IPv6 connectivity, together with the corresponding AAAA records. However, this is out of the control of the operator needing to deploy a NAT64 function. This has been proposed already in [I-D.bp-v6ops-ipv6-ready-dns-dnssec].

An alternative solution, which was the one considered while developing [RFC6147], is that validators will be DNS64-aware, so could perform the necessary discovery and do their own synthesis. That was done under the expectation that it was sufficiently early in the validator-deployment curve that it would be ok to break certain DNSSEC assumptions for networks who were really stuck in a NAT64/ DNS64-needing world.

As already indicated, the scenarios in the previous section, are in fact somehow simplified, looking at the worst possible case. Saying it in a different way: "trying to look for the most perfect approach". DNSSEC breach will not happen if the end-host is not doing validation.

Existing previous studies seems to indicate that the figures of DNSSEC actually broken by using DNS64 will be around 1.7% ([About-DNS64]) of the cases. However, we can't negate that this may increase, as DNSSEC deployment grows. Consequently, a decision point for the operator must depend on "do I really care for that percentage of cases and the impact in my helpdesk or can I provide alternative solutions for them?". Some possible solutions may be taken, as depicted in the next sections.

4.1.1. Not using DNS64

A solution will be to avoid using DNS64, but as already indicated this is not possible in all the scenarios.

The use of DNS64 is a key component for some networks, in order to comply with traffic performance metrics, monitored by some governmental bodies and other institutions ([FCC], [ARCEP]).

One drawback of not having a DNS64 at the network side, is that is not possible to heuristically discover the NAT64 ([RFC7050]). Consequently, an IPv6 host behind the IPv6-only access network, will not be able to detect the presence of the NAT64 function, neither to learn the IPv6 prefix to be used for it, unless it is configured by alternative means.

The discovery of the IPv6 prefix could be solved, as described in [RFC7050], by means of adding the relevant AAAA records to the ipv4only.arpa. zone, of the service provider recursive servers, i.e., if using the WKP (64:ff9b::/96):

```
ipv4only.arpa.  SOA     . . 0 0 0 0 0
ipv4only.arpa.  NS      .
ipv4only.arpa.  AAAA    64:ff9b::192.0.0.170
ipv4only.arpa.  AAAA    64:ff9b::192.0.0.171
ipv4only.arpa.  A       192.0.0.170
ipv4only.arpa.  A       192.0.0.171
```

An alternative option to the above, is the use of DNS RPZ ([I-D.vixie-dns-rpz]) or equivalent functionalities. Note that this may impact DNSSEC if the zone is signed.

One more alternative, only valid in environments with PCP support (for both the hosts or CEs and for the service provider network), is to follow [RFC7225] (Discovering NAT64 IPv6 Prefixes using PCP).

Other alternatives may be available in the future. All them are extensively discussed in [RFC7051], however the deployment evolution has evolved many considerations from that document. New options are being documented, such using Router Advertising ([I-D.ietf-6man-ra-pref64]) or DHCPv6 options ([I-D.li-intarea-nat64-prefix-dhcp-option]).

It may be convenient the simultaneous support of several of the possible approaches, in order to ensure that clients with different ways to configure the NAT64 prefix, successfully obtain it. This is also convenient even if DNS64 is being used.

Of special relevance to this section is also [I-D.cheshire-sudn-ipv4only-dot-arpa].

4.1.2. DNSSEC validator aware of DNS64

In general, by default, DNS servers with DNS64 function will not synthesize AAAA responses if the DNSSEC OK (DO) flag was set in the query.

In this case, as only an A record is available, if a CLAT function is present, it means that the CLAT will take the responsibility, as in the case of literal IPv4 addresses, to keep that traffic flow end-to-end as IPv4, so DNSSEC is not broken.

However, this will not work if a CLAT function is not present as the hosts will not be able to use IPv4 (which is the case for all the scenarios without 464XLAT).

4.1.3. Stub validator

If the DO flag is set and the client device performs DNSSEC validation, and the Checking Disabled (CD) flag is set for a query, the DNS64 recursive server will not synthesize AAAA responses. In this case, the client could perform the DNSSEC validation with the A record and then synthesize the AAAA ([RFC6052]). For that to be possible, the client must have learned beforehand the NAT64 prefix using any of the available methods ([RFC7050], [RFC7225], [I-D.ietf-6man-ra-pref64], [I-D.li-intarea-nat64-prefix-dhcp-option]). This allows the client device to avoid using the DNS64 function and still use NAT64 even with DNSSEC.

If the end-host is IPv4-only, this will not work if a CLAT function is not present (scenarios without 464XLAT).

Some devices or Operating Systems may implement, instead of a CLAT, an equivalent function by using Bump-in-the-Host ([RFC6535]), implemented as part of Happy Eyeballs v2 (Section 7.1 of [RFC8305]). In this case, the considerations in the above paragraphs are also applicable.

4.1.4. CLAT with DNS proxy and validator

If a CE includes CLAT support and also a DNS proxy, as indicated in Section 6.4 of [RFC6877], the CE could behave as a stub validator on behalf of the client devices. Then, following the same approach described in the Section 4.1.3, the DNS proxy actually will "lie" to the client devices, which in most of the cases will not notice it, unless they perform validation by themselves. Again, this allow the client devices to avoid using the DNS64 function and still use NAT64 with DNSSEC.

Once more, this will not work without a CLAT function (scenarios without 464XLAT).

4.1.5. ACL of clients

In cases of dual-stack clients, the AAAA queries typically take preference over A queries. If DNS64 is enabled for those clients, will never get A records, even for IPv4-only servers.

As a consequence, in cases where there are IPv4-only servers, and those are located in the path before the NAT64 function, the clients will not be able to reach them. If DNSSEC is being used for all those flows, specific addresses or prefixes can be left-out of the DNS64 synthesis by means of ACLs.

Once more, this will not work without a CLAT function (scenarios without 464XLAT).

4.1.6. Mapping-out IPv4 addresses

If there are well-known specific IPv4 addresses or prefixes using DNSSEC, they can be mapped-out of the DNS64 synthesis.

Even if this is not related to DNSSEC, this "mapping-out" feature is actually, quite commonly used to ensure that [RFC1918] addresses (for example used by LAN servers) are not synthesized to AAAA.

Once more, this will not work without a CLAT function (scenarios without 464XLAT).

4.2. DNS64 and Reverse Mapping

When a client device, using DNS64 tries to reverse-map a synthesized IPv6 address, the name server responds with a CNAME record pointing the domain name used to reverse-map the synthesized IPv6 address (the one under ip6.arpa), to the domain name corresponding to the embedded IPv4 address (under in-addr.arpa).

This is the expected behavior, so no issues need to be considered regarding DNS reverse mapping.

4.3. Using 464XLAT with/without DNS64

In the case the client device is IPv6-only (either because the stack or application is IPv6-only, or because it is connected via an IPv6-only LAN) and the remote server is IPv4-only (either because the stack is IPv4-only, or because it is connected via an IPv4-only LAN), only NAT64 combined with DNS64 will be able to provide access among both. Because DNS64 is then required, DNSSEC validation will be only possible if the recursive name server is validating the negative response from the authoritative name server and the client is not performing validation.

Note that is not expected at this stage of the transition, that applications, devices or Operating Systems are IPv6-only. It will not be a sensible decision for a developer to work on that direction, unless it is clear that the deployment scenario fully supports it.

On the other hand, an end-user or enterprise network may decide to run IPv6-only in the LANs. In case there is any chance for applications to be IPv6-only, the Operating System may be responsible either for doing a local address synthesis, or alternatively, setting up some kind of on-demand VPN (IPv4-in-IPv6), which need to be

supported by that network. This may become very common in enterprise networks, where "Unique IPv6 Prefix per Host" [RFC8273] is supported.

However, when the client device is dual-stack and/or connected in a dual-stack LAN by means of a CLAT function (or has a built-in CLAT function), DNS64 is an option.

1. With DNS64: If DNS64 is used, most of the IPv4 traffic (except if using literal IPv4 addresses or non-IPv6 compliant APIs) will not use the CLAT, so will use the IPv6 path and only one translation will be done at the NAT64. This may break DNSSEC, unless measures as described in the precedent sections are taken.

2. Without DNS64: If DNS64 is not used, all the IPv4 traffic will make use of the CLAT, so two translations are required (NAT46 at the CLAT and NAT64 at the PLAT), which adds some overhead in terms of the extra NAT46 translation. However, this avoids the AAAA synthesis and consequently will never break DNSSEC.

Note that the extra translation, when DNS64 is not used, takes place at the CLAT, which means no extra overhead for the operator. It however adds potential extra delays to establish the connections, and no perceptible impact for a CE in a broadband network, while it may have some impact in a battery powered device. This cost for a battery powered device, is possibly comparable to the cost when the device is doing a local address synthesis (see Section 7.1 of [RFC8305]).

4.4. Foreign DNS

Clients, devices or applications in a service provider network, may use DNS servers from other networks. This may be the case either if individual applications use their own DNS server, the Operating System itself or even the CE, or combinations of the above.

Those "foreign" DNS servers may not support DNS64, which as a consequence, will mean that those scenarios that require a DNS64 may not work. However, if a CLAT function is available, the considerations in Section 4.3 will apply.

In the case that the foreign DNS supports the DNS64 function, we may be in the situation of providing incorrect configurations parameters, for example, un-matching WKP or NSP, or a case such the one described in Section 3.2.3.

Having a CLAT function, even if using foreign DNS without a DNS64 function, ensures that everything will work, so the CLAT must be considered as an advantage even against user configuration errors.

The cost of this, is that all the traffic will use a double translation (NAT46 at the CLAT and NAT64 at the operator network), unless there is support for EAM (Section 4.9).

An exception to that is the case when there is a CLAT function at the CE, which is not able to obtain the correct configuration parameters (again, un-matching WKP or NSP).

However, it needs to be emphasized, that if there is not a CLAT function (scenarios without 464XLAT), an external DNS without DNS64 support, will disallow any access to IPv4-only destination networks, and will not guarantee the correct DNSSEC validation, so will behave as in the Section 3.2.1.

In summary, it can be said, that the consequences of the use of foreign DNS depend very much in each specific case. However, in general, if a CLAT function is present, most of the time, there will not be any. In the other cases, generally, the access to IPv6-enabled services is still guaranteed for IPv6-enabled hosts, but not for IPv4-only hosts, neither the access to IPv4-only services for any hosts in the network.

The causes of "foreign DNS" could be classified in three main categories, as depicted in the following sub-sections.

4.4.1. Manual Configuration of DNS

It is becoming increasingly common that end-users or even devices or applications configure alternative DNS in their Operating Systems, and sometimes in CEs.

4.4.2. DNS Privacy/Encryption Mechanisms

Clients or applications may use mechanisms for DNS privacy/ encryption, such as DNS over TLS ([RFC7858]), DNS over DTLS ([RFC8094]), DNS queries over HTTPS ([RFC8484]) or DNS over QUIC ([I-D.huitema-quic-dnsoquic]). Those are commonly cited as DoT, DoH and DoQ.

Those DNS privacy/encryption options, currently are typically provided by the applications, not the Operating System vendors. At the time of writing this document, at least DoT and DoH standards have declared DNS64 (and consequently NAT64) out of their scope, so an application using them may break NAT64, unless a correctly configured CLAT function is used.

4.4.3. Split DNS and VPNs

When networks or hosts use "split-DNS" (also called Split Horizon, DNS views or private DNS), the successful use of the DNS64 is not guaranteed. Section 4 of [RFC6950], analyses this case.

A similar situation may happen in case of VPNs that force all the DNS queries through the VPN, ignoring the operator DNS64 function.

4.5. Well-Known Prefix (WKP) vs Network-Specific Prefix (NSP)

Section 3 of [RFC6052] (IPv6 Addressing of IPv4/IPv6 Translators), discusses some considerations which are useful to decide if an operator should use the WKP or an NSP.

Taking in consideration that discussion and other issues, we can summarize the possible decision points as:

a. The WKP MUST NOT be used to represent non-global IPv4 addresses. If this is required because the network to be translated use non-global addresses, then an NSP is required.

b. The WKP MAY appear in inter-domain routing tables, if the operator provides a NAT64 function to peers. However, in this case, special considerations related to BGP filtering are required and IPv4-embedded IPv6 prefixes longer than the WKP MUST NOT be advertised (or accepted) in BGP. An NSP may be a more appropriate option in those cases.

c. If several NAT64 use the same prefix, packets from the same flow may be routed to different NAT64 in case of routing changes. This can be avoided either by using different prefixes for each NAT64 function, or by ensuring that all the NAT64 coordinate their state. Using an NSP could simplify that.

d. If DNS64 is required and users, devices, Operating Systems or applications may change their DNS configuration, and deliberately choose an alternative DNS64 function, most probably alternative DNS64 will use by default the WKP. In that case, if an NSP is used by the NAT64 function, clients will not be able to use the operator NAT64 function, which will break connectivity to IPv4-only destinations.

4.6. IPv4 literals and non-IPv6 Compliant APIs

A host or application using literal IPv4 addresses or older APIs, which aren’t IPv6 compliant, behind a network with IPv6-only access, will not work unless any of the following alternatives is provided:

- o CLAT (or equivalent function).
- o Happy Eyeballs v2 (Section 7.1, [RFC8305]).
- o Bump-in-the-Host ([RFC6535]) with a DNS64 function.

Those alternatives will solve the problem for an end-host. However, if that end-hosts is providing "tethering" or an equivalent service to other hosts, that needs to be considered as well. In other words, in a case of a cellular network, it resolves the issue for the UE itself, but may be not the case for hosts behind it.

Otherwise, the support of 464XLAT is the only valid and complete approach to resolve this issue.

4.7. IPv4-only Hosts or Applications

An IPv4-only hosts or application behind a network with IPv6-only access, will not work unless a CLAT function is present.

464XLAT is the only valid approach to resolve this issue.

4.8. CLAT Translation Considerations

As described in Section 6.3 of [RFC6877] (IPv6 Prefix Handling), if the CLAT function can be configured with a dedicated /64 prefix for the NAT46 translation, then it will be possible to do a more efficient stateless translation.

Otherwise, if this dedicated prefix is not available, the CLAT function will need to do a stateful translation, for example performing stateful NAT44 for all the IPv4 LAN packets, so they appear as coming from a single IPv4 address, and then in turn, stateless translated to a single IPv6 address.

A possible setup, in order to maximize the CLAT performance, is to configure the dedicated translation prefix. This can be easily achieved automatically, if the broadband CE or end-user device is able to obtain a shorter prefix by means of DHCPv6-PD ([RFC8415]), or other alternatives. The CE can then use a specific /64 for the translation. This is also possible when broadband is provided by a cellular access.

The above recommendation is often not possible for cellular networks, when connecting smartphones (as UEs), as generally they don’t use DHCPv6-PD ([RFC8415]). Instead, a single /64 is provided for each PDP context and prefix sharing ([RFC6877]) is used. So, in this case, the UEs typically have a build-in CLAT function which is

performing a stateful NAT44 translation before the stateless NAT46.

4.9. EAM Considerations

Explicit Address Mappings for Stateless IP/ICMP Translation ([RFC7757]) provide a way to configure explicit mappings between IPv4 and IPv6 prefixes of any length. When this is used, for example in a CLAT function, it may provide a simple mechanism in order to avoid traffic flows between IPv4-only nodes or applications and dual-stack destinations to be translated twice (NAT46 and NAT64), by creating mapping entries with the GUA of the IPv6-reachable destination. This optimization of the NAT64 usage is very useful in many scenarios, including CDNs and caches, as described in [I-D.palet-v6ops-464xlat-opt-cdn-caches].

In addition to that, it may provide as well a way for IPv4-only nodes or applications to communicate with IPv6-only destinations.

4.10. Incoming Connections

The use of NAT64, in principle, disallows IPv4 incoming connections, which may be still needed for IPv4-only peer-to-peer applications. However, there are several alternatives that resolve this issue:

a. STUN ([RFC5389]), TURN ([RFC5766]) and ICE ([RFC8445]) are commonly used by peer-to-peer applications in order to allow incoming connections with IPv4 NAT. In the case of NAT64, they work as well. RFC editor note: If in time, replace STUN and TURN with [I-D.ietf-tram-stunbis] / [I-D.ietf-tram-turnbis].

b. PCP ([RFC6887]) allows a host to control how incoming IPv4 and IPv6 packets are translated and forwarded. A NAT64 may implement PCP to allow this service.

c. EAM ([RFC7757]) may also be used in order to configure explicit mappings for customers that require them. This is used for example by SIIT-DC ([RFC7755]) and SIIT-DC-DTM ([RFC7756]).

5. Summary of Deployment Recommendations for NAT64/464XLAT

NAT64/464XLAT has demonstrated to be a valid choice in several scenarios (IPv6-IPv4 and IPv4-IPv6-IPv4), being the predominant mechanism in the majority of the cellular networks, which account for hundreds of millions of users ([ISOC]). NAT64/464XLAT offer different choices of deployment, depending on each network case, needs and requirements. Despite that, this document is not an explicit recommendation for using this choice versus other IPv4aaS transition mechanisms. Instead, this document is a guide that

facilitates evaluating a possible implementation of NAT64/464XLAT and key decision points about specific design considerations for its deployment.

Depending on the specific requirements of each deployment case, DNS64 may be a required function, while in other cases the adverse effects may be counterproductive. Similarly, in some cases a NAT64 function, together with a DNS64 function, may be a valid solution, when there is a certainty that IPv4-only hosts or applications do not need to be supported (Section 4.6 and Section 4.7). However, in other cases (i.e. IPv4-only devices or applications need to be supported), the limitations of NAT64/DNS64, may suggest the operator to look into 464XLAT as a more complete solution.

In the case of broadband managed networks (where the CE is provided or suggested/supported by the operator), in order to fully support the actual user needs (IPv4-only devices and applications, usage of IPv4 literals and non-IPv6 compliant APIs), the 464XLAT scenario should be considered. In that case, it must support a CLAT function.

If the operator provides DNS services, in order to increase performance by reducing the double translation for all the IPv4 traffic, they may support a DNS64 function and avoid, as much as possible, breaking DNSSEC. In this case, if the DNS service is offering DNSSEC validation, then it must be in such way that it is aware of the DNS64. This is considered the simpler and safer approach, and may be combined as well with other recommendations described in this document:

- DNS infrastructure MUST be aware of DNS64 (Section 4.1.2).
- Devices running CLAT SHOULD follow the indications in Section 4.1.3 (Stub Validator). However, this may be out of the control of the operator.
- CEs SHOULD include a DNS proxy and validator (Section 4.1.4).
- Section 4.1.5 (ACL of clients) and Section 4.1.6 (Mapping-out IPv4 addresses) MAY be considered by operators, depending on their own infrastructure.

This "increased performance" approach has the disadvantage of potentially breaking DNSSEC for a small percentage of validating end-hosts versus the small impact of a double translation taking place in the CE. If CE performance is not an issue, which is the most frequent case, then a much safer approach is to not use DNS64 at all, and consequently, ensure that all the IPv4 traffic is translated at the CLAT (Section 4.3).

If DNS64 is not used, at least one of the alternatives described in Section 4.1.1, must be followed in order to learn the NAT64 prefix.

The operator needs to consider that if the DNS configuration can be modified (Section 4.4, Section 4.4.2, Section 4.4.3), which most probably is impossible to avoid, there are chances that instead of configuring a DNS64 a foreign non-DNS64 is used. In a scenario with only a NAT64 function IPv4-only remote host will no longer be accessible. Instead, it will continue to work in the case of 464XLAT.

Similar considerations need to be taken regarding the usage of a NAT64 WKP vs NSP (Section 4.5), as they must match with the configuration of the DNS64. In case of using foreign DNS, they may not match. If there is a CLAT and the configured foreign DNS is not a DNS64, the network will keep working only if other means of learning the NAT64 prefix are available.

As described in Section 4.8, for broadband networks, the CEs supporting a CLAT function, SHOULD support DHCPv6-PD ([RFC8415]), or alternative means for configuring a shorter prefix. The CE SHOULD internally reserve one /64 for the stateless NAT46 translation. The operator must ensure that the customers get allocated prefixes shorter than /64 in order to support this optimization. One way or the other, this is not impacting the performance of the operator network.

Operators may follow Section 7 of [RFC6877] (Deployment Considerations), for suggestions in order to take advantage of traffic engineering requirements.

In the case of cellular networks, the considerations regarding DNSSEC may appear as out-of-scope, because UEs Operating Systems, commonly don't support DNSSEC. However, applications running on them may do, or it may be an Operating System "built-in" support in the future. Moreover, if those devices offer tethering, other client devices behind the UE, may be doing the validation, hence the relevance of a proper DNSSEC support by the operator network.

Furthermore, cellular networks supporting 464XLAT ([RFC6877]) and "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis" ([RFC7050]), allow a progressive IPv6 deployment, with a single APN supporting all types of PDP context (IPv4, IPv6, IPv4v6). This approach allows the network to automatically serve every possible combinations of UEs.

If the operator chooses to provide validation for the DNS64 prefix discovery, it must follow the advice from Section 3.1. of [RFC7050]

(Validation of Discovered Pref64::/n).

One last consideration, is that many networks may have a mix of different complex scenarios at the same time, for example, customers requiring 464XLAT, others not requiring it, customers requiring DNS64, others not, etc. In general, the different issues and the approaches described in this document can be implemented at the same time for different customers or parts of the network. That mix of approaches don't present any problem or incompatibility, as they work well together, being just a matter of appropriate and differentiated provisioning. In fact, the NAT64/464XLAT approach facilitates an operator offering both cellular and broadband services, to have a single IPv4aaS for both networks while differentiating the deployment key decisions to optimize each case. It even makes possible using hybrid CEs that have a main broadband access link and a backup via the cellular network.

In an ideal world we could safely use DNS64, if the approach proposed in [I-D.bp-v6ops-ipv6-ready-dns-dnssec] is followed, avoiding the cases where DNSSEC may be broken. However, this will not solve the issues related to DNS Privacy and Split DNS.

The only 100% safe solution, which also resolves all the issues, will be, in addition to having a CLAT function, not using a DNS64 but instead making sure that the hosts have a built-in address synthesis feature. Operators could manage to provide CEs with the CLAT function, however the built-in address synthesis feature is out of their control. If the synthesis is provided either by the Operating System (via its DNS resolver API) or by the application (via its own DNS resolver), in such way that the prefix used for the NAT64 function is reachable for the host, the problem goes away.

Whenever feasible, using EAM ([RFC7757]) as indicated in Section 4.9, provides a very relevant optimization, avoiding double-translations.

Applications that require incoming connections, typically already provide means for that. However, PCP and EAM, as indicated in Section 4.10, are valid alternatives, even for creating explicit mappings for customers that require them.

6. Deployment of 464XLAT/NAT64 in Enterprise Networks

The recommendations of this document can be used as well in enterprise networks, campus and other similar scenarios (including managed end-user networks).

This include scenarios where the NAT64 function (and DNS64 function, if available) are under the control of that network (or can be

configured manually according to that network specific requirements), and for whatever reasons, there is a need to provide "IPv6-only access" to any part of that network or it is IPv6-only connected to third party-networks.

An example of that is the IETF meetings network itself, where both NAT64 and DNS64 functions are provided, presenting in this case the same issues as per Section 3.1.1. If there is a CLAT function in the IETF network, then there is no need to use DNS64 and it falls under the considerations of Section 3.1.3. Both scenarios have been tested and verified already in the IETF network itself.

Next figures are only meant to represent a few of the possible scenarios, not pretending to be the only feasible ones.

Figure 14 provides an example of an IPv6-only enterprise network connected with dual-stack to Internet and using local NAT64 and DNS64 functions.

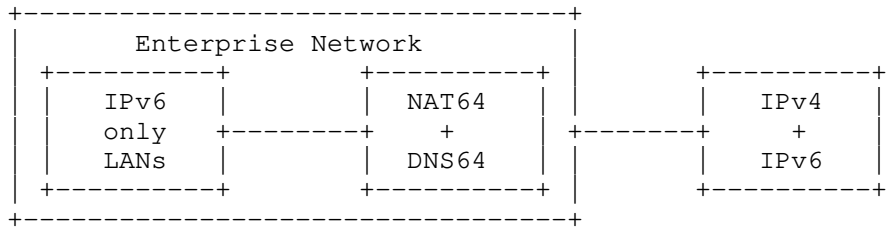

Figure 14: IPv6-only enterprise with NAT64 and DNS64

Figure 15 provides an example of a dual-stack (DS) enterprise network connected with dual-stack (DS) to Internet and using a CLAT function, without a DNS64 function.

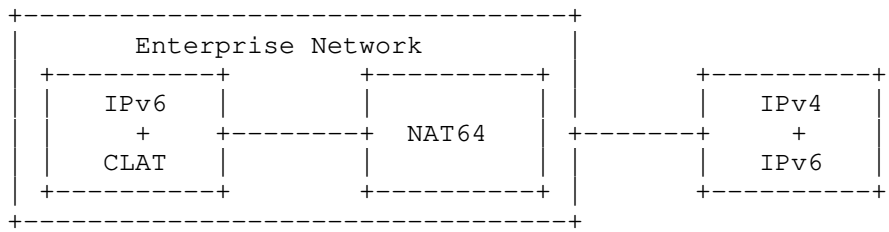

Figure 15: DS enterprise with CLAT, DS Internet, without DNS64

Finally, Figure 16 provides an example of an IPv6-only provider with a NAT64 function, and a dual-stack (DS) enterprise network by means of their own CLAT function, without a DNS64 function.

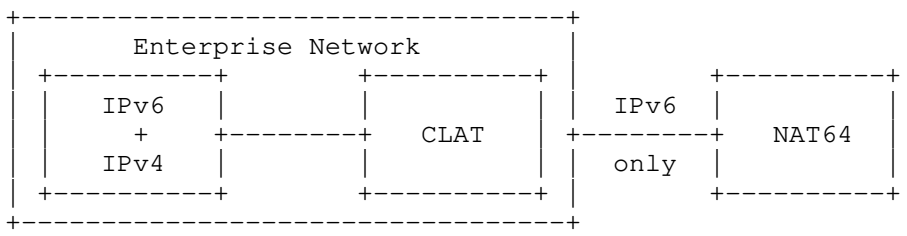

Figure 16: DS enterprise with CLAT, IPv6-only Access, without DNS64

7. Security Considerations

This document does not have new specific security considerations beyond those already reported by each of the documents cited. For example, DNS64 ([RFC6147]) already describes the DNSSEC issues.

Note that, as already described in Section 4.4, there may be undesirable interactions, specially if using VPNs or DNS privacy, which may impact in the correct performance of DNS64/NAT64.

It should be remarked that the use of a DNS64 function has equivalent privacy considerations as in the case of a regular DNS, either located in the service provider or an external one.

8. IANA Considerations

This document does not have any new specific IANA considerations.

Note: This section is assuming that https://www.rfc-editor.org/errata/eid5152 is resolved, otherwise, this section may include the required text to resolve the issue.

Alternatively, this could be fixed also by [I-D.cheshire-sudn-ipv4only-dot-arpa].

9. Acknowledgements

The author would like to acknowledge the inputs of Gabor Lencse, Andrew Sullivan, Lee Howard, Barbara Stark, Fred Baker, Mohamed Boucadair, Alejandro D'Egidio, Dan Wing, Mikael Abrahamsson and Eric Vyncke.

Conversations with Marcelo Bagnulo, one of the co-authors of NAT64 and DNS64, as well as several emails in mailing lists from Mark Andrews, have been very useful for this work.

Christian Huitema inspired working in this document by suggesting

that DNS64 should never be used, during a discussion regarding the deployment of CLAT in the IETF network.

10. ANNEX A: Example of Broadband Deployment with 464XLAT

This section summarizes how an operator may deploy an IPv6-only network for residential/SOHO customers, supporting IPv6 inbound connections, and IPv4-as-a-Service (IPv4aaS) by using 464XLAT.

Note that an equivalent setup could also be provided for enterprise customers. In case they need to support IPv4 inbound connections, several mechanisms, depending on specific customer needs, allow that, for instance [RFC7757].

Conceptually, most part of the operator network could be IPv6-only (represented in the next pictures as "IPv6-only flow"), or even if this part of the network is actually dual-stack, only IPv6-access is available for some customers (i.e. residential customers). This part of the network connects the IPv6-only subscribers (by means of IPv6-only access links), to the IPv6 upstream providers, as well as to the IPv4-Internet by means of the NAT64 (PLAT in the 464XLAT terminology).

The traffic flow from and back to the CE to services available in the IPv6 Internet (or even dual-stack remote services, when IPv6 is being used), is purely native IPv6 traffic, so there are no special considerations about it.

Looking at the picture from the DNS perspective, there are remote networks with are IPv4-only, and typically will have only IPv4 DNS (DNS/IPv4), or at least will be seen as that from the CE perspective. At the operator side, the DNS, as seen from the CE, is only IPv6 (DNS/IPv6) and has also a DNS64 function.

In the customer LANs side, there is actually one network, which of course could be split in different segments. The most common setup will be those segments being dual-stack, using global IPv6 addresses and [RFC1918] for IPv4, as usual in any regular residential/SOHO IPv4 network. In the figure, it is represented as tree segments, just to show that the three possible setups are valid (IPv6-only, IPv4-only and dual-stack).

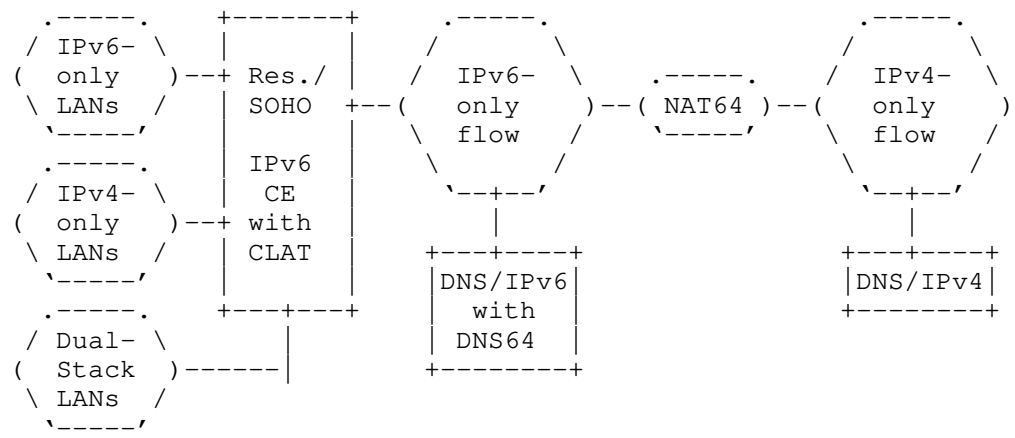

Figure 17: CE setup with built-in CLAT with DNS64

In addition to the regular CE setup, which will be typically access-technology dependent, the steps for the CLAT function configuration can be summarized as:

1. Discovery of the PLAT (NAT64) prefix: It may be done using [RFC7050], or in those networks where PCP is supported, by means of [RFC7225], or other alternatives that may be available in the future, such as Router Advertising ([I-D.ietf-6man-ra-pref64]) or DHCPv6 options ([I-D.li-intarea-nat64-prefix-dhcp-option]).

2. If the CLAT function allows stateless NAT46 translation, a /64 from the pool typically provided to the CE by means of DHCPv6-PD [RFC8415], need to be set aside for that translation. Otherwise, the CLAT is forced to perform an intermediate stateful NAT44 before the a stateless NAT46, as described in Section 4.8.

A more detailed configuration approach is described in [RFC8585].

The operator network needs to ensure that the correct responses are provided for the discovery of the PLAT prefix. It is highly recommended to follow [RIPE-690], in order to ensure that multiple /64s are available, including the one needed for the NAT46 stateless translation.

The operator needs to understand other issues, described across this document, in order to take the relevant decisions. For example, if several NAT64 functions are needed in the context of scalability/ high-availability, an NSP should be considered (Section 4.5).

More complex scenarios are possible, for example, if a network offers

multiple NAT64 prefixes, destination-based NAT64 prefixes, etc.

If the operator decides not to provide a DNS64 function, then this setup turns into the one in the following Figure. This will be also the setup that "will be seen" from the perspective of the CE, if a foreign DNS is used and consequently is not the operator-provided DNS64 function.

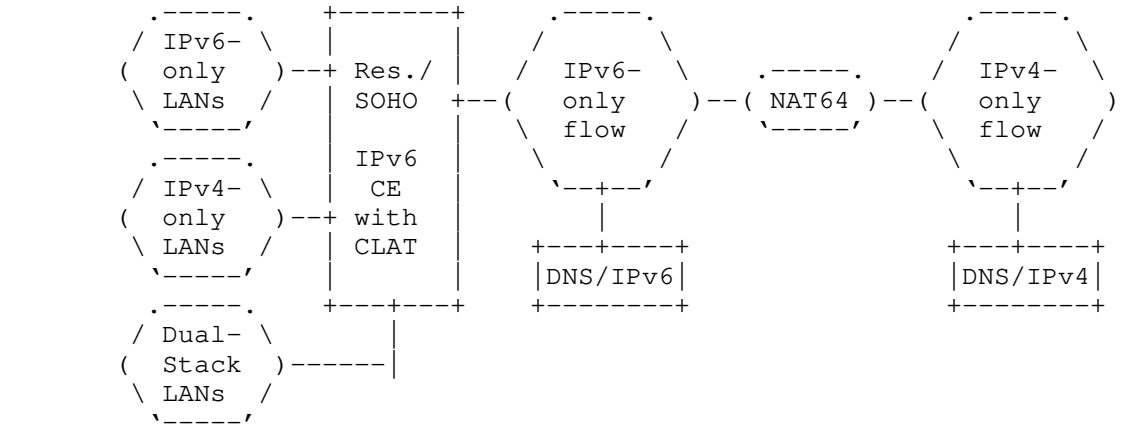

Figure 18: CE setup with built-in CLAT without DNS64

In this case, the discovery of the PLAT prefix needs to be arranged as indicated in Section 4.1.1.

In this case, the CE doesn't have a built-in CLAT function, or the customer can choose to setup the IPv6 operator-managed CE in bridge mode (and optionally use an external router), or for example, there is an access technology that requires some kind of media converter (ONT for FTTH, Cable-Modem for DOCSIS, etc.), the complete setup will look as in Figure 19. Obviously, there will be some intermediate configuration steps for the bridge, depending on the specific access technology/protocols, which should not modify the steps already described in the previous cases for the CLAT function configuration.

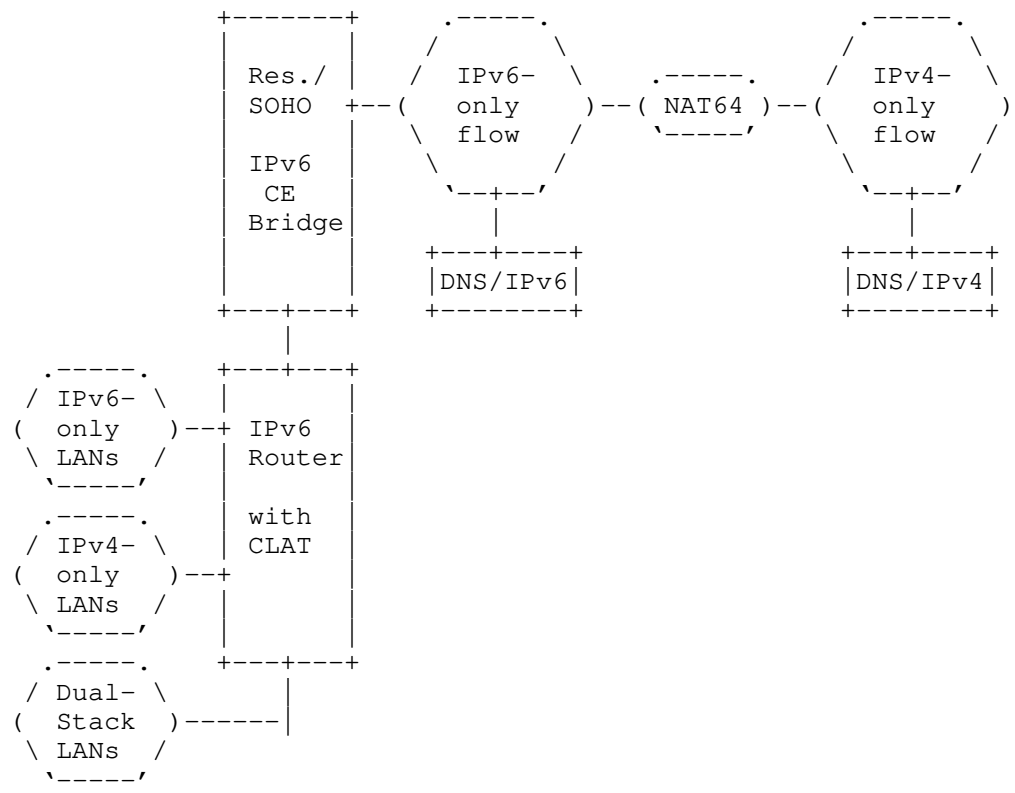

Figure 19: CE setup with bridged CLAT without DNS64

It should be avoided that several routers (i.e., the operator provided CE and a downstream user provided router) enable simultaneously routing and/or CLAT, in order to avoid multiple NAT44 and NAT46 levels, as well as ensuring the correct operation of multiple IPv6 subnets. In those cases, it is suggested the use of HNCP ([RFC8375]).

Note that the procedure described here for the CE setup, can be simplified if the CE follows [RFC8585].

11. ANNEX B: CLAT Implementation

In addition to the regular set of features for a CE, a CLAT CE implementation requires support of:

- [RFC7915] for the NAT46 function.
- [RFC7050] for the PLAT prefix discovery.

- [RFC7225] for the PLAT prefix discovery if PCP is supported.
- [I-D.ietf-6man-ra-pref64] for the PLAT prefix discovery by means of Router Advertising.
- [I-D.li-intarea-nat64-prefix-dhcp-option] for the PLAT prefix discovery by means of DHCP.
- If stateless NAT46 is supported, a mechanism to ensure that multiple /64 are available, such as DHCPv6-PD [RFC8415].

There are several OpenSource implementations of CLAT, such as:

- Android: https://github.com/ddrown/android_external_android-clat.
- Jool: https://www.jool.mx.
- Linux: https://github.com/toreanderson/clatd.
- OpenWRT: https://github.com/openwrt-routing/packages/blob/master/nat46/files/464xlat.sh.
- VPP: https://git.fd.io/vpp/tree/src/plugins/nat.

12. ANNEX C: Benchmarking

[RFC8219] has defined a benchmarking methodology for IPv6 transition technologies. NAT64 and 464XLAT are addressed among the single and double translation technologies, respectively. DNS64 is addressed in Section 9, and the methodology is more elaborated in [DNS64-BM-Meth].

Several documents provide references to benchmarking results, for example in the case of DNS64, [DNS64-Benchm].

13. ANNEX D: Changes from -00 to -01/-02

Section to be removed after WGLC. Significant updates are:

1. Text changes across all the document.

14. ANNEX E: Changes from -02 to -03

Section to be removed after WGLC. Significant updates are:

1. Added references to new cited documents.
2. Reference to RFC8273 and on-demand IPv4-in-IPv6 VPN for IPv6-only LANs w/o DNS64.

3. Overall review and editorial changes.

15. ANNEX F: Changes from -03 to -04

Section to be removed after WGLC. Significant updates are:

1. Added text related to EAM considerations.

16. ANNEX G: Changes from -04 to -05

Section to be removed after WGLC. Significant updates are:

1. Added cross references to EAM section.

2. Reworded "foreing DNS section".

3. Overall editorial review of text, pictures and nits correction.

17. ANNEX H: Changes from -05 to -06

Section to be removed after WGLC. Significant updates are:

1. Corrected EAMT to EAM.

2. Typos and nits.

3. New considerations regarding incoming connections.

18. ANNEX H: Changes from -06 to -07

Section to be removed after WGLC. Significant updates are:

1. Inputs/clarifications from IESG review.

19. References

19.1. Normative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/info/rfc1918>.

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

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <https://www.rfc-editor.org/info/rfc5389>.

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, DOI 10.17487/RFC5625, August 2009, <https://www.rfc-editor.org/info/rfc5625>.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, DOI 10.17487/RFC5766, April 2010, <https://www.rfc-editor.org/info/rfc5766>.

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

[RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144, April 2011, <https://www.rfc-editor.org/info/rfc6144>.

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

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <https://www.rfc-editor.org/info/rfc6147>.

[RFC6535] Huang, B., Deng, H., and T. Savolainen, "Dual-Stack Hosts Using "Bump-in-the-Host" (BIH)", RFC 6535, DOI 10.17487/RFC6535, February 2012, <https://www.rfc-editor.org/info/rfc6535>.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <https://www.rfc-editor.org/info/rfc6877>.

[RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <https://www.rfc-editor.org/info/rfc6887>.

[RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <https://www.rfc-editor.org/info/rfc7050>.

[RFC7225] Boucadair, M., "Discovering NAT64 IPv6 Prefixes Using the Port Control Protocol (PCP)", RFC 7225, DOI 10.17487/RFC7225, May 2014, <https://www.rfc-editor.org/info/rfc7225>.

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

[RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <https://www.rfc-editor.org/info/rfc7915>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8273] Brzozowski, J. and G. Van de Velde, "Unique IPv6 Prefix per Host", RFC 8273, DOI 10.17487/RFC8273, December 2017, <https://www.rfc-editor.org/info/rfc8273>.

[RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

[RFC8375] Pfister, P. and T. Lemon, "Special-Use Domain 'home.arpa.'", RFC 8375, DOI 10.17487/RFC8375, May 2018, <https://www.rfc-editor.org/info/rfc8375>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

[RFC8445] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", RFC 8445, DOI 10.17487/RFC8445, July 2018, <https://www.rfc-editor.org/info/rfc8445>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

19.2. Informative References

[About-DNS64]
Linkova, J., "Let's talk about IPv6 DNS64 & DNSSEC", 2016, <https://blog.apnic.net/2016/06/09/lets-talk-ipv6-dns64-dnssec/>.

[ARCEP] ARCEP, "Service client des operateurs : les mesures de qualite de service", 2018, <https://www.arcep.fr/cartes-et-donnees/nos-publications-chiffrees/service-client-des-operateurs-mesures-de-la-qualite-de-service/service-client-des-operateurs-les-mesures-de-qualite-de-service.html>.

[DNS64-Benchm]
Lencse, G. and Y. Kadobayashi, "Benchmarking DNS64 Implementations: Theory and Practice", Computer Communications , vol. 127, no. 1, pp. 61-74, DOI 10.1016/j.comcom.2018.05.005, September 2018.

[DNS64-BM-Meth]
Lencse, G., Georgescu, M., and Y. Kadobayashi, "Benchmarking Methodology for DNS64 Servers", Computer Communications , vol. 109, no. 1, pp. 162-175, DOI 10.1016/j.comcom.2017.06.004, September 2017.

[FCC] FCC, "Measuring Broadband America Mobile 2013-2018 Coarsened Data", 2018, <https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-broadband-america-mobile-2013-2018>.

[I-D.bp-v6ops-ipv6-ready-dns-dnssec]
Byrne, C. and J. Palet, "IPv6-Ready DNS/DNSSSEC Infrastructure", draft-bp-v6ops-ipv6-ready-dns-dnssec-00 (work in progress), October 2018.

[I-D.cheshire-sudn-ipv4only-dot-arpa]
Cheshire, S. and D. Schinazi, "Special Use Domain Name 'ipv4only.arpa'", draft-cheshire-sudn-ipv4only-dot-arpa-14 (work in progress), November 2018.

[I-D.huitema-quic-dnsoquic]
Huitema, C., Shore, M., Mankin, A., Dickinson, S., and J. Iyengar, "Specification of DNS over Dedicated QUIC Connections", draft-huitema-quic-dnsoquic-06 (work in progress), March 2019.

[I-D.ietf-6man-ra-pref64]
Colitti, L. and J. Linkova, "Discovering PREF64 in Router Advertisements", draft-ietf-6man-ra-pref64-01 (work in progress), June 2019.

[I-D.ietf-tram-stunbis]
Petit-Huguenin, M., Salgueiro, G., Rosenberg, J., Wing, D., Mahy, R., and P. Matthews, "Session Traversal Utilities for NAT (STUN)", draft-ietf-tram-stunbis-21 (work in progress), March 2019.

[I-D.ietf-tram-turnbis]
K, R., Johnston, A., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", draft-ietf-tram-turnbis-27 (work in progress), June 2019.

[I-D.li-intarea-nat64-prefix-dhcp-option]
Li, L., Cui, Y., Liu, C., Wu, J., Baker, F., and J. Palet, "DHCPv6 Options for Discovery NAT64 Prefixes", draft-li-intarea-nat64-prefix-dhcp-option-02 (work in progress), April 2019.

[I-D.lmhp-v6ops-transition-comparison]
Lencse, G., Palet, J., Howard, L., Patterson, R., and I. Farrer, "Pros and Cons of IPv6 Transition Technologies for IPv4aaS", draft-lmhp-v6ops-transition-comparison-03 (work in progress), July 2019.

[I-D.palet-v6ops-464xlat-opt-cdn-caches]
Palet, J. and A. D'Egidio, "464XLAT Optimization", draft-palet-v6ops-464xlat-opt-cdn-caches-02 (work in progress), June 2019.

[I-D.vixie-dns-rpz]
Vixie, P. and V. Schryver, "DNS Response Policy Zones (RPZ)", draft-vixie-dns-rpz-04 (work in progress), December 2016.

[ISOC] ISOC, "State of IPv6 Deployment 2018", 2018, <https://www.internetsociety.org/resources/2018/state-of-ipv6-deployment-2018/>.

[RFC6889] Penno, R., Saxena, T., Boucadair, M., and S. Sivakumar, "Analysis of Stateful 64 Translation", RFC 6889, DOI 10.17487/RFC6889, April 2013, <https://www.rfc-editor.org/info/rfc6889>.

[RFC6950] Peterson, J., Kolkman, O., Tschofenig, H., and B. Aboba, "Architectural Considerations on Application Features in the DNS", RFC 6950, DOI 10.17487/RFC6950, October 2013, <https://www.rfc-editor.org/info/rfc6950>.

[RFC7051] Korhonen, J., Ed. and T. Savolainen, Ed., "Analysis of Solution Proposals for Hosts to Learn NAT64 Prefix", RFC 7051, DOI 10.17487/RFC7051, November 2013, <https://www.rfc-editor.org/info/rfc7051>.

[RFC7269] Chen, G., Cao, Z., Xie, C., and D. Binet, "NAT64 Deployment Options and Experience", RFC 7269, DOI 10.17487/RFC7269, June 2014, <https://www.rfc-editor.org/info/rfc7269>.

[RFC7755] Anderson, T., "SIIT-DC: Stateless IP/ICMP Translation for IPv6 Data Center Environments", RFC 7755, DOI 10.17487/RFC7755, February 2016, <https://www.rfc-editor.org/info/rfc7755>.

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

[RFC7849] Binet, D., Boucadair, M., Vizdal, A., Chen, G., Heatley, N., Chandler, R., Michaud, D., Lopez, D., and W. Haeffner, "An IPv6 Profile for 3GPP Mobile Devices", RFC 7849, DOI 10.17487/RFC7849, May 2016, <https://www.rfc-editor.org/info/rfc7849>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/info/rfc7858>.

[RFC8094] Reddy, T., Wing, D., and P. Patil, "DNS over Datagram Transport Layer Security (DTLS)", RFC 8094, DOI 10.17487/RFC8094, February 2017, <https://www.rfc-editor.org/info/rfc8094>.

[RFC8219] Georgescu, M., Pislaru, L., and G. Lencse, "Benchmarking Methodology for IPv6 Transition Technologies", RFC 8219, DOI 10.17487/RFC8219, August 2017, <https://www.rfc-editor.org/info/rfc8219>.

[RFC8585] Palet Martinez, J., Liu, H., and M. Kawashima, "Requirements for IPv6 Customer Edge Routers to Support IPv4-as-a-Service", RFC 8585, DOI 10.17487/RFC8585, May 2019, <https://www.rfc-editor.org/info/rfc8585>.

[RIPE-690] RIPE, "Best Current Operational Practice for Operators: IPv6 prefix assignment for end-users - persistent vs non-persistent, and what size to choose", October 2017, <https://www.ripe.net/publications/docs/ripe-690>.

[Threat-DNS64] Lencse, G. and Y. Kadobayashi, "Methodology for the identification of potential security issues of different IPv6 transition technologies: Threat analysis of DNS64 and stateful NAT64", Computers & Security , vol. 77, no. 1, pp. 397-411, DOI 10.1016/j.cose.2018.04.012, August 2018.

Author's Address

Jordi Palet Martinez
The IPv6 Company
Molino de la Navata, 75
La Navata - Galapagar, Madrid 28420
Spain

Email: jordi.palet@theipv6company.com
URI: http://www.theipv6company.com/

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: July 13, 2021

G. Lencse
BUTE
J. Palet Martinez
The IPv6 Company
L. Howard
Retevia
R. Patterson
Sky UK
I. Farrer
Deutsche Telekom AG
Jan 9, 2021

Pros and Cons of IPv6 Transition Technologies for IPv4aaS
draft-lmhp-v6ops-transition-comparison-06

Abstract

Several IPv6 transition technologies have been developed to provide customers with IPv4-as-a-Service (IPv4aaS) for ISPs with an IPv6-only access and/or core network. All these technologies have their advantages and disadvantages, and depending on existing topology, skills, strategy and other preferences, one of these technologies may be the most appropriate solution for a network operator.

This document examines the five most prominent IPv4aaS technologies considering a number of different aspects to provide network operators with an easy to use reference to assist in selecting the technology that best suits their needs.

Table of Contents

1. Introduction

As the deployment of IPv6 becomes more prevalent, it follows that network operators will move to building single-stack IPv6 core and access networks to simplify network planning and operations. However, providing customers with IPv4 services continues to be a requirement for the foreseeable future. To meet this need, the IETF has standardized a number of different IPv4aaS technologies for this [LEN2019] based on differing requirements and deployment scenarios.

The number of technologies that have been developed makes it time consuming for a network operator to identify the most appropriate mechanism for their specific deployment. This document provides a comparative analysis of the most commonly used mechanisms to assist operators with this problem.

Five different IPv4aaS solutions are considered. The following IPv6 transition technologies are covered:

1. 464XLAT [RFC6877]

2. Dual Stack Lite [RFC6333]

3. lw4o6 (Lightweight 4over6) [RFC7596]

4. MAP-E [RFC7597]

5. MAP-T [RFC7599]

We note that [RFC6180] gives guidelines for using IPv6 transition mechanisms during IPv6 deployment addressing a much broader topic, whereas this document focuses on a small part of it.

1.1. 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 BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Overview of the Technologies

The following sections introduce the different technologies analyzed in this document, describing some of their most important characteristics.

2.1. 464XLAT

464XLAT is a single/dual translation model, which uses a customer-side translator (CLAT) located in the customer's device to perform stateless NAT64 translation [RFC7915] (more precisely, stateless NAT46, a stateless IP/ICMP translation from IPv4 to IPv6). IPv4-embedded IPv6 addresses [RFC6052] are used for both source and destination addresses. Commonly, a /96 prefix (either the 64:ff9b::/96 Well-Known Prefix, or a Network-Specific Prefix) is used as the IPv6 destination for the IPv4-embedded client traffic.

In the operator's network, the provider-side translator (PLAT) performs stateful NAT64 [RFC6146] to translate the traffic. The destination IPv4 address is extracted from the IPv4-embedded IPv6 packet destination address and the source address is from a pool of public IPv4 addresses.

Alternatively, when a dedicated /64 is not available for translation, the CLAT device uses a stateful NAT44 translation before the stateless NAT46 translation.

Note that we generally do not see state close to the end-user as equally problematic as state in the middle of the network.

In typical deployments, 464XLAT is used together with DNS64 [RFC6147], see Section 3.1.2 of [RFC8683]. When an IPv6-only client or application communicates with an IPv4-only server, the DNS64 server returns the IPv4-embedded IPv6 address of the IPv4-only server. In this case, the IPv6-only client sends out IPv6 packets, thus CLAT functions as an IPv6 router and the PLAT performs a stateful NAT64 for these packets. In this case, there is a single translation.

Alternatively, one can say that the DNS64 + stateful NAT64 is used to carry the traffic of the IPv6-only client and the IPv4-only server, and the CLAT is used only for the IPv4 traffic from applications or devices that use literal IPv4 addresses or non-IPv6 compliant APIs.

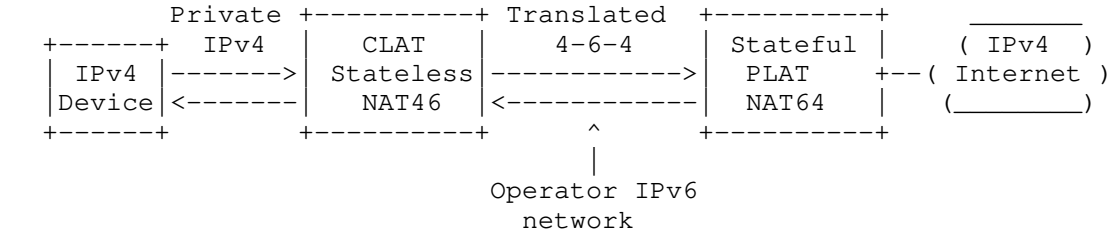

Figure 1: Overview of the 464XLAT architecture

Note: in mobile networks, CLAT is commonly implemented in the user's equipment (UE or smartphone).

2.2. Dual-Stack Lite

Dual-Stack Lite (DS-Lite) [RFC6333] was the first of the considered transition mechanisms to be developed. DS-Lite uses a 'Basic Broadband Bridging' (B4) function in the customer's CE router that encapsulates IPv4 in IPv6 traffic and sends it over the IPv6 native service-provider network to a centralized 'Address Family Transition Router' (AFTR). The AFTR performs encapsulation/decapsulation of the 4in6 traffic and translates the IPv4 payload to public IPv4 source address using a stateful NAPT44 function.

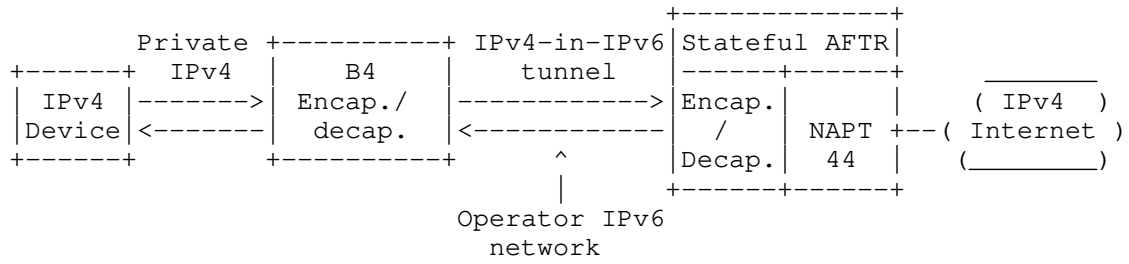

Figure 2: Overview of the DS-Lite architecture

2.3. Lightweight 4over6

Lightweight 4over6 (lw4o6) is a variant of DS-Lite. The main difference is that the stateful NAPT44 function is relocated from the centralized AFTR to the customer's B4 element (called a lwB4). The

AFTR (called a lwAFTR) function therefore only performs A+P routing and 4in6 encapsulation/decapsulation.

Routing to the correct client and IPv4 address sharing is achieved using the Address + Port (A+P) model [RFC6346] of provisioning each lwB4 with a unique tuple of IPv4 address unique range of layer-4 ports. The client uses these for NAPT44.

The lwAFTR implements a binding table, which has a per-client entry linking the customer's source IPv4 address and allocated range of layer-4 ports to their IPv6 tunnel endpoint address. The binding table allows egress traffic from customers to be validated (to prevent spoofing) and ingress traffic to be correctly encapsulated and forwarded. As there needs to be a per-client entry, an lwAFTR implementation needs to be optimized for performing a per-packet lookup on the binding table.

Direct communication between two lwB4s is performed by hair-pinning traffic through the lwAFTR.

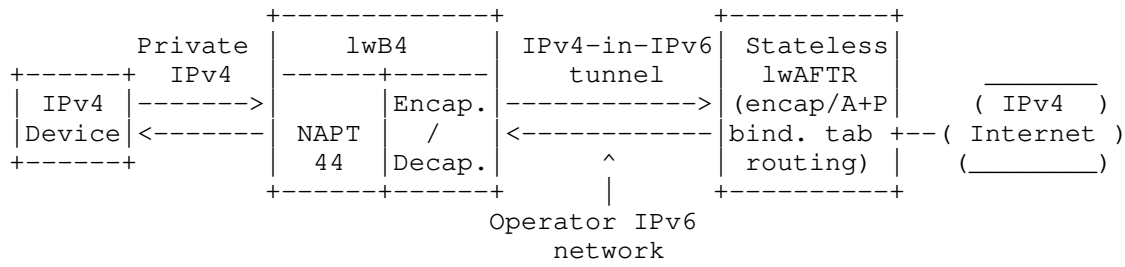

Figure 3: Overview of the lw4o6 architecture

2.4. MAP-E

MAP-E uses a stateless algorithm to embed portions of the customer's allocated IPv4 address (or part of an address with A+P routing) into the IPv6 prefix delegated to the client. This allows for large numbers of clients to be provisioned using a single MAP rule (called a MAP domain). The algorithm also allows for direct IPv4 peer-to-peer communication between hosts provisioned with common MAP rules.

The CE (Customer-Edge) router typically performs stateful NAPT44 [RFC2663] to translate the private IPv4 source addresses and source ports into an address and port range defined by applying the MAP rule applied to the delegated IPv6 prefix. The client address/port allocation size is a design parameter. The CE router then encapsulates the IPv4 packet in an IPv6 packet [RFC2473] and sends it directly to another host in the MAP domain (for peer-to-peer) or to a

Border Router (BR) if the IPv4 destination is not covered in one of the CE's MAP rules.

The MAP BR is provisioned with the set of MAP rules for the MAP domains it serves. These rules determine how the MAP BR is to decapsulate traffic that it receives from client, validating the source IPv4 address and layer 4 ports assigned, as well as how to calculate the destination IPv6 address for ingress IPv4 traffic.

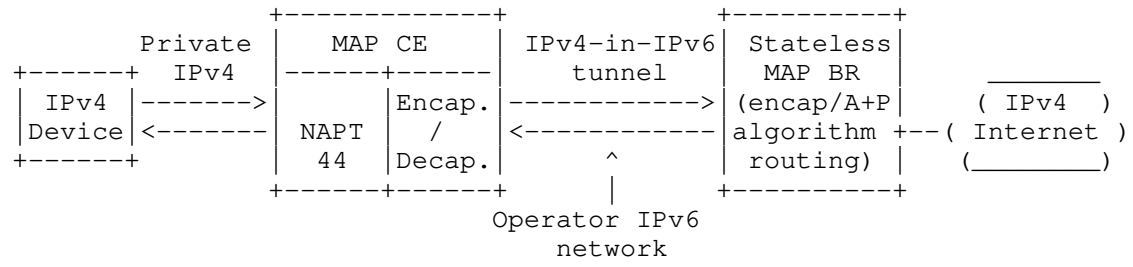

Figure 4: Overview of the MAP-E architecture

2.5. MAP-T

MAP-T uses the same mapping algorithm as MAP-E. The major difference is that double stateless translation (NAT46 in the CE and NAT64 in the BR) is used to traverse the ISP's IPv6 single-stack network. MAP-T can also be compared to 464XLAT when there is a double translation.

A MAP CE typically performs stateful NAPT44 to translate traffic to a public IPv4 address and port-range calculated by applying the provisioned Basic MAP Rule (BMR - a set of inputs to the algorithm) to the delegated IPv6 prefix. The CE then performs stateless translation from IPv4 to IPv6 [RFC7915]. The MAP BR is provisioned with the same BMR as the client, enabling the received IPv6 traffic to be statelessly NAT64 translated back to the public IPv4 source address used by the client.

Using translation instead of encapsulation also allows IPv4-only nodes to correspond directly with IPv6 nodes in the MAP-T domain that have IPv4-embedded IPv6 addresses.

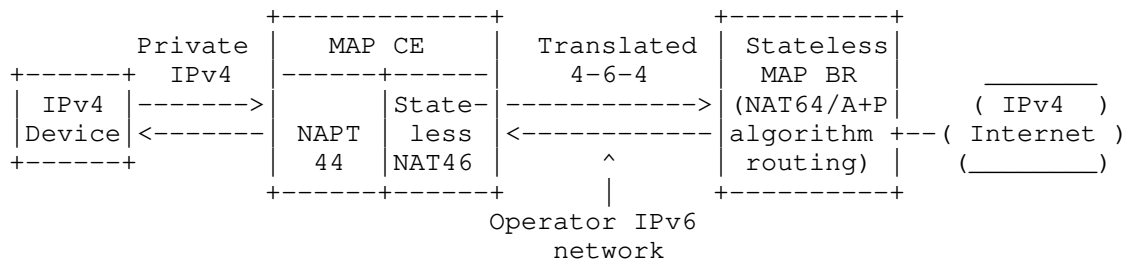

Figure 5: Overview of the MAP-T architecture

3. High-level Architectures and their Consequences

3.1. Service Provider Network Traversal

For the data-plane, there are two approaches for traversing the IPv6 provider network:

- 4-6-4 translation
- 4-in-6 encapsulation

	464XLAT	DS-Lite	lw4o6	MAP-E	MAP-T
4-6-4 trans.	X				X
4-6-4 encap.		X	X	X	

Table 1: Available Traversal Mechanisms

In the scope of this document, all of the encapsulation based mechanisms use IP-in-IP tunnelling [RFC2473]. This is a stateless tunneling mechanism which does not require any additional tunnel headers.

It should be noted that both of these approaches result in an increase in the size of the packet that needs to be transported across the operator's network when compared to native IPv4. 4-6-4 translation adds a 20-bytes overhead (the 20-byte IPv4 header is replaced with a 40-byte IPv6 header). Encapsulation has a 40-byte overhead (an IPv6 header is prepended to the IPv4 header).

The increase in packet size can become a significant problem if there is a link with a smaller MTU in the traffic path. This may result in traffic needing to be fragmented at the ingress point to the IPv6 only domain (i.e., the NAT46 or 4in6 encapsulation endpoint). It may

also result in the need to implement buffering and fragment re-assembly in the BR node.

The advice given in [RFC7597] Section 8.3.1 is applicable to all of these mechanisms: It is strongly recommended that the MTU in the IPv6-only domain be well managed and that the IPv6 MTU on the CE WAN-side interface be set so that no fragmentation occurs within the boundary of the IPv6-only domain.

3.2. Network Address Translation

For the high-level solution of IPv6 service provider network traversal, MAP-T uses double stateless translation. First at the CE from IPv4 to IPv6 (NAT46), and then from IPv6 to IPv4 (NAT64), at the service provider network.

464XLAT may use double translation (stateless NAT46 + stateful NAT64) or single translation (stateful NAT64), depending on different factors, such as the use of DNS by the applications and the availability of a DNS64 function (in the host or in the service provider network). For deployment guidelines, please refer to [RFC8683].

The first step for the double translation mechanisms is a stateless NAT from IPv4 to IPv6 implemented as SIIT (Stateless IP/ICMP Translation Algorithm) [RFC7915], which does not translate IPv4 header options and/or multicast IP/ICMP packets. With encapsulation-based technologies the header is transported intact and multicast can also be carried.

Single and double translation results in native IPv6 traffic with a layer-4 next-header. The fields in these headers can be used for functions such as hashing across equal-cost multipaths or ACLs. For encapsulation, there is an IPv6 header followed by an IPv4 header. This results in less entropy for hashing algorithms, and may mean that devices in the traffic path that perform header inspection (e.g. router ACLs or firewalls) require the functionality to look into the payload header.

Solutions using double translation can only carry port-aware IP protocols (e.g. TCP, UDP) and ICMP when they are used with IPv4 address sharing (please refer to Section 4.3 for more details). Encapsulation based solutions can carry any other protocols over IP, too.

An in-depth analysis of stateful NAT64 can be found in [RFC6889].

3.3. IPv4 Address Sharing

As public IPv4 address exhaustion is a common motivation for deploying IPv6, transition technologies need to provide a solution for allowing public IPv4 address sharing.

In order to fulfill this requirement, a stateful NAPT function is a necessary function in all of the mechanisms. The major differentiator is where in the architecture this function is located.

The solutions compared by this document fall into two categories:

- CGN-based approaches (DS-Lite, 464XLAT)
- A+P-based approaches (lw4o6, MAP-E, MAP-T)

In the CGN-based model, a device such as a CGN/AFTR or NAT64 performs the NAPT44 function and maintains per-session state for all of the active client's traffic. The customer's device does not require per-session state for NAPT.

In the A+P-based model, a device (usually a CE) performs stateful NAPT44 and maintains per-session state only co-located devices, e.g. in the customer's home network. Here, the centralized network function (lwAFTR or BR) only needs to perform stateless encapsulation/decapsulation or NAT64.

Issues related to IPv4 address sharing mechanisms are described in [RFC6269] and should also be considered.

The address sharing efficiency of the five technologies is significantly different, it is discussed in Section 4.2

lw4o6, MAP-E and MAP-T can also be configured without IPv4 address sharing, see the details in Section 4.3. However, in that case, there is no advantage in terms of public IPv4 address saving. In the case of 464XLAT, this can be achieved as well through EAMT [RFC7757].

Conversely, both MAP-E and MAP-T may be configured to provide more than one public IPv4 address (i.e., an IPv4 prefix shorter than a /32) to customers.

Dynamic DNS issues in address-sharing contexts and their possible solutions using PCP (Port Control Protocol) are discussed in detail in [RFC7393].

3.4. CE Provisioning Considerations

All of the technologies require some provisioning of customer devices. The table below shows which methods currently have extensions for provisioning the different mechanisms.

	464XLAT	DS-Lite	lw4o6	MAP-E	MAP-T
DHCPv6 [RFC8415]		X	X	X	X
RADIUS Attr.		X	X	X	X
TR-69		X		X	X
DNS64 [RFC7050]	X				
YANG [RFC7950]	[RFC8512]	X	X	X	X
DHCP4o6			X	X	

Table 2: Available Provisioning Mechanisms

3.5. Support for Multicast

The solutions covered in this document are all intended for unicast traffic. [RFC8114] describes a method for carrying encapsulated IPv4 multicast traffic over an IPv6 multicast network. This could be deployed in parallel to any of the operator's chosen IPv4aaS mechanism.

4. Detailed Analysis

4.1. Architectural Differences

4.1.1. Basic Comparison

The five IPv4aaS technologies can be classified into 2x2=4 categories on the basis of two aspects:

- Technology used for service provider network traversal. It can be single/double translation or encapsulation.
- Presence or absence of NAPT44 per-flow state in the operator network.

	464XLAT	DS-Lite	lw4o6	MAP-E	MAP-T
4-6-4 trans.	X				X
4-in-4 encap.		X	X	X	
Per-flow state in op. network	X	X			

Table 3: Available Provisioning Mechanisms

4.2. Tradeoff between Port Number Efficiency and Stateless Operation

464XLAT and DS-Lite use stateful NAPT at the PLAT/AFTR devices, respectively. This may cause scalability issues for the number of clients or volume of traffic, but does not impose a limitation on the number of ports per user, as they can be allocated dynamically on-demand and the allocation policy can be centrally managed/adjusted.

A+P based mechanisms (Lw4o6, MAP-E, and MAP-T) avoid using NAPT in the service provider network. However, this means that the number of ports provided to each user (and hence the effective IPv4 address sharing ratio) must be pre-provisioned to the client.

Changing the allocated port ranges with A+P based technologies, requires more planning and is likely to involve re-provisioning both hosts and operator side equipment. It should be noted that due to the per-customer binding table entry used by lw4o6, a single customer can be re-provisioned (e.g., if they request a full IPv4 address) without needing to change parameters for a number of customers as in a MAP domain.

It is also worth noting that there is a direct relationship between the efficiency of customer public port-allocations and the corresponding logging overhead that may be necessary to meet data-retention requirements. This is considered in Section 4.7 below.

Determining the optimal number of ports for a fixed port set is not an easy task, and may also be impacted by local regulatory law, which may define a maximum number of users per IP address, and consequently a minimum number of ports per user.

On the one hand, the "lack of ports" situation may cause serious problems in the operation of certain applications. For example, Miyakawa has demonstrated the consequences of the session number limitation due to port number shortage on the example of Google Maps [MIY2010]. When the limit was 15, several blocks of the map were missing, and the map was unusable. This study also provided several

examples for the session numbers of different applications (the highest one was Apple's iTunes: 230-270 ports).

The port number consumption of different applications is highly varying and e.g. in the case of web browsing it depends on several factors, including the choice of the web page, the web browser, and sometimes even the operating system [REP2014]. For example, under certain conditions, 120-160 ports were used (URL: sohu.com, browser: Firefox under Ubuntu Linux), and in some other cases it was only 3-12 ports (URL: twitter.com, browser: Iceweasel under Debian Linux).

There may be several users behind a CE router, especially in the broadband case (e.g. Internet is used by different members of a family simultaneously), so sufficient ports must be allocated to avoid impacting user experience.

Furthermore, assigning too many ports per CE router will result in waste of public IPv4 addresses, which is a scarce and expensive resource. Clearly this is a big advantage in the case of 464XLAT where they are dynamically managed, so that the number of IPv4 addresses for the sharing-pool is smaller while the availability of ports per user don't need to be pre-defined and is not a limitation for them.

There is a direct tradeoff between the optimization of client port allocations and the associated logging overhead. Section 4.7 discusses this in more depth.

We note that common CE router NAT44 implementations utilizing Netfilter, multiplexes active sessions using a 3-tuple (source address, destination address, and destination port). This means that external source ports can be reused for unique internal source and destination address and port sessions. It is also noted, that Netfilter cannot currently make use of multiple source port ranges (i.e. several blocks of ports distributed across the total port space as is common in MAP deployments), this may influence the design when using stateless technologies.

Stateful technologies, 464XLAT and DS-Lite (and also NAT444) can therefore be much more efficient in terms of port allocation and thus public IP address saving. The price is the stateful operation in the service provider network, which allegedly does not scale up well. It should be noticed that in many cases, all those factors may depend on how it is actually implemented.

XXX MEASUREMENTS ARE PLANNED TO TEST IF THE ABOVE IS TRUE. XXX

We note that some CGN-type solutions can allocate ports dynamically "on the fly". Depending on configuration, this can result in the same customer being allocated ports from different source addresses. This can cause operational issues for protocols and applications that expect multiple flows to be sourced from the same address. E.g., ECMP hashing, STUN, gaming, content delivery networks. However, it should be noticed that this is the same problem when a network has a NAT44 with multiple public IPv4 addresses, or even when applications in a dual-stack case, behave wrongly if happy eyeballs is flapping the flow address between IPv4 and IPv6.

The consequences of IPv4 address sharing [RFC6269] may impact all five technologies. However, when ports are allocated statically, more customers may get ports from the same public IPv4 address, which may results in negative consequences with higher probability, e.g. many applications and service providers (Sony PlayStation Network, OpenDNS, etc.) permanently black-list IPv4 ranges if they detect that they are used for address sharing.

Both cases are, again, implementation dependent.

We note that although it is not of typical use, one can do deterministic, stateful NAT and reserve a fixed set of ports for each customer, as well.

4.3. Support for Public Server Operation

Mechanisms that rely on operator side per-flow state do not, by themselves, offer a way for customers to present services on publicly accessible layer-4 ports.

Port Control Protocol (PCP) [RFC6877] provides a mechanism for a client to request an external public port from a CGN device. For server operation, it is required with NAT64/464XLAT, and it is supported in some DS-Lite AFTR implementations.

A+P based mechanisms distribute a public IPv4 address and restricted range of layer-4 ports to the client. In this case, it is possible for the user to configure their device to offer a publicly accessible server on one of their allocated ports. It should be noted that commonly operators do not assign the Well-Known-Ports to users (unless they are allocating a full IPv4 address), so the user will need to run the service on an allocated port, or configure port translation.

Lw4o6, MAP-E and MAP-T may be configured to allocated clients with a full IPv4 address, allowing exclusive use of all ports, and non-port-based layer 4 protocols. Thus, they may also be used to support

server/services operation on their default ports. However, when public IPv4 addresses are assigned to the CE router without address sharing, obviously there is no advantage in terms of IPv4 public addresses saving.

It is also possible to configure specific ports mapping in 464XLAT/ NAT64 using EAMT [RFC7757], which means that only those ports are "lost" from the pool of addresses, so there is a higher maximization of the total usage of IPv4/port resources.

4.4. Support and Implementations

4.4.1. OS Support

A 464XLAT client (CLAT) is implemented in Windows 10, Linux (including Android), Windows Mobile, Chrome OS and iOS, but at the time of writing is not available in MacOS.

The remaining four solutions are commonly deployed as functions in the CE device only, however in general, except DS-Lite, the vendors support is poor.

The OpenWRT Linux based open-source OS designed for CE devices offers a number of different 'opkg' packages as part of the distribution:

- '464xlat' enables support for 464XLAT CLAT functionality
- 'ds-lite' enables support for DSLite B4 functionality
- 'map' enables support for MAP-E and lw4o6 CE functionality
- 'map-t' enables support for MAP-T CE functionality

For the operator side functionality, some free open-source implementations exist:

CLAT, NAT64, EAMT: http://www.jool.mx

MAP-BR, lwAFTR, CGN, CLAT, NAT64: VPP/fd.io
https://gerrit.fd.io/r/#/admin/projects/

lwAFTR: https://github.com/Igalia/snabb

DSLite AFTR: https://www.isc.org/downloads/

4.4.2. Support in Cellular and Broadband Networks

Several cellular networks use 464XLAT, whereas we are not aware of any deployment of the four other technologies in cellular networks, as they are not implemented in UE devices.

In broadband networks, there are some deployments of 464XLAT, MAP-E and MAP-T. Lw4o6 and DS-Lite have more deployments, with DS-Lite being the most common, but lw4o6 taking over in the last years.

Please refer to Table 2 and Table 3 of [LEN2019] for a limited set of deployment information.

4.4.3. Implementation Code Sizes

As hint to the relative complexity of the mechanisms, the following code sizes are reported from the OpenWRT implementations of each technology are 17kB, 35kB, 15kB, 35kB, and 48kB for 464XLAT, lw4o6, DS-Lite, MAP-E, MAP-T, and lw4o6, respectively (https://openwrt.org/packages/start).

We note that the support for all five technologies requires much less code size than the total sum of the above quantities, because they contain a lot of common functions (data plane is shared among several of them).

4.5. Typical Deployment and Traffic Volume Considerations

4.5.1. Deployment Possibilities

Theoretically, all five IPv4aaS technologies could be used together with DNS64 + stateful NAT64, as it is done in 464XLAT. In this case the CE router would treat the traffic between an IPv6-only client and IPv4-only server as normal IPv6 traffic, and the stateful NAT64 gateway would do a single translation, thus offloading this kind of traffic from the IPv4aaS technology. The cost of this solution would be the need for deploying also DNS64 + stateful NAT64.

However, this has not been implemented in clients or actual deployments, so only 464XLAT always uses this optimization and the other four solutions do not use it at all.

4.5.2. Cellular Networks with 464XLAT

Actual figures from existing deployments, show that the typical traffic volumes in an IPv6-only cellular network, when 464XLAT technology is used together with DNS64, are:

- 75% of traffic is IPv6 end-to-end (no translation)
- 24% of traffic uses DNS64 + NAT64 (1 translation)
- Less than 1% of traffic uses the CLAT in addition to NAT64 (2 translations), due to an IPv4 socket and/or IPv4 literal.

Without using DNS64, 25% of the traffic would undergo double translation.

4.6. Load Sharing

If multiple network-side devices are needed as PLAT/AFTR/BR for capacity, then there is a need for a load sharing mechanism. ECMP (Equal-Cost Multi-Path) load sharing can be used for all technologies, however stateful technologies will be impacted by changes in network topology or device failure.

Technologies utilizing DNS64 can also distribute load across PLAT/AFTR devices, evenly or unevenly, by using different prefixes. Different network specific prefixes can be distributed for subscribers in appropriately sized segments (like split-horizon DNS, also called DNS views).

Stateless technologies, due to the lack of per-flow state, can make use of anycast routing for load sharing and resiliency across network-devices, both ingress and egress; flows can take asymmetric paths through the network, i.e., in through one lwAFTR/BR and out via another.

Mechanisms with centralized NAPT44 state have a number of challenges specifically related to scaling and resilience. As the total amount of client traffic exceeds the capacity of a single CGN instance, additional nodes are required to handle the load. As each CGN maintains a stateful table of active client sessions, this table may need to be syncronized between CGN instances. This is necessary for two reasons:

- To prevent all active customer sessions being dropped in event of a CGN node failure.
- To ensure a matching state table entry for an active session in the event of asymmetric routing through different egress and ingress CGN nodes.

4.7. Logging

In the case of 464XLAT and DS-Lite, the user of any given public IPv4 address and port combination will vary over time, therefore, logging is necessary to meet data retention laws. Each entry in the PLAT/ AFTR's generates a logging entry. As discussed in Section 4.2, a client may open hundreds of sessions during common tasks such as web-browsing, each of which needs to be logged so the overall logging burden on the network operator is significant. In some countries, this level of logging is required to comply with data retention legislation.

One common optimization available to reduce the logging overhead is the allocation of a block of ports to a client for the duration of their session. This means that logging entry only needs to be made when the client's port block is released, which dramatically reducing the logging overhead. This comes as the cost of less efficient public address sharing as clients need to be allocated a port block of a fixed size regardless of the actual number of ports that they are using.

Stateless technologies that pre-allocate the IPv4 addresses and ports only require that copies of the active MAP rules (for MAP-E and MAP-T), or binding-table (for lw4o6) are retained along with timestamp information of when they have been active. Support tools (e.g., those used to serve data retention requests) may need to be updated to be aware of the mechanism in use (e.g., implementing the MAP algorithm so that IPv4 information can be linked to the IPv6 prefix delegated to a client). As stateless technologies do not have a centralized stateful element which customer traffic needs to pass through, so if data retention laws mandate per-session logging, there is no simple way of meeting this requirement with a stateless technology alone. Thus a centralized NAPT44 model may be the only way to meet this requirement.

Deterministic CGN [RFC7422] was proposed as a solution to reduce the resource consumption of logging.

4.8. Optimization for IPv4-only devices/applications

When IPv4-only devices or applications are behind a CE connected with IPv6-only and IPv4aaS, the IPv4-only traffic flows will necessarily, be encapsulated/decapsulated (in the case of DS-Lite, lw4o6 and MAP-E) and will reach the IPv4 address of the destination, even if that service supports dual-stack. This means that the traffic flow will cross thru the AFTR, lwAFTR or BR, depending on the specific transition mechanism being used.

Even if those services are directly connected to the operator network (for example, CDNs, caches), or located internally (such as VoIP, etc.), it is not possible to avoid that overhead.

However, in the case of those mechanism that use a NAT46 function, in the CE (464XLAT and MAP-T), it is possible to take advantage of optimization functionalities, such as the ones described in [I-D.ietf-v6ops-464xlat-optimization].

Using those optimizations, because the NAT46 has already translated the IPv4-only flow to IPv6, and the services are dual-stack, they can be reached without the need to translate them back to IPv4.

5. Performance Comparison

We plan to compare the performances of the most prominent free software implementations of the five IPv6 transition technologies using the methodology described in "Benchmarking Methodology for IPv6 Transition Technologies" [RFC8219].

The Dual DUT Setup of [RFC8219] makes it possible to use the existing "Benchmarking Methodology for Network Interconnect Devices" [RFC2544] compliant measurement devices, however, this solution has two kinds of limitations:

- Dual DUT setup has the drawback that the performances of the CE and of the ISP side device (e.g. the CLAT and the PLAT of 464XLAT) are measured together. In order to measure the performance of only one of them, we need to ensure that the desired one is the bottleneck.

- Measurements procedures for PDV and IPDV measurements are missing from the legacy devices, and the old measurement procedure for Latency has been redefined in [RFC8219].

The Single DUT Setup of [RFC8219] makes it possible to benchmark the selected device separately, but it either requires a special Tester or some trick is need, if we want to use legacy Testers. An example for the latter is our stateless NAT64 measurements testing Througput and Frame Loss Rate using a legacy [RFC5180] compliant commercial tester [LEN2020a]

Siitperf, an [RFC8219] compliant DPDK-based software Tester for benchmarking stateless NAT64 gateways has been developed recently and it is available from GitHub [SIITperf] as free software and documented in [LEN2021]. Originally, it literally followed the test frame format of [RFC2544] including "hard wired" source and destination port numbers, and then it has been complemented with the

random port feature required by [RFC4814]. The new version is documented in [LEN2020b]

- It can be used for benchmarking both the CLAT and PLAT of 464XLAT separately, according to the single DUT setup. (We note that the benchmarking prodedures for stateful NAT64 include the stateless tests, plus a few additional tests, which are not implemented yet.)

- It can also be used for benchmarking all five IPv4-as-a-Service technologies according to the Dual DUT setup, because it supports the usage of IPv4 on its both sides, too.

Another software tester for benchmaring the B4 and AFTR components of DS-Lite is currently being developed at the Budapest University of Technology and Economics as a student project. It is planned to be released as free software later this year.

We plan to start an intesive benchmaking campaign using the resources of NICT StarBED, Japan.

6. Acknowledgements

The authors would like to thank Ole Troan for his thorough review of this draft and acknowledge the inputs of Mark Andrews, Edwin Cordeiro, Fred Baker, Alexandre Petrescu, Cameron Byrne, Tore Anderson, Mikael Abrahamsson, Gert Doering, Satoru Matsushima, Mohamed Boucadair, Tom Petch, Yannis Nikolopoulos, and TBD ...

7. IANA Considerations

This document does not make any request to IANA.

8. Security Considerations

According to the simplest model, the number of bugs is proportional to the number of code lines. Please refer to Section 4.4.3 for code sizes of CE implementations.

For all five technologies, the CE device should contain a DNS proxy. However, the user may change DNS settings. If it happens and lw4o6, MAP-E and MAP-T are used with significantly restricted port set, which is required for an efficient public IPv4 address sharing, the entropy of the source ports is significantly lowered (e.g. from 16 bits to 10 bits, when 1024 port numbers are assigned to each subscriber) and thus these technologies are theoretically less resilient against cache poisoning, see [RFC5452]. However, an efficient cache poisoning attack requires that the subscriber

operates an own caching DNS server and the attack is performed in the service provider network. Thus, we consider the chance of the successful exploitation of this vulnerability as low.

An in-depth security analysis of all five IPv6 transition technologies and their most prominent free software implementations according to the methodology defined in [LEN2018] is planned.

As the first step, the theoretical security analysis of 464XLAT was done in [Azz2020].

9. References

9.1. Normative References

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

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <https://www.rfc-editor.org/info/rfc2544>.

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

[RFC4814] Newman, D. and T. Player, "Hash and Stuffing: Overlooked Factors in Network Device Benchmarking", RFC 4814, DOI 10.17487/RFC4814, March 2007, <https://www.rfc-editor.org/info/rfc4814>.

[RFC5180] Popoviciu, C., Hamza, A., Van de Velde, G., and D. Dugatkin, "IPv6 Benchmarking Methodology for Network Interconnect Devices", RFC 5180, DOI 10.17487/RFC5180, May 2008, <https://www.rfc-editor.org/info/rfc5180>.

[RFC5452] Hubert, A. and R. van Mook, "Measures for Making DNS More Resilient against Forged Answers", RFC 5452, DOI 10.17487/RFC5452, January 2009, <https://www.rfc-editor.org/info/rfc5452>.

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

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

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <https://www.rfc-editor.org/info/rfc6147>.

[RFC6180] Arkko, J. and F. Baker, "Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment", RFC 6180, DOI 10.17487/RFC6180, May 2011, <https://www.rfc-editor.org/info/rfc6180>.

[RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, DOI 10.17487/RFC6269, June 2011, <https://www.rfc-editor.org/info/rfc6269>.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <https://www.rfc-editor.org/info/rfc6333>.

[RFC6346] Bush, R., Ed., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, DOI 10.17487/RFC6346, August 2011, <https://www.rfc-editor.org/info/rfc6346>.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <https://www.rfc-editor.org/info/rfc6877>.

[RFC6889] Penno, R., Saxena, T., Boucadair, M., and S. Sivakumar, "Analysis of Stateful 64 Translation", RFC 6889, DOI 10.17487/RFC6889, April 2013, <https://www.rfc-editor.org/info/rfc6889>.

[RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <https://www.rfc-editor.org/info/rfc7050>.

[RFC7393] Deng, X., Boucadair, M., Zhao, Q., Huang, J., and C. Zhou, "Using the Port Control Protocol (PCP) to Update Dynamic DNS", RFC 7393, DOI 10.17487/RFC7393, November 2014, <https://www.rfc-editor.org/info/rfc7393>.

[RFC7422] Donley, C., Grundemann, C., Sarawat, V., Sundaresan, K., and O. Vautrin, "Deterministic Address Mapping to Reduce Logging in Carrier-Grade NAT Deployments", RFC 7422, DOI 10.17487/RFC7422, December 2014, <https://www.rfc-editor.org/info/rfc7422>.

[RFC7596] Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture", RFC 7596, DOI 10.17487/RFC7596, July 2015, <https://www.rfc-editor.org/info/rfc7596>.

[RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <https://www.rfc-editor.org/info/rfc7597>.

[RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July 2015, <https://www.rfc-editor.org/info/rfc7599>.

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

[RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <https://www.rfc-editor.org/info/rfc7915>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8114] Boucadair, M., Qin, C., Jacquenet, C., Lee, Y., and Q. Wang, "Delivery of IPv4 Multicast Services to IPv4 Clients over an IPv6 Multicast Network", RFC 8114, DOI 10.17487/RFC8114, March 2017, <https://www.rfc-editor.org/info/rfc8114>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8219] Georgescu, M., Pislaru, L., and G. Lencse, "Benchmarking Methodology for IPv6 Transition Technologies", RFC 8219, DOI 10.17487/RFC8219, August 2017, <https://www.rfc-editor.org/info/rfc8219>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

[RFC8512] Boucadair, M., Ed., Sivakumar, S., Jacquenet, C., Vinapamula, S., and Q. Wu, "A YANG Module for Network Address Translation (NAT) and Network Prefix Translation (NPT)", RFC 8512, DOI 10.17487/RFC8512, January 2019, <https://www.rfc-editor.org/info/rfc8512>.

[RFC8683] Palet Martinez, J., "Additional Deployment Guidelines for NAT64/464XLAT in Operator and Enterprise Networks", RFC 8683, DOI 10.17487/RFC8683, November 2019, <https://www.rfc-editor.org/info/rfc8683>.

9.2. Informative References

[Azz2020] Al-Azzawi, A. and G. Lencse, "Towards the Identification of the Possible Security Issues of the 464XLAT IPv6 Transition Technology", 43rd International Conference on Telecommunications and Signal Processing (TSP 2020), Milan, Italy, 10.1109/TSP49548.2020.9163487, Jul 2020, <http://www.hit.bme.hu/~lencse/publications/TSP-2020-464XLAT-revised.pdf>.

[I-D.ietf-v6ops-464xlat-optimization] Martinez, J. and A. D'Egidio, "464XLAT/MAT-T Optimization", draft-ietf-v6ops-464xlat-optimization-03 (work in progress), July 2020.

[LEN2018] Lencse, G. and Y. Kadobayashi, "Methodology for the identification of potential security issues of different IPv6 transition technologies: Threat analysis of DNS64 and stateful NAT64", Computers & Security (Elsevier), vol. 77, no. 1, pp. 397-411, DOI: 10.1016/j.cose.2018.04.012, Aug 2018, <http://www.hit.bme.hu/~lencse/publications/ECS-2018-Methodology-revised.pdf>.

[LEN2019] Lencse, G. and Y. Kadobayashi, "Comprehensive Survey of IPv6 Transition Technologies: A Subjective Classification for Security Analysis", IEICE Transactions on Communications, vol. E102-B, no.10, pp. 2021-2035., DOI: 10.1587/transcom.2018EBR0002, Oct 2019, <http://www.hit.bme.hu/~lencse/publications/e102-b_10_2021.pdf>.

[LEN2020a] Lencse, G., "Benchmarking Stateless NAT64 Implementations with a Standard Tester", Telecommunication Systems, vol. 75, pp. 245-257, DOI: 10.1007/s11235-020-00681-x, Jun 2020, <http://www.hit.bme.hu/~lencse/publications/Lencse2020_Article_BenchmarkingStatelessNAT64Impl.pdf>.

[LEN2020b] Lencse, G., "Adding RFC 4814 Random Port Feature to Siitperf: Design, Implementation and Performance Estimation", International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems, vol 9, no 3, pp. 18-26, DOI: 10.11601/ijates.v9i3.291, 2020, <http://www.hit.bme.hu/~lencse/publications/291-1113-1-PB.pdf>.

[LEN2021] Lencse, G., "Design and Implementation of a Software Tester for Benchmarking Stateless NAT64 Gateways", IEICE Transactions on Communications, DOI: 10.1587/transcom.2019EBN0010, 2021, <http://www.hit.bme.hu/~lencse/publications/IEICE-2020-siitperf-revised.pdf>.

[MIY2010] Miyakawa, S., "IPv4 to IPv6 transformation schemes", IEICE Trans. Commun., vol.E93-B, no.5, pp. 1078-1084, DOI:10.1587/transcom.E93.B.10, May 2010, <https://www.jstage.jst.go.jp/article/transcom/E93.B/5/E93.B_5_1078/_article>.

[REP2014] Repas, S., Hajas, T., and G. Lencse, "Port number consumption of the NAT64 IPv6 transition technology", Proc. 37th Internat. Conf. on Telecommunications and Signal Processing (TSP 2014), Berlin, Germany, DOI: 10.1109/TSP.2015.7296411, July 2014.

[SIITperf] Lencse, G., "Siitperf: an RFC 8219 compliant SIIT (stateless NAT64) tester", November 2019, <https://github.com/lencsegabor/siitperf>.

Appendix A. Change Log

A.1. 01 - 02

- Ian Farrer has joined us as an author.
- Restructuring: the description of the five IPv4aaS technologies was moved to a separate section.
- More details and figures were added to the description of the five IPv4aaS technologies.
- Section titled "High-level Architectures and their Consequences" has been completely rewritten.
- Several additions/clarification throughout Section titled "Detailed Analysis".
- Section titled "Performance Analysis" was dropped due to lack of results yet.
- Word based text ported to XML.
- Further text cleanups, added text on state sync and load balancing. Additional comments inline that should be considered for future updates.

A.2. 02 - 03

- The suggestions of Mohamed Boucadair are incorporated.
- New considerations regarding possible optimizations.

A.3. 03 - 04

- Section titled "Performance Analysis" was added. It mentions our new benchmarking tool, siitperf, and highlights our plans.
- Some references were updated or added.

A.4. 04 - 05

- Some references were updated or added.

A.5. 05 - 06

- Some references were updated or added.

Authors' Addresses

Gabor Lencse
Budapest University of Technology and Economics
Magyar Tudosok korutja 2.
Budapest H-1117
Hungary

Email: lencse@hit.bme.hu

Jordi Palet Martinez
The IPv6 Company
Molino de la Navata, 75
La Navata - Galapagar, Madrid 28420
Spain

Email: jordi.palet@theipv6company.com
URI: http://www.theipv6company.com/

Lee Howard
Retevia
9940 Main St., Suite 200
Fairfax, Virginia 22031
USA

Email: lee@asgard.org

Richard Patterson
Sky UK
1 Brick Lane
London EQ 6PU
United Kingdom

Email: richard.patterson@sky.uk

Ian Farrer
Deutsche Telekom AG
Landgrabenweg 151
Bonn 53227
Germany

Email: ian.farrer@telekom.de

IPv6 Operations (v6ops)
Internet-Draft
Obsoletes: 6177 (if approved)
Intended status: Best Current Practice
Expires: April 12, 2019

J. Palet Martinez
The IPv6 Company
L. Roberts
Stanford University, UIT

October 9, 2018

IPv6 Address Assignment to End-Sites
draft-palet-v6ops-rfc6177-bis-02

Abstract

The Regional Internet Registries (RIRs) policies have different views regarding the recommendation of the prefix to be assigned to end-sites. However, all them allow up to a /48 without further justification and clearly state that the exact choice of how much address space should be assigned to end-sites is a decision of each operator.

This document reviews the architectural and operational considerations of end-site assignments, and reiterates that assignment policy and guidelines belong to the RIR community. This revision is being made to emphasize that IPv6 protocol evolution requires an ever-increasing availability of subnets at the end-site, so policy should reflect that assignment of a single subnet is never recommended.

This document obsoletes RFC6177 (IPv6 Address Assignment to End Sites).

Table of Contents

1. Introduction

There are a number of considerations that factor into address and prefix assignment policies. For example, to provide for the long-term health and scalability of the public routing infrastructure, it is important that prefixes aggregate well [Route-Scaling]. Likewise, giving out an excessive amount of address space could result in premature depletion of the address space. This document focuses on the (narrower) question of what is an appropriate IPv6 address assignment size for end-sites. That is, when end-sites request IPv6 address space from ISPs, what is an appropriate assignment size.

[RFC3177] called for a default end-site IPv6 assignment size of /48. Subsequently, the Regional Internet Registries (RIRs) developed and adopted IPv6 address assignment and allocation policies consistent with ISP practices, and it triggered the development of [RFC6177]. Current RIR policies still allow using /48, but leave the decision in the hands of the ISP. In some cases, encourage the assignment /48 blocks for all, while other RIRs encourage the assignment of smaller (e.g., /56) blocks to residential end-sites, while keeping /48 for business.

More recently, a Global IPv6 Deployment Survey (for residential/ household services) has been conducted since May 2016, with responses from over 1.559 ISPs, from 105 different countries, which latest results have been presented in January 2018 [IPv6-Survey]. This survey is showing that 23% of the responders (generally in more advanced countries, in terms of IPv6 deployment) assign a /48, 35% assign a /56 and 33% assign a single /64.

This raises the question of over-zealous interpretation of [RFC6177] by at least one third of the ISP community and consequently, the need to revisit it.

This document obsoletes [RFC6177], updating its recommendations in the following ways:

1. It is extremely discouraged that /128s are assigned. While there may be some applications where assigning only a single address may be tolerated (e.g., an IoT object), a site, by definition, implies multiple subnets and multiple devices. Also, a /128 prevents any form of privacy-based addressing.

2. [RFC3177] specifically recommended using prefix lengths of /48, /64, and /128. Specifying a small number of fixed boundaries has raised concerns that implementations and operational practices might become "hard-coded" to recognize only those fixed boundaries (i.e., a return to "classful addressing"). The actual intention has always been that there must be no hard-coded boundaries within addresses, and that Classless Inter-Domain Routing (CIDR) continues to apply to all bits of the routing prefixes [RFC7608].

3. [RFC6177] moved away from the previous recommendation that a single default assignment size (e.g., a /48) makes sense for all end-sites in the general case. End-sites come in different shapes and sizes, and a one-size-fits-all approach is not necessary or appropriate.

4. This revision has been created to more clearly assert the requirement to ensure that address assignments to end-sites provide a sufficiently big number of subnets (/64 on classic networks) to each end-site, taking under consideration the end-site’s future expected needs, new deployment expectations and new protocol requirements, among others. Once all these are considered, it seems unlikely that a single subnet (/64) or even a small number of them should be assigned.

This document reaffirms, as [RFC6177] did, an important assumption behind [RFC3177], using however, a much stronger and clearer

language:

> A key principle for address management is that end-sites always be able to obtain a reasonable number of /64 subnets for their actual and planned usage, and over time ranges specified in many years, probably decades, rather than just months. In practice, that means that a single /64 subnet is not a choice, even for a small residential network, which following technology trends will need a sufficiently big number of /64 subnets. One particular situation that must be avoided is having an end-site feel compelled to use IPv6-to-IPv6 Network Address Translation or other burdensome address conservation techniques (e.g., [RFC7278]) because it could not get sufficient address space.

This document does not make a formal recommendation on what the exact assignment size should be, beyond what has been indicated in the precedent paragraph. The exact choice of how much address space to assign end-sites is an operational issue and under that context, discussed already in [RIPE-690].

The IETF's role in this case is limited to providing guidance on IPv6 architectural and operational considerations. This document provides input into those discussions. The focus of this document is to examine the architectural issues and some of the operational considerations relating to the size of the end-site assignment.

2. Considerations Regarding the Prefix Length

[RFC7608] already discusses about the IPv6 prefix length recommendations for forwarding, and the need for routing and forwarding implementations to ensure that longest-prefix-match works on any prefix length.

Any prefix length up to /128 is treated identically by routing protocols, however for a given network, end-site, subnet or link, there always exists a Longest Acceptable Prefix (LAP, [I-D.carpenter-6man-lap]), whose length is locally determined, e.g., a site or link that uses SLAAC has a LAP of /64 and will not work with a longer one.

This consideration should be noticed, across this document, in the sense that end-sites usually have subnets that use, by default, SLAAC, and consequently, the LAP is mandatorily a /64. Future technologies, may have a different LAP, which must be used accordingly.

3. On /48 Assignments to End-Sites

Looking back at some of the original motivations behind the /48 recommendation [RFC3177], there were three main concerns. The first motivation was to ensure that end-sites could easily obtain sufficient address space without having to "jump through hoops" to do so. For example, if someone felt they needed more space, just the act of asking would at some level be sufficient justification. As a comparison point, in IPv4, typical home users are given a single public IP address (though even this is not always assured), but getting any more than one address is often difficult or even impossible -- unless one is willing to pay a (significantly) increased fee for what is often considered to be a "higher grade" of service. It should be noted that increased ISP charges to obtain a small number of additional addresses cannot usually be justified by the real per-address cost levied by RIRs, but additional addresses are frequently only available to end users as part of a different type or "higher grade" of service, for which an additional charge is levied. The point here is that the additional cost is not due to the RIR fee structures, but to business choices ISPs make. An important goal in IPv6 is to significantly change the default and minimal end site assignment, from "a single address" to "multiple networks" and to ensure that end-sites can easily obtain address space.

Furthermore, as the operational costs of carrier-grade NAT and address+port sharing have shown, availability of multiple addresses and prefixes to end-sites will be a considerable saving to their ISPs.

It might be tempting to give home sites a single /64, since that is already significantly more address space compared with today’s IPv4 practice. However, this precludes the certainty that even home sites will grow to support multiple subnets going forward, as expected by the "IPv6 Home Networking Architecture" [RFC7368]. Hence, it is strongly intended that even home sites be given a big number of subnets worth of space, by default. Hence, this document still recommends giving home sites significantly many more than a single /64.

A second motivation behind the original /48 recommendation was to simplify the management of an end-site’s addressing plan in the presence of renumbering (e.g., when switching ISPs). In IPv6, a site may simultaneously use multiple prefixes, including one or more public prefixes from ISPs as well as Unique Local Addresses [RFC4193]. In the presence of multiple prefixes, it is significantly less complex to manage a numbering plan if the same subnet numbering plan can be used for all prefixes. That is, for a link that has (say) three different prefixes assigned to it, the subnet portion of

those prefixes would be identical for all assigned addresses. In contrast, renumbering from a larger set of "subnet bits" into a smaller set is often painful, as it can require making changes to the network itself (e.g., collapsing subnets). Hence, renumbering a site into a prefix that has (at least) the same number of subnet bits is more straightforward, because only the top-level bits of the address need to change. A key goal of the recommendations in [RFC3177] was to ensure that upon renumbering, one does not have to deal with renumbering into a smaller subnet size. In particular, this would apply to any site that switches to an ISP that provides a longer prefix.

It should be noted that similar arguments apply to the management of zone files in the DNS. In particular, managing the reverse (ip6.arpa) tree is simplified when all links are renumbered using the same subnet ids.

Furthermore, to keep addressing plans usable and understandable, and to align with DNS reverse zone delegations, the size of the assigned prefix should be aligned with a nibble boundary. Each hexadecimal character in an IPv6 prefix represents one nibble, which is 4 bits. The length of a delegated prefix should therefore always be a multiple of 4, so the possible choices are /48, /52, /56 and /60.

A third motivation behind the /48 recommendation was to better support network growth common at many sites. In IPv4, it is usually difficult (or impossible) to obtain public address space for more than a few months' worth of projected growth. Thus, even slow growth over several years can lead to the need to renumber into a larger address block. With IPv6's vast address space, end-sites can easily be given more address space (compared with IPv4) to support expected growth over multi-year time periods.

While the /48 recommendation does simplify address space management for end-sites, it has also been widely criticized as being wasteful. While reasonable people may disagree over whether all end sites should get a /48 assignment by default, reasonable people do agree that an end-site should be able to get up to a /48 by request. It is important to stress that the strength of IPv6 is the vast size of its address space, which should allow users to easily acquire as many subnets as required for their applications, plus room to grow. Math's show that even assuming 32 billion of humans (2^35), and assigning each of them 4 /48's, with a 50% routing efficiency, we can repeat that 2^10 times, so if average life span of each human is 100 years, and we don't recover back the /48's, we will be able to use IPv6 addressing space for over 102.400 years.

This document does not advocate careless use of address space, but

shows that there is objectively no reason to be restrictive. It is important to leave behind the mind set of IPv4 address scarcity and embrace the wealth of IPv6 address abundance.

A large business (which may have thousands of employees) would, by default, receive the same amount of address space, for each of its end-sites, as a home user. However, it is clear that that for each corporate end-site it is perfectly feasible to justify further needs if that becomes the case, and RIR policies allow that.

Today typically, a home has already a considerable number of possible subnets (a common CE has 4 LAN ports, 2 WiFi radios which support several SSIDs each one, VoIP subnet, IPTV subnet, additional VLANs) and if downstream routers are used, there is a need for further subnets. This means that in a short term, assigning a /60 (16 subnets), it is already a really bad decision.

It will become very common that homes start using technologies like HNCP [RFC7788], and this increases the need for more subnets, which means that /56 (256 subnets) may be too short also in very few years.

Finally, considerations about multiple addresses per host [RFC7934] and techniques to allow a single /64 per host/interface [RFC7934], means that we will see in the short term, many home devices that will take advantage of that, either for security reasons, or because they may need to run internally multiple virtual machines, or many other reasons, which will again, push the limit of the regular home needs, beyond the /56, and consequently, suggesting that /48 is a smarter choice.

The above-mentioned goals of [RFC3177] could be met by giving home users a default assignment of less than /48, such as a /56, as suggested in [RFC6177], however, there are new motivations and technologies, to reconsider that a /48 is a much better choice.

4. Summary

The exact choice of how much address space to assign end-sites is an issue for the operational community, discussed with much more detail in a recent document [RIPE-690].

While the recommendation to assign /48s as a default, is not a requirement of the IPv6 architecture and anything of length /64 or shorter works from a standards perspective, there are important operational considerations as well, some of which are important if users are to share in the key benefit of IPv6: expanding the usable address space of the Internet.

The IETF recommends that any policy on IPv6 address assignment to end-sites take into consideration the following:

a. It should be easy and inexpensive for an end-site to obtain address space to deploy a sufficiently big number of subnets (i.e., a big number of /64's) and to support reasonable growth projections over long time periods (e.g., more than a decade).

b. An end-user should be able to receive any prefix length up to /48 simply by asking. it is critical that the community shed the restrictive view of IP addresses that grew up during the end of IPv4. IPv6 addresses should be freely available, not a tiered cost structure.

c. The default assignment size should take into consideration the likelihood that an end-site will have need for multiple subnets in the near future and many more in a medium and longer terms, avoiding the IPv4 practice of having frequent and continual justification for obtaining small amounts of additional space.

d. Although a /64 can (in theory) address an almost unlimited number of devices, end-sites should be given sufficient address space to be able to lay out subnets as appropriate, and not be forced to use address conservation techniques such as using bridging, NAT, proxy or other techniques. Whether or not those techniques are an appropriate choice is an end-site matter.

e. Assigning a longer prefix to an end-site, compared with the existing prefixes the end-site already has assigned to it, is likely to increase operational costs and complexity for the end-site, with insufficient benefit to anyone.

f. The operational considerations of managing and delegating the reverse DNS tree under ip6.arpa on nibble versus non-nibble boundaries should be given adequate consideration.

As a consequence, it is strongly discouraged to assign to end-sites a single /64 or even a reduced number of them.

Instead, it is strongly suggested considering a /48, or alternatively, a trade-off choice is to reserve a /48 for each end-site, and actually assign them only the first /56, so in the future renumbering is not needed and either in a case by case, by demand, or across all the network, the complete /48 can be re-assigned to each end-site.

The considerations of this document are meant mainly for end-sites, regardless of being connected by cellular, wired or other

technologies. They don’t apply to single cellular devices such as smartphones, which typically will get a single /64 for each connection (PDP context). Even in the case of a handset, assigning a single /64 may require some hacks on the handset, for example to support tethering. This is typically the reason [RFC7849] argued for aligning prefix assignments practices in both cellular and wired networks. Indeed, cellular networks are more and more perceived as an alternative to non-cellular networks for home IP-based services delivery; especially with the advent of 3GPP data dongles. There is a need for an efficient mechanism to assign larger prefixes to cellular hosts so that each LAN segment can get its own /64 prefix and multi-link subnet issues to be avoided. The support of this functionality in both cellular and non-cellular networks is key for fixed-mobile convergence.

A broadband cellular router connecting an end-site falls within the scope of this document.

5. Security Considerations

A shorter prefix has more potential space for scanning attacks, which would require more time, especially if the subnets and hosts are "sparsely assigned".

More prefix space allows the use of slightly randomized prefixes and/or prefix-per-host [RFC8273].

The use of /128 would prevent any form of privacy-based addressing.

6. IANA Considerations

This document has no actions for IANA.

7. Acknowledgements

The authors of this document will like to acknowledge the authors and contributors of previous versions ([RFC3177], [RFC6177]) and the inputs to this version from Brian Carpenter, Mohamed Boucadair, Dusan Mudric, David Farmer, Fred Baker, Barbara Stark, Rajiv Asati, Owen DeLong, ... TBD.

8. Informative References

[I-D.carpenter-6man-lap]
Carpenter, B., "The Longest Acceptable Prefix for IPv6 Links", draft-carpenter-6man-lap-01 (work in progress), June 2018.

[IPv6-Survey]
Palet Martinez, J., "IPv6 Deployment Survey (Residential/ Household Services)", January 2018, <https://indico.uknof.org.uk/event/41/contribution/5/material/slides/0.pdf>.

[RFC3177] IAB and IESG, "IAB/IESG Recommendations on IPv6 Address Allocations to Sites", RFC 3177, DOI 10.17487/RFC3177, September 2001, <https://www.rfc-editor.org/info/rfc3177>.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <https://www.rfc-editor.org/info/rfc4193>.

[RFC6177] Narten, T., Huston, G., and L. Roberts, "IPv6 Address Assignment to End Sites", BCP 157, RFC 6177, DOI 10.17487/RFC6177, March 2011, <https://www.rfc-editor.org/info/rfc6177>.

[RFC7278] Byrne, C., Drown, D., and A. Vizdal, "Extending an IPv6 /64 Prefix from a Third Generation Partnership Project (3GPP) Mobile Interface to a LAN Link", RFC 7278, DOI 10.17487/RFC7278, June 2014, <https://www.rfc-editor.org/info/rfc7278>.

[RFC7368] Chown, T., Ed., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", RFC 7368, DOI 10.17487/RFC7368, October 2014, <https://www.rfc-editor.org/info/rfc7368>.

[RFC7608] Boucadair, M., Petrescu, A., and F. Baker, "IPv6 Prefix Length Recommendation for Forwarding", BCP 198, RFC 7608, DOI 10.17487/RFC7608, July 2015, <https://www.rfc-editor.org/info/rfc7608>.

[RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <https://www.rfc-editor.org/info/rfc7788>.

[RFC7849] Binet, D., Boucadair, M., Vizdal, A., Chen, G., Heatley, N., Chandler, R., Michaud, D., Lopez, D., and W. Haeffner, "An IPv6 Profile for 3GPP Mobile Devices", RFC 7849, DOI 10.17487/RFC7849, May 2016, <https://www.rfc-editor.org/info/rfc7849>.

[RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", BCP 204, RFC 7934, DOI 10.17487/RFC7934, July 2016, <https://www.rfc-editor.org/info/rfc7934>.

[RFC8273] Brzozowski, J. and G. Van de Velde, "Unique IPv6 Prefix per Host", RFC 8273, DOI 10.17487/RFC8273, December 2017, <https://www.rfc-editor.org/info/rfc8273>.

[RIPE-690] RIPE, "Best Current Operational Practice for Operators: IPv6 prefix assignment for end-users - persistent vs non-persistent, and what size to choose", October 2017, <https://www.ripe.net/publications/docs/ripe-690>.

[Route-Scaling] "Routing and Addressing Problem Statement", February 2010, <https://tools.ietf.org/html/draft-narten-radir-problem-statement-05>.

Authors' Addresses

Jordi Palet Martinez
The IPv6 Company
Molino de la Navata, 75
La Navata - Galapagar, Madrid 28420
Spain

EMail: jordi.palet@theipv6company.com
URI: http://www.theipv6company.com/

Rosalea G Roberts
Stanford University, UIT
P.O. Box 19131
Stanford CA 94309-9131

Phone: +1-650-723-3352
EMail: lea.roberts@stanford.edu