Pros and Cons of IPv6 Transition Technologies for IPv4aaS

draft-ietf-v6ops-transition-comparison-04

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 24 November 2022.
Internet-Draft Pros and Cons of IPv4aaS Technologies May 2022

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction ................................................. 3
2. Overview of the Technologies ................................. 4
   2.1. 464XLAT ................................................. 4
   2.2. Dual-Stack Lite ......................................... 5
   2.3. Lightweight 4over6 ..................................... 6
   2.4. MAP-E .................................................. 7
   2.5. MAP-T .................................................. 8
3. High-level Architectures and their Consequences .......... 9
   3.1. Service Provider Network Traversal ...................... 9
   3.2. Network Address Translation among the Different IPv4aaS Technologies ............................................. 10
   3.3. IPv4 Address Sharing ................................... 11
   3.4. IPv4 Pool Size Considerations .......................... 12
   3.5. CE Provisioning Considerations .......................... 14
   3.6. Support for Multicast ................................... 14
4. Detailed Analysis ............................................. 15
   4.1. Architectural Differences ............................... 15
       4.1.1. Basic Comparison .................................. 15
   4.2. Tradeoff between Port Number Efficiency and Stateless Operation ..................................................... 15
   4.3. Support for Public Server Operation ..................... 18
   4.4. Support and Implementations ............................. 19
       4.4.1. Vendor Support ..................................... 19
       4.4.2. Support in Cellular and Broadband Networks .......... 19
       4.4.3. Implementation Code Sizes .......................... 20
   4.5. Typical Deployment and Traffic Volume Considerations .. 20
       4.5.1. Deployment Possibilities .......................... 20
As the deployment of IPv6 continues to be prevalent, it becomes clearer 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 standardised 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 IPv4aaS 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.

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 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).

The customer-side translator (CLAT) is located in the customer's device, and it performs stateless NAT46 translation [RFC7915] (more precisely, 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 deployments where NAT64 load-balancing [RFC7269] section 4.2 is
enabled, multiple WKPs [RFC8215] may be used.

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.

In general, keeping state in devices close to the end-user network (i.e. at the CE - Customer Edge router) is not perceived as problematic as keeping state in the operator's 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, and the CLAT functions as an IPv6 router and the PLAT performs a stateful NAT64 for these packets. In this case, there is a single translation.

Similarly, when an IPv4-only client or application communicates with an IPv4-only server, the CLAT will statelessly translate to IPv6 so it can traverse the ISP network up to the PLAT (NAT64), which in turn will translate to IPv4.

Alternatively, one can say that 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.
Note: in mobile networks, CLAT is commonly implemented in the user's equipment (UE or smartphone), please refer to Figure 2 of [RFC6877].

Often NAT64 vendors support direct communication (that is, without translation) between two CLATs by means of hair-pinning through the NAT64.

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 Bridging BroadBand' (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 an 'Address Family Transition Router' (AFTR). The AFTR performs encapsulation/decapsulation of the 4in6 [RFC2473] traffic and translates the IPv4 source address in the inner IPv4 packet to public IPv4 source address using a stateful NAPT44 [RFC2663] function.

Some AFTR vendors support direct communication between two B4s by means of hair-pinning through the AFTR.
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 [RFC6346] 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 and a unique range of transport layer 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 transport layer 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 (that is, without translation) between two lwB4s is performed by hair-pinning traffic through the lwAFTR.
2.4. MAP-E

Like 464XLAT (Section 2.1), MAP-E and MAP-T use [RFC6052] IPv4-embedded IPv6 addresses to represent IPv4 hosts outside the MAP domain.

MAP-E and MAP-T use 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 to the delegated IPv6 prefix. The client address/port allocation size is a configuration 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 transport layer ports assigned, as well as how to calculate the destination IPv6 address for ingress IPv4 traffic.
Some BR vendors support direct communication between two MAP CEs by means of hair-pinning through the BR.

### 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 router 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

Some BR vendors support direct communication between two MAP CEs by means of hair-pinning through the BR.

3. High-level Architectures and their Consequences

3.1. Service Provider Network Traversal

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

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

| +================================+=+================================+=+================================+=+================================+=+================================+=+ |
| | 464XLAT | DS-Lite | lw4o6 | MAP-E | MAP-T | |
| +================================+=+================================+=+================================+=+================================+=+================================+=+ |
| 4-6-4 trans. | | | | | |
| +================================+=+================================+=+================================+=+================================+=+================================+=+ |
| 4-6-4 encap. | | | | | |
| +================================+=+================================+=+================================+=+================================+=+================================+=+ |

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 overhead.

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 PLAT/AFTR/lwAFTR/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 (it should have sufficiently large MTU to support tunneling and/or translation) 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 among the Different IPv4aaS Technologies

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 transport layer next-header. The fields in these headers can be used for functions such as hashing across equal-cost multipaths or access control list (ACL) filtering. Encapsulation technologies, in contrast, may hinder hashing algorithms or other functions relying on header inspection.

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].

As stateful NAT interferes with the port numbers, [I-D.ietf-tsvwg-natsupp] explains how NATs can handle SCTP (Stream Control Transmission Protocol).

3.3. IPv4 Address Sharing

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

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

The solutions compared by this document fall into two categories:

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

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

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

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

The address sharing efficiency of the five technologies is significantly different, it is discussed in Section 4.2.
lw4o6, MAP-E and MAP-T can also be configured without IPv4 address sharing, see the details in Section 4.3. However, in that case, there is no advantage in terms of public IPv4 address saving. In the case of 464XLAT, this can be achieved as well through EAMT (Explicit Address Mapping Table) [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. IPv4 Pool Size Considerations

In this section we do some simple calculations regarding port numbers, however, technical limitations are not the only point to consider for port sharing, there are also local regulations or BCP.

Note: under "port numbers", we mean TCP/UDP port numbers or ICMP identifiers.

In most networks, it is possible to, using existing data about flows to CDNs/caches or other well-known IPv6-enabled destinations, calculate the percentage of traffic that would turn into IPv6 if it is enabled on that network or part of it.

Knowing that, it is possible to calculate the IPv4 pool size required for a given number of subscribers, depending on the IPv4aaS technology being used.

According to [MIY2010], each user-device (computer, tablet, smartphone) behind a NAT, could simultaneously use up to 300 ports. (Table 1 of [MIY2010] lists the port number usage of various applications. According to [REP2014] the downloading of some web pages may consume up to 200 port numbers.) If the extended NAPT algorithm is used, which includes the full five tuple into the connection tracking table, then the port numbers are reused, when the
destinations are different. Therefore, we need to consider the number of "port hungry" applications that are accessing the same destination simultaneously. We estimate that in the case of a residential subscriber, there will be typically no more than 4 of port hungry applications communicating with the same destination simultaneously, which means a total of 1,200 ports.

If for example, 80% of the traffic is expected towards IPv6 destinations, only 20% will actually be using IPv4 ports, so in our example, that will mean 240 ports required per subscriber.

From the 65,535 ports available per IPv4 address, we could even consider reserving 1,024 ports, in order to allow customers that need EAMT entries for incoming connections to System Ports (0-1023, also called well-known ports) [RFC7605], which means 64,511 ports actually available per each IPv4 address.

According to this, a /22 (1,024 public IPv4 addresses) will be sufficient for over 275,000 subscribers (1,024x64,511/240=275,246.93).

Similarly, a /18 (16,384 public IPv4 addresses) will be sufficient for over 4,403,940 subscribers, and so on.

This is a conservative approach, which is valid in the case of 464XLAT, because ports are assigned dynamically by the NAT64, so it is not necessary to consider if one user is actually using more or less ports: Average values work well.

As the deployment of IPv6 progresses, the use of NAT64, and therefore of public IPv4 addresses, decreases (more IPv6/ports, less IPv4/ports), so either more subscribers can be accommodated with the same number of IPv4 addresses, or some of those addressed can be retired from the NAT64.

For comparison, if dual-stack is being used, any given number of users will require the same number of public IPv4 addresses. For instance, a /14 will provide 262,144 IPv4 public addresses for 262,144 subscribers, versus 275,000 subscribers being served with only a /22.

In the other IPv4aaS technologies, this calculation will only match
if the assignment of ports per subscriber can be done dynamically, which is not always the case (depending on the vendor implementation).

An alternative approximation for the other IPv4aaS technologies, when dynamically assignment of addresses is not possible, must ensure sufficient number of ports per subscriber. That means 1,200 ports, and typically, it comes to 2,000 ports in many deployments. In that case, assuming 80% of IPv6 traffic, as above, which will allow only 30 subscribers per each IPv4 address, so the closer approximation to 275,000 subscribers per our example with 464XLAT (with a /22), will be using a /19, which serves 245,760 subscribers (a /19 has 8,192 addresses, 30 subscribers with 2,000 ports each, per address).

If the CGN (in case of DS-Lite) or the CE (in case of lw4o6, MAP-E and MAP-T) make use of a 5-tuple for tracking the NAT connections, the number of ports required per subscriber can be limited as low as 4 ports per subscriber. However, the practical limit depends on the desired limit for parallel connections that any single host behind the NAT can have to the same address and port in Internet. Note that it is becoming more common that applications use AJAX (Asynchronous JavaScript and XML) and similar mechanisms, so taking that extreme limit is probably not a safe choice.

This extremely reduced number of ports "feature" could also be used in case the CLAT-enabled CE with 464XLAT makes use of the 5-tuple NAT connections tracking, and could also be further extended if the NAT64 also use the 5-tuple.

3.5. 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.

| +===============================================================+ |
Table 2: Available Provisioning Mechanisms


X: Supported by the provisioning method.

3.6. 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.
4.1.1. Basic Comparison

The five IPv4aaS technologies can be classified into \(2 \times 2 = 4\) categories on the basis of two aspects:

* Technology used for service provider network traversal. It can be single/double translation or encapsulation.

* Presence or absence of NAPT44 per-flow state in the operator network.

<table>
<thead>
<tr>
<th>Translation (T) or Encapsulation (E)</th>
<th>464XLAT</th>
<th>DS-Lite</th>
<th>lw4o6</th>
<th>MAP-E</th>
<th>MAP-T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per-flow state in op. network</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Basic comparison among the analyzed technologies

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 (and in the Belgian case it is not a law but more a MoU / BCP), 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.

In general, assigning too few source port numbers to an end user may result in unexpected and hard to debug consequences, therefore, if the number of ports per end user is fixed, then we recommend to assign a conservatively large number of ports. E.g. the developers of Jool used 2048 ports per user in their example for MAP-T [MEX2022].

However, 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.

Measurements have been started to examine the scalability of a few stateful solutions in two areas:

* How their performance scales up with the number of CPU cores?
* To what extent their performance degrades with the number of concurrent connections?

The details of the measurements and their results are available from [I-D.lencse-v6ops-transition-scalability].

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 blocking 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 transport layer ports.

Port Control Protocol (PCP) [RFC6887] 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 transport layer 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 transport layer 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. Vendor Support

In general, router vendors support AFTR, MAP-E/T BR and NAT64. Load balancer and firewall vendors usually support NAT64 as well, while not all of them have support for the other protocols.

A 464XLAT client (CLAT) is implemented in Windows 10, Linux (including Android), Windows Mobile, Chrome OS and iOS, but it is not available in macOS 12.3.1.

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

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

* '464xlat' enables support for 464XLAT CLAT functionality
* 'ds-lite' enables support for DSLite B4 functionality
* 'map' enables support for MAP-E and lw4o6 CE functionality
* 'map-t' enables support for MAP-T CE functionality
At the time of publication some free open-source implementations exist for the operator side functionality:

* Jool [jool](jool) (CLAT, NAT64, EAMT, MAP-T CE, MAP-T BR).
* VPP/fd.io [vpp](vpp) (MAP-BR, lwAFTR, CGN, CLAT, NAT64).
* Snabb [snabb](snabb) (lwAFTR).
* AFTR [aftr](aftr) (DS-Lite AFTR).

### 4.4.2. Support in Cellular and Broadband Networks

Several cellular networks use 464XLAT, whereas there are no deployments of the four other technologies in cellular networks, as they are neither standardised nor 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](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](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

Figures from existing deployments (end of 2018), show that the typical traffic volumes in an IPv6-only cellular network, when 464XLAT technology is used together with DNS64, are:

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

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

4.5.3. Wireline Networks with 464XLAT

Figures from several existing deployments (end of 2020), mainly with residential customers, show that the typical traffic volumes in an IPv6-only network, when 464XLAT is used with DNS64, are in the following ranges:

* 65%-85% of traffic is IPv6 end-to-end (no translation)
* 14%-34% of traffic uses DNS64 + NAT64 (1 translation)
* Less than 1-2% of traffic uses the CLAT in addition to NAT64 (2 translations)
translations), due to an IPv4 socket and/or IPv4 literal.

Without using DNS64, 16%-35% of the traffic would undergo double translation.

This data is consistent with non-public information of actual deployments, which can be easily explained. When a wireline ISP has mainly residential customers, content providers and CDNs which are already IPv6 enabled (Google/Youtube, Netflix, Facebook, Akamai, etc) typically account for the 65-85% of the traffic in the network, so when the subscribers are IPv6 enabled, about the same figures of traffic will become IPv6.

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,
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/AFTTR'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 a logging entry only needs to be made when the client's port block is released, which dramatically reduces the logging overhead. This comes as the cost of less efficient public address sharing as clients need to be allocated a port block of a fixed size regardless of the actual number of ports that they are using.

Stateless technologies that pre-allocate the IPv4 addresses and ports only require that copies of the active MAP rules (for MAP-E and MAP-T), or binding-table (for lw4o6) are retained along with timestamp information of when they have been active. Support tools (e.g., those used to serve data retention requests) may need to be updated to be aware of the mechanism in use (e.g., implementing the MAP algorithm so that IPv4 information can be linked to the IPv6 prefix delegated to a client). As stateless technologies do not have a centralized stateful element which customer traffic needs to pass through, so if data retention laws mandate per-session logging, there is no simple way of meeting this requirement with a stateless technology alone. Thus, a centralized NAPT44 model may be the only way to meet this requirement.
Deterministic CGN [RFC7422] was proposed as a solution to reduce the resource consumption of logging.

Please also refer to Section 4 of [RFC6888] for more information about requirements for logging Carrier-Grade NAT gateways.

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

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

5. Performance Comparison

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

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

* Dual DUT setup has the drawback that the performances of the CE and of the ISP side device (e.g. the CLAT and the PLAT of 464XLAT) are measured together. In order to measure the performance of only one of them, we need to ensure that the desired one is the bottleneck.
Measurements procedures for PDV and IPDV measurements are missing from the legacy devices, and the old measurement procedure for Latency has been redefined in [RFC8219].

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

Further DPDK-based, [RFC8219] compliant software testers are being developed at the Budapest University of Technology and Economics as student projects. They are planned to be released as free software, too.

Information about the benchmarking tools, measurements and results will be made available in [I-D.lencse-v6ops-transition-benchmarking].

6. Acknowledgements

The authors would like to thank Ole Troan, Warren Kumari, Dan Romascanu, Brian Trammell, Joseph Salowey, Roman Danyliw, Erik Kline, Lars Eggert, Zaheduzzaman Sarker, Robert Wilton, Eric Vyncke and Martin Duke for their 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, Yutianpeng (Tim), Mohamed Boucadair, Nick Hilliard, Joel Jaeggli, Kristian McColm, Nick Hilliard, Tom Petch, Yannis Nikolopoulos, Havard Eidnes, Yann-Ju Chu, Barbara Stark, Vasilenko Eduard, Chongfeng Xie, Henri Alves de Godoy, Magnus Westerlund, Michael Tuexen, Philipp S. Tiesel, Brian E Carpenter and Joe Touch.
7. IANA Considerations

This document does not make any request to IANA.

8. Security Considerations

As discussed in section 4.7, the different technologies have varying logging capabilities and limitations. Care should be taken when storing, transmitting, and providing access to log entries that may be considered personally identifiable information. However, it should be noticed that those issues are not specific to the IPv4aaS IPv6 transition technologies, but in general to logging functionalities.

For all five technologies, the CE device typically contains 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.

IPv4aaS technologies based on encapsulation have not DNSSEC implications. However, those based on translation may have implications as discussed in Section 4.1 of [RFC8683].

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, an initial security analysis of 464XLAT was done in [Azz2021].

The implementers of any of the five IPv4aaS solutions should consult
the Security Considerations of the respective RFCs documenting them.

9. References

9.1. Normative References


[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,

[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,

Transition Mechanisms during IPv6 Deployment", RFC 6180,
DOI 10.17487/RFC6180, May 2011,

P. Roberts, "Issues with IP Address Sharing", RFC 6269,
DOI 10.17487/RFC6269, June 2011,

[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,

the IPv4 Address Shortage", RFC 6346,
DOI 10.17487/RFC6346, August 2011,

[RFC6519] Maglione, R. and A. Durand, "RADIUS Extensions for Dual-
Stack Lite", RFC 6519, DOI 10.17487/RFC6519, February

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT:
Combination of Stateful and Stateless Translation",


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,


9.2. Informative References


[MII2010] Miyakawa, S., "IPv4 to IPv6 transformation
Appendix A. Change Log

A.1. 01 - 02

* Ian Farrer has joined us as an author.

* Restructuring: the description of the five IPv4aaS technologies was moved to a separate section.

* More details and figures were added to the description of the five IPv4aaS technologies.

* Section titled "High-level Architectures and their Consequences" has been completely rewritten.
* Several additions/clarification throughout Section titled "Detailed Analysis".

* Section titled "Performance Analysis" was dropped due to lack of results yet.

* Word based text ported to XML.

* Further text cleanups, added text on state sync and load balancing. Additional comments inline that should be considered for future updates.

A.2. 02 - 03

* The suggestions of Mohamed Boucadair are incorporated.

* New considerations regarding possible optimizations.

A.3. 03 - 04

* Section titled "Performance Analysis" was added. It mentions our new benchmarking tool, siitperf, and highlights our plans.

* Some references were updated or added.

A.4. 04 - 05

* Some references were updated or added.

A.5. 05 - 06

* Some references were updated or added.

A.6. 06 - 00-WG Item

* Stats dated and added for Broadband deployments.

* Other clarifications and references.

* New section: IPv4 Pool Size.
* Typos.

A.7. 00 - 01

To facilitate WGLC, the unfinished parts were moved to two new drafts:

* New I-D for scale up measurements. (Including the results of iptables.)

* New I-D for benchmarking measurements. (Only a stub.)

A.8. 01 - 02

Update on the basis of the AD review.

Update of the references.


Internet-Draft Pros and Cons of IPv4aaS Technologies May 2022

A.9. 02 - 03

Nits and changes from IESG review.

Updated wrong reference to PCP.

A.10. 03 - 04

Update to address the comments of further reviews.

Updated Acknowledgements.

Authors' Addresses

Gabor Lencse
Budapest University of Technology and Economics
Budapest
Magyar tudosok korutja 2.
H-1117
Hungary
Email: lencse@hit.bme.hu
URI: http://www.hit.bme.hu/~lencse/index_en.htm