

NAT64/DNS64 detection via SRV Records
draft-ietf-v6ops-nat64-srv-00

Abstract

This document specifies the way of discovering the NAT64 pools in use as well as DNS servers providing DNS64 service to the local clients. The discovery is done via SRV records, which also allows assignment of priorities to the NAT64 pools as well as DNS64 servers. It also allows clients to have different DNS providers than NAT64 provider, while providing a secure way via DNSSEC validation of provided SRV records. This way, it provides DNS64 service even in case where DNS over HTTPS is used.

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 September 11, 2019

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

| | |
|---------------------------------------|---|
| 1. Introduction | 3 |
| 1.1. Requirements Language | 3 |
| 2. Terminology | 3 |
| 3. NAT64 service SRV record | 3 |
| 4. DNS64 service SRV record | 4 |
| 5. Node Behavior | 4 |
| 5.1. Example | 5 |
| 6. IANA Considerations | 7 |
| 7. Security considerations | 7 |
| 8. References | 7 |
| 8.1. Normative References | 7 |
| 8.2. Informative References | 8 |

1. Introduction

The simultaneous use of NAT64/DNS64 and DNSSEC outlined by [RFC7050], does not solve all the aspects of such use. Namely [RFC7050] does not allow assignment of NAT64 priorities in case when multiple network prefixes are in use. [RFC7050] also doesn't work in the case when network operator and DNS operator are not the same subject, like in the case when the end node is using some public DNS resolvers. This document describes the way how to circumvent that limitation while maintaining added security provided by DNSSEC.

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

2. Terminology

End node: Either DNS stub resolver or the DNS recursive resolver serving a local area network or station.

Pref64::: an IPv6 prefix used for IPv6 address synthesis [RFC6146].

Pref64::WKA: an IPv6 address consisting of Pref64:: and WKA at any of the locations allowed by [RFC6052].

Well-Known IPv4 Address (WKA): an IPv4 address that is well-known and present in an A record for the well-known name. Two well-known IPv4 addresses are defined for Pref64:: discovery purposes: 192.0.0.170 and 192.0.0.171.

3. NAT64 service SRV record

This document specifies two new well-known SRV records. The one for NAT64 prefix which validation end node MUST implement:

nat64. ipv6.Name TTL Class SRV Priority Weight Port Target

The TTL, Class, Priority and Weight follows the same scheme as defined in [RFC2782] and have theirs standard meaning.

Port: IPv6 as L3 protocol doesn't use port numbers. Because of that this field SHOULD be either set to zero, or SHOULD be used to indicate length of network prefix mask in both IPv6 and IPv4 protocol, used NAT64. In such case the port 16b integer MUST be constructed by directly appending IPv4 pool prefix mask after the

IPv6 prefix mask decadically. Usually this would mean 9632 stating that IPv6 prefix with mask /96 is translated into single IPv4 address (/32).

Target: MUST point to AAAA record formed from Pref64::/n prefix and WKA same way as in [RFC7050] (Pref64::WKA). Target MAY also point to A record, in which case it SHOULD point to IPv4 address used for NAT64 (or base address of the NAT64 IPv4 prefix).

4. DNS64 service SRV record

The second SRV record is for the discovery of DNS64 service. Support of this record is OPTIONAL but end node SHOULD implement it.

dns64.Protocol.Name TTL Class SRV Priority Weight Port Target

Record informs about location of DNS64 service. This might be used in case that network operator doesn't want to deploy DNS64 in their main DNS infrastructure. A DNS64 SRV record follows the rules specified by [RFC2782] and does not modify meaning of any field.

Server provided by this record SHOULD only be used for domain names which have returned NODATA for AAAA record.

5. Node Behavior

In early stage of end node connection to the network - after the end node is configured with IP address, the end node MUST get local domains used in the network. Method of obtaining such information is out of scope of this document, but it might contain one or more methods, like the SLAAC-DNSSL [RFC8106], the DHCPv6 - option 24 or a manual configuration. In case, when no local domain can be discovered, the end node SHOULD continue NAT64/DNS64 detection by other means, like [RFC7050].

After the list of local domains has been established, the end node MUST ask for NAT64 SRV record for every domain in the list. Result of such queries SHOULD be ordered by following the rules of [RFC2782]. In case when multiple records do have a same values of both priority and weight, the records SHOULD maintain the same order as its domain in the discovered domain list.

For every domain with NAT64 SRV record the end node SHOULD perform query for DNS64 SRV record. If such a record is obtained and the end node is not configured to make DNS64 synthesis itself, the end node SHOULD use preferred target of DNS64 SRV record to query for FQDN without AAAA record - when it received NODATA response.

If the end node is capable of validation of DNS records via DNSSEC, the end node MUST perform validation of NAT64/DNS64 SRV record. Default behavior of end node SHOULD be to ignore any NAT64/DNS64 SRV records which cannot be validated or did not pass the validation.

5.1. Example

The end node is a home router connected to the ISP network in which the NAT64/DNS64 is used and the ISP has the following SRV records in their zones:

```
- nat64.ipv6.example.com. IN SRV 5 10 9632 nat64-pool-1.example.com.
- nat64-pool-1.example.com. IN AAAA 2001:db8:64:ff9b:1::c000:aa
- nat64-pool-1.example.com. IN A 192.0.2.64
- nat64.ipv6.example.com.
      IN SRV 10 10 9632 nat64-pool-2.example.com.
- nat64-pool-2.example.com. IN AAAA 2001:db8:64:ff9b:2::c000:aa
- nat64-pool-2.example.com. IN A 192.0.2.164
- nat64.ipv6.example.net. IN SRV 10 10 9624 nat64-pool.example.net.
- nat64-pool.example.net. IN AAAA 2001:db8:64:ff9b:abc::c000:aa
- nat64-pool.example.net. IN A 198.51.100.0
- nat64.ipv6.example.invalid.
      IN SRV 10 10 9624 nat64-pool.example.org.
- nat64-pool.example.org. IN AAAA 2001:db8:64:ff9b:def::c000:aa
- nat64-pool.example.org. IN A 203.0.113.0
```

In addition the zones "example.net" and "example.invalid" has got DNS64 SRV records:

```
- dns64.tcp.example.net. IN SRV 5 10 53 dns64.example.net.
- dns64.udp.example.net. IN SRV 10 10 53 dns64.example.net.
- dns64.example.net. IN AAAA 2001:db8::53
- dns64.udp.example.invalid. IN SRV 10 10 53 dns64.example.org.
- dns64.example.org. IN AAAA 2001:db8:123::53
```

The zones "example.com" and "example.net" are secured and successfully validated by the DNSSEC. Domain "example.invalid" is either not secured by the DNSSEC or its validation failed. Domain "example.org" is DNSSEC secured but does not have any NAT64/DNS64 SRV records.

The end node has been supplied with the following list of domains via SLAAC-DNSSSL:

1. example.net
2. example.invalid
3. example.com
4. example.org

The end node would fetch all available SRV records and its A and AAAA counterparts and sort it in following order:

| pool | DNSSEC | priority | reason |
|---------------------------|--------|----------|-----------------------|
| nat64-pool-1.example.com. | yes | 5 | lowest priority field |
| nat64-pool.example.net. | yes | 10 | discovered first |
| nat64-pool-2.example.com. | yes | 10 | higher priority field |
| nat64-pool.example.org. | no | 10 | no valid DNSSEC chain |

After sorting, the end node SHOULD graylist any record which cannot be validated by the DNSSEC. In this example it would be "nat64-pool.example.org." because it has been obtained from insecure domain "example.invalid". A such pool SHOULD NOT be used if it is not confirmed by other DNSSEC secured record.

If the end node is capable to act as recursive or caching DNS server and it is configured to provide the DNS64 service, it MUST provide this service using sorted list of NAT64 pools. For such end node a process of the NAT64/DNS64 ends here.

However, when the end node is not capable of record synthesis or it is not configured to provide DNS64 service, it SHOULD perform detection of DNS64 by querying for "ipv4only.arpa" like in the case of [RFC7050]. If the reply contains a pool listed in the NAT64 pool list, the corresponding entry is marked as having DNS64 provided by recursive DNS.

When the end node supports DNS64 SRV record and there is at least one non-graylisted NAT64 pool, which is not reachable by using the end node's recursive DNS, the end node MUST make a sorted list of DNS64 servers from the DNS64 SRV records. The DNS64 sorted list would look like this:

| server | proto | DNSSEC | priority | reason |
|--------------------|-------|--------|----------|-----------------------|
| dns64.example.net. | tcp | yes | 5 | lowest priority field |
| dns64.example.net. | udp | yes | 10 | higher priority field |
| dns64.example.org. | udp | no | 10 | no valid DNSSEC chain |

Sorting is done in the same fashion as any other SRV record with the same exception of graylisting records without valid DNSSEC chain. Those SHOULD NOT be used when not confirmed by DNSSEC validated record and SHOULD be kept in the end of the list.

For example when ISP is providing DNS64 service in their main DNS infrastructure only for pools in the domains "example.com" and "example.org" and the pool "nat64-pool.example.net" is used only with corresponding DNS64 server. The final sorted list of NAT64 prefixes used by the end node in the ISP network would be:

| pool | state | priority | reason |
|---------------------------|----------|----------|-----------------------|
| nat64-pool-1.example.com. | active | 5 | lowest priority field |
| nat64-pool-2.example.com. | backup | 10 | higher priority field |
| nat64-pool.example.net. | backup* | 10 | main DNS has priority |
| nat64-pool.example.org. | inactive | 10 | no valid DNSSEC chain |

As the pool "nat64-pool.example.net" is used only with the server "dns64.example.net" this would effectively put this pool to the end of the list. Because it would be used only for FQDN for which the regular DNS infrastructure returns NODATA.

Now the end node has successfully identified NAT64 pools and the DNS64 servers in the ISP infrastructure. The discovered prefixes SHOULD be considered safe and DNSSEC validation of answers in these prefixes MUST be either disabled or performed by validating only the suffix.

6. IANA Considerations.

This document proposes a usage of "ipv6" in Proto field and two services "nat64" and "dns64" in Service field of SRV RR ([RFC2782]).

7. Security considerations

Method proposed by this document relies on security principles based on DNSSEC and secure discovery of local domain. In order to be secure, the network operator MUST deploy DNSSEC on at least one domain (advertised to end node) and establish secure channel to this advertisement.

8. References

8.1. Normative References

- [RFC2119] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. RFC 2119. RFC Editor, Mar. 1997, pp. 1-3. url: <https://www.rfc-editor.org/rfc/rfc2119.txt>.
- [RFC2782] A. Gulbrandsen, P. Vixie, and L. Esibov. A DNS RR for specifying the location of services (DNS SRV). RFC 2782. RFC Editor, Feb. 2000, pp. 1-12. url: <https://www.rfc-editor.org/rfc/rfc2782.txt>.
- [RFC6146] M. Bagnulo, P. Matthews, and I. van Beijnum. Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers. RFC 6146. RFC Editor, Apr. 2011, pp. 1-45. url: <https://www.rfc-editor.org/rfc/rfc6146.txt>.
- [RFC7050] T. Savolainen, J. Korhonen, and D. Wing. Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis. RFC 7050. RFC Editor, Nov. 2013, pp. 1-22. url: <https://www.rfc-editor.org/rfc/rfc7050.txt>.
- [RFC8174] B. Leiba. Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words. RFC 8174. RFC Editor, May 2017, pp. 1-4. url: <https://www.rfc-editor.org/rfc/rfc8174.txt>.

8.2. Informative References

- [RFC6052] C. Bao et al. IPv6 Addressing of IPv4/IPv6 Translators.
RFC 6052. RFC Editor, Oct. 2010, pp. 1-18.
url: <https://www.rfc-editor.org/rfc/rfc6052.txt>.
- [RFC8106] J. Jeong et al. IPv6 Router Advertisement Options for DNS
Configuration. RFC 8106. RFC Editor, Mar. 2017, pp. 1-19.
url: <https://www.rfc-editor.org/rfc/rfc8106.txt>.

Acknowledgments

This work has been supported by Student Grant Scheme (SGS 2019) at
Technical University of Liberec.

Authors' Addresses

Martin Hunek
Technical University of Liberec
Studentska 1402/2
Liberec, 46017 Czech Republic

phone: +420 485 35 3792
e-mail: martin.hunek@tul.cz

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.

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 July 13, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

| | |
|--|----|
| 1. Introduction | 3 |
| 1.1. Requirements Language | 4 |
| 2. Overview of the Technologies | 4 |
| 2.1. 464XLAT | 4 |
| 2.2. Dual-Stack Lite | 5 |
| 2.3. Lightweight 4over6 | 5 |
| 2.4. MAP-E | 6 |
| 2.5. MAP-T | 7 |
| 3. High-level Architectures and their Consequences | 8 |
| 3.1. Service Provider Network Traversal | 8 |
| 3.2. Network Address Translation | 9 |
| 3.3. IPv4 Address Sharing | 10 |
| 3.4. CE Provisioning Considerations | 11 |
| 3.5. Support for Multicast | 11 |
| 4. Detailed Analysis | 11 |
| 4.1. Architectural Differences | 11 |
| 4.1.1. Basic Comparison | 11 |
| 4.2. Tradeoff between Port Number Efficiency and Stateless Operation | 12 |
| 4.3. Support for Public Server Operation | 14 |
| 4.4. Support and Implementations | 15 |
| 4.4.1. OS Support | 15 |
| 4.4.2. Support in Cellular and Broadband Networks | 16 |
| 4.4.3. Implementation Code Sizes | 16 |
| 4.5. Typical Deployment and Traffic Volume Considerations | 16 |
| 4.5.1. Deployment Possibilities | 16 |
| 4.5.2. Cellular Networks with 464XLAT | 16 |
| 4.6. Load Sharing | 17 |
| 4.7. Logging | 18 |
| 4.8. Optimization for IPv4-only devices/applications | 18 |
| 5. Performance Comparison | 19 |

| | |
|---------------------------------------|----|
| 6. Acknowledgements | 20 |
| 7. IANA Considerations | 20 |
| 8. Security Considerations | 20 |
| 9. References | 21 |
| 9.1. Normative References | 21 |
| 9.2. Informative References | 24 |
| Appendix A. Change Log | 26 |
| A.1. 01 - 02 | 26 |
| A.2. 02 - 03 | 26 |
| A.3. 03 - 04 | 27 |
| A.4. 04 - 05 | 27 |
| A.5. 05 - 06 | 27 |
| Authors' Addresses | 27 |

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.

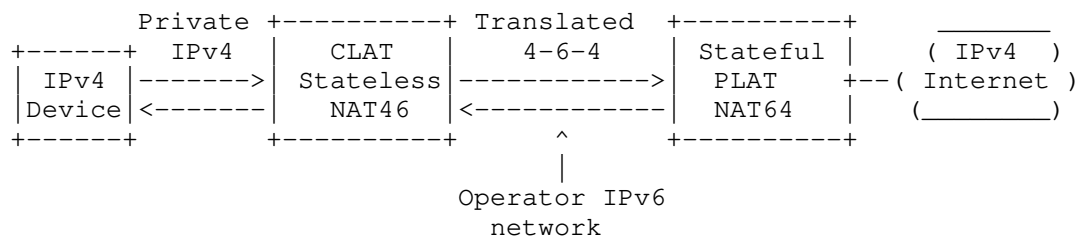


Figure 1: Overview of the 464XLAT architecture

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.

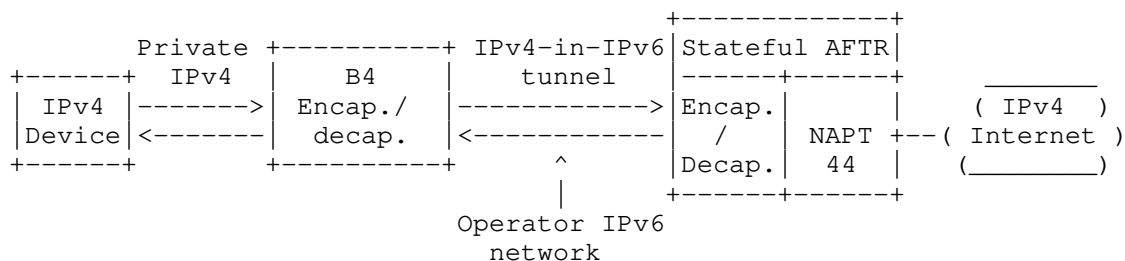


Figure 2: Overview of the DS-Lite architecture

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.

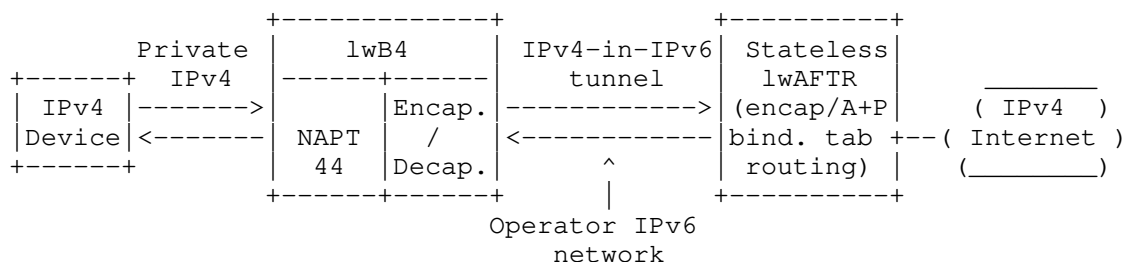


Figure 3: Overview of the lw4o6 architecture

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.

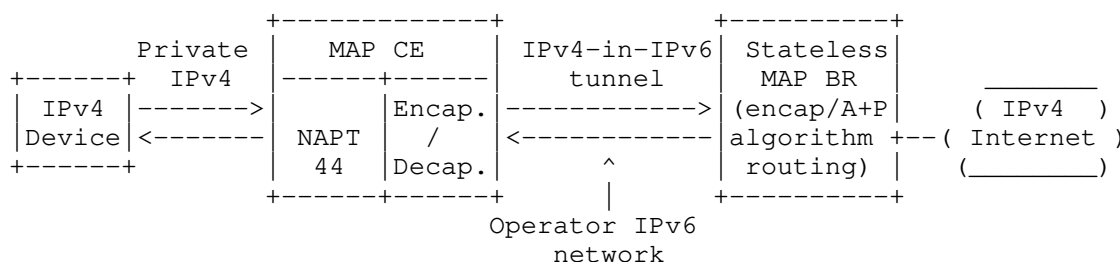


Figure 4: Overview of the MAP-E architecture

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.

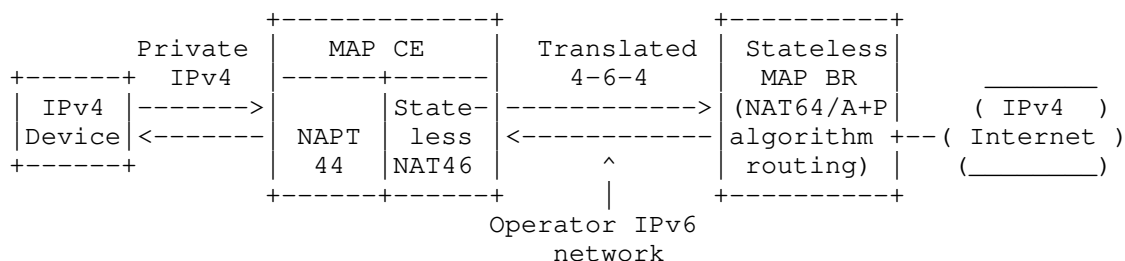


Figure 5: Overview of the MAP-T architecture

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:

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

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

- o CGN-based approaches (DS-Lite, 464XLAT)
- o 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 $2 \times 2 = 4$ categories on the basis of two aspects:

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

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

- o '464xlat' enables support for 464XLAT CLAT functionality
- o 'ds-lite' enables support for DSLite B4 functionality
- o 'map' enables support for MAP-E and lw4o6 CE functionality
- o '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.fdn.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:

- o 75% of traffic is IPv6 end-to-end (no translation)
- o 24% of traffic uses DNS64 + NAT64 (1 translation)
- o 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 synchronized between CGN instances. This is necessary for two reasons:

- o To prevent all active customer sessions being dropped in event of a CGN node failure.
- o 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 at 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:

- o 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.
- o 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 Throughput 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]

- o 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 procedures for stateful NAT64 include the stateless tests, plus a few additional tests, which are not implemented yet.)
- o 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 benchmarking 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 intensive benchmarking 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., Pislariu, 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

- o Ian Farrer has joined us as an author.
- o Restructuring: the description of the five IPv4aaS technologies was moved to a separate section.
- o More details and figures were added to the description of the five IPv4aaS technologies.
- o Section titled "High-level Architectures and their Consequences" has been completely rewritten.
- o Several additions/clarification throughout Section titled "Detailed Analysis".
- o Section titled "Performance Analysis" was dropped due to lack of results yet.
- o Word based text ported to XML.
- o 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

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

A.3. 03 - 04

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

A.4. 04 - 05

- o Some references were updated or added.

A.5. 05 - 06

- o 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

v6ops
Internet-Draft
Intended status: Informational
Expires: May 7, 2020

J. Palet Martinez
The IPv6 Company
A. D'Egidio
Telecentro
November 4, 2019

464XLAT Optimization
draft-palet-v6ops-464xlat-opt-cdn-caches-04

Abstract

This document proposes possible solutions to avoid certain drawbacks of IP/ICMP Translation Algorithm (SIIT) when the destinations are available with IPv6. When SIIT is used as a NAT46 and IPv4-only devices or applications initiate traffic flows to dual-stack CDNs (Content Delivery Networks), Caches or other network resources (in the operator network or Internet), those flows will be translated back to IPv4 by a NAT64. This is the case for 464XLAT and MAP-T.

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 May 7, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|---|----|
| 1. Introduction | 2 |
| 2. Requirements Language | 4 |
| 3. Problem Statement | 4 |
| 4. Solution Approaches | 6 |
| 4.1. Approach 1: DNS/Routing-based Solution | 6 |
| 4.2. Approach 2: NAT46/CLAT/DNS-proxy-EAM-based Solution | 7 |
| 4.2.1. Detection of IPv4-only devices or applications | 7 |
| 4.2.2. Detection of IPv6-enabled service | 8 |
| 4.2.3. Creation of EAMT entries | 8 |
| 4.2.4. Forwarding path via stateful NAT for existing EAMT entries | 10 |
| 4.2.5. Maintenance of the EAMT entries | 10 |
| 4.2.6. Usage example | 10 |
| 4.2.7. Behavior in case of multiple A/AAAA RRs | 11 |
| 4.2.8. Behavior in presence/absence of DNS64 | 11 |
| 4.2.9. Behavior when using literal addresses or non IPv6-compliant APIs | 11 |
| 4.2.10. False detection of a dual-stack host as IPv4-only | 11 |
| 4.2.11. Behaviour in presence of Happy Eyeballs | 12 |
| 4.2.12. Behavior in case of Foreign DNS | 12 |
| 4.3. Approach 3: NAT46/CLAT-provider-EAM-based Solution | 13 |
| 5. IPv6-only Services become accessible to IPv4-only devices/apps | 14 |
| 6. Conclusions | 14 |
| 7. Security Considerations | 15 |
| 8. IANA Considerations | 15 |
| 9. Acknowledgements | 15 |
| 10. References | 15 |
| 10.1. Normative References | 15 |
| 10.2. Informative References | 16 |
| Authors' Addresses | 17 |

1. Introduction

Different transition mechanisms, typically in the group of the so-called IPv6-only with IPv4aaS (IPv4-as-a-Service), such as 464XLAT ([RFC6877]) or MAP-T ([RFC7599]), allow IPv4-only devices or applications to connect with IPv4 services in Internet, by means of a NAT46 SIIT (IP/ICMP Translation Algorithm) as described by [RFC7915].

This is done by the implementation of SIIT at the CE (Customer Edge) Router or sometimes at the end-device, for example, the UE (User

Equipment) in cellular networks. This functionality is the CLAT (Customer Translator) in the case of 464XLAT.

The NAT46/CLAT (WAN side) is connected by IPv6-only to the operator network, which in turn, will have a reverse function, the NAT64 ([RFC6146]), known as PLAT (Provider Translator) in the case of 464XLAT. This allows to translate the IPv6-only flow back to IPv4, in order to forward it to Internet.

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 following [RFC6052].

In the case of 464XLAT, a DNS64 ([RFC6147]) optionally is in charge of the synthesis of AAAA records from the A records, so they can use a NAT64, without the need of doing a double-translation by means of the CLAT. However, the DNS64 is not useful for the IPv4-only devices or applications in the LANs, as they will not be able to use the AAAA records.

A typical 464XLAT deployment is depicted in Figure 1.

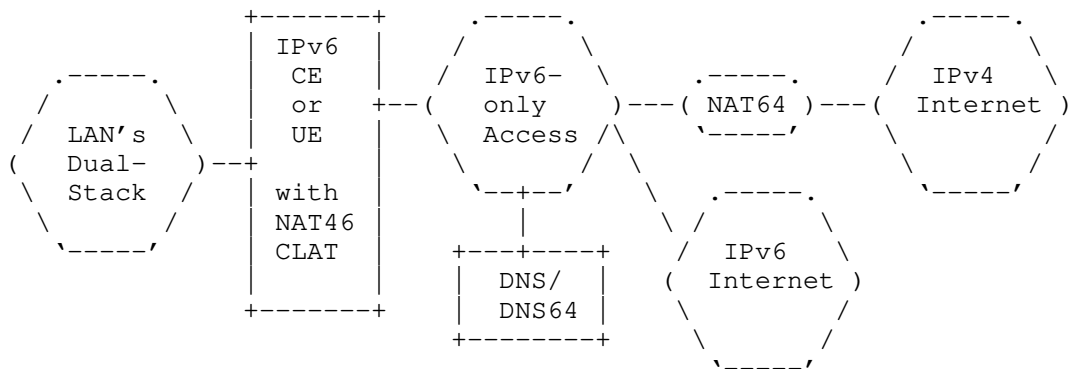


Figure 1: Typical 464XLAT Deployment

As it can be observed in the preceding picture, the situation is the same, regardless of in case of a wired network with a CE Router or a cellular network where a UE is connecting other devices (which may be IPv4-only or have IPv4-only apps), by means of a tethering functionality.

If the operator is providing direct access to Content Delivery Networks (CDNs), caches, or other resources, and they are dual-stacked, the situation can be described as shown in Figure 2.

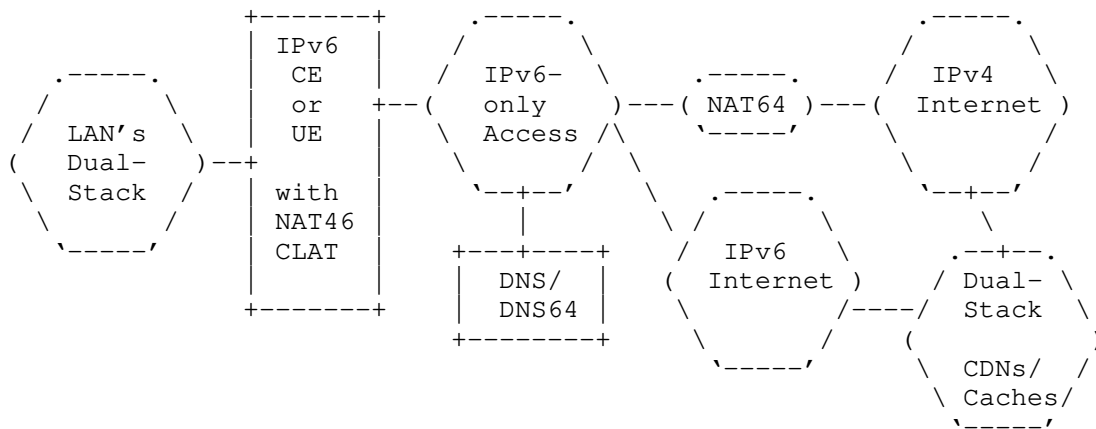


Figure 2: Typical 464XLAT Deployment with CDNs/Caches

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. Problem Statement

If the devices or applications in the customer LAN are IPv6-capable, then the access to the CDNs, caches or other resources, will be made in an optimized way, by means of IPv6-only, not using the NAT64, as depicted in Figure 3.

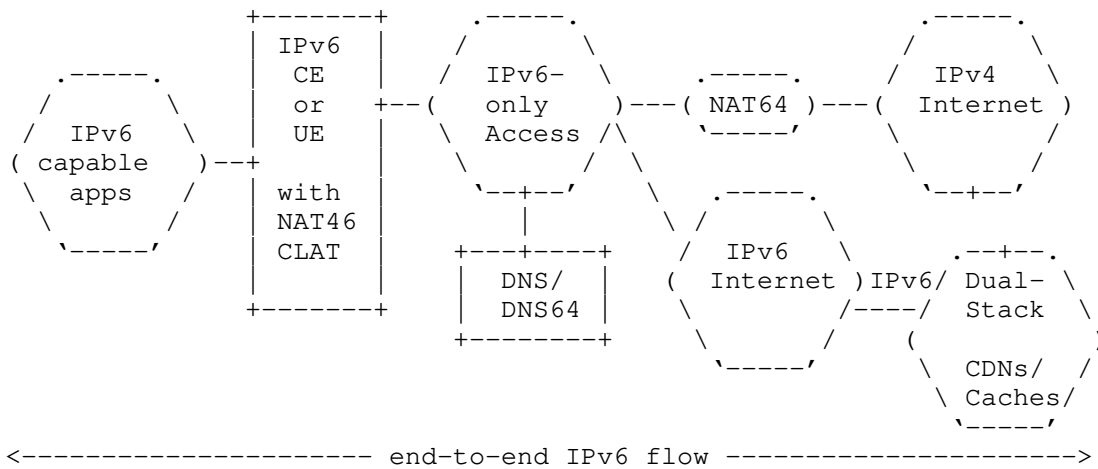


Figure 3: 464XLAT access to CDNs/Caches by IPv6-capable apps

However, if the devices or applications are IPv4-only, for example, most of the SmartTVs and Set-Top-Boxes available today, a non-optimal double translation will occur (NAT46 at the CLAT and NAT64 at the PLAT), as illustrated in Figure 4.

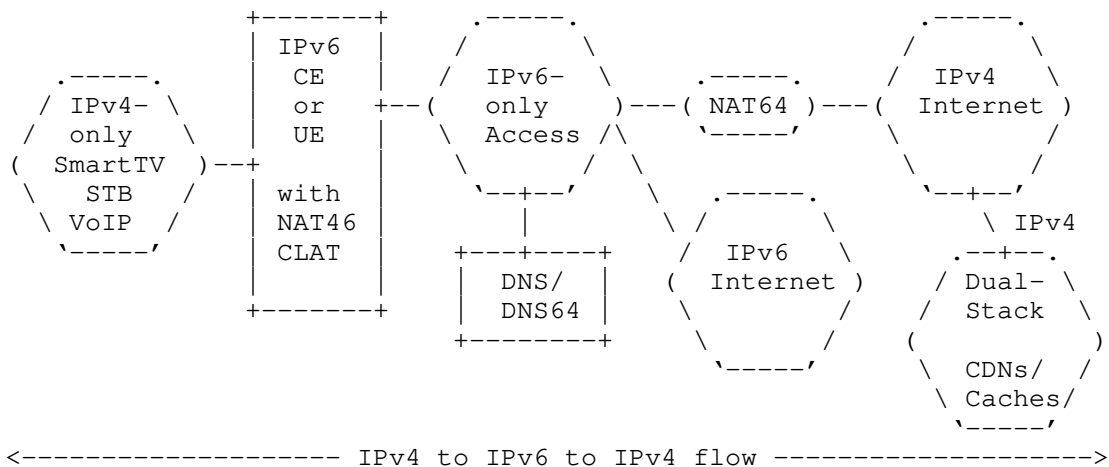


Figure 4: 464XLAT access to CDNs/Caches by IPv4-only apps

Clearly, this is a non-optimal situation, as it means that even if there is a dual-stack service, the NAT46/CLAT translated IPv4 to IPv6 traffic flow, is unnecessarily translated back to IPv4, traversing the stateful NAT64. This has a direct impact in the need to scale the NAT64 beyond what will be actually needed if possible solutions,

in order to keep using the IPv6 path towards those services, are considered.

As shown in the Figure 4, this is also the case for many other services, not just CDNs or caches, such as VoIP access to the relevant operator infrastructure, which may be also dual-stack. This is true as well for many other dual-stack or IPv6-enabled services, which may be directly reachable from the operator infrastructure, even if they are not part of it, for example peering agreements, services in IXs, etc. In general, this will become a more frequent situation for many other services, which are not yet dual-stack.

For simplicity, across the rest of this document, references to CDNs/caches, should be understood, unless otherwise stated, as any dual-stacked resources.

This document looks into different possible solution approaches in order to optimize the IPv4-only SIIT translation providing a direct path to IPv6-capable services, as depicted in Figure 5.

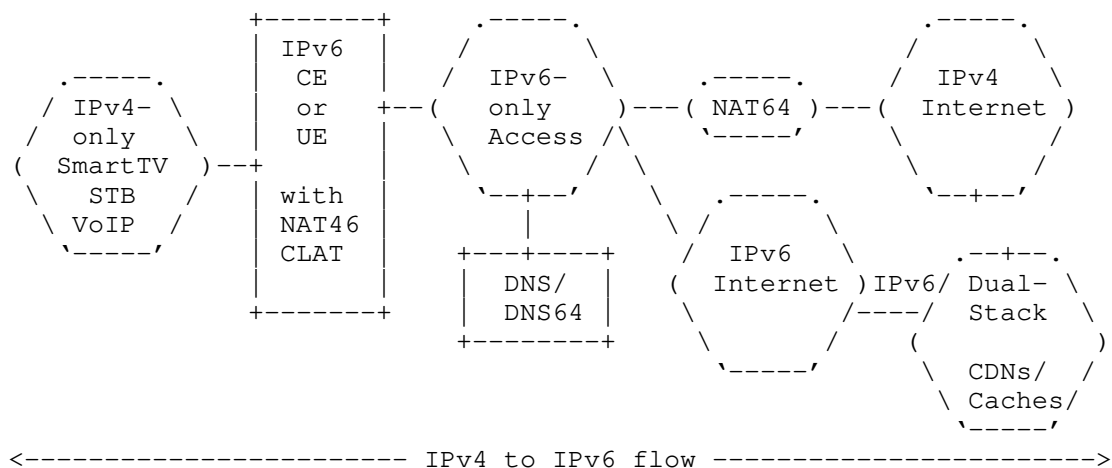


Figure 5: Optimized 464XLAT access to CDNs/Caches by IPv4-only apps

4. Solution Approaches

4.1. Approach 1: DNS/Routing-based Solution

Because the IPv4-only devices will not be able to query for AAAA records, the NAT46/CLAT/CE will translate the IPv4 addresses from the A record for the CDN/cache destination, using the WKP or NSP, as configured by the operator.

If the CDN/cache provider is able to configure, in the relevant interfaces of the CDN/caches, the same IPv6 addresses that will naturally result as the translated destination addresses for the queried A records, preceded by the WKP or NSP, then having more specific routing prefixes, will result in traffic to those destinations being directly forwarded towards those interfaces, instead of needing to traverse the NAT64.

For example, let's suppose a provider using the WKP (64:ff9b::/96) and a SmartTV querying for www.example.com:

| | | |
|--|---|--------------------|
| www.example.com | A | 192.0.2.1 |
| NAT46/CLAT translated to | | 64:ff9b::192.0.2.1 |
| CDN IPv6 interface must be | | 64:ff9b::192.0.2.1 |
| Operator must have a specific route to | | 64:ff9b::192.0.2.1 |

Note: Examples using text representation as per Section 2.3 of [RFC6052].

Because the WKP is non-routable, this solution will only be possible if the CDN/cache is in the same ASN as the provider network, or somehow interconnected without routing thru Internet.

This solution has the additional drawback of the operational complexity/issues added to the operation of the CDN/cache, and the need to synchronize any IPv4 interface address changes with the relevant IPv6 ones, and possibly with routing.

4.2. Approach 2: NAT46/CLAT/DNS-proxy-EAM-based Solution

If the NAT46/CLAT/CE, as commonly is the case, is also a DNS proxy/stub resolver, it is possible to modify the behavior and create an "internal" interaction among both of them.

This approach uses the existing IPv4 and IPv6 addresses in the A and AAAA records, respectively, so no additional complexity/issues added to the CDN/caches operations.

The following sub-sections detail this approach and provide a step-by-step example case.

4.2.1. Detection of IPv4-only devices or applications

The assumption is that, typically a dual-stack device will prefer using IPv6 as the DNS transport. So, when there is a DNS query, transported with IPv4, for an A record, and there is not a query for the AAAA record from the same IPv4 source (to the same destination), the DNS proxy/stub resolver can infer that, most probably, it is an

IPv4-only device or application.

It needs to be remarked that, if the detection of the IPv4-only device or application is done incorrectly (either not detecting it or by a false detection), no harm is caused. In the worst case, optimization will not be performed, at least, at the time being. However, optimization maybe performed later on, if a new detection succeeds (for example, another device using the same A record).

4.2.2. Detection of IPv6-enabled service

In the case of an IPv4-only detected device or application, the DNS proxy/stub resolver MUST actually perform an additional AAAA query, unless the information is already present in the Additional Section, as per Section 3 of [RFC3596]. Note that the NAT46/CLAT MUST already know the WKP or NSP being used in that network. If the response contains at least one IPv6 address not using the WKP/NSP, it means that the destination is IPv6-enabled (because at least one of the IPv6 addresses is not synthesized). This means that it is possible for the NAT46/CLAT, to create an Explicit Address Mapping ([RFC7757]).

4.2.3. Creation of EAMT entries

This way, an EAM Table (EAMT used for short, across the rest of this document) is created/maintained automatically by the DNS proxy/stub resolver in the NAT46/CLAT, and the NAT46/CLAT is responsible to prioritize any available entries in the EAMT, versus the use of any synthetic AAAA.

In order to create the EAMT entry, to determine if there is an AAAA record after an A record query, it is suggested to use the same delay value (50 milliseconds) as the "Resolution Delay" indicated by Happy Eyeballs [RFC8305]. This avoids a slight NAT64 overload and flapping between destination addresses (IPv4/IPv6), which may impact some applications, at the cost of a small extra delay for the initial communication setup, when the EAMT entry doesn't yet exist.

Each EAMT entry will contain, the fields already described in [RFC7757] and a few new ones:

1. ID: EAMT Entry Index (optional).
2. IPv4 address/prefix: By default, the prefix length is 32 bits.
3. IPv6 address/prefix: By default, the prefix length is 128 bits.
4. TTL: Because the optimization will make use of the AAAA (IPv6

address), the TTL for the EAMT entry must be the one of the AAAA RR. In normal conditions the TTL for both A and AAAA records, of a given FQDN, should be the same, so this ensures a proper behavior if there is any DNS mismatch.

5. FQDN: The one that originated the A query for this EAMT entry. Required in order to ensure a correct detection of cases such as the use of reverse-proxy with a single IPv4 address to multiple IPv6 addresses.
6. Valid/Invalid: When set to 1, means that this EAMT entry MUST NOT be used and consequently no optimization performed. It may be used also for an explicit configuration (GUI, CLI, provisioning system, etc.) to disallow optimization for any IPv4 addresses.
7. Auto/Static: When set to 1, means that this EAMT entry has been manually/statically configured, for example by means of an explicit configuration (GUI, CLI, provisioning system, etc.), so it doesn't expire with TTL.

When a new EAMT entry is first automatically created, it is marked as "Valid" and "Auto" (both bits cleared). If a subsequent A query, with a different FQDN, results in an IPv4 address that has already an EAMT entry and a different IPv6 address, it means that some reverse-proxy or similar functionality is being used by the IPv6-enabled service. In this case, the existing EAMT entry will be marked as "Invalid" (bit set). No new EAMT entry is created for that IPv4 address. Otherwise, the optimization will only allow to access the first set of IPv4/IPv6/FQDN, which may break the access to other FQDN that share the same IPv4 address and different IPv6 addresses.

In this case the EAMT entry will still expire according the TTL, which allows to re-enable optimization if a new query for the A record has changed the situation. For example, maybe the reverse-proxy has been removed, or there is now only a single device using it, so at the time being, the optimization is again possible without creating troubles to other hosts.

Note that when an EAMT entry is marked as "invalid", it will not affect the devices or applications, as they will still be able to use the regular CLAT+NAT64 flow, of course, without the optimization.

***** Open question regarding TTL and maybe FQDN and valid/auto bits. Is this always a good thing to do for EAM? Should this document update [RFC7757] to support this by default? Or it is just an "extension" as per section 3.1 of [RFC7757].

4.2.4. Forwarding path via stateful NAT for existing EAMT entries

Following this approach, if there is a valid EAMT entry, for a given IPv4-destination, the IPv6-native path pointed by the IPv6 address of that EAMT entry, will take precedence versus the NAT64 path, so the traffic will not be forwarded to the NAT64.

However, this is not sufficient to ensure that individual applications are able to keep existing connections. In many cases, audio and video streaming may use a single TCP connection lasting from minutes to hours. Instead, the CDN TTLs may be configured in the range from 10 to 300 seconds in order to allow new resolutions to switch quickly and to handle large recursive resolvers (with hundreds of thousands of clients behind them).

Consequently, the EAMT entries should not be used directly to establish a forwarding path, but instead, to create a stateful NAT entry for the 4-tuple for the duration of the session/connection.

4.2.5. Maintenance of the EAMT entries

The information in the EAMT MUST be kept timely-synchronized with the AAAA records TTL's, so the EAMT entries MUST expire on the AAAA TTL expiry and consequently be deleted.

However, EAMT entries with the Auto/Static bit set, will not be deleted.

4.2.6. Usage example

Using the same example as in the previous approach:

| | | |
|--|-----------|-------------------|
| www.example.com | A | 192.0.2.1 |
| | AAAA | 2001:db8::a:b:c:d |
| EAMT entry | 192.0.2.1 | 2001:db8::a:b:c:d |
| NAT46/CLAT translated to | | 2001:db8::a:b:c:d |
| CDN IPv6 interface already is | | 2001:db8::a:b:c:d |
| Operator already has a specific route to | | 2001:db8::a:b:c:d |

The following is an example of the CE behavior after the previous case has already created an EAMT entry and a reverse-proxy is detected:

1. A query for www.another-example.com A RR is received
2. www.another-example.com A 192.0.2.1
3. www.another-example.com AAAA 2001:db8::e:e:f:f

4. A conflict has been detected

5. The existing EAMT entry for 192.0.2.1 is set as invalid

4.2.7. Behavior in case of multiple A/AAAA RRs

If multiple A and/or AAAA records are available, the DNS proxy/stub resolver MUST follow existing procedures to choose each one. In other words, the chosen pair of A/AAAA records doesn't present any different result compared with a situation when this mechanism is not used.

4.2.8. Behavior in presence/absence of DNS64

This mechanism performs the same in both cases, if a DNS64 is present/used and if it is not present/used. This is explained because the mechanism is only relevant for destinations which don't have AAAA records, and in those cases DNS64 is not relevant. Furthermore, because as indicated in Section 4.2.2, the EAMT entry is not created when the service is IPv6-enabled. This is relevant because 464XLAT can be deployed/used with and without a DNS64.

4.2.9. Behavior when using literal addresses or non IPv6-compliant APIs

Because the EAMT entries are only created when the NAT46/CLAT/CE proxy/stub DNS is being used, any devices or applications that don't use DNS, will not create the relevant entries.

They will be however optimized if devices or applications using DNS, at some point, query for the same A RRs, or if EAMT entries are statically configured.

4.2.10. False detection of a dual-stack host as IPv4-only

If a dual-stack host is issuing the A query using IPv4 transport, and the AAAA query using IPv6 transport, or using different IPv4 addresses for the A and AAAA queries, the EAMT entry will be created. However, this EAMT entry may not be used by dual-stack devices or applications, because those devices or applications should prefer IPv6. If the host is preferring IPv4 for connecting to the CDN/cache or IPv6-enabled service, it will be actually using the NAT46/CLAT, including the EAMT entry and consequently IPv6, so this mechanism will be correcting an undesirable behavior. This is a special case, which actually seems to be an incoherent host or application implementation.

However, if other IPv4-only devices or applications subsequently need to connect to the same IPv6-enabled service, they will take advantage

of the already existing EAMT entry, and consequently use the IPv6-optimised path.

4.2.11. Behaviour in presence of Happy Eyeballs

Happy Eyeballs [RFC8305] is only available in dual-stack hosts. Consequently, is not affected by this mechanism because both, the A and the AAAA queries should be issued by the host as soon after one another as possible. However, if the same NAT46/CLAT/CE is serving IPv4-only hosts and dual-stack hosts and both of them are using the same destinations, an EAMT entry will be created for that destination. Consequently, a Happy Eyeballs fallback to IPv4 will actually be using the relevant EAMT entry IPv6 destination. This has the disadvantage that the IPv4-IPv6-IPv4 translation path can't be used by Happy Eyeballs-enabled applications. However, this may be actually considered as a good thing, in the sense that an operator is interested in knowing as soon as possible, if the IPv6-only network is not performing correctly, because that means also IPv4 will not be working. If the issue is related to extra IPv6 delay versus the IPv4 delay, Happy Eyeballs will not be able to offer a significative advantage here, but it looks like an acceptable trade-off.

Note that when using 464XLAT, the WAN link of the NAT46/CLAT/CE is IPv6-only. So even if Happy Eyeballs is present, the fallback to IPv4-only typically, will be slower than native IPv6 itself, because the added detail in the NAT46+NAT64 translations, when not using this optimization.

4.2.12. Behavior in case of Foreign DNS

Devices or applications may use DNS servers from other networks. For a complete description of reasons for that, refer to Section 4.4 of [I-D.ietf-v6ops-nat64-deployment]. In the case the DNS is modified, or some devices or applications use other DNS servers, the possible scenarios and the implications are:

- a. Devices configured to use a DNS proxy/resolver which is not the CE/NAT46/CLAT. In this case this optimization will not work, because the EAMT entry will not be created based on their own flows. Nevertheless, the EAMT entry may be created by other devices using the same destinations. However, the lack of EAMT entry, will not impact negatively in the user's devices/applications (the optimization is not performed). It should be noticed that users commonly, don't change the configuration of devices such as SmartTVs or STBs (if they do, some other functionalities, such as CDN/caches optimizations may not work as well), so this only happens typically if the vendor is doing it on-purpose and for good well-known reasons.

- b. DNS privacy/encryption. Hosts or applications that use mechanisms for DNS privacy/encryption, such as DoT ([RFC7858], [RFC8094]), DoH ([RFC8484]) or DoQ ([I-D.huitema-quic-dnsquic]), will not make use of the stub/proxy resolver, so the same considerations as for the previous case apply.
- c. Users that modify the DNS in their Operating Systems. This is quite frequent, however commonly Operating Systems are dual-stack, so aren't part of the problem statement described by this document and will not be adversely affected.
- d. Users that modify the DNS in the CE. This is less common. In this case, this optimization is not adversely affected, because it doesn't depend on the operator DNS, it works only based on the internal CE interaction between the NAT46/CLAT and the stub/proxy resolver. Note that it may be affected if the operator offers different "DNS views" or "split DNS", however this is not related to this optimization and will anyway impact in the other possible operator optimizations (i.e. CDN/cache features).
- e. Combinations of the above ones. No further impact, than the one already described, is observed.

4.3. Approach 3: NAT46/CLAT-provider-EAM-based Solution

Instead of using the DNS proxy/stub resolver to create the EAMT entries, the operator may push this table (or parts of it) into the CE/NAT46/CLAT, by using configuration/management mechanisms.

This solution has the advantage of not being affected by any DNS changes from the user (the EAMT is created by the operator) and ensures a complete control from the operator. However, it may impact the cases of devices with a DNS configured by the vendor.

In general, most of the considerations from the previous approach will apply.

One more advantage of this solution is that the EAMT pairs doesn't need to match the "real" IPv4/IPv6 addresses available in the A/AAAA records, as shown in the next example.

| | | |
|--|-----------|-------------------|
| www.example.com | A | 192.0.2.1 |
| | AAAA | 2001:db8::a:b:c:d |
| EAMT pulled/pushed entry | 192.0.2.1 | 2001:db8::f:e:d:c |
| NAT46/CLAT translated to | | 2001:db8::f:e:d:c |
| CDN IPv6 interface already is | | 2001:db8::f:e:d:c |
| Operator already has a specific route to | | 2001:db8::f:e:d:c |

EAMT may contain TTLs which probably are derived from DNS ones, or alternatively, a global TTL for the full table.

An alternative way to configure the table, is that the CE is actually pulling the table (or parts of it) from the operator infrastructure. In this case it will be mandatory that the entries have individual TTLs, again probably derived from the DNS ones.

The major drawback of this approach is that it requires a new protocol, or an extension to existing ones, in order to push or pull the EAMT, in addition to the possible impact in terms of bandwidth each time the CEs reboot, or an EAMT must be pushed to all the CEs, etc.

5. IPv6-only Services become accessible to IPv4-only devices/apps

One of the issues with the IPv6 deployment, is that those services which become IPv6-only in Internet, aren't reachable by IPv4-only devices and applications. This means that new content providers must support dual-stack even for new services, even while IPv4 public addresses aren't available.

If NAT46/CLAT/DNS-proxy-EAM approach (Section 4.2) is chosen, it can be complemented to resolve this issue, by means of making sure that IPv6-only destinations have one A resource record (even an invalid one), despite they aren't actually connected to IPv4. This will mean that those services will work fine if there is a NAT46/CLAT, and will have no impact if that one doesn't exist, not a different situation than not having an A resource record.

In fact, it may become an incentive for the IPv6 deployment in Internet services and provides the option to use an IPv4 address (maybe anycast) for the "non-valid" A resource record, that points to a "universal" web page (maybe hosted by IETF?) that displays a warning such as "Sorry, you don't IPv6 support in your operator, so this service is not available for you".

6. Conclusions

NAT46/CLAT/DNS-proxy-EAM approach (Section 4.2) seems the right solution for optimizing the access to dual-stack services, whether they are located inside or outside the ISP.

Having this type of optimization facilitates and increases the usage of IPv6, even for IPv4-only devices and applications, at the same time that decreases the use of the NAT64.

SIIT already has a SHOULD for EAM support. Should 464XLAT be updated

by this document so the CLAT has a MUST for EAM support?.

Should we recommend having A records for IPv6-only services in Internet? The A record may point to a "reserved" or "special" IPv4 address. A web page IPv4-only hosted by IETF(?) showing "sorry this web page/service is only available from IPv6 enabled operators"?.

Open question: Should we consider any other risks? If CE's implementing this optimization create troubles, it may bring the content providers to switch back to IPv4-only. So possible failure cases need to be carefully considered for every possible solution approach.

7. Security Considerations

This document does not have any new specific security considerations.

8. IANA Considerations

This document does not have any new specific IANA considerations, unless we decide to define a "special reserved IPv4 address".

9. Acknowledgements

The authors would like to acknowledge the inputs of Erik Nygren, Fred Baker, Martin Hunek and TBD ...

10. References

10.1. 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>>.
- [RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", STD 88, RFC 3596, DOI 10.17487/RFC3596, October 2003, <<https://www.rfc-editor.org/info/rfc3596>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

10.2. Informative References

- [I-D.huitema-quic-dnsquic]
Huitema, C., Shore, M., Mankin, A., Dickinson, S., and J. Iyengar, "Specification of DNS over Dedicated QUIC Connections", draft-huitema-quic-dnsquic-07 (work in progress), September 2019.

- [I-D.ietf-v6ops-nat64-deployment]
Palet, J., "Additional NAT64/464XLAT Deployment Guidelines in Operator and Enterprise Networks", draft-ietf-v6ops-nat64-deployment-08 (work in progress), July 2019.
- [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>>.
- [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>>.

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/>

Alejandro D'Egidio
Telecentro
Argentina

Email: adegidio@telecentro.net.ar