Pros and Cons of IPv6 Transition Technologies for IPv4aaS
draft-lmhp-v6ops-transition-comparison-06

Abstract

   Several IPv6 transition technologies have been developed to provide
   customers with IPv4-as-a-Service (IPv4aaS) for ISPs with an IPv6-only
   access and/or core network. All these technologies have their
   advantages and disadvantages, and depending on existing topology,
   skills, strategy and other preferences, one of these technologies may
   be the most appropriate solution for a network operator.

   This document examines the five most prominent IPv4aaS technologies
   considering a number of different aspects to provide network
   operators with an easy to use reference to assist in selecting the
   technology that best suits their needs.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at https://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on July 13, 2021.

Table of Contents

1. Introduction

As the deployment of IPv6 becomes more prevalent, it follows that
network operators will move to building single-stack IPv6 core and
access networks to simplify network planning and operations.
However, providing customers with IPv4 services continues to be a
requirement for the foreseeable future. To meet this need, the IETF
has standardized a number of different IPv4aaS technologies for this
[LEN2019] based on differing requirements and deployment scenarios.

The number of technologies that have been developed makes it time
consuming for a network operator to identify the most appropriate
mechanism for their specific deployment. This document provides a
comparative analysis of the most commonly used mechanisms to assist
operators with this problem.

Five different IPv4aaS solutions are considered. The following IPv6
transition technologies are covered:

1. 464XLAT [RFC6877]

2. Dual Stack Lite [RFC6333]

3. lw4o6 (Lightweight 4over6) [RFC7596]

4. MAP-E [RFC7597]

5. MAP-T [RFC7599]

We note that [RFC6180] gives guidelines for using IPv6 transition
mechanisms during IPv6 deployment addressing a much broader topic,
whereas this document focuses on a small part of it.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in
BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all
capitals, as shown here.

2. Overview of the Technologies

The following sections introduce the different technologies analyzed
in this document, describing some of their most important
characteristics.

2.1. 464XLAT

464XLAT is a single/dual translation model, which uses a customer-
side translator (CLAT) located in the customer's device to perform
stateless NAT64 translation [RFC7915] (more precisely, stateless
NAT46, a stateless IP/ICMP translation from IPv4 to IPv6).
IPv4-embedded IPv6 addresses [RFC6052] are used for both source and
destination addresses. Commonly, a /96 prefix (either the
64:ff9b::/96 Well-Known Prefix, or a Network-Specific Prefix) is used
as the IPv6 destination for the IPv4-embedded client traffic.

In the operator's network, the provider-side translator (PLAT)
performs stateful NAT64 [RFC6146] to translate the traffic. The
destination IPv4 address is extracted from the IPv4-embedded IPv6
packet destination address and the source address is from a pool of
public IPv4 addresses.

Alternatively, when a dedicated /64 is not available for translation,
the CLAT device uses a stateful NAT44 translation before the
stateless NAT46 translation.

Note that we generally do not see state close to the end-user as
equally problematic as state in the middle of the network.

In typical deployments, 464XLAT is used together with DNS64
[RFC6147], see Section 3.1.2 of [RFC8683]. When an IPv6-only client
or application communicates with an IPv4-only server, the DNS64
server returns the IPv4-embedded IPv6 address of the IPv4-only
server. In this case, the IPv6-only client sends out IPv6 packets,
thus CLAT functions as an IPv6 router and the PLAT performs a
stateful NAT64 for these packets. In this case, there is a single
translation.

Alternatively, one can say that the DNS64 + stateful NAT64 is used to
carry the traffic of the IPv6-only client and the IPv4-only server,
and the CLAT is used only for the IPv4 traffic from applications or
devices that use literal IPv4 addresses or non-IPv6 compliant APIs.

```
            Private +----------+ Translated  +----------+    _____
  +------+  IPv4  |   CLAT   |   4-6-4     | Stateful |   ( IPv4  )
  | IPv4 |------->| Stateless|------------>|   PLAT   +--( Internet )
  |Device|<-------|  NAT46   |<------------|  NAT64   |  (_____)
  +------+        +----------+     ^       +----------+
                                   |
                              Operator IPv6
                                network
```
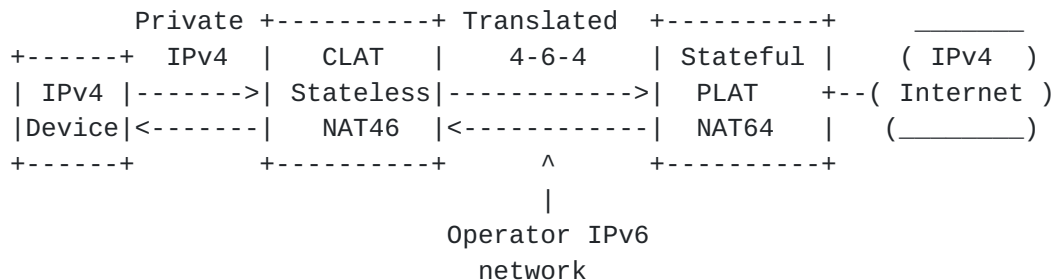
              Figure 1: Overview of the 464XLAT architecture

Note: in mobile networks, CLAT is commonly implemented in the user's
equipment (UE or smartphone).

2.2. Dual-Stack Lite

Dual-Stack Lite (DS-Lite) [RFC6333] was the first of the considered
transition mechanisms to be developed. DS-Lite uses a 'Basic
Broadband Bridging' (B4) function in the customer's CE router that
encapsulates IPv4 in IPv6 traffic and sends it over the IPv6 native
service-provider network to a centralized 'Address Family Transition
Router' (AFTR). The AFTR performs encapsulation/decapsulation of the
4in6 traffic and translates the IPv4 payload to public IPv4 source
address using a stateful NAPT44 function.

```
                                         +-------------+
         Private +----------+ IPv4-in-IPv6|Stateful AFTR|
  +------+  IPv4  |    B4    |   tunnel    |------+------+    _____
  | IPv4 |------->| Encap./  |------------>|Encap.|      |   ( IPv4  )
  |Device|<-------|  decap.  |<------------|  /   | NAPT +--( Internet )
  +------+        +----------+     ^       |Decap.|  44  |  (_____)
                                   |       +------+------+
                              Operator IPv6
                                network
```

              Figure 2: Overview of the DS-Lite architecture

2.3. Lightweight 4over6

Lightweight 4over6 (lw4o6) is a variant of DS-Lite. The main
difference is that the stateful NAPT44 function is relocated from the
centralized AFTR to the customer's B4 element (called a lwB4). The

AFTR (called a lwAFTR) function therefore only performs A+P routing
and 4in6 encapsulation/decapsulation.

Routing to the correct client and IPv4 address sharing is achieved
using the Address + Port (A+P) model [RFC6346] of provisioning each
lwB4 with a unique tuple of IPv4 address unique range of layer-4
ports. The client uses these for NAPT44.

The lwAFTR implements a binding table, which has a per-client entry
linking the customer's source IPv4 address and allocated range of
layer-4 ports to their IPv6 tunnel endpoint address. The binding
table allows egress traffic from customers to be validated (to
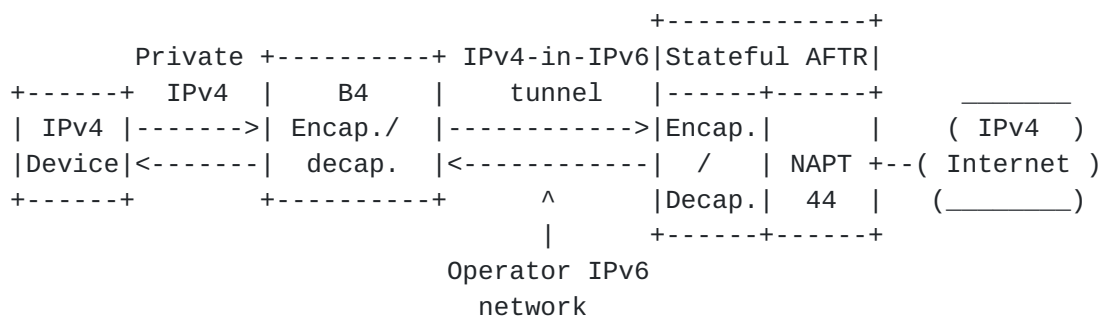prevent spoofing) and ingress traffic to be correctly encapsulated
and forwarded. As there needs to be a per-client entry, an lwAFTR
implementation needs to be optimized for performing a per-packet
lookup on the binding table.

Direct communication between two lwB4s is performed by hair-pinning
traffic through the lwAFTR.

```
                +-------------+              +----------+
         Private |    lwB4     | IPv4-in-IPv6| Stateless|
 +------+  IPv4  |------+------|    tunnel   |  lwAFTR  |     _____
 | IPv4 |------->|      |Encap.|------------>|(encap/A+P|    (  IPv4  )
 |Device|<-------| NAPT |  /   |<------------|bind. tab +--( Internet )
 +------+        |  44  |Decap.|      ^      | routing) |  (_____)
                 +------+------+      |      +----------+
                             Operator IPv6
                                network
```
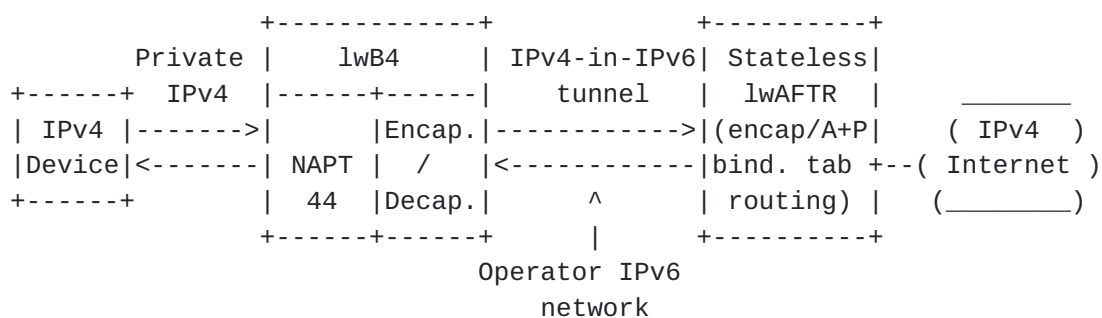
Figure 3: Overview of the lw4o6 architecture

2.4. MAP-E

MAP-E uses a stateless algorithm to embed portions of the customer's
allocated IPv4 address (or part of an address with A+P routing) into
the IPv6 prefix delegated to the client. This allows for large
numbers of clients to be provisioned using a single MAP rule (called
a MAP domain). The algorithm also allows for direct IPv4 peer-to-
peer communication between hosts provisioned with common MAP rules.

The CE (Customer-Edge) router typically performs stateful NAPT44
[RFC2663] to translate the private IPv4 source addresses and source
ports into an address and port range defined by applying the MAP rule
applied to the delegated IPv6 prefix. The client address/port
allocation size is a design parameter. The CE router then
encapsulates the IPv4 packet in an IPv6 packet [RFC2473] and sends it
directly to another host in the MAP domain (for peer-to-peer) or to a

Border Router (BR) if the IPv4 destination is not covered in one of
the CE's MAP rules.

The MAP BR is provisioned with the set of MAP rules for the MAP
domains it serves. These rules determine how the MAP BR is to
decapsulate traffic that it receives from client, validating the
source IPv4 address and layer 4 ports assigned, as well as how to
calculate the destination IPv6 address for ingress IPv4 traffic.

```
                   +--------------+           +----------+
          Private |    MAP CE    | IPv4-in-IPv6| Stateless|
 +------+  IPv4   |------+------|    tunnel   |  MAP BR  |     _____
 | IPv4 |------->|       |Encap.|----------->|(encap/A+P|    ( IPv4  )
 |Device|<-------| NAPT |  /   |<------------|algorithm +--( Internet )
 +------+         |  44  |Decap.|     ^        | routing) |   (_____)
                   +------+------+     |        +----------+
                            Operator IPv6
                              network
```
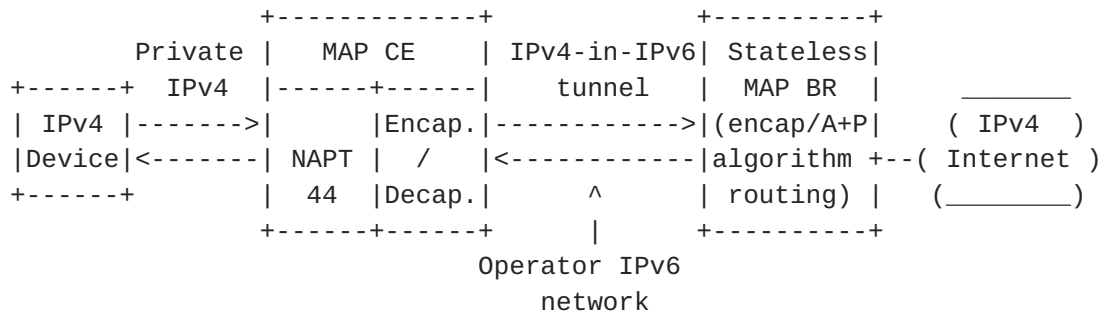
Figure 4: Overview of the MAP-E architecture

2.5. MAP-T

MAP-T uses the same mapping algorithm as MAP-E. The major difference
is that double stateless translation (NAT46 in the CE and NAT64 in
the BR) is used to traverse the ISP's IPv6 single-stack network.
MAP-T can also be compared to 464XLAT when there is a double
translation.

A MAP CE typically performs stateful NAPT44 to translate traffic to a
public IPv4 address and port-range calculated by applying the
provisioned Basic MAP Rule (BMR - a set of inputs to the algorithm)
to the delegated IPv6 prefix. The CE then performs stateless
translation from IPv4 to IPv6 [RFC7915]. The MAP BR is provisioned
with the same BMR as the client, enabling the received IPv6 traffic
to be statelessly NAT64 translated back to the public IPv4 source
address used by the client.

Using translation instead of encapsulation also allows IPv4-only
nodes to correspond directly with IPv6 nodes in the MAP-T domain that
have IPv4-embedded IPv6 addresses.

```
                +-------------+            +----------+
        Private |   MAP CE    | Translated | Stateless|
  +------+  IPv4 |------+------|    4-6-4   |  MAP BR  |     _____
  | IPv4 |------->|      |State-|----------->|(NAT64/A+P|    ( IPv4  )
  |Device|<-------| NAPT | less |<-----------|algorithm +--( Internet )
  +------+        |  44  |NAT46 |     ^      | routing) |   (_____)
                +------+------+     |      +----------+
                           Operator IPv6
                           network
```
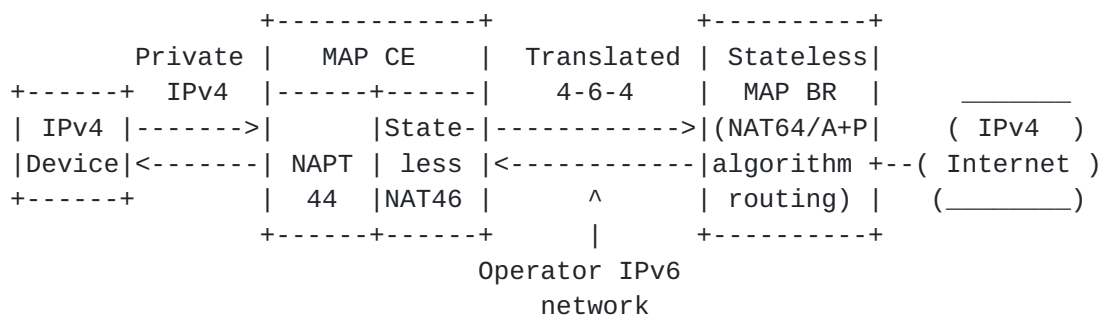
Figure 5: Overview of the MAP-T architecture

3. High-level Architectures and their Consequences

3.1. Service Provider Network Traversal

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

o   4-6-4 translation

o   4-in-6 encapsulation

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

Table 1: Available Traversal Mechanisms

In the scope of this document, all of the encapsulation based
mechanisms use IP-in-IP tunnelling [RFC2473]. This is a stateless
tunneling mechanism which does not require any additional tunnel
headers.

It should be noted that both of these approaches result in an
increase in the size of the packet that needs to be transported
across the operator's network when compared to native IPv4. 4-6-4
translation adds a 20-bytes overhead (the 20-byte IPv4 header is
replaced with a 40-byte IPv6 header). Encapsulation has a 40-byte
overhead (an IPv6 header is prepended to the IPv4 header).

The increase in packet size can become a significant problem if there
is a link with a smaller MTU in the traffic path. This may result in
traffic needing to be fragmented at the ingress point to the IPv6
only domain (i.e., the NAT46 or 4in6 encapsulation endpoint). It may

also result in the need to implement buffering and fragment re-
assembly in the BR node.

The advice given in [RFC7597] Section 8.3.1 is applicable to all of
these mechanisms: It is strongly recommended that the MTU in the
IPv6-only domain be well managed and that the IPv6 MTU on the CE WAN-
side interface be set so that no fragmentation occurs within the
boundary of the IPv6-only domain.

3.2. Network Address Translation

For the high-level solution of IPv6 service provider network
traversal, MAP-T uses double stateless translation. First at the CE
from IPv4 to IPv6 (NAT46), and then from IPv6 to IPv4 (NAT64), at the
service provider network.

464XLAT may use double translation (stateless NAT46 + stateful NAT64)
or single translation (stateful NAT64), depending on different
factors, such as the use of DNS by the applications and the
availability of a DNS64 function (in the host or in the service
provider network). For deployment guidelines, please refer to
[RFC8683].

The first step for the double translation mechanisms is a stateless
NAT from IPv4 to IPv6 implemented as SIIT (Stateless IP/ICMP
Translation Algorithm) [RFC7915], which does not translate IPv4
header options and/or multicast IP/ICMP packets. With encapsulation-
based technologies the header is transported intact and multicast can
also be carried.

Single and double translation results in native IPv6 traffic with a
layer-4 next-header. The fields in these headers can be used for
functions such as hashing across equal-cost multipaths or ACLs. For
encapsulation, there is an IPv6 header followed by an IPv4 header.
This results in less entropy for hashing algorithms, and may mean
that devices in the traffic path that perform header inspection (e.g.
router ACLs or firewalls) require the functionality to look into the
payload header.

Solutions using double translation can only carry port-aware IP
protocols (e.g. TCP, UDP) and ICMP when they are used with IPv4
address sharing (please refer to Section 4.3 for more details).
Encapsulation based solutions can carry any other protocols over IP,
too.

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

3.3. IPv4 Address Sharing

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

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

   The solutions compared by this document fall into two categories:

   o CGN-based approaches (DS-Lite, 464XLAT)

   o A+P-based approaches (lw4o6, MAP-E, MAP-T)

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

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

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

   The address sharing efficiency of the five technologies is
   significantly different, it is discussed in Section 4.2

   lw4o6, MAP-E and MAP-T can also be configured without IPv4 address
   sharing, see the details in Section 4.3. However, in that case,
   there is no advantage in terms of public IPv4 address saving. In the
   case of 464XLAT, this can be achieved as well through EAMT [RFC7757].

   Conversely, both MAP-E and MAP-T may be configured to provide more
   than one public IPv4 address (i.e., an IPv4 prefix shorter than a
   /32) to customers.

   Dynamic DNS issues in address-sharing contexts and their possible
   solutions using PCP (Port Control Protocol) are discussed in detail
   in [RFC7393].

3.4. CE Provisioning Considerations

All of the technologies require some provisioning of customer
devices. The table below shows which methods currently have
extensions for provisioning the different mechanisms.

```
+------------------+-----------+---------+-------+-------+-------+
|                  |  464XLAT  | DS-Lite | lw4o6 | MAP-E | MAP-T |
+------------------+-----------+---------+-------+-------+-------+
| DHCPv6 [RFC8415] |           |    X    |   X   |   X   |   X   |
| RADIUS Attr.     |           |    X    |   X   |   X   |   X   |
| TR-69            |           |    X    |       |   X   |   X   |
| DNS64 [RFC7050]  |     X     |         |       |       |       |
| YANG [RFC7950]   | [RFC8512] |    X    |   X   |   X   |   X   |
| DHCP4o6          |           |         |   X   |   X   |       |
+------------------+-----------+---------+-------+-------+-------+
```

Table 2: Available Provisioning Mechanisms

3.5. Support for Multicast

The solutions covered in this document are all intended for unicast
traffic. [RFC8114] describes a method for carrying encapsulated IPv4
multicast traffic over an IPv6 multicast network. This could be
deployed in parallel to any of the operator's chosen IPv4aaS
mechanism.

4. Detailed Analysis

4.1. Architectural Differences

4.1.1. Basic Comparison

The five IPv4aaS technologies can be classified into 2x2=4 categories
on the basis of two aspects:

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

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

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

Table 3: Available Provisioning Mechanisms

4.2. Tradeoff between Port Number Efficiency and Stateless Operation

464XLAT and DS-Lite use stateful NAPT at the PLAT/AFTR devices,
respectively. This may cause scalability issues for the number of
clients or volume of traffic, but does not impose a limitation on the
number of ports per user, as they can be allocated dynamically on-
demand and the allocation policy can be centrally managed/adjusted.

A+P based mechanisms (Lw4o6, MAP-E, and MAP-T) avoid using NAPT in
the service provider network. However, this means that the number of
ports provided to each user (and hence the effective IPv4 address
sharing ratio) must be pre-provisioned to the client.

Changing the allocated port ranges with A+P based technologies,
requires more planning and is likely to involve re-provisioning both
hosts and operator side equipment. It should be noted that due to
the per-customer binding table entry used by lw4o6, a single customer
can be re-provisioned (e.g., if they request a full IPv4 address)
without needing to change parameters for a number of customers as in
a MAP domain.

It is also worth noting that there is a direct relationship between
the efficiency of customer public port-allocations and the
corresponding logging overhead that may be necessary to meet data-
retention requirements. This is considered in Section 4.7 below.

Determining the optimal number of ports for a fixed port set is not
an easy task, and may also be impacted by local regulatory law, which
may define a maximum number of users per IP address, and consequently
a minimum number of ports per user.

On the one hand, the "lack of ports" situation may cause serious
problems in the operation of certain applications. For example,
Miyakawa has demonstrated the consequences of the session number
limitation due to port number shortage on the example of Google Maps
[MIY2010]. When the limit was 15, several blocks of the map were
missing, and the map was unusable. This study also provided several

examples for the session numbers of different applications (the
highest one was Apple's iTunes: 230-270 ports).

The port number consumption of different applications is highly
varying and e.g. in the case of web browsing it depends on several
factors, including the choice of the web page, the web browser, and
sometimes even the operating system [REP2014]. For example, under
certain conditions, 120-160 ports were used (URL: sohu.com, browser:
Firefox under Ubuntu Linux), and in some other cases it was only 3-12
ports (URL: twitter.com, browser: Iceweasel under Debian Linux).

There may be several users behind a CE router, especially in the
broadband case (e.g. Internet is used by different members of a
family simultaneously), so sufficient ports must be allocated to
avoid impacting user experience.

Furthermore, assigning too many ports per CE router will result in
waste of public IPv4 addresses, which is a scarce and expensive
resource. Clearly this is a big advantage in the case of 464XLAT
where they are dynamically managed, so that the number of IPv4
addresses for the sharing-pool is smaller while the availability of
ports per user don't need to be pre-defined and is not a limitation
for them.

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

We note that common CE router NAT44 implementations utilizing
Netfilter, multiplexes active sessions using a 3-tuple (source
address, destination address, and destination port). This means that
external source ports can be reused for unique internal source and
destination address and port sessions. It is also noted, that
Netfilter cannot currently make use of multiple source port ranges
(i.e. several blocks of ports distributed across the total port space
as is common in MAP deployments), this may influence the design when
using stateless technologies.

Stateful technologies, 464XLAT and DS-Lite (and also NAT444) can
therefore be much more efficient in terms of port allocation and thus
public IP address saving. The price is the stateful operation in the
service provider network, which allegedly does not scale up well. It
should be noticed that in many cases, all those factors may depend on
how it is actually implemented.

XXX MEASUREMENTS ARE PLANNED TO TEST IF THE ABOVE IS TRUE. XXX

We note that some CGN-type solutions can allocate ports dynamically
"on the fly". Depending on configuration, this can result in the
same customer being allocated ports from different source addresses.
This can cause operational issues for protocols and applications that
expect multiple flows to be sourced from the same address. E.g.,
ECMP hashing, STUN, gaming, content delivery networks. However, it
should be noticed that this is the same problem when a network has a
NAT44 with multiple public IPv4 addresses, or even when applications
in a dual-stack case, behave wrongly if happy eyeballs is flapping
the flow address between IPv4 and IPv6.

The consequences of IPv4 address sharing [RFC6269] may impact all
five technologies. However, when ports are allocated statically,
more customers may get ports from the same public IPv4 address, which
may results in negative consequences with higher probability, e.g.
many applications and service providers (Sony PlayStation Network,
OpenDNS, etc.) permanently black-list IPv4 ranges if they detect that
they are used for address sharing.

Both cases are, again, implementation dependent.

We note that although it is not of typical use, one can do
deterministic, stateful NAT and reserve a fixed set of ports for each
customer, as well.

4.3. Support for Public Server Operation

Mechanisms that rely on operator side per-flow state do not, by
themselves, offer a way for customers to present services on publicly
accessible layer-4 ports.

Port Control Protocol (PCP) [RFC6877] provides a mechanism for a
client to request an external public port from a CGN device. For
server operation, it is required with NAT64/464XLAT, and it is
supported in some DS-Lite AFTR implementations.

A+P based mechanisms distribute a public IPv4 address and restricted
range of layer-4 ports to the client. In this case, it is possible
for the user to configure their device to offer a publicly accessible
server on one of their allocated ports. It should be noted that
commonly operators do not assign the Well-Known-Ports to users
(unless they are allocating a full IPv4 address), so the user will
need to run the service on an allocated port, or configure port
translation.

Lw4o6, MAP-E and MAP-T may be configured to allocated clients with a
full IPv4 address, allowing exclusive use of all ports, and non-port-
based layer 4 protocols. Thus, they may also be used to support

server/services operation on their default ports. However, when
public IPv4 addresses are assigned to the CE router without address
sharing, obviously there is no advantage in terms of IPv4 public
addresses saving.

It is also possible to configure specific ports mapping in 464XLAT/
NAT64 using EAMT [RFC7757], which means that only those ports are
"lost" from the pool of addresses, so there is a higher maximization
of the total usage of IPv4/port resources.

4.4. Support and Implementations

4.4.1. OS Support

A 464XLAT client (CLAT) is implemented in Windows 10, Linux
(including Android), Windows Mobile, Chrome OS and iOS, but at the
time of writing is not available in MacOS.

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

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

o '464xlat' enables support for 464XLAT CLAT functionality

o 'ds-lite' enables support for DSLite B4 functionality

o 'map' enables support for MAP-E and lw4o6 CE functionality

o 'map-t' enables support for MAP-T CE functionality

For the operator side functionality, some free open-source
implementations exist:

CLAT, NAT64, EAMT: http://www.jool.mx

MAP-BR, lwAFTR, CGN, CLAT, NAT64: VPP/fd.io
                    https://gerrit.fd.io/r/#/admin/projects/

lwAFTR: https://github.com/Igalia/snabb

DSLite AFTR: https://www.isc.org/downloads/

4.4.2. Support in Cellular and Broadband Networks

Several cellular networks use 464XLAT, whereas we are not aware of
any deployment of the four other technologies in cellular networks,
as they are not implemented in UE devices.

In broadband networks, there are some deployments of 464XLAT, MAP-E
and MAP-T. Lw4o6 and DS-Lite have more deployments, with DS-Lite
being the most common, but lw4o6 taking over in the last years.

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

4.4.3. Implementation Code Sizes

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

We note that the support for all five technologies requires much less
code size than the total sum of the above quantities, because they
contain a lot of common functions (data plane is shared among several
of them).

4.5. Typical Deployment and Traffic Volume Considerations

4.5.1. Deployment Possibilities

Theoretically, all five IPv4aaS technologies could be used together
with DNS64 + stateful NAT64, as it is done in 464XLAT. In this case
the CE router would treat the traffic between an IPv6-only client and
IPv4-only server as normal IPv6 traffic, and the stateful NAT64
gateway would do a single translation, thus offloading this kind of
traffic from the IPv4aaS technology. The cost of this solution would
be the need for deploying also DNS64 + stateful NAT64.

However, this has not been implemented in clients or actual
deployments, so only 464XLAT always uses this optimization and the
other four solutions do not use it at all.

4.5.2. Cellular Networks with 464XLAT

Actual figures from existing deployments, show that the typical
traffic volumes in an IPv6-only cellular network, when 464XLAT
technology is used together with DNS64, are:

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

   o 24% of traffic uses DNS64 + NAT64 (1 translation)

   o Less than 1% of traffic uses the CLAT in addition to NAT64 (2
      translations), due to an IPv4 socket and/or IPv4 literal.

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

4.6. Load Sharing

   If multiple network-side devices are needed as PLAT/AFTR/BR for
   capacity, then there is a need for a load sharing mechanism. ECMP
   (Equal-Cost Multi-Path) load sharing can be used for all
   technologies, however stateful technologies will be impacted by
   changes in network topology or device failure.

   Technologies utilizing DNS64 can also distribute load across PLAT/
   AFTR devices, evenly or unevenly, by using different prefixes.
   Different network specific prefixes can be distributed for
   subscribers in appropriately sized segments (like split-horizon DNS,
   also called DNS views).

   Stateless technologies, due to the lack of per-flow state, can make
   use of anycast routing for load sharing and resiliency across
   network-devices, both ingress and egress; flows can take asymmetric
   paths through the network, i.e., in through one lwAFTR/BR and out via
   another.

   Mechanisms with centralized NAPT44 state have a number of challenges
   specifically related to scaling and resilience. As the total amount
   of client traffic exceeds the capacity of a single CGN instance,
   additional nodes are required to handle the load. As each CGN
   maintains a stateful table of active client sessions, this table may
   need to be syncronized between CGN instances. This is necessary for
   two reasons:

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

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

4.7. **Logging**

   In the case of 464XLAT and DS-Lite, the user of any given public IPv4
   address and port combination will vary over time, therefore, logging
   is necessary to meet data retention laws. Each entry in the PLAT/
   AFTR's generates a logging entry. As discussed in Section 4.2, a
   client may open hundreds of sessions during common tasks such as web-
   browsing, each of which needs to be logged so the overall logging
   burden on the network operator is significant. In some countries,
   this level of logging is required to comply with data retention
   legislation.

   One common optimization available to reduce the logging overhead is
   the allocation of a block of ports to a client for the duration of
   their session. This means that logging entry only needs to be made
   when the client's port block is released, which dramatically reducing
   the logging overhead. This comes 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.

4.8. **Optimization for IPv4-only devices/applications**

   When IPv4-only devices or applications are behind a CE connected with
   IPv6-only and IPv4aaS, the IPv4-only traffic flows will necessarily,
   be encapsulated/decapsulated (in the case of DS-Lite, lw4o6 and MAP-
   E) and will reach the IPv4 address of the destination, even if that
   service supports dual-stack. This means that the traffic flow will
   cross thru the AFTR, lwAFTR or BR, depending on the specific
   transition mechanism being used.

Even if those services are directly connected to the operator network
(for example, CDNs, caches), or located internally (such as VoIP,
etc.), it is not possible to avoid that overhead.

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

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

5. Performance Comparison

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

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

o Dual DUT setup has the drawback that the performances of the CE
   and of the ISP side device (e.g. the CLAT and the PLAT of 464XLAT)
   are measured together. In order to measure the performance of
   only one of them, we need to ensure that the desired one is the
   bottleneck.

o Measurements procedures for PDV and IPDV measurements are missing
   from the legacy devices, and the old measurement procedure for
   Latency has been redefined in [RFC8219].

The Single DUT Setup of [RFC8219] makes it possible to benchmark the
selected device separately, but it either requires a special Tester
or some trick is need, if we want to use legacy Testers. An example
for the latter is our stateless NAT64 measurements testing Throuput
and Frame Loss Rate using a legacy [RFC5180] compliant commercial
tester [LEN2020a]

Siitperf, an [RFC8219] compliant DPDK-based software Tester for
benchmarking stateless NAT64 gateways has been developed recently and
it is available from GitHub [SIITperf] as free software and
documented in [LEN2021]. Originally, it literally followed the test
frame format of [RFC2544] including "hard wired" source and
destination port numbers, and then it has been complemented with the

random port feature required by [RFC4814]. The new version is
documented in [LEN2020b]

o It can be used for benchmarking both the CLAT and PLAT of 464XLAT
   separately, according to the single DUT setup. (We note that the
   benchmarking prodedures for stateful NAT64 include the stateless
   tests, plus a few additional tests, which are not implemented
   yet.)

o It can also be used for benchmarking all five IPv4-as-a-Service
   technologies according to the Dual DUT setup, because it supports
   the usage of IPv4 on its both sides, too.

Another software tester for benchmaring the B4 and AFTR components of
DS-Lite is currently being developed at the Budapest University of
Technology and Economics as a student project. It is planned to be
released as free software later this year.

We plan to start an intesive benchmaking campaign using the resources
of NICT StarBED, Japan.

6. Acknowledgements

The authors would like to thank Ole Troan for his thorough review of
this draft and acknowledge the inputs of Mark Andrews, Edwin
Cordeiro, Fred Baker, Alexandre Petrescu, Cameron Byrne, Tore
Anderson, Mikael Abrahamsson, Gert Doering, Satoru Matsushima,
Mohamed Boucadair, Tom Petch, Yannis Nikolopoulos, and TBD ...

7. IANA Considerations

This document does not make any request to IANA.

8. Security Considerations

According to the simplest model, the number of bugs is proportional
to the number of code lines. Please refer to Section 4.4.3 for code
sizes of CE implementations.

For all five technologies, the CE device should contain a DNS proxy.
However, the user may change DNS settings. If it happens and lw4o6,
MAP-E and MAP-T are used with significantly restricted port set,
which is required for an efficient public IPv4 address sharing, the
entropy of the source ports is significantly lowered (e.g. from 16
bits to 10 bits, when 1024 port numbers are assigned to each
subscriber) and thus these technologies are theoretically less
resilient against cache poisoning, see [RFC5452]. However, an
efficient cache poisoning attack requires that the subscriber

operates an own caching DNS server and the attack is performed in the service provider network. Thus, we consider the chance of the successful exploitation of this vulnerability as low.

An in-depth security analysis of all five IPv6 transition technologies and their most prominent free software implementations according to the methodology defined in [LEN2018] is planned.

As the first step, the theoretical security analysis of 464XLAT was done in [Azz2020].

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <https://www.rfc-editor.org/info/rfc2544>.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, DOI 10.17487/RFC2663, August 1999, <https://www.rfc-editor.org/info/rfc2663>.

[RFC4814] Newman, D. and T. Player, "Hash and Stuffing: Overlooked Factors in Network Device Benchmarking", RFC 4814, DOI 10.17487/RFC4814, March 2007, <https://www.rfc-editor.org/info/rfc4814>.

[RFC5180] Popoviciu, C., Hamza, A., Van de Velde, G., and D. Dugatkin, "IPv6 Benchmarking Methodology for Network Interconnect Devices", RFC 5180, DOI 10.17487/RFC5180, May 2008, <https://www.rfc-editor.org/info/rfc5180>.

[RFC5452] Hubert, A. and R. van Mook, "Measures for Making DNS More Resilient against Forged Answers", RFC 5452, DOI 10.17487/RFC5452, January 2009, <https://www.rfc-editor.org/info/rfc5452>.

   [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X.
              Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052,
              DOI 10.17487/RFC6052, October 2010,
              <https://www.rfc-editor.org/info/rfc6052>.

   [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful
              NAT64: Network Address and Protocol Translation from IPv6
              Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146,
              April 2011, <https://www.rfc-editor.org/info/rfc6146>.

   [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van
              Beijnum, "DNS64: DNS Extensions for Network Address
              Translation from IPv6 Clients to IPv4 Servers", RFC 6147,
              DOI 10.17487/RFC6147, April 2011,
              <https://www.rfc-editor.org/info/rfc6147>.

   [RFC6180] Arkko, J. and F. Baker, "Guidelines for Using IPv6
              Transition Mechanisms during IPv6 Deployment", RFC 6180,
              DOI 10.17487/RFC6180, May 2011,
              <https://www.rfc-editor.org/info/rfc6180>.

   [RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and
              P. Roberts, "Issues with IP Address Sharing", RFC 6269,
              DOI 10.17487/RFC6269, June 2011,
              <https://www.rfc-editor.org/info/rfc6269>.

   [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-
              Stack Lite Broadband Deployments Following IPv4
              Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011,
              <https://www.rfc-editor.org/info/rfc6333>.

   [RFC6346] Bush, R., Ed., "The Address plus Port (A+P) Approach to
              the IPv4 Address Shortage", RFC 6346,
              DOI 10.17487/RFC6346, August 2011,
              <https://www.rfc-editor.org/info/rfc6346>.

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

   [RFC6889] Penno, R., Saxena, T., Boucadair, M., and S. Sivakumar,
              "Analysis of Stateful 64 Translation", RFC 6889,
              DOI 10.17487/RFC6889, April 2013,
              <https://www.rfc-editor.org/info/rfc6889>.

   [RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of
              the IPv6 Prefix Used for IPv6 Address Synthesis",
              RFC 7050, DOI 10.17487/RFC7050, November 2013,
              <https://www.rfc-editor.org/info/rfc7050>.

   [RFC7393] Deng, X., Boucadair, M., Zhao, Q., Huang, J., and C. Zhou,
              "Using the Port Control Protocol (PCP) to Update Dynamic
              DNS", RFC 7393, DOI 10.17487/RFC7393, November 2014,
              <https://www.rfc-editor.org/info/rfc7393>.

   [RFC7422] Donley, C., Grundemann, C., Sarawat, V., Sundaresan, K.,
              and O. Vautrin, "Deterministic Address Mapping to Reduce
              Logging in Carrier-Grade NAT Deployments", RFC 7422,
              DOI 10.17487/RFC7422, December 2014,
              <https://www.rfc-editor.org/info/rfc7422>.

   [RFC7596] Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I.
              Farrer, "Lightweight 4over6: An Extension to the Dual-
              Stack Lite Architecture", RFC 7596, DOI 10.17487/RFC7596,
              July 2015, <https://www.rfc-editor.org/info/rfc7596>.

   [RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S.,
              Murakami, T., and T. Taylor, Ed., "Mapping of Address and
              Port with Encapsulation (MAP-E)", RFC 7597,
              DOI 10.17487/RFC7597, July 2015,
              <https://www.rfc-editor.org/info/rfc7597>.

   [RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S.,
              and T. Murakami, "Mapping of Address and Port using
              Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July
              2015, <https://www.rfc-editor.org/info/rfc7599>.

   [RFC7757] Anderson, T. and A. Leiva Popper, "Explicit Address
              Mappings for Stateless IP/ICMP Translation", RFC 7757,
              DOI 10.17487/RFC7757, February 2016,
              <https://www.rfc-editor.org/info/rfc7757>.

   [RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont,
              "IP/ICMP Translation Algorithm", RFC 7915,
              DOI 10.17487/RFC7915, June 2016,
              <https://www.rfc-editor.org/info/rfc7915>.

   [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language",
              RFC 7950, DOI 10.17487/RFC7950, August 2016,
              <https://www.rfc-editor.org/info/rfc7950>.

   [RFC8114] Boucadair, M., Qin, C., Jacquenet, C., Lee, Y., and Q.
              Wang, "Delivery of IPv4 Multicast Services to IPv4 Clients
              over an IPv6 Multicast Network", RFC 8114,
              DOI 10.17487/RFC8114, March 2017,
              <https://www.rfc-editor.org/info/rfc8114>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

   [RFC8219] Georgescu, M., 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>.

   [RFC8683] Palet Martinez, J., "Additional Deployment Guidelines for
              NAT64/464XLAT in Operator and Enterprise Networks",
              RFC 8683, DOI 10.17487/RFC8683, November 2019,
              <https://www.rfc-editor.org/info/rfc8683>.

9.2. Informative References

   [Azz2020] Al-Azzawi, A. and G. Lencse, "Towards the Identification
              of the Possible Security Issues of the 464XLAT IPv6
              Transition Technology", 43rd International Conference on
              Telecommunications and Signal Processing (TSP 2020),
              Milan, Italy, 10.1109/TSP49548.2020.9163487, Jul 2020,
              <http://www.hit.bme.hu/~lencse/publications/TSP-2020-
              464XLAT-revised.pdf>.

   [I-D.ietf-v6ops-464xlat-optimization]
              Martinez, J. and A. D'Egidio, "464XLAT/MAT-T
              Optimization", draft-ietf-v6ops-464xlat-optimization-03
              (work in progress), July 2020.

[LEN2018]   Lencse, G. and Y. Kadobayashi, "Methodology for the
            identification of potential security issues of different
            IPv6 transition technologies: Threat analysis of DNS64 and
            stateful NAT64", Computers & Security (Elsevier), vol.
            77, no. 1, pp. 397-411, DOI: 10.1016/j.cose.2018.04.012,
            Aug 2018, <http://www.hit.bme.hu/~lencse/publications/ECS-
            2018-Methodology-revised.pdf>.

[LEN2019]   Lencse, G. and Y. Kadobayashi, "Comprehensive Survey of
            IPv6 Transition Technologies: A Subjective Classification
            for Security Analysis", IEICE Transactions on
            Communications, vol. E102-B, no.10, pp. 2021-2035., DOI:
            10.1587/transcom.2018EBR0002, Oct 2019,
            <http://www.hit.bme.hu/~lencse/publications/
            e102-b_10_2021.pdf>.

[LEN2020a]
            Lencse, G., "Benchmarking Stateless NAT64 Implementations
            with a Standard Tester", Telecommunication Systems, vol.
            75, pp. 245-257, DOI: 10.1007/s11235-020-00681-x, Jun
            2020, <http://www.hit.bme.hu/~lencse/publications/
            Lencse2020_Article_BenchmarkingStatelessNAT64Impl.pdf>.

[LEN2020b]
            Lencse, G., "Adding RFC 4814 Random Port Feature to
            Siitperf: Design, Implementation and Performance
            Estimation", International Journal of Advances in
            Telecommunications, Electrotechnics, Signals and Systems,
            vol 9, no 3, pp. 18-26, DOI: 10.11601/ijates.v9i3.291,
            2020, <http://www.hit.bme.hu/~lencse/
            publications/291-1113-1-PB.pdf>.

[LEN2021]   Lencse, G., "Design and Implementation of a Software
            Tester for Benchmarking Stateless NAT64 Gateways", IEICE
            Transactions on Communications, DOI:
            10.1587/transcom.2019EBN0010, 2021,
            <http://www.hit.bme.hu/~lencse/publications/IEICE-2020-
            siitperf-revised.pdf>.

[MIY2010]   Miyakawa, S., "IPv4 to IPv6 transformation
            schemes", IEICE Trans. Commun., vol.E93-B, no.5, pp.
            1078-1084, DOI:10.1587/transcom.E93.B.10, May 2010,
            <https://www.jstage.jst.go.jp/article/transcom/E93.B/5/
            E93.B_5_1078/_article>.

   [REP2014]   Repas, S., Hajas, T., and G. Lencse, "Port number
               consumption of the NAT64 IPv6 transition
               technology", Proc. 37th Internat. Conf. on
               Telecommunications and Signal Processing (TSP 2014),
               Berlin, Germany, DOI: 10.1109/TSP.2015.7296411, July
               2014.

   [SIITperf]
               Lencse, G., "Siitperf: an RFC 8219 compliant SIIT
               (stateless NAT64) tester", November 2019,
               <https://github.com/lencsegabor/siitperf>.

Appendix A. Change Log

A.1. 01 - 02

   o Ian Farrer has joined us as an author.

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

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

   o Section titled "High-level Architectures and their Consequences"
      has been completely rewritten.

   o Several additions/clarification throughout Section titled
      "Detailed Analysis".

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

   o Word based text ported to XML.

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

A.2. 02 - 03

   o The suggestions of Mohamed Boucadair are incorporated.

   o New considerations regarding possible optimizations.

[A.3](#). **03 - 04**

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

   o Some references were updated or added.

[A.4](#). **04 - 05**

   o Some references were updated or added.

[A.5](#). **05 - 06**

   o Some references were updated or added.

Authors' Addresses

   Gabor Lencse
   Budapest University of Technology and Economics
   Magyar Tudosok korutja 2.
   Budapest H-1117
   Hungary

   Email: lencse@hit.bme.hu

   Jordi Palet Martinez
   The IPv6 Company
   Molino de la Navata, 75
   La Navata - Galapagar, Madrid 28420
   Spain

   Email: jordi.palet@theipv6company.com
   URI: http://www.theipv6company.com/

   Lee Howard
   Retevia
   9940 Main St., Suite 200
   Fairfax, Virginia 22031
   USA

   Email: lee@asgard.org

Richard Patterson
Sky UK
1 Brick Lane
London EQ 6PU
United Kingdom

Email: richard.patterson@sky.uk

Ian Farrer
Deutsche Telekom AG
Landgrabenweg 151
Bonn   53227
Germany

Email: ian.farrer@telekom.de