A Comprehensive Survey on the Most Important IPv4aaS IPv6 Transition Technologies, their Implementations and Performance Analysis

Omar D'yab

***Abstract*—As the central public IPv4 address pool has already been exhausted, the deployment of IPv6 has become inevitable. However, the users still require IPv4 Internet access due to some IPv4-only applications. The IPv4aaS (IPv4-as-a-Service) IPv6 transition technologies facilitate that ISPs provide IPv4 service to their customers while using only IPv6 in their access and core networks. This paper discusses the widely used IPv4aaS IPv6 transition technologies in ISP/enterprise networks; we explain their operations, advantages, properties and consider their performances. There are currently many IPv6 transition technologies, nevertheless, in this paper, the five most prominent IPv4aaS IPv6 transition technologies are discussed, namely 464XLAT, Dual-Stack Lite, Lightweight 4over6, MAP-E, and MAP-T. Moreover, the deployment and implementations of these technologies are being analysed and inspected. This paper also overviews the benchmarking methodology for IPv6 transition technologies and surveys several papers that investigated metrics and tools utilized in analysing the performance of different IPv6 transition technologies.**

***Index Terms*—464XLAT, DS-Lite, Lw4o6, MAP-E, MAP-T**

I. INTRODUCTION

WE have already given an overview of the five IPv4aaS technologies, their operation, advantages and disadvantages, as well as their most important implementations [1]. As expected, years ago, the world is now running out of IPv4 addresses. In February 2011, IANA, the global body responsible for managing Internet addresses, distributed the last five "/8" sets of IPv4 Internet addresses to the five regional Internet registries [2]. IPv4 uses a 32-bit addressing scheme, which was thought to be enough to support billions of devices, yet the more devices connected, the more we need IPv6 to solve the problem many predicted.

IPv6 activation is the main solution to the problem of lack of IPv4 addresses. IPv6 is the next generation of Internet Protocol and is designed to replace the existing IPv4 protocol, however, it is still not easy to deploy, and as the network environment needs to be converted from IPv4 to IPv6, especially that IPv4 is still widely used, it may take a long time because of some factors, such as the inability of the IPv4 network devices to be completely replaced.

O. D'yab is with Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary (e-mail: omardyab@hit.bme.hu).

The deployment of the IPv6 protocol around the world is relatively slow, the following may address the possibilities of this issue:

- Service providers do not want to activate IPv6, because there is no demand from subscribers, and subscribers do not request IPv6 because of the lack of content that works on it, hence content providers do not want to activate IPv6 until it becomes a demand from users.
- IPv6 hosting provides a greater number of available internet addresses and many other features, however, ISPs do not yet offer IPv6 services or support many of the features of this version of the IP.
- If one wants to deploy something new into the network, there is an impact on the stability of the network, routers must be upgraded, sometimes firmware must be changed, so an upgrade is needed more often, debugging this software is necessary and it costs extra efforts. IPv6 and IPv4 are incompatible protocols meaning that if one has at least one application that does not support it, then both protocols must be run.

This paper [3] surveys some tools and methods for measuring the deployment of IPv6, grouping them into different categories and comparing them from different aspects, distinguishing sources of data, whether public, private, or restricted, and the extent of the measurement duration, aiming to give an estimation of the IPv6 portion.

There are plenty of IPv6 technologies that have been developed to facilitate the co-operation of the two incompatible versions of IP (IPv4 and IPv6) for different scenarios [4]. One important scenario is, when IPv4 addresses ran out and only IPv6 addresses are being distributed to the clients, but there are still many old servers, which have only IPv4 addresses. A suitable solution for this scenario is the combination of NAT64 [5] and DNS64 [6]. This technology works well with the majority of the generally used client-server network applications [7]; however, there are some applications such as Skype which unable to use IPv6. For this reason, many providers, who would like to forget about IPv4 in the access and

DOI: 10.36244/ICJ.2022.3.5

core network, still must provide IPv4 to the customers, while they use solely IPv6 in their access and core network. It is called "IPv4 as a Service" (IPv4aaS) and there are several solutions were developed for this purpose. The advantages and disadvantages of the five most important IPv4aaS technologies are discussed in the following Internet Draft [8].

The remainder of this paper is organized as follows: Section II introduces the five most important IPv4aaS technologies and their proposed applied systems. Section III deals with their implementations. Section IV gives an introduction about benchmarking methodologies for IPv6 Transition Technologies. Section V is a conclusion of this paper.

II. THE FIVE MOST IMPORTANT IPV4AAS TECHNOLOGIES

A. 464XLAT

IPv6 hosts cannot communicate directly with IPv4 hosts and for this reason, several transitions methods have been developed: 464XLAT (RFC 6877) technology [9] is essentially an extension to NAT64 that provides the IPv4 access by combining stateful (RFC 6146) and stateless translation (RFC 6145).

464XLAT as a combination of stateless NAT64 (RFC 6145) and stateful NAT64 (RFC 6146) provides a lot of benefits [10] such as:

- It is easy to deploy and troubleshoot, using open-source standard technologies and based on RFC.
- It is efficient in terms of using IPv4 at minimum resource requirements and maximum efficiency.
- 464XLAT allows for full functionality and solves IPv4 numbering issues.
- IPv6-only networks are less expensive and simpler to operate, already proven by multi-vendor: Cisco, Juniper and F5.

464XLAT main components as shown in Fig. 1 are:

- CLAT (customer side translator) is a small piece of code that enables the client to have an IPv4 address. It translates 1:1 private IPv4 addresses to global IPv6 addresses, and vice versa [9].
- PLAT provider-side translator translates statefully IPv6 to IPv4 using stateful NAT64, it translates N:1 global IPv6 addresses to public IPv4 addresses and vice versa [9].

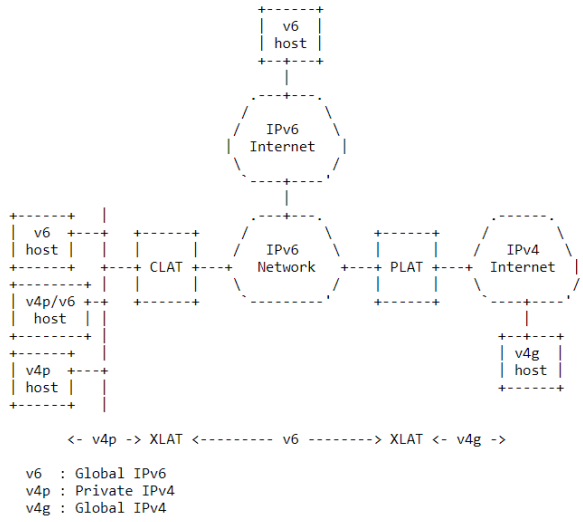

Fig. 1 464XLAT Wireline Network Topology [9]

We have elaborated an example (taken from RFC 6877 [9]) of IPv4/IPv6 address translation on the 464XLAT architecture:

- The IPv4 client with 192.168.1.2/24 private IP address is aiming to access the IPv4 server with 198.51.100.1 public IP address across an IPv6-only network.
- At the CLAT, IP routing is performed and different IPv6 prefixes are used for translation, the CLAT and the PLAT at this stage both know their IPv6 prefixes (the CLAT IPv6 prefix is 2001:8db:aaaa::/96, the PLAT IPv6 prefix is 2001:8db:1234::/96 in our example).
- The CLAT must do the translation process for the IPv4 packet to reach the IPv4 server, which means and as per our example, the destination address will be translated to 2001:db8:1234::198.51.100.1 and the source address will be translated to 2001:db8:aaaa::192.168.1.2. However, for reaching IPv6 hosts, the CLAT function is clearly dispensable.
- At the PLAT, before reaching the IPv4 server, the destination address is being extracted reversely back to its original 198.51.100.1, for the source IP address, 192.0.2.1 was chosen.
- At the server, the packets have successfully reached their destination.

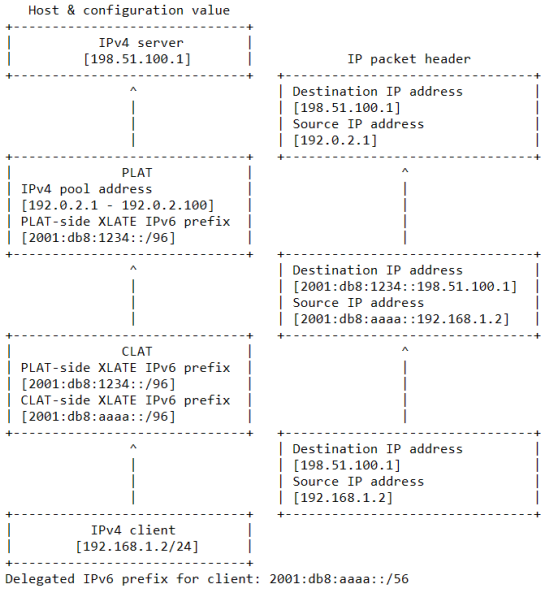

Fig. 2 464XLAT scenario [9]

As shown in Fig. 2, the example mentioned above requires double translation, however, if the client is an IPv6, then a single translation is necessary only at the PLAT, in which single stateful translation is enabled and in conjunction with a configured DNS64, as in RFC6146 in [9], CLAT is no longer needed. The DNS64 server in this case is responsible for constructing and returning a special IPv6 address called IPv4-Embedded IPv6 Address [9].

This solution is similar to NAT64, but the main difference is that the CLAT service needs to be installed on the mobile equipment. For example: Skype is an IPv4 only application, so it does not work with IPv6, the CLAT is to translate Skype clients IPv4 packets into IPv6 packets, the packets are then sent over an IPv6 only network to a NAT64 translator which translates them back into IPv4 and sends the packets to an IPv4 only server (Skype server). 464XLAT have helped a lot of mobile providers with the IPv6 implementations, because customers with 464XLAT can have an IPv6 only connection and still access all IPv4 only applications and content.

This recent paper [11] has been analysing the security aspects of this transition technology using STRIDE and Data Flow Diagram (DFD) methods, observing threats that the PLAT might face.

B. Dual Stack Lite or DS-Lite

DS-Lite stands for dual stack light (dual stack environment is one that has version 4 and version 6 addresses, too), DS-Light combines IP in IP (IPv4-in-IPv6) and Network Address Translation (NAT) technologies and allows IPv4 traffic to be encapsulated into IPv6 [12].

There are mainly two elements -as shown in Fig. 3- of DS-Lite as following:

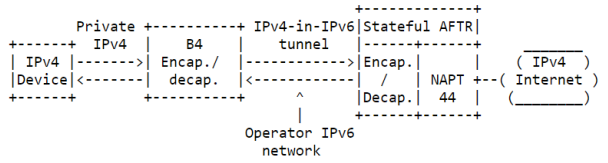

Fig. 3 Overview of the DS-Lite architecture [8]

1. B4, Basic Bridging Broadband element, which encapsulates IPv4 within IPv6, those IPv4 packets will go through an IPv6 network, B4 creates a multipoint-to-point IPv4-in-IPv6 tunnel to an AFTR [8].
2. AFTR, Address Family Transition Router receives the packets handled by the B4 and de-capsulates them. AFTR can reconstruct IPv6 when IPv4 packets come back from the Internet by doing a reverse lookup in the NAT binding table. AFTR is combination of IPv4-in-IPv6 tunnel endpoint and an IPv4-IPv4 NAT implemented on the same node [8].

IANA has defined a well-known range, 192.0.0.0/29 for numbering the interfaces of both B4 and AFTR [12].

As explained in RFC 6333 [12] and shown in Fig. 4, the goal is to carry IPv4 traffic over the IPv6 access and core network. In the case of outbound traffic, the message is first sent to the DS-Lite home router (B4) as “IPv4 datagram 1”, in which it's encapsulated and forwarded to the AFTR as “IPv6 datagram 2”. The AFTR decapsulates the IPv4 datagram from the IPv6 datagram and then the carrier-grade NAT44 is performed, “IPv4 datagram 3” is sent out [12]. In the case of inbound traffic, “IPv4 datagram 3” is received by the AFTR, NAT checks the information of its translation table and changes TCP destination port and sets the IP destination address, then it’s encapsulated into and IPv6 packet and forwarded to the home router B4. B4 decapsulates and extracts the IPv4 datagram and forwards it to the host. However, the packets are being dropped at the AFTR, when addresses are out of range, or the information does not match the NAT table.

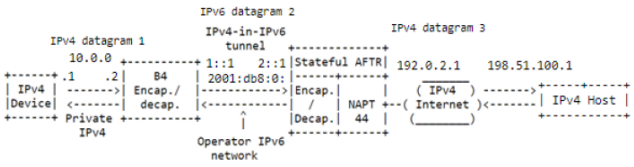

Fig. 4 Inbound and Outbound Datagram [based on 12]

C. Lightweight 4over6

Lightweight 4over6 is a transition mechanism as an extension of DS-lite; it has some of its concepts in providing IPv4 connectivity over IPv6, as well as the following main components:

- Lightweight B4 is the Lightweight Basic Bridging Broadband "lwB4" element that performs NAPT44 and creates a tunnel to a lwAFTR.

- Lightweight AFTR is the Lightweight Address Family Transition Router that is an IPv4-in-IPv6 tunnel.

The Lightweight 4over6 mechanism of sharing the addresses among clients is different from that of DS-Lite, as lw4o6 gives a portion of the port space to each client, the CPE is going to do a NAT and encapsulates the packets into IPv6, and the packets get through the border router. The lwAFTR task is to do a lookup in the binding table, which is a static table, once there is a match, lwAFTR de-capsulates the packets and forwards it to the Internet.

Lightweight 4over6 is a technology that flips the complexity of the dynamic address translation (between the LAN interface and the given public address) back to the client, where every CPE does the address translation and port-based NAT-ting, the difference in lightweight AFTR is that customers share the public IP address and each client gets the same IP address over limited port range to use, which makes this function stateless, where they all share the same binding table.

Lightweight 4over6 is a scalable solution where all routers are configured equally to load balance the traffic, for single flow packets can be distributed, and once the routing updates do not get through an instance fails then it's quickly picked up as another negligible hop and the traffic gets distributed to the other one.

As explained in RFC 7596 [13] and shown in Fig. 5, the following are the working scenarios of lwB4 and lwAFTR:

- LwB4 performs a NAPT44 function once receives an IPv4 packet, encapsulates it with an IPv6 header and forwards it to the lwAFTR as configured, while for the packet coming back from lwAFTR, lwB4 obtains IPv4 packet and performs NAPT44 translation based on the information in its NAPT44 table including the destination and port number, however, and when there is no match within what is configured, whether it's IPv6 packet at the lwAFTR, or its IPv4 packet at the lwB4, in both cases the packet is being discarded [13].

- LwAFTR performs a decapsulation and verification once receives an IPv4-in-IPv6 packet coming from lwB4. Based on the information in the binding table, lwAFTR verifies its source addresses and port, once there is a match the packet is forwarded to the IPv4 destination, otherwise it is discarded [13].

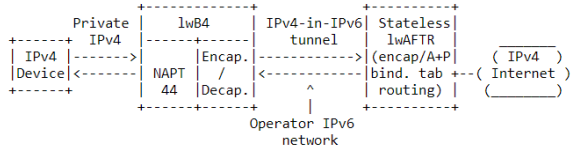

Fig. 5 Overview of the Lightweight 4over6 mechanism [8]

Paper [14] has been dealing with the design of an RFC 8219 compliant software tester for the performance analysis of the lw4o6 transition technology, disclosing the first lw4o6 tester, design considerations and important details of its operational requirements.

D. MAP

MAP stands for mapping of address and port, it's another transition mechanism; it basically maps the addresses and ports of IPv4 into the IPv6 addresses to serve IPv4 connectivity over IPv6 network, where IPv4aaS on top of IPv6 is being delivered using this stateless technology.

MAP (as in Fig. 6) is targeted access customer of broadband service providers, where it allows them to deploy an IPv6 infrastructure, MAP main components are:

- MAP Customer Edge: A home gateway acts as a CE router and provides IPv4/IPv6 stateless translation.

- MAP Border Relay (BR): a router from provider side supports stateless translation.

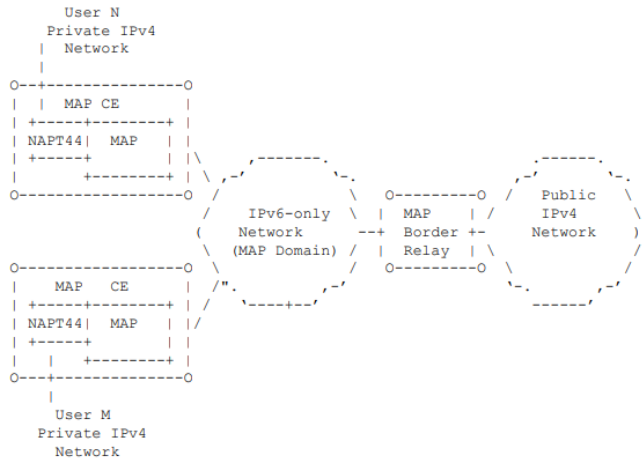

Fig. 6 MAP Network Topology [15]

MAP has two subtypes:

- MAP-T: Mapping of Address and Port using Translation.

- MAP-E. Mapping of Address and Port with Encapsulation.

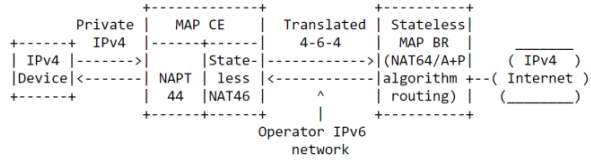

Fig. 7 MAP-T architecture [8]

TABLE I
DIFFERENT IMPLEMENTATIONS OF IPV6 TRANSITION TECHNOLOGIES

Name	*System*	*License*	*Function(s)*	*Technology*
CLATD [18]	Linux	open source	CLAT / SIIT-DC Edge Relay implementation	464XLAT
Android CLAT [19]	Android OS 4.3 jellybean or above.	open source	CLAT services through Wi-Fi connection	464XLAT
Cisco CGv6 [20]	Cisco	Licensed hardware and software	Supports stateless MAP technology to deliver both IPv4 and IPv6 services.	Stateless MAP Technology
Map [21]	Linux and OpenWrt	open-source repository	Supports both MAP-T and MAP-E and can be configured with or without NAPT44 function	MAP-T and MAP-E
SNABB [22]	Linux	open-source software	Has a large binding table with high performance.	Lw4o6
MAEMO [23]	MAEMO (OS2008 version)	licensed and open source	Tunnelling IPv6 through a tunnel broker.	DS-lite
PF [24]	BSD systems	Free software	Filter and manipulate IP packets.	464XLAT
Thunder CGN [25]	A10	Licensed hardware and software	Managing transition technologies, enabling providers to smoothly extend IPv4 connectivity and transition to IPv6	DS-Lite, lw4o6, MAP-T and MAP-E
Jool SIIT/NAT64 [27]	Linux	open-source software	BR as PLAT is stateful NAT64 and CLAT is an SIIT. High availability across Jool instances.	Stateful 464XLAT expected to support MAP-T
TAYGA[28]	Linux-based	open-source software	TAYGA is fast, flexible, and secure implementation.	Stateless NAT64
BIG-IP (CGNAT) [29]	F5	Licensed hardware and software	Stateful translation	464XLAT
Cisco ASR 9000[30]	Cisco ISM	Licensed hardware and software	ISM provides scalability in delivering services which supports CGN.	DS-Lite, Stateful NAT64, (MAP-T)
ASAMAP Vyatta [31]	Linux	Open source	Stateless address auto-configuration.	MAP-E, MAP-T, DS-Lite and 464XLAT
FD.io VPP [32]	Linux	Open source	Vector processing graph	Stateful NAT64, MAP-E, MAP-T and lw4o6.

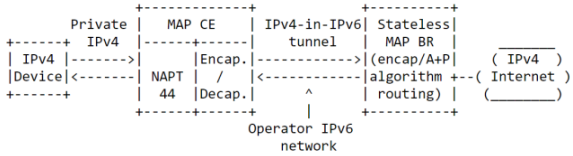

Fig. 8 MAP-E architecture [8]

MAP-T works as follows. When a CE (Customer Edge) device receives a packet that is destined to the public Internet, performs two transformations:

1. It does NAPT but with limited set of ports, the source port is being replaced, the source IP address is also being replaced with public IP address, while the destination address and port number remain the same [16].
2. Then it translates the IPv4 header into an IPv6 header using a stateless NAT46 translation [16].

Then the CE forwards the IPv6 packet to the MAP BR (Border Relay) device. BR performs just the inverse of the second translation (that is, a stateless NAT64) and forwards the resulting IPv4 packet to the public Internet. (Fig. 7).

MAP-E operates similarly to MAP-T, but it uses encapsulation and de-encapsulation instead of stateless NAT46 and stateless NAT64, respectively. (Fig. 8).

Their operation determined by various mapping rules, (Basic Mapping Rule, Forwarding Mapping Rule, Default Mapping Rule). All the details can be found in their RFCs.

The main benefit of this technology is that it is stateless at the center of the network, where no additional hardware is required even with the growth of the traffic. MAP-T [17] is one of two transform modes of the parent technology MAP, which aims to transport IPv4 over an IPv6 domain, while MAP-E [15] uses encapsulation, similarly to DS-Lite, where an IPv4 Packet is prepended with an IPv6 header and transported across the network, MAP-T uses IPv4 and IPv6 stateless translation, so the header translation as opposed to encapsulation. With MAP-T, the IPv4 addresses are embedded within the corresponding IPv6 address.

III. IMPLEMENTATIONS

There are many implementations for the most IPv4aaS (IPv4-as-a-Service) technologies, most of the implementations are free open source and they are usually preferred. **Table I** provides a summary for IPv6 transition technologies implementations.

A. CLATD

CLATD [18] - a CLAT / SIIT-DC Edge Relay implementation for Linux is free software available to implement the CLAT component of the 464XLAT network.

B. Android CLAT

Android CLAT [19] is an open -source application already installed for any Android OS 4.3 jellybean or above. This solution relies on the routing table in order to separate traffic, this implementation does not support IPv6 content only; it was mainly designed to offer CLAT services through Wi-Fi connection.

C. Cisco CGv6

Cisco CGv6 [20] supports stateless MAP technology to deliver both IPv4 and IPv6 services more efficiently on a high scale at a lower cost and less latency. Machine to Machine services is an advantage of this technology.

D. Map

Map [21] is an open-source repository supports both MAP-T and MAP-E and can be configured with or without NAPT44 function. This software is also compatible with AFTR of DS-Lite and NAT64 (stateful and stateless), this CPE implementation runs on Linux and OpenWrt.

E. SNABB

SNABB [22] is fully compatible open-source software with Lightweight 4over6 that has a large binding table with high performance, it consists mainly three elements: APP (Filter, lwAFTR), Programs and links to connect applications together. The 3rd version of SNABB supports YANG IETF. The 4-th version of SNABB supports RSS (Receive Side Scaling) multiprocessor and YANG Alarm Module.

F. MAEMO

MAEMO, this implementation requires an N810 Nokia tablet as hardware and one of the supported software's listed in [23], the idea behind this is tunnelling IPv6 through a tunnel broker sending and receiving IPv4 Packets.

G. OpenBSD Packet Filter

PF [24] is an abbreviation of Packet Filter subsystem which is a free software released with OpenBSD 3.0 in 2001, and contained a rather complete implementation of packet filtering, including network address translation (NAT64) [25]. Packet filter controls the flow of the packets on interfaces, it differentiates whether its TCP or UDP, it recognizes the source and destination IP addresses or layer 3 addresses.

H. Thunder CGN:

The Thunder CGN [26] is a scalable secure implementation thorough hardware and software solutions provided by A10, managing transition technologies, and enabling providers to smoothly extend IPv4 connectivity and transition to IPv6, that is including DS-Lite, lw4o6, MAP-T, and MAP-E transition technologies. However, it's worth mentioning that it's not free and prices vary from device to another that usually comes with extra yearly service and maintenance cost.

I. Jool SIIT/NAT64

Jool [27] is an open-source software and reflects an implementation of 464XLAT transition technology in which PLAT is a stateful NAT64, whereas CLAT is an SIIT. One of the most important features is the high availability across Jool instances. It's worth mentioning that Jool is presently in late development for MAP-T transition technology.

J. TAYGA

TAYGA [28] is a Linux-based stateless NAT64 implementation, packets are exchanged with the help of TUN driver. TAYGA is also: fast, flexible, compatible, secure, and most importantly it is free, however, it could not offer stateful solution. It is usually combined with iptables (stateful NAT44 for Linux) to implement a stateful NAT64 solution.

K. F5 BIG-IP Carrier-Grade NAT (CGNAT)

Widely deployed, provides scalable and high-performance network, F5 [29] implemented 464XLAT transition mechanism to deliver IPv4 and IPv6 connectivity.

L. Cisco ASR 9000

Cisco ASR 9000 [30] is an Integrated Service Module (ISM) that provides scalability in delivering services which supports Carrier Grade Network Address Translation (NAT) or CGN, Dual-Stack Lite, Stateful NAT64, and Mapping of Address and Port Translation (MAP-T), in which multiple can coexist on multiple ISMs with a lot of major features and benefits, yet this option is costly.

M. ASAMAP Vyatta

Vyatta [31] is a system that supports stateless configuration with the help of SLAAC protocol which has a host and a router as main components; however, DHCPv6 is not supported by Vyatta. ASAMAP Vyatta supports MAP-E, MAP-T, DS-Lite and 464XLAT.

N. FD.io VPP

Vector Packet Processing [32] is the heart of FD.io; it is the open-source version of Cisco's Vector Packet Processing (VPP) technology, the VPP is faster than current technologies; as it processes through vector processing graph at extreme performance, VPP supports Stateful NAT64, MAP-E, MAP-T and lw4o6.

IV. BENCHMARKING METHODOLOGY FOR IPV6 TRANSITION TECHNOLOGIES

In this section, a short introduction is given to the benchmarking methodology for IPv6 transition technologies, furthermore, a summary of several papers that investigated the performance of IPv6 transition technologies are added.

The goal of RFC 8219 [33] is to provide meaningful and unbiased results by measuring performance characteristics of various IPv6 transition technologies. There are two well-known RFCs about the benchmarking methodology for network interconnect devices: RFC 2544, which is theoretically IP version independent, but relies on IPv4 and the specificities of IPv6 are addressed in RFC 5280, in which IPv6 transition

technologies excluded. On the other hand, RFC 8219 is a new one handling IPv6 transition technologies.

RFC 8219 [33] helped in classifying a massive number of transition technologies into a much smaller number of categories. Model of production network was used to achieve this purpose, in which we have two different domains IPvX and IPvY and called domain A and domain B respectively. The scenarios are as following:

- Single translation: Domain A needs to be translated to be able to communicate to Domain B. Stateful NAT64, SIIT, and IVI are transition technologies that can solve this problem.
- Double translation: There are three domains, in which domain A and B are version 4 specific and the core domain is version 6 specific. 464XLAT and MAP-T both are examples of this production model.
- Encapsulation: Any version can be encapsulated/decapsulated into another version, DS-Lite, and MAP-E both are examples of this production model.

DNS64 is an additional protocol and does not transfer data packets, which is just required to support NAT64, thus it does not fit into any of these categories mentioned above, and it is to be dealt with separately. The problem with these scenarios, the packet format and size can be changed during the process of translation, that is why those methods must be calibrated, two test setups are defined to solve this issue:

Test Setup 1, for Single-Translation where the DUT (device under test) is translating the IPvX packets into IPvY packets as shown in Fig. 9. As for Test Setup 2 (Double-Translation), there are two DUTs. One DUT implements the reverse operation of the other one: if one DUT does encapsulation the other is decapsulation, if one is translating 4 to 6, the other is translating 6 to 4 as in Fig. 10. In case of testing as peers together we may use RFC 2544 [33] tester, however, if there is asymmetric behaviour, then we will not be able to observe it and in this case, we should use Test Setup 1.

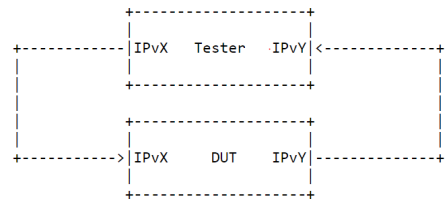

Fig. 9 Single DUT Test Setup for benchmarking[33]

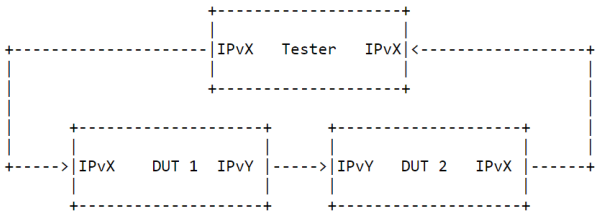

Fig. 10 Dual DUT Test Setup for benchmarking [33]

RFC 8219 recommended important benchmarking measurement tests, each with different requirements, such as: Throughput, Latency, Frame Loss Rate, Packet Delay Variation.

For double translation (either in stateless or stateful) same tests can be used, as well as different test setups for example dual and single DUT, the latter is recommended to observe asymmetric behaviour. Similar procedures for encapsulation, however packets that are encapsulated must be provided to prepare a tester. For stateless tests, UDP is used, for stateful tests, (all RFC 3511) TCP is used.

As for DNS64 benchmarking, based on RFC8219 in [33], the tester implements two different logical functions: version 6 only-client and an authoritative DNS server, it can be implemented by two different devices or similar devices. The test traffic of the DNS64 benchmarking is as following (shown in Fig. 11):

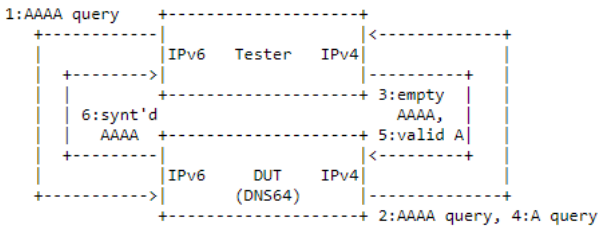

Fig. 11 DNS64 DUT Test Setup for benchmarking [33]

1. The IPv6-only client sends “AAAA” record query (IPv6 address) for a domain name.
2. The DNS64 server receives the request, sends “AAAA” record query for the given domain name to the authoritative DNS server.
3. If there is no such “AAAA” record, then an empty “AAAA” record is being returned.
4. The DNS64 server sends another query asking for “A” record of the same domain name.
5. The authoritative DNS System replies with a valid “A” record (IPv4 address).
6. The DNS64 server synthesizes an *IPv4-embedded in IPv6 address*, which is returned to the IPv6-only client.

When the DNS64 server implements caching and there is a cache hit, then step 1 is followed by step 6, and for message 1 the answer is message 6. The goal here again is to determine performance (requests processed per second), in other words, the *rate* between messages sent and received. A test should last at least 60 seconds and *timeout* should be not more than 1 second. However, the measurement may be influenced by the tasks executed by the device in the background, so the *median* of the results of the repetitive measurements is calculated to get a better understanding of the performance.

As mentioned before, while the IPv6 demands in solving the IP address shortage is expanding, there are several papers experimenting transition technologies of IPv6, utilizing

TABLE II
SUMMARY OF TRANSITION TECHNOLOGIES COVERED BY EACH REFERENCE

IPv6 Transition	*[39]*	*[41]*	*[42]*	*[43]*	*[44]*	*[45]*	*[46]*
464XLAT	✓						
MAP-E	✓						
MAP-T	✓						
Dual Stack		✓	✓		✓	✓	✓
DS-lite	✓			✓			
4over6				✓			✓
6to4		✓					
4rd				✓			

different implementations and network environments, each depending on different factor such as: type of test, configuration, technology, topology and more as discussed in [34].

Moving toward benchmarking methodologies and tools, siitperf is "an RFC 8219 compliant SIIT (stateless NAT64) tester written in C++ using DPDK" [35]. The accuracy of siitperf is examined in [36], by structuring an error model and discussing what could influence the measurements and cause inaccuracy, observing the effect of Ethernet flow-control, concluding that calibrating siitperf is a necessity.

There is a novel Internet Draft about an RFC 8219 compliant methodology for benchmarking stateful NATxy (x, y are in {4, 6}) gateways [37]. Its proposed benchmarking procedures are implemented as an extension of siitperf for stateful tests [38].

IPv6NET is a network evaluation testbed built as a combination of closed and open environments [39]. For the closed environment, ASAMAP Vyatta implementation was used, multiple IPv6 transition technologies were considered including MAP (both MAP-E and MAP-T), 464XLAT, and DS-Lite. The traffic was generated by a distributed Internet traffic generator (D-ITG), two functions were performed in each computer, one to send (ITGSend) and one to receive (ITGRecv). During the process and based on the recommendation of RFC5180, frame size and frame rates were considered. They monitored the following network performance metrics: Round-trip-delay, jitter, packet loss, and throughput. Overall MAP-E achieved the best performance in a closed environment.

As for the open environment, three associated operational feasibility metrics were introduced: configuration, troubleshooting, and application capabilities. Inspired by [40] three configuration task groups were organized associated with a task code: initial setup, reconfiguration, and confirmation. They concluded that applications capability was running smoothly for all four technologies, in regard to configuration capability, an addition of a guided self-configuration would be beneficial, for troubleshooting capabilities improvements are needed.

Based on the empirical results, it was found that MAP-E was more feasible compared to other transition technologies. MAP-T and 464XLAT had a better performance in terms of latency as translation-based technology, on the other hand, MAP-E and Ds-Lite had a better performance in terms of throughput as encapsulation-based technology, IPv6NET has shown that it has a high level of repeatability, one flaw in IPv6NET is the lack of control data.

This research [41] examined three IPv4/IPv6 transition mechanisms of dual stack, the manual tunnel, and the 6to4 automatic tunnel through three metrics: delay, delay variation, and packet loss by using the Optimized Network Engineering Tool (OPNET) Modeler simulator, on a real-time application (video conferencing). The performance results show that dual-stack had better performance than the others with the lowest average delay, dual-stack has shown efficiency in terms of packet delay variation and with a lower loss rate. Hence, the Dual-Stack was the best. Consequently, both tunnelling mechanisms results were deficient, and this is due to the encapsulation and decapsulation processes.

Network analysis was performed in [42] for three different transition technologies, namely: Dual-Stack, 6in4, and NAT-PT, they were compared using Cisco packet tracer, and for this purpose, three main performance metrics were taken into consideration: Round Trip Delay Time (RTT), Bandwidth and Throughput. The results have shown that NAT-PT due to its high latency and low throughput, was neglected, dual-stack had better performance and was preferred.

Chuangchunsong et al. [43] compared delay time, and reliability for four different transition scenarios: 4over6, DS-lite, 4rd: NAT Centralization, and 4rd: NAT Distribution by using OPNET. Results have shown that both 4rd have high performance and high reliability, but both are inflexible in IP address allocation. 4over6 has also shown a similar result to 4rd, but with lower performance compared to other transition mechanisms. On the other hand, DS-Lite only on inter-communication has shown relatively high performance and reliability, but also high flexibility. Conversely, for intra-communication, the DS-Lite has low performance and low reliability, but rather has less complexity and higher compatibility compared to other mechanisms.

This empirical measurement in [44] conducted a performance study of IPv6 and IPv4 through dual-stack sites from all over the world, using performance metrics: connectivity, throughput, packet loss, hop count, and round-trip time (RTT), considering different regions and times. Compared with IPv6, IPv4 had higher latency and lower throughput with intangible improvements since 2004, IPv6, however, had lower packet loss rate and better connectivity. The average hop count of the IPv6 network is very similar to that of IPv4.

Proving that dual stack is the best technique, achieving better performance in solving the limitations of IPv4, [45] proposed a methodology of four phases: Build & Design network, Statistics, Simulator, and the results of the analysis. The analysis and based on three different scenarios (IPv4, IPv6, and

Dual-Stack), were compared using Riverbed simulator, evaluating five performance metrics*:* Delay, Traffic dropped, Jitter, Packet delay, and CPU Utilization. The results have shown that Dual-stack surpassed IPv4 and gave a better performance.

This paper [46] proposed comparing 4over6 and dual stack by tracking seven nodes and measuring the average time, the results have shown that the average time for Dual stack had a higher performance by over 17%.

Table II concludes a summary of those different references that has been analysed for different IPv6 transition technologies.

V. CONCLUSION

The future of communications and networks, IPv6 transition technologies are the key to solving the shortage and limitations of IPv4, the question remains how both would be compatible and coexist together effectively, paving the way to develop transition technologies based on different metrics, and factors such as throughput, jitter, packet loss, delay and so forth. Consequently, this paper has fully disclosed the necessity of deploying IPv6 technologies, explained their most promising IPv4aaS transition technologies, namely, 464XLAT, DS-Lite, lw4over6, MAP-E, MAP-T, their operation mechanism, advantages, and disadvantages, analysed and examined existing solutions, collected their most important implementation cases, and gave an introduction about benchmarking methodologies, surveyed some papers, considering and analysing their outcomes. To summarize, 464XLAT is easy to deploy and efficient in using minimum resources, dual-stack lite can work with Interoperability; allowing IPv4 and IPv6 content to reach hosts simultaneously, MAP is a stateless scalable transition technology. Contrarily, 464XLAT needs additional service on the client or on the network, Lw4o6, MAP-E, and MAP-T require more planning and re-provisioning, 464XLAT and Ds-lite may have scalability issues being a stateful and keeping per-flow mapping information between IPv4 and IPv6 addresses.

VI. ACKNOWLEDGEMENTS

The author thanks his supervisor; Dr. Gabor Lencse, Budapest University of Technology & Economics, for helping him by reviewing and commenting the manuscript.

REFERENCES

[1] O. D'yab, "An overview of the most important implementations of IPv4aaS technologies", *AIS 2019*, University of Óbuda, Székesfehérvár, Hungary (2019) pp. 143–146.

[2] IANA IPv4 Address Space Registry. [Online]. Available: https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml

[3] V. Iró, G. Lencse, "Survey on Measurement Methods for IPv6 Deployment", *Acta Technica Jaurinensis*, vol 13, no 2, pp. 112–130, May 27, 2020, DOI: 10.14513/actatechjaur.v13.n2.544

[4] G. Lencse 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

[5] M. Bagnulo, P. Matthews, I. V. Beijnum, Stateful NAT64: "Network address and protocol translation from IPv6 clients to IPv4 servers", IETF RFC 6146 (2011). DOI: 10.17487/RFC6146

[6] M. Bagnulo, A. Sullivan et al., "DNS64: DNS extensions for network address translation from IPv6 clients to IPv4 servers", IETF RFC 6147 (2011), DOI: 10.17487/RFC6147

[7] S. Répás, T. Hajas, G. Lencse, "Application compatibility of the NAT64 IPv6 transition technology", in *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, 2015, pp. 1–7. DOI: 10.1109/TSP.2015.7296383

[8] G. Lencse, J. P. Martinez et al., "Pros and cons of IPv6 transition technologies for IPv4aaS", approved Internet Draft, May 23, 2022. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-v6ops-transition-comparison-03

[9] M. Mawatari, M. Kawashima, C. Byrne, "464XLAT: Combination of stateful and stateless translation", IETF RFC 6877 (2013). DOI: 10.17487/RFC6877

[10] UK IPv6 Council: 464xlat for mobile operators. [Online]. Available: https://www.ipv6.org.uk/wp-content/uploads/2018/11/Nick-Heatley_BT_EE_464xlat_UKv6Council_20180925.pdf

[11] A. Al-Azzawi and G. Lencse, "Identification of the Possible Security Issues of the 464XLAT IPv6 Transition Technology", *Infocommunications Journal*, vol. 13, no. 4, pp. 10–18, December 2021, DOI: 10.36244/ICJ.2021.4.2

[12] A. Durand, R. Droms et al., "Dual-stack lite broadband deployments following IPv4 exhaustion", IETF RFC 6333 (2011). DOI: 10.17487/RFC6333

[13] Y. Cui, Q. Sun et al., "Lightweight 4over6: An extension to the dual-stack lite architecture", IETF RFC 7596 (2015). DOI: 10.17487/RFC7596

[14] A. Al-hamadani and G. Lencse, "Design of a Software Tester for Benchmarking Lightweight 4over6 Devices", *44th International Conference on Telecommunications and Signal Processing (TSP 2021)*, Brno, Czech Republic, July 26-28, 2021, pp. 157–161, DOI: 10.1109/TSP52935.2021.9522607

[15] E. O. Troan, W. Dec et al., "Mapping of address and port with encapsulation (MAP-E)", IETF RFC 7597 (2015). DOI: 10.17487/RFC7597

[16] MAP - Solving IPv6 Deployment and IPv4 Address Exhaustion without Stateful CGN: CKN TechAdvantage Webinar. [Online]. Available: https://community.cisco.com/t5/networking-knowledge-base/ckn-techadvantage-webinar-map-solving-ipv6-deployment-and-ipv4/ta-p/3639138

[17] X. Li, C. Bao, W. Dec (ed), O. Troan, S. Matsushima, T. Murakami, "Mapping of address and port using translation (MAP-T)", IETF RFC 7599, July 2015. DOI: 10.17487/RFC7599

[18] Clatd - a CLAT / SIIT-DC Edge Relay implementation for Linux. [Online]. Available: https://github.com/toreanderson/clatd

[19] What is Android CLAT. [Online]. Available: https://dan.drown.org/android/clat

[20] Carrier Grade IPv6 over Integrated Services Module (ISM). [Online]. Available: https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k_r5-2/cg-nat/configuration/guide/b_cgnat_cg52xasr9k/b_cgnat_cg52xasr9k_chapter_010.html

[21] MAP, source code of MAP CE. [Online]. Available: https://github.com/cernet/MAP

[22] Lightweight4over6, one step further dual-stack lite networks. [Online]. Available: https://ripe76.ripe.net/presentations/105-lw4o6-ripe.pdf

[23] DS-Lite Host Profile for MAEMO (OS2008 version), HW and SW Requirements. [Online]. Available: http://ds-lite.garage.maemo.org/

[24] P. N. M. Hansteen, the Book of PF: A No-Nonsense Guide to the OpenBSD Firewall, 2nd ed., San Francisco: No Starch Press, 2010. ISBN: 978-1593272746.

[25] OpenBSD manual page server. [Online]. Available: https://man.openbsd.org/pf.4

[26] Extend IPv4 Investment and Transition from IPv4 to IPv6 Seamlessly. [Online]. Available: https://www.a10networks.com/wp-content/uploads/A10-SB-19104-EN.pdf

[27] Introduction to IPv4/IPv6 Translation. [Online]. Available: https://www.jool.mx/en/intro-xlat.html#ipv4ipv6-translation.html

[28] TAYGA, Simple, no-fuss NAT64 for Linux. [Online]. Available: http://www.litech.org/tayga/

[29] Using DS-Lite with CGNAT. [Online]. Available: https://techdocs.f5.com/kb/en-us/products/big-ip_ltm/manuals/product/cgn-implementations-11-5-0/14.html

[30] Cisco ASR 9000 Series Aggregation Services Routers. [Online]. Available: https://www.cisco.com/c/en/us/products/routers/asr-9000-series-aggregation-services-routers/index.html

[31] Guide to IPv6 Support. [Online]. Available: https://docs.huihoo.com/vyatta/6.0/Vyatta_IPv6_R6.0_v03.pdf

[32] The Vector Packet Processor (VPP). [Online]. Available: https://s3-docs.fd.io/vpp/22.02/

[33] M. Georgescu, L. Pislaru, G. Lencse, "Benchmarking Methodology for IPv6 transition technologies", IETF RFC 8219 (2017). DOI: 10.17487/RFC8219

[34] A.T.H.Al-hamadani,G.Lencse,"Asurveyontheperformanceanalysis of IPv6 transition technologies", *Acta Technica Jaurinensis*, vol. 14, no. 2, pp. 186–211, May 26, 2021. DOI: 10.14513/actatechjaur.00577

[35] G. Lencse, "Design and Implementation of a Software Tester for Benchmarking Stateless NAT64 Gateways", IEICE Transactions on Communications, vol. E104-B, no. 2, pp. 128–140. February 1, 2021. DOI: 10.1587/transcom.2019EBN0010

[36] G. Lencse, "Checking the Accuracy of Siitperf", Infocommunications Journal, vol. 13, no. 2, pp. 2–9, June 2021, DOI: 10.36244/ICJ.2021.2.1

[37] G. Lencse, K. Shima, "Benchmarking methodology for stateful NATxy gateways using RFC 4814 pseudorandom port numbers", active Internet Draft, May 17, 2021, draft-lencse-bmwg-benchmarking-stateful-00

[38] G. Lencse, "Design and implementation of a software tester for benchmarking stateful NATxy gateways: Theory and practice of extending siitperf for stateful tests", *Computer Communications*, vol. 172, no. 1, pp. 75–88, August 1, 2022, DOI: 10.1016/j.comcom.2022.05.028

[39] M. Georgescu, H. Hazeyama et al., "Empirical analysis of IPv6 transition technologies using the IPv6 Network Evaluation Testbed", *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol 2, no. 2, (2015). DOI: 10.4108/inis.2.2.e1

[40] D. Harrington, "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions." RFC 5706 (Informational), Nov. 2009.

[41] G. Altangerel, E. Tsogbaatar, D. Yamkhin, "Performance analysis on IPv6 transition technologies and transition method", in *2016 11th International Forum on Strategic Technology (IFOST)*, 2016, pp. 465–469. DOI: 10.1109/IFOST.2016.7884155

[42] J. L. Shah, J. Parvez, "An examination of next generation IP migration techniques: Constraints and evaluation", in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2014, pp. 776–781. DOI: 10.1109/ICCICCT.2014.6993064

[43] N. Chuangchunsong, S. Kamolphiwong, T. Kamolphiwong, R. Elz and P. Pongpaibool, "Performance evaluation of IPv4/IPv6 transition mechanisms: IPv4-in-IPv6 tunneling techniques", *The International Conference on Information Networking 2014 (ICOIN2014)*, 2014, pp. 238–243. DOI: 10.1109/ICOIN.2014.6799698

[44] Li, K.-H.; Wong, K.-Y. "Empirical Analysis of IPv4 and IPv6 Networks through Dual-Stack Sites". *Information 2021*, 12, 246. DOI: 10.3390/info12060246

[45] M. R. A. Ahmed and S. S. A. Shaikhedris, "Network Migration and Performance Analysis of IPv4 and IPv6", *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2021, pp. 1–6. DOI: 10.1109/ICCCEEE49695.2021.9429664

[46] Lu, T.T., Wu, C.Y., Lin, W.Y., Chen, H.P., Hsueh, K.P. (2017). "Comparison of IPv4-over-IPv6 (4over6) and Dual Stack Technologies in Dynamic Configuration for IPv4/IPv6 Address". In: Pan, JS., Tsai, PW., Huang, HC. (eds) Advances in Intelligent Information Hiding and Multimedia Signal Processing. Smart Innovation, Systems and Technologies, vol 63. Springer, Cham. DOI: 10.1007/978-3-319-50209-0_32

Omar D'yab received his MSc in Computer Science Engineering from Óbuda University, Budapest, Hungary in 2019. He is now a PhD student at the Budapest University of Technology and Economics, Department of Networked Systems and Services in the MédiaNets Laboratory.

His research area covers IPv6 transition technologies for IPv4aaS and their performance analysis.