Internet Engineering Task Force (IETF)
Request for Comments: 9313
Category: Informational
ISSN: 2070-1721

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

October 2022

# Pros and Cons of IPv6 Transition Technologies for IPv4-as-a-Service (IPv4aaS)

## 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. These technologies have their advantages and disadvantages. 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 and considers 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 document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9313.



## Table of Contents

## 1. Introduction

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 standardized a number of different IPv4aaS technologies for this (see [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:

1. 464XLAT [RFC6877]

2. Dual-Stack Lite [RFC6333]

3. Lightweight 4over6 (lw4o6) [RFC7596]

4. Mapping of Address and Port with Encapsulation (MAP-E) [RFC7597]

5. Mapping of Address and Port using Translation (MAP-T) [RFC7599]

We note that [RFC6180] gives guidelines for using IPv6 transition mechanisms during IPv6 deployment; that document addresses 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 and describe 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 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 (WKP) or a Network-Specific Prefix) is used as the IPv6 destination for the IPv4-embedded client traffic.

In deployments where NAT64 load balancing (see Section 4.2 of [RFC7269]) 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 to be 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, the CLAT functions as an IPv6 router, and the PLAT performs a stateful NAT64 for these packets. 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.

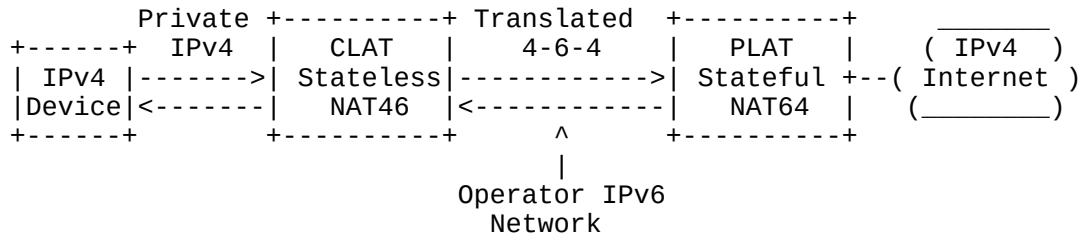


Figure 1: Overview of the 464XLAT Architecture

Note: In mobile networks, the CLAT is commonly implemented in the user equipment (UE) or smartphone; please refer to Figure 2 in [RFC6877].

Some NAT64 vendors support direct communication (that is, without translation) between two CLATs by means of hairpinning 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 a public IPv4 source address using a stateful NAPT44 [RFC2663] function.

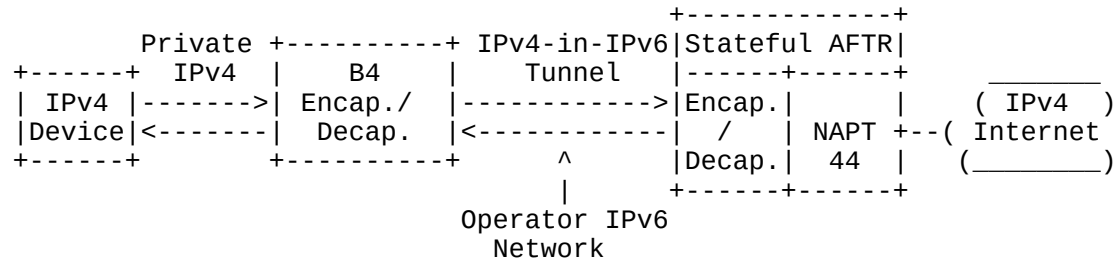


Figure 2: Overview of the DS-Lite Architecture

Some AFTR vendors support direct communication between two B4s by means of hairpinning 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 an "lwB4"). The AFTR (called an "lwAFTR") function therefore only performs A+P (Address plus Port) routing [RFC6346] and 4in6 encapsulation/ decapsulation.

Routing to the correct client and IPv4 address sharing are achieved using the 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 an 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 hairpinning traffic through the lwAFTR.

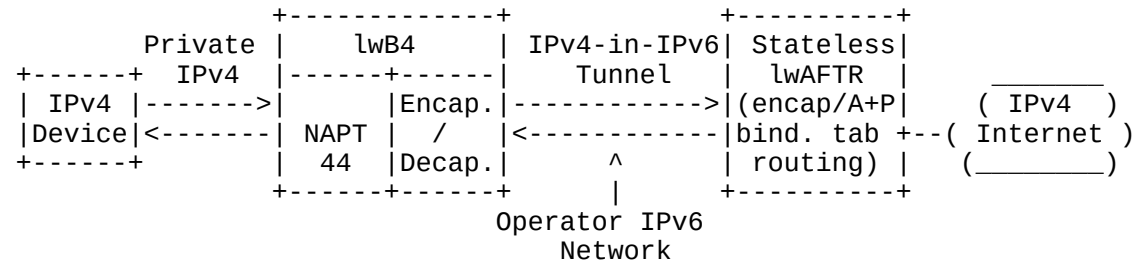


Figure 3: Overview of the lw4o6 Architecture

## 2.4. MAP-E

Like 464XLAT (Section 2.1), MAP-E and MAP-T use IPv4-embedded IPv6 addresses [RFC6052] 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 direct IPv4 peer-to-peer communication between hosts provisioned with common MAP rules.

The CE 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 the client, validate the source IPv4 address and transport-layer ports assigned, and calculate the destination IPv6 address for ingress IPv4 traffic.

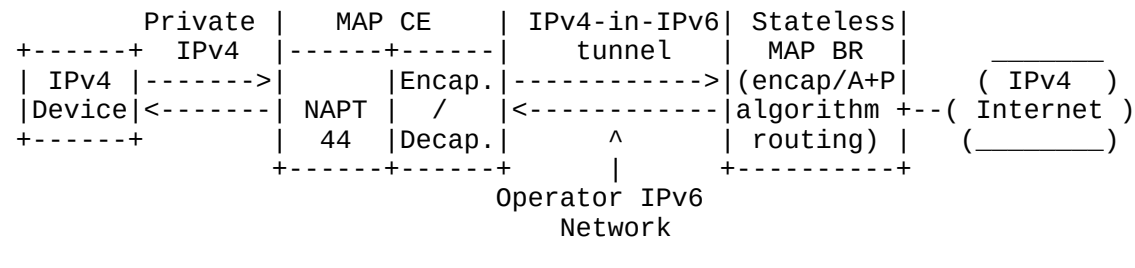


Figure 4: Overview of the MAP-E Architecture

Some BR vendors support direct communication between two MAP CEs by means of hairpinning 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), which is 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 translated (using stateless NAT64) 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.

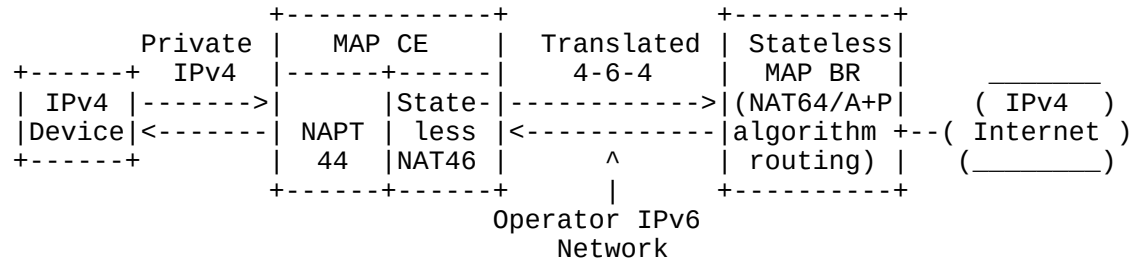


Figure 5: Overview of the MAP-T Architecture

Some BR vendors support direct communication between two MAP CEs by means of hairpinning 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

* 4in6 encapsulation

| | 464XLAT | DS-Lite | lw4o6 | MAP-E | MAP-T |
|---|---|---|---|---|---|
| 4-6-4 translation | X | | | | X |
| 4in6 encapsulation | | X | X | X | |

Table 1: Available Traversal Mechanisms

In the scope of this document, all of the encapsulation-based mechanisms use IP-in-IP tunneling [RFC2473]. This is a stateless tunneling mechanism that 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-byte 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 the need for traffic 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 reassembly in the PLAT/AFTR/lwAFTR/BR node.

The advice given in Section 8.3.1 of [RFC7597] 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. The first translation is from IPv4 to IPv6 (NAT46) at the CE, and the second translation is 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 and UDP) and ICMP when they are used with IPv4 address sharing (please refer to Section 4.3 for more details). Encapsulation-based solutions can also carry any other protocols over IP.

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

As stateful NAT interferes with the port numbers, [NAT-SUPP] 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 that allows 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:

* Approaches based on Carrier-Grade NAT (CGN) (DS-Lite, 464XLAT)

* Approaches based on A+P (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 for 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 and 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, one-to-one mapping can also be achieved 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 address with 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 and best current practices.

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

In most networks, it is possible to use existing data about flows to Content Delivery Networks (CDNs), caches, or other well-known IPv6-enabled destinations to calculate the percentage of traffic that would turn into IPv6 if IPv6 is enabled on that network or on 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 5-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 four port-hungry applications communicating with the same destination simultaneously, which is a total of 1,200 ports.

For example, if 80% of the traffic is expected towards IPv6 destinations, only 20% will actually be using IPv4 ports. Thus, in our example, 240 ports are required for each subscriber.

From the 65,535 ports available per IPv4 address, we could even consider reserving 1,024 ports for customers that need EAMT entries for incoming connections to System ports (0-1023, also called "well-known ports") [RFC7605]. This means that 64,511 ports are actually available for 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. Therefore, it is not necessary to consider if one user is actually using more or fewer 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, fewer IPv4 ports). Thus, 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).

When dynamic assignment of addresses is not possible, an alternative approximation for the other IPv4aaS technologies must ensure a 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% is IPv6 traffic (as above), only 30 subscribers will be allowed per each IPv4 address; thus, 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 and 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 four 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 feature of extremely reduced number of ports could also be used in case the CLAT-enabled CE with 464XLAT makes use of tracking the 5-tuple NAT connections and could also be further extended if the NAT64 also uses the 5-tuple.

Please also refer to [RFC6888] for in-depth information about the

requirements for sizing CGN gateways.

### 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.

| Provisioning Method | 464XLAT | DS-Lite | lw4o6 | MAP-E | MAP-T |
|---|---|---|---|---|---|
| DHCPv6 [RFC8415] | | X | X | X | X |
| RADIUS [RFC8658] | | [RFC6519] | X | X | X |
| TR-069 [TR-069] | * | X | * | X | X |
| DNS64 [RFC7050] | X | | | | |
| YANG [RFC7950] | [RFC8512] | [RFC8513] | [RFC8676] | [RFC8676] | [RFC8676] |
| DHCP 4o6 [RFC7341] | | | X | X | |

Table 2: Available Provisioning Mechanisms

*: Work started at Broadband Forum (2021)

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. Detailed Analysis

### 4.1. Architectural Differences

#### 4.1.1. Basic Comparison

The five IPv4aaS technologies can be classified based on two aspects:

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

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

| | 464XLAT | DS-Lite | lw4o6 | MAP-E | MAP-T |
|---|---|---|---|---|---|
| Translation (T) or Encapsulation (E) | T | E | E | E | T |
| Presence (+) of Per-Flow State in Operator Network | + | + | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | |

Table 3: Basic Comparison among the Analyzed Technologies

## 4.2. Trade-Off between Port Number Efficiency and Stateless Operation

464XLAT and DS-Lite use stateful NAPT at the PLAT and AFTR devices, respectively. This may cause scalability issues for the number of clients or volume of traffic, but it 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 and 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 reprovisioning 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 reprovisioned (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 public port allocations for customers and the corresponding logging overhead that may be necessary to meet data-retention requirements. This is considered in Section 4.7.

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 memorandum of understanding or best current practice), 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 in 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 at 230-270 ports).

The port number consumption of different applications is highly varying. In the case of web browsing, it depends on several factors, including the choice of the web page, the web browser, and sometimes 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, only 3-12 ports were used (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 assigning a conservatively large number of ports. For example, the developers of Jool used 2048 ports per user in their example for MAP-T [JOOL-MAPT].

However, assigning too many ports per CE router will result in waste of public IPv4 addresses, which are scarce and expensive resources. 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 doesn't need to be pre-defined and is not a limitation.

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

We note that common NAT44 implementations utilizing Netfilter at the CE router multiplex 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 addresses 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, DS-Lite, and 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 noted 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 [IPv4aaS-SCALE-TECH].

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, and content delivery networks). However, it should be noted 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. For example, many applications and service providers (Sony PlayStation Network, OpenDNS, etc.) can permanently block 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.

The 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 464XLAT/NAT64, 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 operators commonly 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, there is obviously 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 BR, MAP-T BR, and NAT64. Vendors of load balancers and firewalls 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, the vendors' support is poor in general (except for DS-Lite).

OpenWRT is a Linux-based open-source OS designed for CE devices. It 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] (CLAT, NAT64, EAMT, MAP-T CE, MAP-T BR)

* VPP/fd.io [VPP] (MAP-BR, lwAFTR, CGN, CLAT, NAT64)

* Snabb [SNABB] (lwAFTR)

* AFTR [AFTR] (DSLite 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 standardized 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 deployments of lw4o6 have been rapidly increasing in the last few years.

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

#### 4.4.3. Implementation Code Sizes

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

We note that the support for all five technologies requires a much smaller code size than the total sum of the above quantities, because they contain a lot of common functions (e.g., 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 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 to also deploy 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 (through the end of 2018) show the typical traffic volumes in an IPv6-only cellular network when 464XLAT technology is used together with DNS64:

* 75% of traffic is IPv6 end-to-end (no translation).

* 24% of traffic uses DNS64 + NAT64 (one translation).

* Less than 1% of traffic uses the CLAT in addition to NAT64 (two 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 (through the end of 2020), mainly with residential customers, show the ranges of typical traffic volumes in an IPv6-only network, when 464XLAT is used with DNS64:

* 65%-85% of traffic is IPv6 end-to-end (no translation).

* 14%-34% of traffic uses DNS64 + NAT64 (one translation).

* Less than 1-2% of traffic uses the CLAT in addition to NAT64 (two 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 that are already IPv6 enabled (Google/YouTube, Netflix, Facebook, Akamai, etc.) typically account for 65-85% of the traffic in the network. Thus, when the subscribers are IPv6 enabled, about the same percentage 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 Multipath) 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. Each CGN maintains a stateful table of active client sessions, and this table may need to be synchronized between CGN instances. This is necessary for two reasons:

* To prevent all active customer sessions from being dropped in the event of a CGN node failure.

* To ensure a matching state table entry for an active session in the event of asymmetric routing through different egress and ingress CGN nodes.

### 4.7. Logging

In the case of 464XLAT and DS-Lite, the user of any given public IPv4 address and port combination will vary over time; therefore, logging is necessary to meet data-retention laws. Each entry in the PLAT/AFTR 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). Stateless technologies do not have a centralized stateful element that 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 CGN gateways.

### 4.8. Optimization for IPv4-Only Devices and 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 (e.g., CDNs and 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 [OP-464XLAT/MAP-T].

Because the NAT46 has already translated the IPv4-only flow to IPv6 and the services are dual-stack, using these optimizations allows the services to be reached without the need to translate the flow 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 Device Under Test (DUT) setup of [RFC8219] makes it possible to use the existing measurement devices compliant with "Benchmarking Methodology for Network Interconnect Devices" [RFC2544]; however,

this solution has two kinds of limitations:

* Dual DUT setup has the drawback that the performances of the CE and the ISP-side device (e.g., the CLAT and 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.

* Measurement procedures for Packet Delay Variation (PDV) and Inter-Packet Delay Variation (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 either special Tester is required or some trick is needed 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 commercial Tester [LEN2020a] that is compliant with [RFC5180].

Siitperf, a DPDK-based software Tester that is compliant with [RFC8219] and used for benchmarking stateless NAT64 gateways, has been developed recently. Siitperf is available from GitHub [SIITPERF] as free software and is documented in [LEN2021]. Originally, it literally followed the test frame format of [RFC2544], including "hard-wired" source and destination port numbers, and then it was complemented with the pseudorandom port feature required by [RFC4814]. The new version is documented in [LEN2020b].

Further DPDK-based software Testers that are compliant with [RFC8219] 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 [IPv4aaS-BENCHMARK-TECH].

## 6. IANA Considerations

This document has no IANA actions.

## 7. 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 noted that those issues are not specific to the IPv4aaS IPv6 transition technologies but apply to logging functionalities in general.

For all five technologies, the CE device typically contains a DNS proxy. However, the user may change DNS settings. If this happens and lw4o6, MAP-E, and MAP-T are used with a significantly restricted port set (which is required for 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 these technologies are thus theoretically less resilient against cache poisoning (see [RFC5452]). However, an efficient cache poisoning attack requires that the subscriber operates its 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 to be low.

IPv4aaS technologies based on encapsulation have no 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.

# 8. References

## 8.1. Normative References


[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>.

[RFC6519] Maglione, R. and A. Durand, "RADIUS Extensions for Dual-Stack Lite", RFC 6519, DOI 10.17487/RFC6519, February 2012, <https://www.rfc-editor.org/info/rfc6519>.

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

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

[RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <https://www.rfc-editor.org/info/rfc6888>.

[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>.

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

[RFC7341] Sun, Q., Cui, Y., Siodelski, M., Krishnan, S., and I. Farrer, "DHCPv4-over-DHCPv6 (DHCP 4o6) Transport", RFC 7341, DOI 10.17487/RFC7341, August 2014, <https://www.rfc-editor.org/info/rfc7341>.

[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>.

[RFC7605] Touch, J., "Recommendations on Using Assigned Transport Port Numbers", BCP 165, RFC 7605, DOI 10.17487/RFC7605, August 2015, <https://www.rfc-editor.org/info/rfc7605>.

[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>.

[RFC8215] Anderson, T., "Local-Use IPv4/IPv6 Translation Prefix", RFC 8215, DOI 10.17487/RFC8215, August 2017, <https://www.rfc-editor.org/info/rfc8215>.

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

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

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

[RFC8513] Boucadair, M., Jacquenet, C., and S. Sivakumar, "A YANG Data Model for Dual-Stack Lite (DS-Lite)", RFC 8513, DOI 10.17487/RFC8513, January 2019, <https://www.rfc-editor.org/info/rfc8513>.

[RFC8658] Jiang, S., Ed., Fu, Y., Ed., Xie, C., Li, T., and M.

Boucadair, Ed., "RADIUS Attributes for Softwire Mechanisms Based on Address plus Port (A+P)", RFC 8658, DOI 10.17487/RFC8658, November 2019, <https://www.rfc-editor.org/info/rfc8658>.

[RFC8676] Farrer, I., Ed. and M. Boucadair, Ed., "YANG Modules for IPv4-in-IPv6 Address plus Port (A+P) Softwires", RFC 8676, DOI 10.17487/RFC8676, November 2019, <https://www.rfc-editor.org/info/rfc8676>.

[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>.

## 8.2. Informative References

[AFTR] ISC, "ISC Implementation of AFTR", <https://downloads.isc.org/isc/aftr/>.

[AZZ2021] Al-Azzawi, A. and G. Lencse, "Identification of the Possible Security Issues of the 464XLAT IPv6 Transition Technology", Infocommunications Journal, Vol. 13, No. 4, pp. 10-18, DOI 10.36244/ICJ.2021.4.2, December 2021, <https://www.infocommunications.hu/2021_4_2>.

[IPv4aaS-BENCHMARK-TECH] Lencse, G., "Performance Analysis of IPv6 Transition Technologies for IPv4aaS", Work in Progress, Internet-Draft, draft-lencse-v6ops-transition-benchmarking-01, 2 May 2022, <https://datatracker.ietf.org/doc/html/draft-lencse-v6ops-transition-benchmarking-01>.

[IPv4aaS-SCALE-TECH] Lencse, G., "Scalability of IPv6 Transition Technologies for IPv4aaS", Work in Progress, Internet-Draft, draft-lencse-v6ops-transition-scalability-03, 30 June 2022, <https://datatracker.ietf.org/doc/html/draft-lencse-v6ops-transition-scalability-03>.

[JOOL] "Jool: SIIT & NAT64", <http://www.jool.mx>.

[JOOL-MAPT] "MAP-T Run", <https://www.jool.mx/en/run-mapt.html>.

[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, Vol. 77, No. 1, pp. 397-411, DOI 10.1016/j.cose.2018.04.012, August 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, October 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, June 2020, <https://link.springer.com/article/10.1007/s11235-020-00681-x>.

[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, <https://ijates.org/index.php/ijates/article/view/291>.

[LEN2021] Lencse, G., "Design and Implementation of a Software Tester for Benchmarking Stateless NAT64 Gateways", IEICE Transactions on Communications, Vol. E104.B, Issue 2, pp. 128-140, DOI 10.1587/transcom.2019EBN0010, 2021, <https://doi.org/10.1587/transcom.2019EBN0010>.

[MIY2010] Miyakawa, S., "IPv4 to IPv6 Transformation Schemes", IEICE Transactions on Communications, Vol. E93-B, Issue 5, pp. 1078-1084, DOI 10.1587/transcom.E93.B.1078, 2010, <https://www.jstage.jst.go.jp/article/transcom/E93.B/5/E93.B_5_1078/_article>.

[NAT-SUPP] Stewart, R. R., Txen, M., and I. Ruengeler, "Stream Control Transmission Protocol (SCTP) Network Address Translation Support", Work in Progress, Internet-Draft, draft-ietf-tsvwg-natsupp-23, 25 October 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-natsupp-23>.

[OP-464XLAT/MAP-T] Palet Martinez, J. and A. D'Egidio, "464XLAT/MAT-T Optimization", Work in Progress, Internet-Draft, draft-ietf-v6ops-464xlat-optimization-03, 28 July 2020, <https://datatracker.ietf.org/doc/html/draft-ietf-v6ops-464xlat-optimization-03>.

[REP2014] Rps, S., Hajas, T., and G. Lencse, "Port Number Consumption of the NAT64 IPv6 Transition Technology", 37th International Conference on Telecommunications and Signal Processing, DOI 10.1109/TSP.2015.7296411, 2014, <http://www.hit.bme.hu/~lencse/publications/TSP-2014-PC.pdf>.

[SIITPERF] "Siitperf: an RFC 8219 compliant SIIT (stateless NAT64) tester", commit bdce0f, February 2021, <https://github.com/lencsegabor/siitperf>.

[SNABB] "Snabb implementation of lwAFTR", commit 1ef72ce, January 2022, <https://github.com/Igalia/snabb>.

[TR-069] Broadband Forum, "CPE WAN Management Protocol", Technical Report TR-069, June 2020, <https://www.broadband-forum.org/technical/download/TR-069.pdf>.

[VPP] "VPP", July 2022, <https://wiki.fd.io/index.php?title=VPP&oldid=11809>.

## 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, ric Vyncke and Martin Duke for their review of this document 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, Tom Petch, Yannis

Nikolopoulos, Havard Eidnes, Yann-Ju Chu, Barbara Stark, Vasilenko Eduard, Chongfeng Xie, Henri Alves de Godoy, Magnus Westerlund, Michael Txen, Philipp S. Tiesel, Brian E. Carpenter, and Joe Touch.

## Authors' Addresses


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

Jordi Palet Martinez
The IPv6 Company
Molino de la Navata, 75
28420 La Navata - Galapagar Madrid
Spain
Email: jordi.palet@theipv6company.com
URI: http://www.theipv6company.com/

Lee Howard
Retevia
9940 Main St., Suite 200
Fairfax, Virginia 22031
United States of America
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
53227 Bonn
Germany
Email: ian.farrer@telekom.de