v6ops
Internet-Draft
Intended status: Informational
Expires: November 5, 2019

J. Palet Martinez
The IPv6 Company

May 4, 2019

Additional NAT64/464XLAT Deployment Guidelines in Operator and Enterprise Networks
draft-ietf-v6ops-nat64-deployment-06

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 November 5, 2019.

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".

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, as 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.

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. 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, 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.

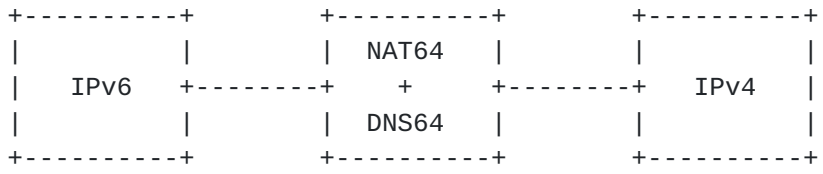

Figure 1: NAT64 with DNS64

A similar scenario 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.

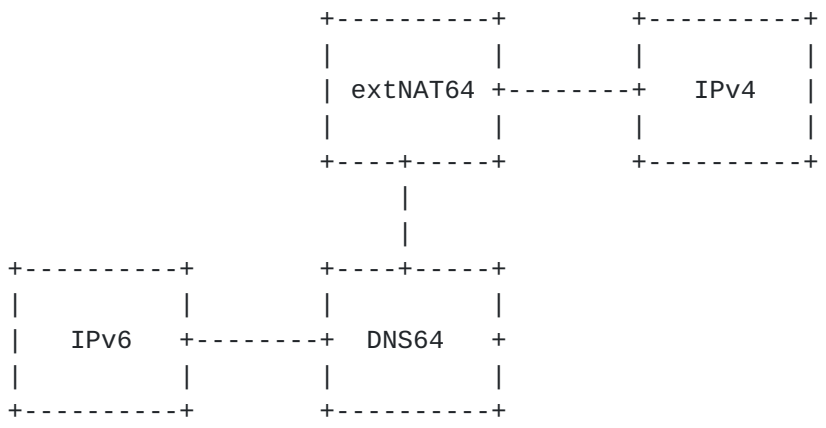

Figure 2: NAT64 in external service provider

This is equivalent to the scenario 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.

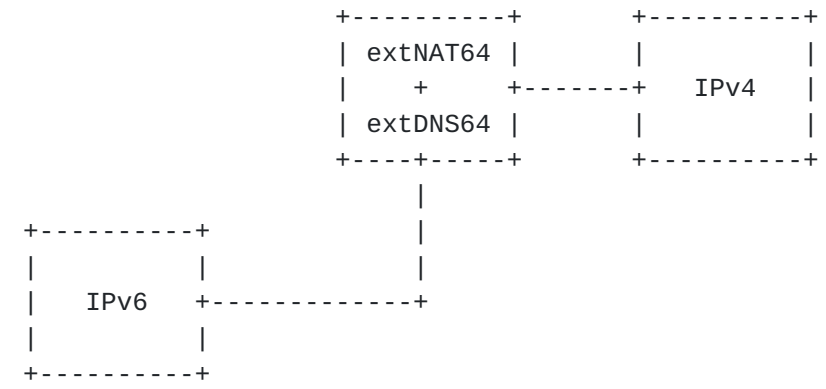

Figure 3: NAT64 and DNS64 in external provider

One more equivalent scenario 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.

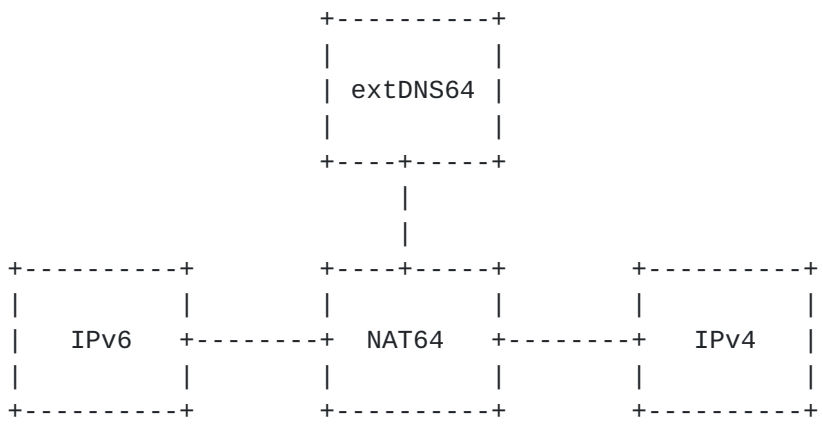

Figure 4: NAT64; DNS64 by external provider

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 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 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:

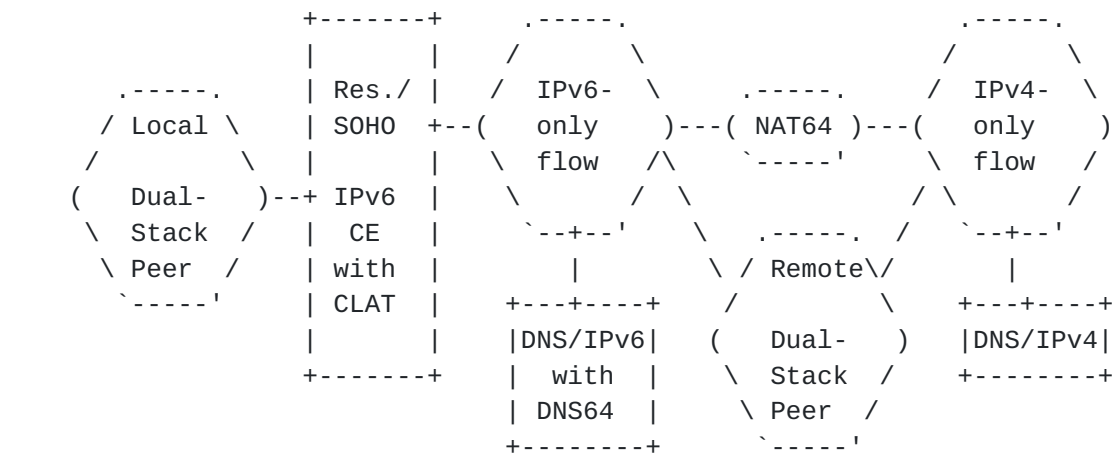

Figure A: Representation of 464XLAT among two peers with DNS64

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 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.

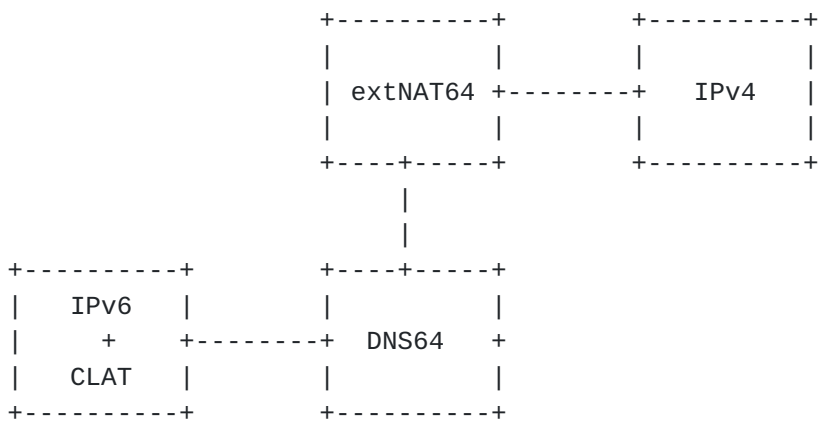

Figure 6: 464XLAT with DNS64; NAT64 in external provider

As well, is equivalent to the scenario 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.

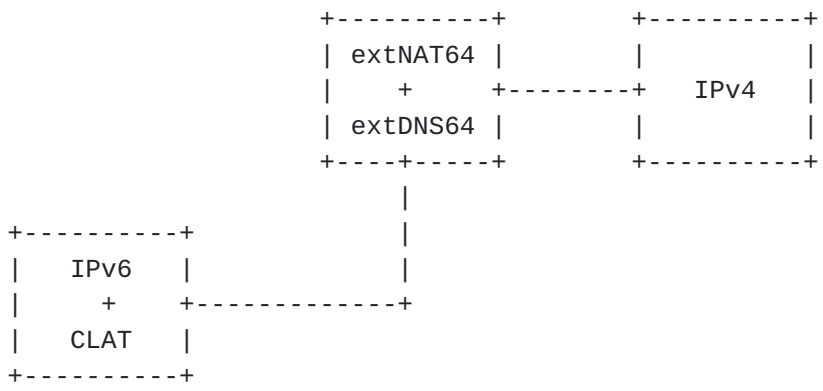

Figure 7: 464XLAT with DNS64; NAT64 and DNS64 in external provider

3.1.3. Service Provider Offering 464XLAT, without DNS64

The major advantage of this scenario, 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 (HEv2, [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 offers one advantage:

o 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 HEv2 ([RFC8305]), which is very common.

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

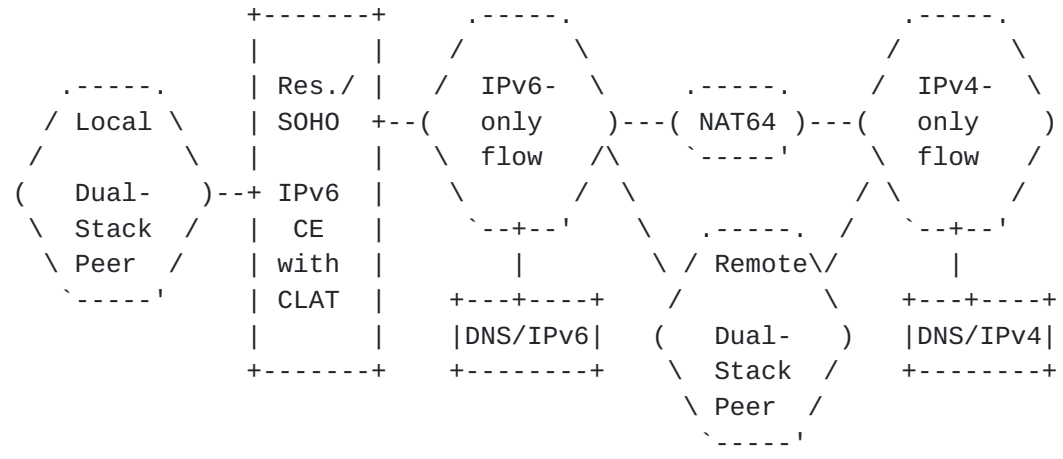

Figure B: Representation of 464XLAT among two peers without DNS64

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.

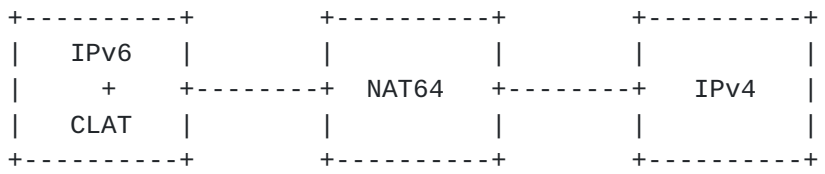

Figure 8: 464XLAT without DNS64

This is equivalent to the scenario 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.

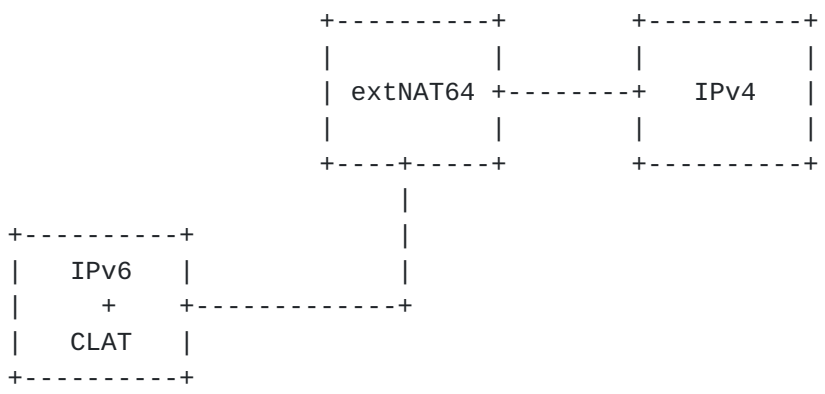

Figure 9: 464XLAT without DNS64; NAT64 in external provider

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, 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.

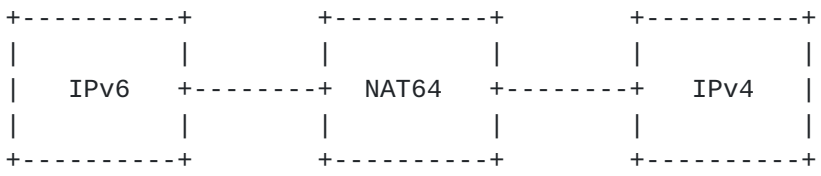

Figure 10: NAT64 without DNS64

3.2.2. Service Provider NAT64; DNS64 in the IPv6 hosts

In this scenario, 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, which is not the actual situation, but 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 HEv2 ([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.

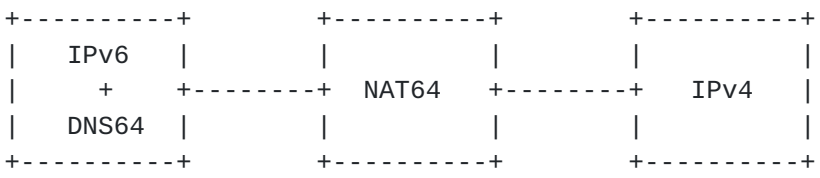

Figure 11: NAT64; DNS64 in IPv6 hosts

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

In this scenario, 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 HEv2 ([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 HEv2. 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 WAN, 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 not 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.

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 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.

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, 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 (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 HEv2 (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, if the IPv4-only servers are in the path before the NAT64 function, the clients will never 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 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 DNSSEC, so will behave as in the Section 3.2.1.

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 Foreign 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

A new trend is for clients or applications to 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

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 old APIs

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

- CLAT (or equivalent function).
- HEv2 (Section 7.1, [RFC8305]).
- 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.

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), with hundreds of millions of users, offering 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 old 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 he 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 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.

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

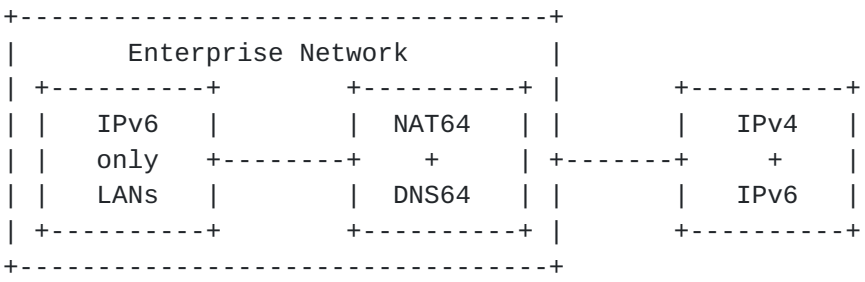

Figure 14: IPv6-only enterprise with NAT64 and DNS64

The following figure 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.

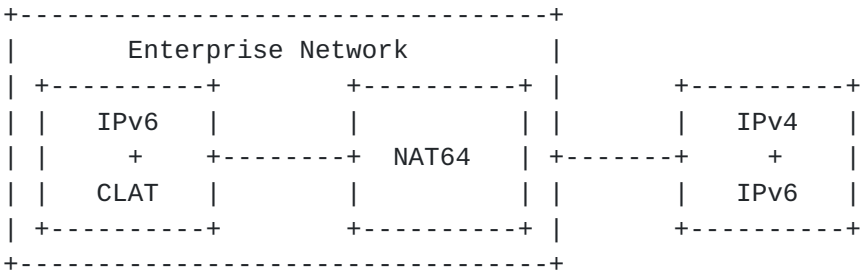

Figure 15: DS enterprise with CLAT, DS Internet, without DNS64

Finally, the following figure 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.

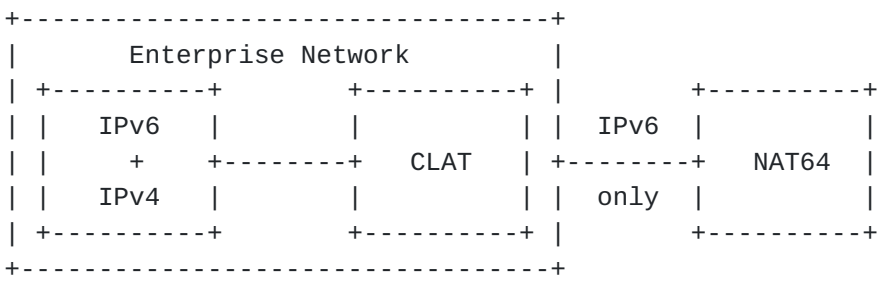

Figure 16: DS enterprise with CLAT, IPv6-only Access, without DNS64

7. Security Considerations

This document does not have new specific security considerations beyond those already reported by each of the documents cited.

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.

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 and Mikael Abrahamsson.

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).

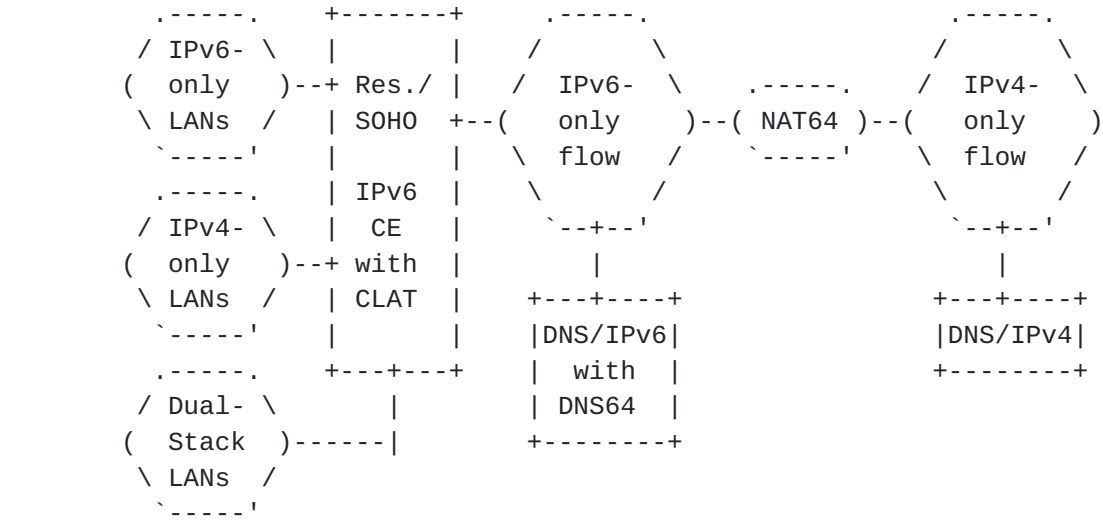

Figure 17: CE setup with built-in CLAT with DNS64

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.

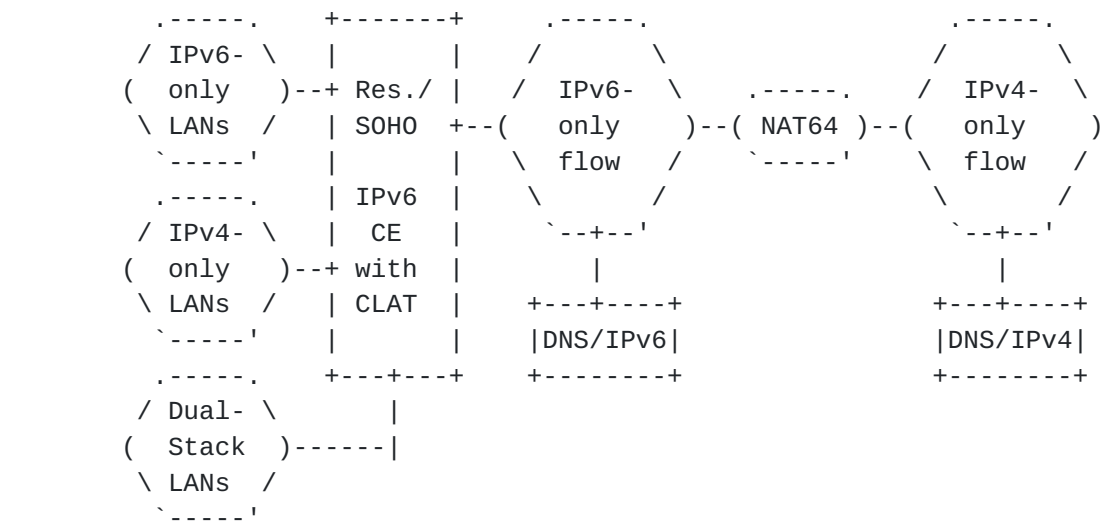

Figure 18: CE setup with built-in CLAT without DNS64

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 the next figure. 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.

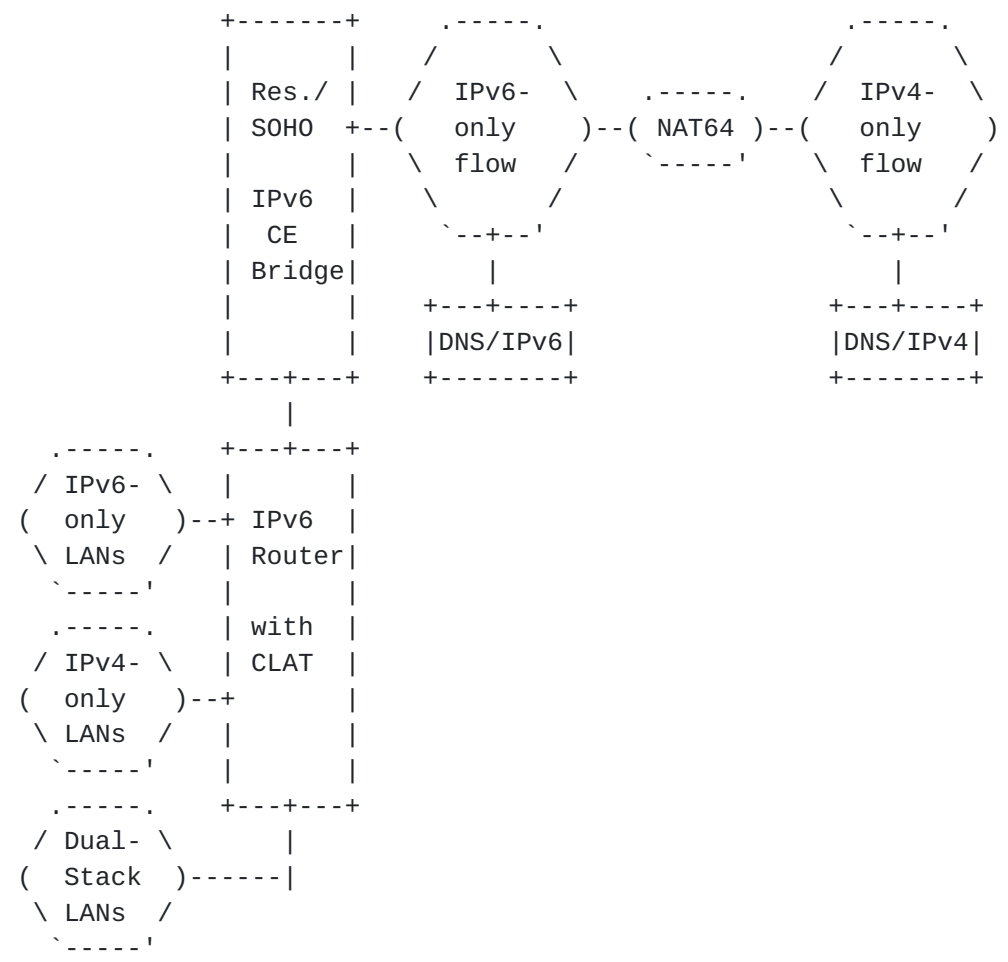

Figure 19: CE setup with bridged CLAT without DNS64

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.
- 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. References

18.1. 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>.

18.2. 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/>.

[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.

[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.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., Kline, E., and J. Linkova, "Discovering PREF64 in Router Advertisements", draft-ietf-6man-ra-pref64-00 (work in progress), March 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-02 (work in progress), January 2019.

[I-D.palet-v6ops-464xlat-opt-cdn-caches]
Palet, J. and A. D'Egidio, "464XLAT Optimization for CDNs/Caches", draft-palet-v6ops-464xlat-opt-cdn-caches-01 (work in progress), March 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.

[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/