v6ops
Internet-Draft
Intended status: Informational
Expires: January 8, 2021

J. Palet Martinez
The IPv6 Company
A. D'Egidio
Telecentro

July 7, 2020

464XLAT/NAT64 Optimization
draft-ietf-v6ops-464xlat-optimization-01

Abstract

This document proposes possible solutions to avoid certain drawbacks of IP/ICMP Translation Algorithm (SIIT) when the destinations are already available with IPv6. When SIIT is used as a stateless NAT46 and IPv4-only devices or applications initiate traffic flows to dual-stack services (in the operator network or Internet), those flows will be translated back to IPv4 by a NAT64. Avoiding this dual-translation will significantly reduce the usage of the NAT64 and the relevant logging, which may be a high impact when traffic flows are towards CDNs, caches or similar resources. 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 January 8, 2021.

Table of Contents

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

stateless 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, while in the case of MAP-T is called NAT46.

The NAT46/CLAT (WAN side) is connected by IPv6-only to the operator network, which in turn, will have a reverse translation, 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.

In both cases (NAT46 and NAT64), the translation of the packet headers is done using the IP/ICMP translation algorithm defined in [RFC7915] and algorithmically translating the IPv4 addresses to IPv6 addresses following [RFC6052].

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. Possible Optimization

In the case of 464XLAT, a DNS64 ([RFC6147]) is (optionally) 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 NAT46/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, so they are always forced to use the double-translation.

This is the expected behavior, as the original design of NAT64 was targeted to connect IPv6-only devices (using DNS) to IPv4-only services. 464XLAT expanded the solution to also allow IPv4-only devices (even if not using DNS) to connect to IPv4-only services by means of IPv6-only access networks.

The optimization solutions presented by this document try to avoid this double-translation, in the cases when the Internet services, are already IPv6-enabled. So, in those cases, if the NAT46 already

translated the IPv4-only flow to IPv6, it doesn't look necessary to translate this back to IPv6.

A typical 464XLAT deployment is depicted in Figure 1.

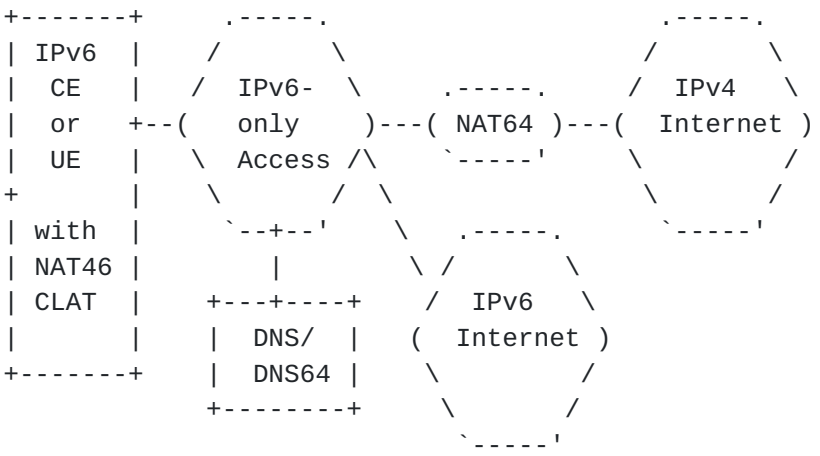


Figure 1: Typical 464XLAT Deployment

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, for example, to Content Delivery Networks (CDNs), caches, or other resources, and they are dual-stacked, the situation can be described as shown in Figure 2.

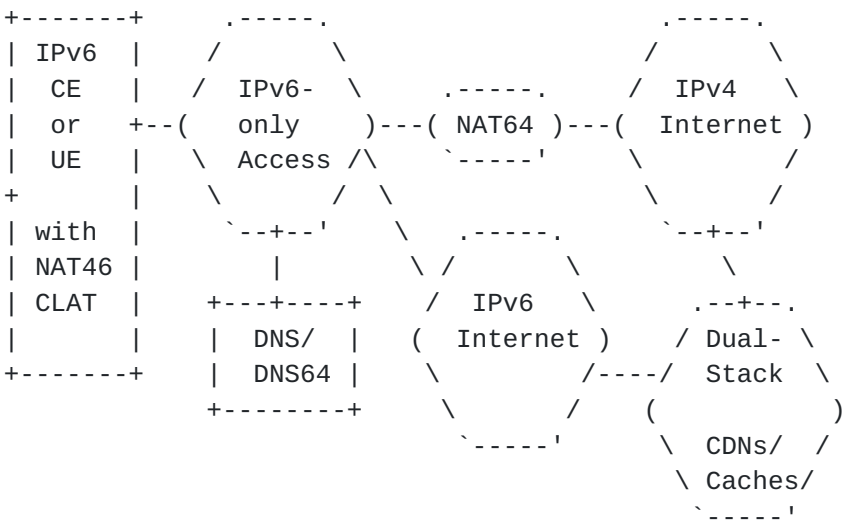


Figure 2: Typical 464XLAT Deployment with CDNs/Caches

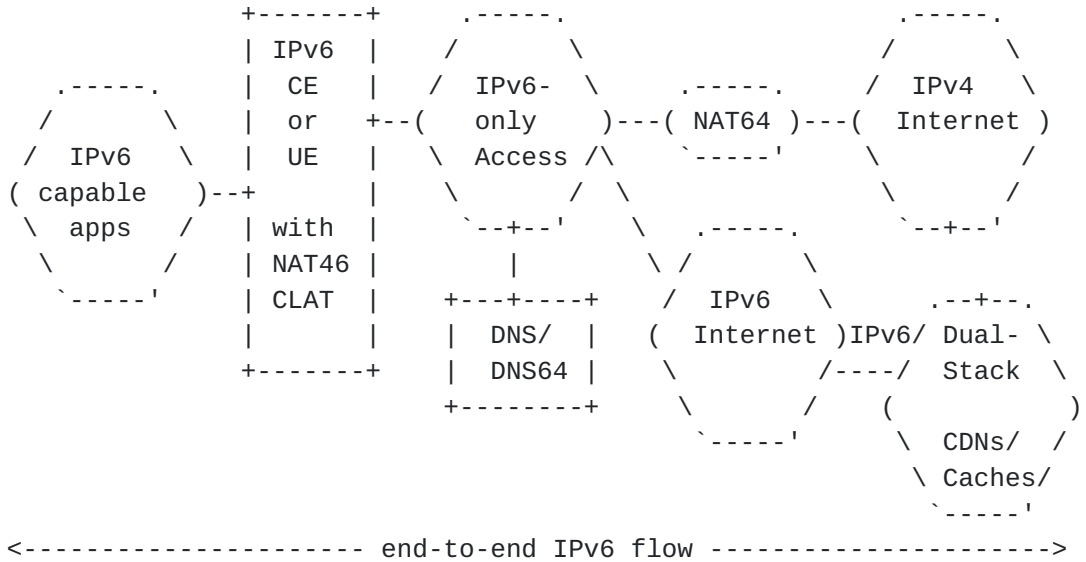
In this case if the flows initiated in the LANs come from IPv4-only devices or applications, even if the destination resources are IPv6-enabled, the double-translation is enforced, which has the following consequences:

- More traffic needs to pass thru the NAT64 devices.
- More NAT64 devices may be needed to handle the additional traffic.
- Additional usage of IPv4 addresses.
- Additional state at the NAT64 devices.
- Additional logging, its relevant storage and processing resources.

Clearly, all those aspects have impact in both, CapEx and OpEx. This is extremely important when considering that most of the time, the contents stored in CDNs, caches, and so on, is there for a good reason: It is frequently accessed resources and/or big. Examples such as video, audio, software and updates, are very common. So, this optimization can be highly impacting in many networks.

4. 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.



However, if the devices or applications are IPv4-only, for example, many 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.

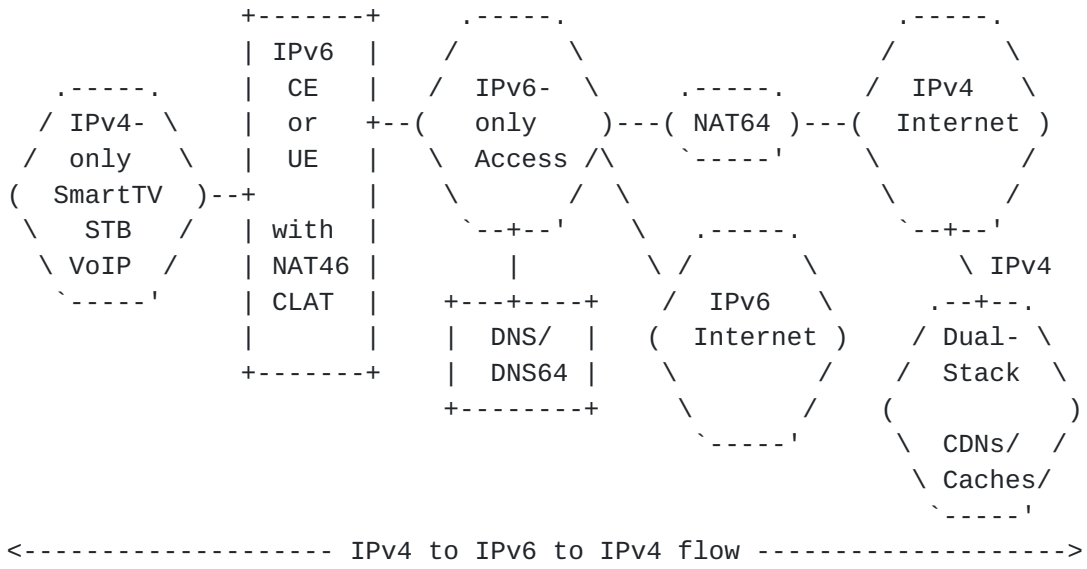


Figure 4: 464XLAT access to CDNs/Caches by IPv4-only apps

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.

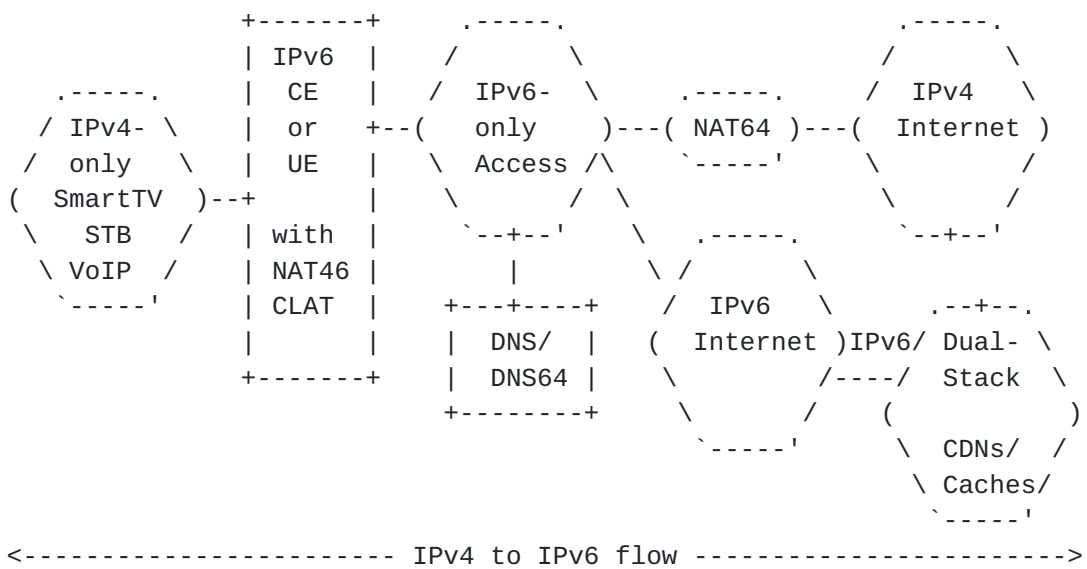


Figure 5: Optimized 464XLAT access to CDNs/Caches by IPv4-only apps

5. Solution Approaches

5.1. 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] and IPv4 documentation addresses following [RFC5737].

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.

5.2. 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.

5.2.1. 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).

5.2.2. 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]).

5.2.3. 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 set to the same value as in 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 explicitly configured 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.

Note the newly defined EAMT fields, follow the "extensions" approach as per section 3.1 of [RFC7757].

5.2.4. 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.

5.2.5. 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. This allows users/operators to set explicit rules for diagnostics or resolution of issues in special situations.

5.2.6. 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

5.2.7. 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.

5.2.8. Behavior in presence/absence of DNS64

This optimization performs the same in both cases: if a DNS64 is present/used or if it is not present/used. This is explained because the optimization is only relevant for destinations which already have

AAAA records, and in those cases DNS64 is not relevant. Furthermore, because as indicated in Section 5.2.2, 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.

5.2.9. 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 maybe optimized if devices or applications using DNS, at some point, query for the same A RRs, or if EAMT entries are statically configured.

5.2.10. 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.

5.2.11. Behavior 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 delay in the NAT46+NAT64 translations, when not using this optimization.

5.2.12. 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 [RFC8683]. 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-dprive-dnsoquic]), 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.

5.3. 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.

6. 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 5.2) is chosen, it also offers the chance to resolve this issue. This is possible if IPv6-only services get configured with an A resource record pointing to a well-known IPv4 address despite they aren't actually connected to IPv4. This is out of scope for this document, as it will require further work and a reservation by IANA, This will mean that those services will work fine if there is a NAT46/CLAT supporting the optimization. This A RR has no negative impact if the NAT46/CLAT doesn't exist, or it is not optimized, because is not reachable via IPv4-only, so is not a different situation compared with not having an A RR.

The result of this is equivalent to the approach taken by MAP-T (Section 12.3 of [RFC7599]). However, it has the advantage that the MAP-T approach is restricted to services in the MAP-T domain.

In fact, it may become an incentive for the IPv6 deployment in Internet services, as it provides the option to use a well-known IPv4 address (maybe anycast) for the "non-valid" A RR, that points, in case of port 80/443 to a web page or service that returns a warning such as "This service is only available if the network is properly connected to Internet with IPv6".

7. Conclusions

NAT46/CLAT/DNS-proxy-EAM approach (Section 5.2) seems the right solution for optimizing the access to dual-stack services, whether they are located inside or outside the ISP. It is also the only approach which has no additional requirements for the network operators (both ISPs and CDN/cache operators).

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, so it is not a high additional burden the support required for existing implementations to be updated for this optimization.

8. Security Considerations

This document does not have any new specific security considerations, besides the ones already known for DNS64.

9. IANA Considerations

This document does not have any new specific IANA considerations.

10. Acknowledgements

The authors would like to acknowledge the inputs of Erik Nygren, Fred Baker, Martin Hunek, Chongfeng Xie, Fernando Gont, Fernando Frediani and TBD ...

11. References

11.1. 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>.

11.2. Informative References

[I-D.huitema-dprive-dnsoquic] Huitema, C., Mankin, A., and S. Dickinson, "Specification of DNS over Dedicated QUIC Connections", draft-huitema-dprive-dnsoquic-00 (work in progress), March 2020.

[RFC5737] Arkko, J., Cotton, M., and L. Vegoda, "IPv4 Address Blocks Reserved for Documentation", RFC 5737, DOI 10.17487/RFC5737, January 2010, <https://www.rfc-editor.org/info/rfc5737>.

[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>.

[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>.

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