SURVEY PAPER

Comprehensive Survey of IPv6 Transition Technologies: A Subjective Classification for Security Analysis

Gábor LENCSE[†a)] *and* Youki KADOBAYASHI[††b)], *Members*

SUMMARY Due to the depletion of the public IPv4 address pool, the transition to IPv6 became inevitable. However, this ongoing transition is taking a long time, and the two incompatible versions of the Internet Protocol must coexist. Different IPv6 transition technologies were developed, which can be used to enable communication in various scenarios, but they also involve additional security issues. In this paper, first, we introduce our methodology for analyzing the security of IPv6 transition technologies in a nutshell. Then, we develop a priority classification method for the ranking of different IPv6 transition technologies and their most important implementations, so that the vulnerabilities of the most crucial ones may be examined first. Next, we conduct a comprehensive survey of the existing IPv6 transition technologies by describing their application scenarios and the basics of their operation and we also determine the priorities of their security analysis according to our ranking system. Finally, we show that those IPv6 transition technologies that we gave high priorities, cover the most relevant scenarios.

key words: *IPv6 transition technologies, network security, survey*

1. Introduction

Although IPv6, the new version of the Internet Protocol, was defined in 1998 (by a *Draft Standard* state RFC [1]), it has become an *Internet Standard* only in 2017 [2]. Similarly, the deployment of IPv6 was very slow at the beginning, and it started to accelerate only in the latest years for several reasons [3]. Unfortunately, the old version, IPv4, and the new version, IPv6, are incompatible with each other. To resolve this issue, several IPv6 transition technologies [4] have been developed, which address various *communication scenario*s. (Under communication scenario, we mean the problem to be solved, e.g. a client, which can use only IPv6, needs to communicate with a server, which can use only IPv4.)

In our workshop paper [5], we have surveyed the IPv6 transition technologies to have a general picture, what kind of solutions exist. Our results helped us to develop a methodology for the identification of potential security issues of the various IPv6 transition technologies [6].

In this paper, we extend our workshop paper [5] by conducting a comprehensive survey of the IPv6 transition technologies (including any protocols that can be used to enable communication in any scenario despite the incompatibility of IPv4 and IPv6) and identifying those of them, which would be worth submitting to a detailed security analysis. To achieve this goal, first, we give a short introduction to our methodology for the security analysis of IPv6 transition technologies [6], then we develop a priority classification method for both the technologies and their most important implementations, and after that, we present an exhaustive overview of the existing IPv6 transition technologies together with their priority classification.

The aim of this paper is twofold:

- Its primary goal is to serve as a reference for all IPv6 transition technologies defined up to now.
- Its secondary goal is to select those technologies that will play the most important role in the transition to IPv6, which we are headed with for several years or perhaps decades.

In this way, our current paper is the next step of the research that targets to identify and mitigate the security vulnerabilities of the most important IPv6 transition technologies.

We contend that an up-to-date comprehensive survey of IPv6 technologies is needed, because other surveys than our workshop paper [5] are either too old, like [7], [8] and [9] (published in 2006, 2010 and 2011, thus may not contain the most relevant technologies), or cover only a low number of technologies, like [10] and [11]. The best survey on IPv6 transition technologies we have found includes a thorough classification of the methods [12], however, it was published in 2013, thus it also omits some important novel technologies defined since then. Another excellent paper [13] also covers several IPv6 transition technologies, but it focuses on the IPv4 address sharing mechanisms. Therefore, we conclude that there is a need for an up-to-date comprehensive survey of IPv6 transition technologies.

The remainder of this paper is organized as follows. In Section 2, we give a very brief introduction to our methodology for the identification of potential security issues of different IPv6 transition technologies. In Section 3, we disclose our priority classification method. In Section 4, we survey all the existing IPv6 technologies and classify the importance of their analysis. In Section 5, we discuss our recommendations by reconsidering the most important scenarios from the viewpoints of the users, ISPs and content providers. We check the sufficiency and parsimony of our

Manuscript received September 7, 2018.
Manuscript revised February 2, 2019.

[†]The author is with the Department of Networked Systems and Services, Budapest University of Technology and Economics, Magyar tudósok körútja 2, H-1117 Budapest, Hungary.

[††]The author is with Laboratory for Cyber Resilience, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 630-0192, Japan.

a) E-mail: lencse@hit.bme.hu
b) E-mail: youki-k@is.naist.jp

DOI: 10.1587/transcom.E0.B.1

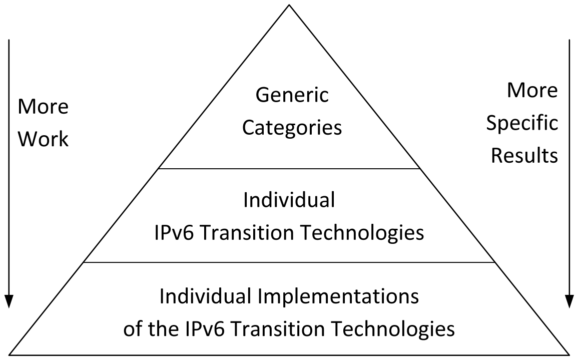
More
Work
Generic
Categories
More
Specific
Results
Individual
IPv6 Transition Technologies
Individual Implementations
of the IPv6 Transition Technologies

Fig. 1 Method hierarchy: Costs and benefits of the different threat analysis methods. [6]

selections. Section 6 concludes our paper.

2. Our Methodology for the Security Analysis of IPv6 Transition Technologies in a Nutshell

We have developed a methodology for the identification of potential security issues of different IPv6 transition technologies [6]. This methodology is based on STRIDE, which is the abbreviation of Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, and Elevation of privilege. STRIDE was developed for software design, and uses a systematic approach to help uncovering potential vulnerabilities [14]. STRIDE operates on the DFD (Data Flow Diagram) model of the system and it examines whether the building blocks of the DFD are susceptible to the above mentioned six vulnerabilities. Marius Georgescu recommended one approach to applying the STRIDE approach to the security analysis of IPv6 transition technologies [15]. That paper used the STRIDE method for examining the possible vulnerabilities of the following four categories of IPv6 transition technologies: dual stack, single translation, double translation, and encapsulation. We found that approach very promising, and we have complemented it in two ways [6]:

- We have pointed out that DNS64 was not covered by the above mentioned four categories, and added a new category for DNS64 [16].
- We have also shown that the general categories, which are useful for a comprehensive analysis at basic level, are worth complementing with deeper analysis at two levels: at the level of the individual IPv6 transition technologies and at the level of their most prominent implementations, see Fig. 1.

Please refer to our paper [6] for an in-depth description of our new methodology and for the demonstration of its operability on the example of DNS64 [16] and Stateful NAT64 [17].

From our survey point of view, our methodology results in a few constraints (or consequences). First of all, the operation of the IPv6 transition technologies selected for deeper analysis needs to be public and well-defined to be able to apply the STRIDE approach. Furthermore, we decided in [6] to consider only those implementations that are free software [18] (also called open source [19]) for multiple reasons:

- “Free software comes with source code and free software licenses explicitly allow the study of the source code, which can be essential for security analysis.
- Proprietary software usually does not include source code, and the licenses of certain vendors (e.g. [20] and [21]) do not allow reverse engineering and sometimes even the publication of benchmarking results is prohibited.
- Free software can be used by anyone for any purposes thus our results can be helpful for anyone.
- Free software is available free of charge for us, too.” [6]

3. Our Priority Classification Method

3.1 General Considerations

IETF has standardized several technologies and occupied a neutral position trusting the selection of the most appropriate ones to the market. Therefore, several IPv6 transition technologies exist even for the same scenarios, and some of them have many implementations, thus the thorough analysis of all of them would require a huge amount of resources. Therefore, we develop a simple method for their priority classification both at IPv6 transition technology level and at implementation level. Our aim is to choose only a few number of technologies into the highest priority classes to be able to start our security analysis with the most important technologies and their most promising implementations. We contend that on the one hand, using only formal criteria would not lead to meaningful results (e.g. too many technologies would satisfy them) but on the other hand, complex expert deliberation always contains arguable elements. We are aware that any such ranking systems have their limits. (E.g. considering too few factors, we may oversimplify the problem, whereas considering too many factors, we may make the problem too complex.) The choice of the examined factors, the determination of their relative priority (or using a weighting system) are also subjective decisions.

3.2 Conditions for Classification

First, we define some expressions that will be used in the classification as conditions. We consider that the operation of an IPv6 transition technology is satisfactorily public and *well-defined*, if it is *defined by a valid* (non-obsoleted) *IETF RFC*. We call a well-defined solution also *standard*, if the IETF RFC is of *at least proposed standard state*. We call the solution *obsolete* if the *defining RFC was obsoleted by another RFC* (and no new version was defined). In all other cases, we call the solution *not well-defined*. (For a further fine grain classification of the not well-defined solutions, we introduce the expression *fairly defined* for those solutions, which have at least some kind of public and stable formal definition, such as an expired and abandoned Internet Draft.)

Table 1 Priority classes (smaller value means higher importance)

Class	Category	Description
1	Important/essential	The only standard solution for a relevant scenario.
2	Important/replaceable and selected	A well-defined solution for a relevant scenario, which we selected.
3	Important/replaceable and not selected	A well-defined solution for a relevant scenario, which we did not select.
4	Optional	A well-defined solution for a non-relevant scenario or a fairly defined solution for a relevant scenario with significant deployment.
-	Negligible	Solution is obsolete or it does not meet the requirements of class 4.

As for communication scenarios (that is the problem to be solved by a given IPv6 transition technology), we call a scenario *relevant*, if the scenario is *common* (there are or there will be a high number of users) and *unavoidable* (its usage is not based on someone's unwise selection, but it is really necessary). Of course, both being common and being unavoidable are questionable, but this is the nature of the beast, as they both refer to real life situations.

We are aware that the actual and future *deployments* of the different IPv6 transition technologies (and of their implementations) are important factors of the usefulness of their security analysis. The problem is that we have only very limited information of the current deployment of the different IPv6 transition technologies (see Section 3.4) and any prediction of their future deployment is questionable. Thus, we consider our incomplete deployment information with restrictions.

3.3 Priority Classes

We classify an IPv6 transition technology as *important*, if it is at least *well-defined* and the communication scenario is *relevant*. An *important* technology is also *essential*, if the technology is the only known *standard* solution for the given scenario, otherwise it is *replaceable*. The security analysis of essential technologies will have the very first priority (class 1).

When there are more than one well-defined solutions exist for a relevant scenario, formally they could be treated as equal as they all belong to the *replaceable* category. Our policy is to select one (or a few) of them for each communication scenario and deal with them first (class 2), and the others may follow later on (class 3). For this selection, we consider the deployment of the technology if such information is available. However, we may not rely on deployment information as primary decision factor, because of the incompleteness of the available information. Therefore, we also consider different properties of the solutions. We are fully aware that these decisions are disputable, but we contend that they are still better than putting the solutions in a random order for examination. (We consider that any formally defined deterministic orders, such as alphabetical order or chronological order are similarly useless.)

We note that class 3 is a subcategory of *important* and all the technologies in it are definitely to be covered by the security analysis. If a solution is *well-defined* but the communication scenario is not considered relevant, we classify the security analysis of the IPv6 technology as *optional*. The optional classification means the lowest priority (class 4). We also put those *fairly defined* solutions for a relevant scenario into this category, which have significant known deployment.

Technologies in class 4 are withheld for later decision whether there are good enough reasons to deal with their security analysis.

We do not deal with the security analysis of *obsolete* technologies. Neither with those that do not meet the requirements of class 4.

Table 1 summarizes the priority classes.

For a more fine-grained classification, we will also use the secondary term *aging*, to express that the communication scenario is expected to be no more common in the near future.

As for the implementations, we usually consider them for class 1 or class 2 technologies. Within the category of the free software implementations, we give further priority to those, which are used widespread and/or are known to be stable and have high performance (if such information is available).

3.4 Deployment of IPv6 Transition Solutions

Unfortunately, we have found that the publicly available information about the deployment of the different IPv6 transition technologies is very much deficient. We have found only two major sources that attempted to give a world-wide picture of the proportions of the deployments of the different IPv6 transition technologies. One of the sources is a survey of Jordi Palet Martinez started in 2016 [22]. Slide 17 of his APNIC 44 presentation (September 2017) contains data about the deployment of the different IPv6 transition mechanisms. We have summarized it in Table 2. Unfortunately, its representativeness is rather questionable for several reasons:

- The sample size was too small.
- Some answers were given by ISP employees and some others by customers (their proportion is not stated).
- According to slide 6, the distribution of the answers did not follow the population of the different countries, e.g. there were 231 answers from Brazil and only 62 from China.
- The controllable deployment results seems to contradict to Google statistics. (The survey reported 3% deployment for 6to4, whereas no more than 0.05% of the traffic was 6to4 or Teredo between January 1, 2016 and December 31, 2017 according to Google IPv6 statistics [23].)

Lee Howards has shared a Google Docs spreadsheet on

Table 2 Deployment of the different IPv6 transition mechanisms according to the Global IPv6 survey of Jordi Palet Martinez [22]

IPv6 transition mechanism	Number of cases	Proportion
464XLAT	4	1%
6rd	11	3%
6to4	13	3%
CGN (Carrier Grade NAT)	33	8%
Dual stack	282	71%
DS-Lite	13	3%
Light-weight 4over6	1	0%
MAP-T	1	0%
MAP-E	5	1%
NAT64	4	1%
Other	17	4%
Softwires (L2TP)	2	1%
Tunnel broker	11	3%

the v6ops IETF mailing list [24], which contains the IPv6 transition technology usage of several ISPs. His comments include:

"Our impression was that of the 26+ transition mechanisms defined, only a few have any modern relevance (editorial comments are mine, not consensus positions):

6rd It may be that its light is waning, with early deployments moving to native IPv6, and no new deployments.

DS-Lite Widely deployed, existing support among home gateway manufacturers.

NAT64/464XLAT Implies NAT64, SIIT, which may be used elsewhere. Handset CLATs. No home gateway CLAT yet.

MAP-T Announced trials and lots of buzz, but no large-scale deployments, no home gateway support yet.

MAP-E Some buzz, no announced trials or deployments, no home gateway support yet.

Native dual-stack Still the gold standard, but doesn't solve IPv4 address shortage."

As the spreadsheet may be updated at any time, we use its snapshot version included in a later e-mail on the same mailing list [25]. We have counted the number of occurrences of the different IPv6 transition technologies and present our results in Table 3. (Some ISPs use two technologies, in these cases we counted both.) Of course, this survey is also surely not representative.

We note that the data in the Google Docs spreadsheet were collected in relation of an IETF v6ops working group Internet Draft [26], which lists the following technologies in the following order (to be supported by customer edge routers): 464XLAT, Dual Stack Lite (DS-Lite), Lightweight 4over6 (lw4o6), MAP-E, MAP-T. All of them are so-called *IPv4-as-a-Service* (IPv4aaS) technologies aiming to provide customers with IPv4 connectivity, while ISPs use IPv6-only access and core networks.

Although dual stack dominates in both tables, we expect that its high share cannot be sustained because of the exhaustion of the public IPv4 address pool.

Table 3 Deployment of the different IPv6 transition mechanisms according to the table appeared at IETF v6ops mailing list [25]

IPv6 transition mechanism	Number of occurrences	Proportion
464XLAT	9	15%
6rd	6	10%
Dual stack	26	44%
DS-Lite	11	19%
IPv4 only	1	2%
Light-weight 4over6	1	2%
MAP-T	2	3%
MAP-E	1	2%
NAT64	2	3%

4. Survey of IPv6 Transition Technologies

We give a comprehensive survey of all known IPv6 transition technologies, presenting their purpose and the basics of their operation. We classify them, and if they are considered as class 1 or class 2, we address their implementations, too.

As there are a high number of IPv6 transition technologies, we follow the categories presented in [15], namely *dual* stack, *single translation*, *double translation* and *encapsulation*. However, we do not deal with the category of dual stack, because it means that both IPv4 and IPv6 can be used. Of course, it also means that vulnerabilities of both the IPv4 and the IPv6 protocol stack may be exploited by the attackers, however, usually no specific "IPv6 transition technology" is used. The only aiding tool we can mention is the so-called "Happy-eyeballs" solution [27], which aims to help the dual-stack clients to choose the IP version that ensures better user experience.

We note that we cover an IPv6 transition technology in Section 4.1.9, which was designed for dual stack hosts, but we address it among the single translation technologies, because we believe that it belongs to there on the basis of how it works.

We give an exhaustive survey for all the other three categories, that is, single translation, double translation and encapsulation solutions. For each category, we summarize our findings in a table.

4.1 Single Translation Type Solutions and DNS64

The aim of these transition mechanisms is to enable a client, which can use only IPvX, to communicate with a server, which can use only IPvY, where $X, Y \in \{4, 6\}$ and $X \neq Y$. They translate the IP data packets arriving from the client from IPvX to IPvY, and also do the reverse translation from IPvY to IPvX for the packets arriving from the server. Although DNS64 [16] does not belong to them, we discuss it among them together with stateful NAT64 [17] because these two technologies are used together.

4.1.1 DNS64 and Stateful NAT64

Both DNS64 [16] and stateful NAT64 [17] are *standard* solutions, and can be used together for enabling IPv6-only clients to communicate with IPv4-only servers. This communication scenario is expressly *relevant* because the ISPs cannot distribute public IPv4 addresses to their high number of new customers due the depletion of the global IPv4 address pool, and we consider it a commendable practice if they go ahead and deploy IPv6 instead of using NAT444 (also called Carrier Grade NAT [28] or Large Scale NAT) or any other solutions, which would keep their customers in the IPv4 world and thus make the transition period longer. However, still there are, and there will be servers, which can use only IPv4. Thus, we consider the analysis of DNS64 and NAT64 *important* and also *essential*, because no other standard IPv6 transition technology exists for this scenario since NAT-PT was moved to historic status [29].

We note that alternatively, the IPv4aaS solutions may be used, which are discussed in Section 4.4.

Now, we summarize the operation of DNS64 and NAT64 in a nutshell.

The DNS64 server acts as a proxy: when it receives a request for an IPv6 address (AAAA record) for a given domain name, it asks the normal DNS system about it. If the DNS64 server receives a valid answer, then it simply returns the answer. If it does not receive a valid answer, then it asks the normal DNS system about the IPv4 address (A record) of the given domain name. The DNS64 server uses the received IPv4 address to synthesize a so-called *IPv4-embedded IPv6 address* [30], which contains the IPv4 address at a well-defined position. Finally, the DNS64 server returns the resulted IPv6 address (or an error message, if it had not received an IPv4 address).

When a stateful NAT64 gateway receives an IPv6 packet, which belongs to a new communication session, the NAT64 gateway constructs an IPv4 packet with the destination IPv4 address taken from the appropriate position of the destination IPv6 address and with its own public IPv4 address as source address, and it registers the new session into its connection tracking table. When it receives a reply packet, it identifies the communication session, which the IPv4 packet belongs to and constructs an IPv6 packet. It is an important restriction of stateful NAT64 that a communication session may be initiated only from the IPv6 side.

Most client-server applications can work well with the DNS64 + NAT64 solution, for more information see [31].

There are three major free software DNS64 implementations exist: BIND [32], PowerDNS [33] and Unbound [34].

In [6], we have given a detailed security analysis of DNS64. Now, we mention only two important threats: DNS cache poisoning and DoS (Denial of Service) attack. As for DNS cache poisoning, we have shown in [35] that all three before mentioned DNS64 servers implemented the three most important countermeasures against DNS cache poisoning defined in RFC 5452 [36]. As for DoS attacks, we have also pointed out that high performance can be a kind of mitigation against DoS attacks [6]. We have examined the performance of the three above mentioned DNS64 implementations according to the methodology defined in RFC 8219 [37] (and detailed in [38]) using the `dns64perf++` [39] free software tool. We have found that Unbound has the highest single core performance, PowerDNS scales up the best with the number of CPU cores and BIND has the lowest DNS64 performance [40].

As for free software stateful NAT64 implementations, we have experience with PF (Packet Filter) [41] of OpenBSD [42], which supports NAT64 since version 5.1, and the combination of the stateless TAYGA [43] and the Netfilter [44] of Linux (also called iptables after name of its user interface tool). We have examined and compared their stability and performance [45]. Ecdysis [46] and Jool [47] are two other free software stateful NAT64 implementations, which we did not test yet. We plan to compare the performance of these four NAT64 implementations, before selecting some of them for detailed security analysis, however, presently we do not have a stateful NAT64 benchmarking tool, which complies with the relevant RFC [37]. There was an RFC 8219 compliant benchmarking tool reported, but it implemented only the stateless NAT64 tests [48].

4.1.2 NAT-PT/NAPT-PT

Basic NAT-PT and NAPT-PT were defined in a standard track RFC 2766 [49] in 2000. These rather complex solutions addressed bidirectional translation between the IPv4 realm and the IPv6 realm, but they were moved to historic status for several reasons in 2007 [29], therefore we do not deal with them.

4.1.3 SIIT

The stateless IP/ICMP translation algorithm can be used to translate between the IPv4 and the IPv6 headers (including ICMP headers) in both directions. Although, its previously defining standard track RFCs (RFC 2765, RFC 6145) have been obsoleted, it is considered as *standard* (defined by a proposed standard state RFC) [50]. Being stateless, it cannot be used as a solution for the IPv4 address shortage problem, but we still consider it *relevant*, because it can be used as a building element of more complex technologies, thus we classify its security analysis as *important*. As SIIT is the only standard solution for this scenario, we select it into class 1. Please see Section 5.1.3 for the justification of this decision.

As for free software implementations, the above mentioned TAYGA and Jool implement SIIT, too. Another example is map646 [50], which was designed as a stateless NAT46 gateway solution for the WIDE project [51]. However, it implements only one half of SIIT, as it provides only an access for IPv4-only clients to IPv6-only servers, that is stateless NAT46, but it does not support stateless NAT64.

4.1.4 SIIT-DC

The *well-defined* SIIT-DC [53] is an application of SIIT in IPv6 data centers (DC). Its goal is the enable DC operators to use IPv6-only servers, while their system is also available for IPv4-only clients. Despite of its limited area of application,

Table 4 Priority classification of IPv6 transition technologies: single translation technologies (including DNS64 and DNS46)

Technology	Scenario	Operation basics	Class
DNS64, RFC 6147	IPv6 client and IPv4 server	IPv4-embedded IPv6 address synthesis	1
Stateful NAT64, RFC 6146	IPv6 client and IPv4 server	stateful network address and port translation	1
NAT-PT/NAPT-PT, RFC 2766 *Obsoleted by RFC 4966*	IPvX client and IPvY server	translation between IPvX and IPvY ($X, Y \in \{4, 6\}$ and $X \neq Y$) including ALGs for DNS and FTP	-
SIIT, RFC 7915	IPvX client and IPvY server	stateless IP/ICMP translation between IPv4 and IPv6 in both directions (including ICMP headers)	1
SIIT-DC, RFC 7755	IPv4 client and IPv6 server	application of SIIT for Data Centers	-
IVI, RFC 6219	IPvX client and IPvY server	similar to the standard SIIT	4
SA46T-AT, exp. I-D [55]	IPv6 client and IPv4 server	too complex solution to support private IPv4 addresses	-
TRT, RFC 3142	IPv6 client and IPv4 server	rather a concept, later realized as stateful NAT64	-
DNS46, exp. I-D [57]	IPv4 client and IPv6 server	dynamic mapping between IPv4 and IPv6 addresses	-
NAT46, exp. I-D [57]	IPv4 client and IPv6 server	network address an protocol translation from IPv4 to IPv6 using the dynamic mapping created by DNS46	-
BIH, RFC 6535	IPv4 application running on a dual stack host and IPv6 server	intercepts with DNS (synthesizes fake IPv4 address) and SIIT (may be implemented either socket API layer or network layer)	4
BIS, RFC 2767 *Obsoleted by RFC 6535*	IPv4 client application running on a dual stack host and IPv6 server	intercepts with DNS (synthesizes fake IPv4 address) and SIIT (implemented at the network layer)	-
BIA, RFC 3338 *Obsoleted by RFC 6535*	IPv4 client application running on a dual stack host and IPv6 server	intercepts with DNS (synthesizes fake IPv4 address) and SIIT (implemented at the socket API layer)	-

we consider SIIT-DC useful and its security analysis could be classified as *important*, but we do not deal with it separately, as SIIT has already been addressed before.

4.1.5 IVI

IVI [54] (the name is the contraction of the Roman numbers IV and VI) is a *well-defined* stateless translation solution between IPv4 and IPv6, where the translation may be initiated from both directions. As the *standard* SIIT can used for the same purpose and we do have any deployment information of IVI, we classify the security analysis of IVI as *optional*.

4.1.6 SA46T-AT

The aim of the *fairly defined* SA46T-AT [55] was to enable an IPv6-only host to access to an IPv4-only host. The scenario is similar to that of DNS64+NAT64, but this technology would have been worked also with private IPv4 addresses, which added significant complexity to the solution. Its Internet Draft expired and it did not became an RFC. We do not deal with it.

4.1.7 TRT

TRT [56] is an old *well-defined* solution (or rather concept) aimed to enable IPv6-only hosts to exchange TCP or UDP traffic with IPv4-only hosts. The concept was good and it was later realized as stateful NAT64. We do not deal with it.

4.1.8 DNS46 + NAT46

Although it is not typical now, later on it may be a realistic scenario that some old IPv4-only clients will need help in accessing IPv6-only servers. The *fairly defined* DNS46 + NAT46 [57] solution addresses this problem. Unfortunately, the logic of the DNS64 + NAT64 solution can not be followed, because IPv6 addresses cannot be embedded into IPv4 address. Therefore, dynamic mappings are created between some elements of the IPv4 address range and of the IPv6 address range, which implies that the DNS46 server and the NAT46 gateway have to use a common database. Regrettably, the Internet Draft has never became an RFC, thus we are waiting for a well-defined solution, as we do not know any other workable solutions for this scenario. We are aware that some implementations exist, but as we do not know of any deployment, thus we do not deal with it.

4.1.9 BIH (BIS, BIA)

The *standard* BIH (Bump-in-the-Host) [58] aims to enable IPv4-only client applications running on dual stack hosts to connect to IPv6-only servers.

Unlike the DNS46 + NAT46 solution, BIH is executed on the same host where the IPv4-only application is running. BIH intercepts with DNS requests, and if no usable “A” record is returned by the DNS system, then BIH asks for a “AAAA” record, and if it receives one, then BIH fakes an “A” record and returns it to the IPv4 application and also stores the mapping of the received IPv6 address and faked IPv4 address. BIH also intercepts with the network traffic sent towards the faked IPv4 addresses and performs SIIT to reach the corresponding IPv6 server.

RFC defining BIH obsoleted the BIS (Bump-in-the-Stack) [59] and BIA (Bump-in-the-API) [60] solutions.

Whereas BIH could have been a usable solution for enabling IPv4-only client applications to interoperate with IPv6 servers, we do not know of its deployment, therefore, even though we consider its scenario *relevant*, we classify the security analysis of BIH as *optional*.

4.2 Double Translation Type Solutions

The aim of these transition mechanisms is to carry IPv4

Table 5 Priority classification of IPv6 transition technologies: double translation technologies

Technology	Scenario	Operation basics	Class
464XLAT, RFC 6877	support for IPv4 applications in an IPv6-only network	CLAT: stateless translation from IPv4 to IPv6 and proxy for DNS; PLAT: stateless NAT64	2
MAP-T, RFC 7599	support for IPv4 applications in an IPv6-only network	very complex solution with multiple translations, see Section 4.2.2	3
4rd, RFC 7600	supports public IPv4 for users over an IPv6-only network	very complex solution with multiple translations, see Section 4.2.3	4
dIVI, expired I-D [70]	supports several scenarios	namely "Dual stateless IPv4/IPv6 translation", practically similar to MAP-T	-

packets through IPv6 networks. They translate the IPv4 data packets to IPv6 data packets when they enter into the IPv6 network, and back to IPv4 when they leave the IPv6 network.

We note that double translation can not be used for carrying IPv6 packets through IPv4 networks, because the longer IPv6 addresses cannot be stored in the places of IPv4 addresses.

4.2.1 464XLAT

464XLAT [61] is a *well-defined* solution, which allows clients on IPv6-only networks to access IPv4-only Internet services, such as Skype or Spotify.[†] It can be a legitimate decision of the ISPs that they use only IPv6 in their network, because of both the higher operational costs and more security vulnerabilities of a dual stack network. However, they need to satisfy their users' demand for the operability of their legacy IPv4-only applications. (We note that this scenario is called *IPv4aaS*, and it is addressed by several other solutions, too. We discuss this scenario further in Section 4.4.) Thus, the scenario is definitely relevant, therefore, we classify the security analysis of 464XLAT as *important* but *replaceable* (as other solutions also exist) and we give the basics of its operation.

464XLAT performs two translations. The CLAT device operates on the client side: it translates the IPv4 packets of the IPv4-only client software to IPv6 and also performs the translation of the reply packets in the other direction. (It actually performs SIIT.) The PLAT device operates at the ISP side, and it actually performs stateful NAT64.

We note that CLAT acts as a router for IPv6 traffic. Thus, 464XLAT can be used together with DNS64+NAT64 as follows: IPv6 capable clients receive IPv6 addresses, and they can reach IPv6 servers natively, whereas they can reach IPv4-only servers using DNS64+NAT64. Only the traffic of the legacy IPv4-only clients or applications undergoes the double translation. As for DNS traffic, CLAT acts as a DNS proxy.

As for the deployment of 464XLAT, the first very significant step was reported in 2014, since then T-Mobile USA uses only IPv6 in its network and provides IPv4 access by using 464XLAT [62]. A strategic whitepaper from 2015 [63] states: "For MNOs, transitioning their networks to IPv6 using 464XLAT offers several advantages. IPv6-only networks are simpler to deploy, operate, and manage, which reduces OPEX. The 464XLAT also delivers reductions in CAPEX because it benefits from the increasing ratio of IPv6-to-IPv4 Internet traffic, lowering CAPEX for CG-NAT. And, for the end customer, the offered service is never compromised." In 2017, a RIPE 74 presentation [64] recommended 464XLAT deployment for residential networks, too. 464XLAT has significant deployment according to Table 3, and it is listed in the first place in the afore-mentioned IETF v6ops working group Internet Draft specifying requirements for customer edge routers [26]. Because of its relevant current and expected further future deployment, we selected 464XLAT into class 2. As for CLAT implementations, clatd [65] exists for Linux, and there is an implementation for Android, but we have no experience with them. Unfortunately, different CLATs will have to be tested for each mobile platform (e.g. Android, iPhone, Windows Phone, etc.) because the CLAT runs on the user's device.

4.2.2 MAP-T

MAP-T [66] is a *standard* solution for the IPv4aaS problem. The operation of the solution is rather complex, we give only some highlights. First, the MAP-T CE (Customer Edge) device performs a NAT44 operation to restrict the available TCP/UDP port numbers for the user.[††] Then the CE device performs a special stateless translation from IPv4 to IPv6, where the source IPv4 address and the selected port bits are encoded into the source IPv6 address according to the MAP-T rules. The IPv6 packets can be destined to other users, where similar CEs perform the necessary transformations, or to the outside IPv4 Internet, in which case the MAP-T Border Relay performs the necessary transformations.

The scenario is deliberately *relevant*, thus the security analysis of MAP-T is classified *important*, but also *replaceable*, because other solutions exist. Because of the complexity of the solution and also its low deployment, we prioritize other solutions and classify MAP-T as class 3. In addition to the security considerations provided in Section 13 of RFC 7599 [66], we would like to mention one more thing: being

[†] For a list of IPv4-only applications, please refer to slide 10 of [64].

[††] At this point, we must mention that we have serious doubts with this design. A proper upper bound for the port number need of a user may be much higher than the average. Thus, the statistical multiplexing of stateful NAT64 could be more advantageous. For the consequences of the port number shortage situation, see [67], and for the port number requirements of web browsing see [68] and its references.

a complex solution, there are a lot of room for security holes in the implementations.

4.2.3 4rd

4rd [69] is a *well-defined* stateless solution to provide residual IPv4 deployment to the users over IPv6 networks. Depending on the actual scarcity of the public IPv4 addresses, it is possible with 4rd to share a single public IPv4 address among multiple customers in a way that the users get only a limited port set, or to assign one or even more than one public IPv4 addresses to a customer. The solution is similar to MAP-T in the sense that it also uses multiple translations and port restriction. Its distinguishing features are "that TCP/UDP IPv4 packets are valid TCP/UDP IPv6 packets during domain traversal and that IPv4 fragmentation rules are fully preserved end to end" [69]. Although the scenario is *relevant*, but as there are much more well-known and significantly deployed IPv4aaS solutions exist, and as 4rd is rather complex and we do not know of any deployment, we consider its analysis as *optional*.

4.2.4 dIVI

The scenario of the *fairly defined* dIVI [70] is the same as that of the standard MAP-T, and it also uses similar solution of encoding the port range into the IPv6 address. As it is defined by an expired Internet Draft and we do not know of any deployment, we do not deal with it.

4.3 Encapsulation Type Solutions

These solutions carry the packets of either IP version encapsulated into the packets of the other IP version. The tunnel may be explicitly created or automatic.

4.3.1 6in4

The aim of the *standard* 6in4 [71] solution is to carry IPv6 packets using IPv4 networks. (The idea behind is to connect the IPv6 "islands" using the IPv4 Internet, until the IPv6 infrastructure is completely built.) It is done by using static tunnels. Between the endpoints of the tunnels, the IPv6 packets are encapsulated in IPv4 packets using the 41 protocol identifier for IPv6 in the IPv4 header. Due to the slow deployment of the IPv6 protocol, the scenario is still common and sometimes unavoidable, thus we classify the security analysis of 6in4 as important. Although different other tunneling technologies exist, 6in4 is so widely used (also as a component of other technologies) that we select it for security analysis in class 2.

As for implementations, all major network operating systems support it. E.g. a 6in4 tunnel endpoint may be statically configured under Linux by using the `ip` command and the `sit` tunnel interface. (Note: SIT stands for Simple Internet Transition.)

4.3.2 4in6

The *standard* 4in6 solution is a tunnel, which carries IPv4 datagrams over IPv6 networks. It can be said that is was defined in [72], though this RFC defines a general encapsulation scheme, where packets of various protocols can be encapsulated into IPv6, e.g. IPv4, IPX, etc.

As it is widely used (also as a building block of other technologies) its security analysis is *important*, but *replaceable*, because there are alternative solutions, e.g. double translation may be used instead, and we do not select it into class 2.

4.3.3 6to4

The *standard* 6to4 [73] solution aims (or aimed) to enable IPv6 sites, which have only IPv4 Internet connection, to communicate with other IPv6 sites being in the same situation or with native IPv6 hosts. The only prerequisite is that the sites must have a public IPv4 address. The solution provides globally routable IPv6 addresses for the IPv6 sites using the 2002::/16 prefix and the public IPv4 address. The sites are made available through the node that has the public IPv4 address, functioning as a 6to4 router. The IPv6 packets are carried as encapsulated into IPv4 packets (using 6in4) between two 6to4 routers, or between a 6to4 router and a 6to4 relay, if the other party is a native IPv6 node. The solution has a very important advantage over using configured tunnels that here the tunnels are created automatically and no action form the site's administrator is needed. However, several problems with 6to4 were reported. Some of them are documented in [74] together with their possible mitigation. Some other problems could not be solved [75] and finally, the anycast prefix for 6to4 was deprecated [76], which means that 6to4 can not be used for accessing the native IPv6 internet from host having only an IPv4 connection.

As there are still many parts of the world, where the ISPs do not provide IPv6 Internet access to the customers, 6to4 is still in use and can be the most convenient way of easily getting IPv6 Internet access, thus we could formally classify its analysis as *important* but *replaceable* and *aging*. However, as the before mentioned Google IPv6 statistics [23] reported negligible 6to4 traffic for the last two years, we rather classify the security analysis of 6to4 as *optional*. Of course, 6to4 is *replaceable* by explicit tunnels (from tunnel brokers).

4.3.4 Teredo

The *standard* Teredo [77] can be used instead of 6to4, if no public IPv4 address is available for the site. It was designed to be a last resort if no others solutions available. Similarly to 6to4, we could formally classify the security analysis of Teredo as *important* but *replaceable* (tunnel brokers) and *aging*, however, we also classify Teredo as *optional* for the same reason.

4.3.5 6rd

The aim of the *standard* 6rd [78] is to provide an easy and fast method for ISPs to provide IPv6 Internet access for its customers using the IPv4 infrastructure of the ISP. The solution operates similarly to 6to4 with the important difference that it does not use the 2002::/16 prefix, but rather the own IPv6 prefix of the ISP and it eliminates all the operational and QoS issues, which arose from the broken reverse path relays in the case of 6to4 [75].

Although we admit that 6rd is currently in use by several ISPs and formally classify it as *important* and *replaceable*, but we recommend the use of native IPv6 for new deployments and considering also the comment of Lee Howards on the v6ops IETF mailing list about the lack of new deployments of 6rd [24], we do not select it for analysis in class 2.

4.3.6 6to4-PMT

The aim of the now *obsolete* 6to4-PMT [79] was that ISPs may provide a better quality 6to4 IPv6 Internet access for their customers without investing into native IPv6 or even 6rd. Being obsoleted by RFC 7526 [76], we do not deal with it.

4.3.7 ISATAP

The *well-defined* ISATAP [80] aims to connect dual stack nodes over IPv4 networks. By not having any information of its deployment, we classify its security analysis as *optional*.

4.3.8 6PE

The aim of the *standard* 6PE [81] is to connect IPv6 islands over IPv4 MPLS routers. The 6PE abbreviation denotes the IPv6 Provider Edge routers, “which are dual stack in order to connect to IPv6 islands and to the MPLS core, which is only required to run IPv4 MPLS” [81]. Similarly to 6rd, this solution provides an easy way for an ISP to implement IPv6. We could classify it in the same way into class 3, but considering native IPv6 deployment a better way and not having any information of its deployment, we rather classify its security analysis as *optional*.

4.3.9 6VPE

The aim of the *standard* 6VPE [82] is to provide IPv6 VPN over IPv4 MPLS routers. This method extends the BGP/MPLS IP VPN solution method to support IPv6. Similarly to 6PE, we classify it into class 4.

4.3.10 MAP-E

The *standard* MAP-E [83] aims to address the same scenario as MAP-T, that is, IPv4aaS, and the two solutions are also similar, but MAP-E uses encapsulation and decapsulation instead of double translation.

Similarly to MAP-T, we classify the security analysis of MAP-E *important*, but *replaceable*, and prefer other solutions.

4.3.11 DS-Lite

The *standard* DS-Lite [84] aims to address the same scenario as 464XLAT, that is, IPv4aaS, and the solution is somewhat similar to 464XLAT, but DS-Lite use encapsulation and decapsulation and then CGN (carrier grade NAT) for the IPv4 traffic. It carries the IPv6 traffic of the user unmodified, and its CPE also provides a DNS proxy for the IPv4 applications as the CPE of 464XLAT does. We classify the security analysis of DS-Lite *important*, but *replaceable*, and prefer other solutions due to the problems described in [85].

4.3.12 Public 4over6

The *well-defined* public 4over6 [86] aims to provide IPv4 Internet connectivity over native IPv6 network using global IPv4 addresses. The defining informational RFC [86] recommends lightweight 4over6 for new deployments, thus we mention this solution only for completeness, and we do not deal with it.

4.3.13 Lightweight 4over6

The *standard* lightweight 4over6 [87] addresses the same scenario as DS-Lite, that is, IPv4aaS, and the solution itself is an extension of DS-Lite. We classify its security analysis *important* but *replaceable*, and prefer other solutions.

4.3.14 SA46T

The *fairly defined* SA46T (Stateless Automatic IPv4 over IPv6 Encapsulation/Decapsulation Technology) [88] is another technology aims to provide a way to carry IPv4 packets over the single-stack IPv6 backbone of ISPs. Its Internet Draft expired and was not published as an RFC (and we do not have any deployment information), thus, we do not deal with it.

4.3.15 Tunnel Broker

The aim of the *well-defined* tunnel broker [89] is to provide end users with IPv6 internet access over IPv4 infrastructure. This is done by managed tunnels using the earlier mentioned 6in4 encapsulation. In this sense, tunnel broker is not another protocol, but rather an architecture for tunnel management.

Please note that in this paper, we use “tunnel broker” for IPv6 tunnels over IPv4 as defined in RFC 3053 [89], but it may be used in a wider sense, including also IPv4 tunnel over IPv6.

As tunnel broker does not define a new protocol, and 6in4 was already selected, no more work is needed. (We

Table 6 Priority classification of IPv6 transition technologies: encapsulation technologies

Technology	Scenario	Operation basics	Class
6in4, RFC 4213	IPv6 packet over IPv4 network	preconfigured tunnel: IPv6 packets are encapsulated into IPv4 packets	2
4in6, RFC 2473	IPv4 packet over IPv6 network	preconfigured tunnel: IPv4 packets are encapsulated into IPv6 packets	3
6to4, RFC 3056	IPv6 capable hosts in IPv4 environment	an “automatic” tunnel, which provides also IPv6 addresses, requires public IPv4 address for the sites	4
Teredo, RFC 4380	IPv6 capable hosts in IPv4 environment	an “automatic” tunnel, which provides also IPv6 addresses, last resort if site has no IPv4 address	4
6rd, RFC 5969 IPv6	Internet access over IPv4 network infrastructure	IPv6 rapid deployment on IPv4 infrastructures	3
6to4-PMT, RFC 6732 *Obs.'d by RFC 7526*	IPv6 capable hosts in IPv4 environment	improvement of 6to4 by provider support	-
ISATAP, RFC 5214	connect dual stack nodes over IPv4 networks	automatic intra-site tunnel with automatic addressing	4
6PE, RFC 4798	connect IPv6 islands over IPv4 MPLS	IPv6 packets are transmitted over IPv4 MPLS network without the insertion of IPv4 headers	4
6VPE, RFC 4659	provide IPv6 VPN over IPv4 MPLS	different tunneling techniques are supported, see the details in RFC 4659	4
MAP-E, RFC 7597	support for IPv4 applications in an IPv6 only network	very complex solution with encapsulation, see Section 4.3.10	3
Dual-Stack Lite (DS-Lite), RFC 6333	support for IPv4 applications in an IPv6 only network	for IPv4 traffic: encapsulation, decapsulation and CGN	3
Public 4over6, RFC 7040	IPv4 connectivity over IPv6 network (public IPv4 addr.)	IPv4 in IPv6 tunnel, provides bidirectional connectivity	-
Lightweight 4over6 (lw4o6), RFC 7596	support for IPv4 applications in an IPv6 only network	extension of DS-Lite	3
SA46T, exp. I-D [88]	support for IPv4 applications in an IPv6 only network	stateless automatic IPv4 over IPv6 encapsulation / decapsulation	-
Tunnel broker, RFC 3053	provide IPv6 Internet over IPv4 infrastructure	defines an architecture for tunnel management	-
TSP, RFC 5572	set up IPv4 or IPv6 tunnels over IPv4 or IPv6 networks (client may reside behind NAT)	defines tunnel set up protocol for tunnel brokers	4
AYIYA, exp. I-D [91]	carry IPvX packets over IPvY network	can encapsulate any IP version in any IP version, works through NAT	4
Softwire: L2TPv2: RFC 2661 L2TPv3: RFC 3931	original aim: to provide wire emulation may be used as a tunneling technology	defines a Layer 2 tunneling protocol	4
6over4, RFC 2529	isolated IPv6 host in an IPv4 network	carries IPv6 packets over multicast capable IPv4 domains without establishment of explicit tunnel	-
MPT, active I-D [95]	carry IPvX packets over IPvY network aim: provide multipath	provides IPvX tunnel over one or more IPvY paths ($X, Y \in \{4, 6\}$)	-

attribute the market share of tunnel broker in Table 2 to 6in4.)

4.3.16 TSP

The aim of the *well-defined* TSP (Tunnel Setup Protocol) [90] is to enable the establishment any kind of tunnels. Its main application area is definitely the IPv6 tunnel brokers. It also support devices behind NAT.

Whereas using tunnel brokers was an important solution to get IPv6 Internet access in the past, the solution is *aging* now, and some market leader tunnel brokers have already closed their services. Therefore, we consider the security analysis of TSP as *optional*. (We have already selected 6in4 into class 2.)

4.3.17 AYIYA

The *fairly defined* AYIYA (Anything In Anything) [91] makes it possible to use tunnels, which carries any version IP packets in any version IP packets even over several NAT devices. The solution is deployed and used by tunnel brokers, therefore, we put it into class 4, even though the Internet Draft expired long time ago and it never became an RFC.

4.3.18 Softwire (L2TPv2, L2TPv3)

The original aim of the *standard* L2TPv2 (Layer Two Tunneling Protocol) [92] was to provide “wire emulation” over packet switched networks (in order to provide a “generalized PPP”). The standard L2TPv3 [93] extends it in different ways. The L2TP solution evolved for a long time and there is a collection of RFCs describing its different features and extensions. L2TP may also be used as a tunneling technology, and it appears in Table 2 with a small market share. We consider its potential in the field of IPv6 transition technologies as marginal, and therefore we classify its security analysis as *optional*.

4.3.19 6over4

The *standard* 6over4 [94] aimed to carry the IPv6 packets of isolated IPv6 hosts over multicast capable IPv4 domains without the establishment of an explicit tunnel. Its security analysis could formally be classified as *optional*, however, it has not been deployed, and therefore, we mention it only for completeness, and we do not deal with it.

4.3.20 MPT

The not well-defined MPT (Multi-Path Technology) [95] is a novel network layer multipath communication technology, which can be used as a tunnel solution, because it supports both IPv4 or IPv6 tunnels over single or multiple IPv4 or IPv6 paths. MPT has one implementation [96], which has been successfully applied for different tasks, e.g. path throughput capacity aggregation ref97, fast connection recovery [98] or elimination of the stalling events on YouTube video playback ref99. Its performance can compete with the well-known MPTCP as shown in [100] and [101], but MPT has not been standardized yet. As MPT is yet immature, thus we do not deal with its security analysis.

4.4 Short Discussion of IPv4aaS Solutions

Theoretically, all five IPv4aaS technologies could be used together with DNS64 + stateful NAT64, thus offloading not only the native IPv6 traffic but also the traffic between an IPv6-only client and an IPv4-only server and thus using the actual IPv4aaS solution only for the traffic of IPv4-only applications (and in the case of IPv4 only literals). Its benefit is that with DNS64+NAT64, the vast majority of the traffic undergoes only a single stateful translation (instead of double translation or encapsulation and decapsulation), whereas its cost is the need for deploying DNS64 and Stateful NAT64. However, in actual deployments, this optimization is used only with 464XLAT, where the additional cost is only the deployment of DNS64, as the PLAT of 464XLAT is actually a stateful NAT64 gateway.

We note that this is a philosophical question, how we call this solution. On the one hand, calling it as DNS64+NAT64 with 464XLAT expresses that the lion's share of the work is done by DNS64+NAT64 and only a small proportion of the traffic needs 464XLAT. On the other hand, it is worded in RFC 6877 [61] that 464XLAT is used together with DNS64, as 464XLAT contains the stateful NAT64 gateway as PLAT, thus DNS64 is the only extra feature.

All five technologies have their specific advantages and disadvantages, and depending on different conditions, any of them may be the most suitable solution for a specific application scenario. For an in depth analysis of the pros and cons of the five most prominent IPv4aaS technologies, please refer to [102].

5. Discussion

5.1 Scenarios Revisited

For the description of the scenarios, we often used the terms and approach of the writers of the RFCs or Internet Drafts. Now, we follow a different approach. Our intent is to generalize the description of the scenarios and to address only those that are relevant to the users, ISPs or content providers and omit those, which are relevant to the Internet technology developers only. To achieve this goal, let us consider the interests of these parties.

5.1.1 Users

An average user is not interested in the IP version, rather wants to reach content and/or use network applications. The user faces with a problem if the content is not available using his/her IP version, or some of the required network applications cannot be used with his/her IP version.

Of course, an average user does not want to deal with any of the IPv6 transition technologies, rather expects a workable solution from his/her ISP.

5.1.2 Internet Service Providers

As for the ISPs, they are faced with a technical challenge: the depletion of the public IPv4 address pool. Those that are not trying to conserve the situation by using CGN (Carrier Grade NAT), but rather go ahead and deploy IPv6, encounter two problems:

1. a large portion of the contents is distributed by IPv4-only servers
2. some of the network applications are not IPv6 capable.

These problem situations match the following previously mentioned scenarios:

Scenario 1: IPv6 client and IPv4 server

Scenario 2: support for IPv4 applications in an IPv6-only ISP network

We contend that these scenarios are very much relevant, and they must be covered by IPv6 transition technologies, the security issues of which will be analyzed. Let us consider the revers ones, too:

Scenario 3: IPv4 client and IPv6 server

Scenario 4: support for IPv6 applications in an IPv4-only ISP network

Scenario 3 can be handled in various ways without using any of the IPv6 transition technologies listed in Section 4.

1. As for the possible new servers of the content providers, their number is orders of magnitude less than that of the new clients, thus they can still get IPv4 addresses.
2. As for the client, why is it not IPv6 capable?
    a. If the problem is a lack of hardware or operating system support, it will be solved soon by the replacement of the device. As for mobile devices, their life cycle is only a few years. The life cycle of desktop and notebook computers is longer, but they are also replaced within 4-8 years due to the end of support of their operating systems (it is especially true for Windows) and all major operating systems support IPv6 for several years.

b. If the problem is the IPv6 incompatibility of the applications, then we are actually talking about scenario 2.
c. If the problem is that the ISP does not support IPv6, then it is in fact scenario 4.

Let us consider scenario 4. Is it a real problem from the viewpoint of the users? Except for some advanced users, who insist on using IPv6 for some reasons, the vast majority of the users is not interested in the version of IP, and we do not know any widely used network applications, which are available for IPv6 only. Therefore, we prioritize scenarios 1 and 2 over scenarios 3 and 4.

5.1.3 Content Providers

Content providers may operate at different levels. Basic level content providers perhaps ask public IPv4 addresses from their ISPs and put the burden of IP version compatibility on the shoulders of the ISPs. This is covered by scenario 1. If they are IPv6 aware, they probably use dual stack.

Advanced content providers may operate server farms, which may contain high number of computers. They very likely provide dual stack access for their users, but they may want to reduce administrative work by using their internal infrastructure as single stack. Thus, for accessing the service dual stack they need stateless translation between clients with and IP version different than that of the server farm. We call this scenario as:

Scenario 5: one-to-one mapping between IPvX and IPvY.

And SIIT is a perfect solution for this scenario. (In fact, this application was an important motivation in the selection of SIIT into class 1 in Section 4.1.3.)

5.2 Sufficiency and Parsimony of the Selected Solutions

Sufficiency means that we must have at least one class 1 or class 2 candidate(s) for each relevant scenario. Parsimony means that the number of selected solutions per relevant scenario should not be significantly higher than one.

As for scenario 1, we have recommended DNS64 and NAT64 as class 1 candidates. As both of them are necessary, the requirement of parsimony is also satisfied.

As for scenario 2, we recommended 464XLAT as class 2 candidate, and we have no more class 1 or class 2 candidates for the scenario.

Thus, we have successfully covered both prioritized scenarios.

As for scenario 3, only the combination of DNS46 and NAT46 is a real match, but we did not select them for security analysis, because they were defined by an expired Internet Draft [57], and we do not know of significant deployment of their existing implementations. We note that SIIT can not be used for this scenario without appropriate DNS support, and SIIT-DC was intended for something else. Having no other solutions, this scenario remains uncovered.

We note that BIH is not a solution for scenario 3, because BIH requires the host executing the client application to have IPv6 enabled.

As for scenario 4, we have recommended 6in4 as a single class 2 solution.

As for scenario 5, we have covered it by SIIT, which was selected into class 1.

Thus, the selected low number of class 1 or class 2 solutions could cover all those scenarios that we found important for the users, ISPs and content providers.

We note that we were striving to find a possibly minimal set of technologies to cover the most important scenarios. However, market may prefer other solutions, therefore we would like to emphasize that we consider the security analysis of all class 3 solutions *important*.

6. Conclusions

We have developed a priority classification method for the ranking of different IPv6 transition technologies and their most important implementations by defining four priority classes so that the vulnerabilities of the most crucial technologies may be examined first. We have conducted a comprehensive and up-to-date survey of the available IPv6 transition technologies. For each technology, we have supplied a pointer to its defining document (RFC or Internet Draft), a brief description of its application scenario, the basics of its operation and some deployment information if it was available. We have also determined the importance of their security analysis according to our ranking system. For class 1 and class 2 technologies, we have provided information about their most important free software implementations. We have revisited the importance of the scenarios from the viewpoints of the users, ISPs, and content providers and we have shown that each important scenario was covered with IPv6 transition technologies selected into the first two priority classes, whereas we also complied with the requirement of parsimony.

Acknowledgment

This work was supported by the International Exchange Program of the National Institute of Information and Communications Technology (NICT), Japan.

References

[1] S. Deering and R. Hinden, "Internet Protocol, version 6 (IPv6) specification", IETF RFC 2460, Dec. 1998.

[2] S. Deering and R. Hinden, "Internet Protocol, version 6 (IPv6) specification", IETF RFC 8200, Jul. 2017.

[3] M. Nikkhah and R. Guérin, "Migrating the internet to IPv6: An exploration of the when and why", IEEE/ACM Transactions on Networking, vol. 24, no. 4, pp. 2291–2304, Aug. 2016. DOI: 10.1109/TNET.2015.2453338

[4] M. Georgescu, H. Hazeyama, Y. Kadobayashi and S. Yamaguchi, "Empirical analysis of IPv6 transition technologies using the IPv6 Network Evaluation Testbed", EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, vol. 15, no. 2, e1, DOI:

10.4108/inis.2.2.e1

[5] G. Lencse, Y. Kadobayashi, "Survey of IPv6 transition technologies for security analysis", IEICE Technical Committee on Internet Architecture (IA) Workshop, Tokyo Japan, Aug. 28, 2017, IEICE Tech. Rep. vol. 117, no. 187, pp. 19–24.

[6] G. Lencse, Y. Kadobayashi, "Methodology for the identification of potential security issues of different IPv6 transition technologies: Threat analysis of DNS64 and stateful NAT64", Computers & Security, vol. 77. no. 1, pp. 397–411, Aug. 2018. DOI: 10.1016/j.cose.2018.04.012

[7] X. Wang, J. Wu, Y Cui, "Survey of Internet IPv6 Transition Technologies", Journal of Chinese Computer Systems, vol. 2006, no. 3, [Online]. Available: http://en.cnki.com.cn/Article_en/CJFDTOTAL-XXWX200603000.htm

[8] X. Zhang, Y. Li, "Survey of IPv6 Transition Mechanisms and Security Review", Computer Knowledge and Technology, vol. 2010, no. 19. [Online]. Available: http://en.cnki.com.cn/Article_en/CJFDTOTAL-DNZS201019041.htm

[9] M. Li, J-H. Yang, H. Wang, "Survey and analysis on IPv6 transition technologies", Journal of Guangxi University(Natural Science Edition), vol. 2011. no. S1, [Online]. Available: http://en.cnki.com.cn/Article_en/CJFDTOTAL-GXKZ2011S1041.htm

[10] S. Kalwar, N. Bohra, A. A. Memon, "A survey of transition mechanisms from IPv4 to IPv6 — Simulated test bed and analysis", In Proc. 2015 Third International Conference on Digital Information, Networking, and Wireless Communications (DINWC), Moscow, Russia, Feb. 3-5, 2015, pp. 30–34, DOI: 10.1109/DINWC.2015.7054212

[11] A. Albkerat, B. Issac, "Analysis of IPv6 transition technologies", International Journal of Computer Networks & Communications, vol. 6, no.5, pp. 19–38, Sep. 2014, DOI: 10.5121/ijcnc.2014.6502

[12] P. Wu, Y. Cui, J. Wu, J. Liu, C. Metz, "Transition from IPv4 to IPv6: A state-of-the-art survey", IEEE Communications Surveys & Tutorials, vol. 15. no 3. pp. 1407–1424, 2013, DOI: 10.1109/SURV.2012.110112.00200

[13] N. Škoberne, O. Maennel, I. Phillips, R. Bush, J. Zorz, and M. Ciglaric, "IPv4 address sharing mechanism classification and tradeoff analysis", IEEE/ACM Transactions on Networking, vol. 22, no. 2, pp. 391–404, Apr. 2014, DOI: 10.1109/SURV.2012.110112.00200

[14] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Threat modeling: uncover security design flaws using the STRIDE approach" MSDN Magazine, Nov. 2006, pp. 68–75.

[15] M. Georgescu, H. Hazeyama, T. Okuda, Y. Kadobayashi, and S. Yamaguchi, "The STRIDE towards IPv6: A comprehensive threat model for IPv6 transition technologies", Proc. 2nd International Conference on Information Systems Security and Privacy, Rome, Feb. 2016. DOI: 10.13140/RG.2.1.2781.6085

[16] M. Bagnulo, A Sullivan, P. Matthews and I. Beijnum, "DNS64: DNS extensions for network address translation from IPv6 clients to IPv4 servers", RFC 6147, Apr. 2011.

[17] M. Bagnulo, P. Matthews and I. Beijnum, "Stateful NAT64: Network address and protocol translation from IPv6 clients to IPv4 servers", IETF RFC 6146, Apr. 2011.

[18] Free Software Foundation, "The free software definition", [Online]. Available: http://www.gnu.org/philosophy/free-sw.en.html

[19] Open Source Initiative, "The open source definition", [Online]. Available: http://opensource.org/docs/osd

[20] Cisco, "End user license agreement", [Online]. Available: http://www.cisco.com/c/en/us/products/end-user-license-agreement.html

[21] Juniper Networks, "End user license agreement", [Online]. Available: http://www.juniper.net/support/eula/

[22] J. Palet, "Global IPv6 deployment survey", APNIC 44, Taichung, Taiwan, Sept. 2017, (presentation slides), [Online]. Available: https://www.slideshare.net/apnic/global-ipv6-deployment-survey

[23] Google, "IPv6 Statistics", [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html

[24] Lee Howard, "[v6ops] discussion of transition technologies", IETF v6ops mailing list, Jan. 19, 2018. [Online]. Available: https://www.ietf.org/mail-archive/web/v6ops/current/msg28889.html

[25] Lee Howard, "[v6ops] Transition mechanisms in use", IETF v6ops mailing list, Mar. 22, 2018. [Online]. Available: https://www.ietf.org/mail-archive/web/v6ops/current/msg29096.html

[26] J. Palet Martinez, H. M.-H. Liu, M. Kawashima, "Requirements for IPv6 customer edge routers to support IPv4 connectivity as-a-service", active Internet Draft, https://tools.ietf.org/html/draft-ietf-v6ops-transition-ipv4aas-14

[27] D. Schinazi, T. Pauly, "Happy eyeballs version 2: Better connectivity using concurrency", IETF RFC 8305, Dec. 2017.

[28] S. Perreault (Ed), I. Yamagata, S. Miyakawa, A. Nakagawa, H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", IETF RFC 6888, Apr. 2013.

[29] C. Aoun and E. Davies, "Reasons to move the Network Address Translator - Protocol Translator (NAT-PT) to historic status", IETF RFC 4966, Jul. 2007.

[30] C. Bao, C. Huitema, M. Bagnulo, M Boucadair and X. Li, "IPv6 addressing of IPv4/IPv6 translators", IETF RFC 6052, Oct. 2010.

[31] S. Répás, T. Hajas and G. Lencse, "Application compatibility of the NAT64 IPv6 transition technology", in Proc. 37th International Conference on Telecommunications and Signal Processing (TSP 2014), Berlin, Germany, Jul. 1-3, 2014, pp. 49–55. DOI: 10.1109/TSP.2015.7296383

[32] Internet Systems Consortium, "BIND: Versatile, classic, complete name server software", [Online]. Available: https://www.isc.org/downloads/bind

[33] Powerdns.com BV, "PowerDNS", [Online]. Available: http://www.powerdns.com

[34] NLnet Labs, Unbound, [Online]. Available: http://unbound.net

[35] G. Lencse and Y. Kadobayashi, "Methodology for DNS cache poisoning vulnerability analysis of DNS64 implementations", Infocommunications Journal, vol. 10, no. 2. pp. 13–25, Jun. 2018.

[36] A. Hubert, R. van Mook, "Measures for making DNS more resilient against forged answers", IETF RFC 5452, Jan. 2009.

[37] M. Georgescu, L. Pislaru, and G. Lencse, "Benchmarking methodology for IPv6 transition technologies", IETF RFC 8219, Aug. 2017.

[38] G. Lencse, M. Georgescu, and Y. Kadobayashi, "Benchmarking methodology for DNS64 servers", Computer Communications, vol. 109, no. 1, pp. 162–175, Sept. 1, 2017, DOI: 10.1016/j.comcom.2017.06.004

[39] G. Lencse and D. Bakai, "Design and implementation of a test program for benchmarking DNS64 servers", IEICE Transactions on Communications, vol. E100-B, no. 6. pp. 948–954, Jun. 2017. DOI: 10.1587/transcom.2016EBN0007

[40] G. Lencse and Y. Kadobayashi, "Benchmarking DNS64 implementations: Theory and practice", Computer Communications, vol. 127, no. 1, pp. 61–74, September 1, 2018, DOI: 10.1016/j.comcom.2018.05.005

[41] 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

[42] Theo de Raadt, The OpenBSD 5.1 Release, May 1, 2012, ISBN 978-0-9784475-9-5, [Online]. Available: http://www.openbsd.org/51.html

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

[44] H. Welte, P. N. Ayuso, "The netfilter.org project", [Online]. Available: http://www.netfilter.org/

[45] G. Lencse and S. Répás, "Performance analysis and comparison of the TAYGA and of the PF NAT64 Implementations", in Proc. 36th International Conference on Telecommunications and Signal Processing (TSP 2013), Rome, Italy, Jul. 2013. pp. 71–76. DOI: 10.1109/TSP.2013.6613894

[46] S. Perreault, J.-P. Dionne, and M. Blanchet, “Ecdysis: Open-source DNS64 and NAT64”, in Proc. AsiaBSDCon 2010, Tokyo, Japan, March 11-14, 2010. pp. 53–59 [Online]. Available: https://2010.asiabsdcon.org/papers/abc2010-P4B-paper.pdf

[47] Network Information Center Mexico, “Jool: SIIT & NAT64”, [Online]. Available: http://jool.mx/en/index.html

[48] P. Bálint, “Test software design and implementation for benchmarking of stateless IPv4/IPv6 translation implementations”, in Proc. 40th International Conference on Telecommunications and Signal Processing (TSP 2017), Barcelona, Spain, Jul. 5-7, 2017, pp. 74–78.

[49] G. Tsirtsis and P. Srisuresh, “Network Address Translation - Protocol Translation (NAT-PT)”, IETF RFC 2766, Feb. 2000.

[50] C. Bao, X. Li, F. Baker, T. Anderson, F. Gont, “IP/ICMP translation algorithm”, IETF RFC 7915, Jun. 2016.

[51] K. Shima, “map646 source code”, [Online]. Available: https://github.com/keiichishima/map646

[52] K. Shima, W. Ishida, Y. Sekiya, “Designing an IPv6-oriented datacenter with IPv4-IPv6 translation technology for future datacenter operation”, In: I.I. Ivanov, M. van Sinderen, F. Leymann, T. Shan (eds) Cloud Computing and Services Science. (CLOSER 2012). Porto, Portugal, Apr. 2012. pp. 39–53, Communications in Computer and Information Science, vol 367. Springer, DOI: 10.1007/978-3-319-04519-1_3

[53] T. Anderson, “SIIT-DC: Stateless IP/ICMP translation for IPv6 data center environments”, IETF RFC 7755, Feb. 2016.

[54] X. Li, C. Bao, M. Chen, H. Zhang, J. Wu, “The China Education and Research Network (CERNET) IVI translation design and deployment for the IPv4/IPv6 coexistence and transition”, IETF RFC 6219, May 2011.

[55] N. Matsuhira, K. Horiba, Y Ueno, O. Nakamura, “SA46T address translator”, expired Internet Draft, https://tools.ietf.org/html/draft-matsuhira-sa46t-at-06

[56] J. Hagino, K. Yamamoto, “An IPv6-to-IPv4 transport relay translator”, IETF RFC 3142, Jun. 2001.

[57] D. Liu, H. Deng, “NAT46 consideration”, expired Internet Draft, https://tools.ietf.org/html/draft-liu-behave-nat46-02

[58] B. Huang, H. Deng, T. Savolainen, “Dual-Stack Hosts Using "Bump-in-the-Host" (BIH)”, IETF RFC 6535, Feb. 2012.

[59] K. Tsuchiya, H. Higuchi, Y. Atarashi, “Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)”, IETF RFC 2767, Feb. 2000.

[60] S. Lee, M-K. Shin, Y-J. Kim, E. Nordmark, A. Durand, “Dual Stack Hosts Using "Bump-in-the-API" (BIA)”, IETF RFC 3338, Oct. 2002.

[61] M. Mawatari, M. Kawashima, C. Byrne, “464XLAT: Combination of stateful and stateless translation”, IETF RFC 6877, Apr. 2013.

[62] A. McConachie, “Case study: T-Mobile US goes IPv6-only using 464XLAT”, Internet Society Case Studies, [Online]. Available: http://www.internetsociety.org/deploy360/resources/case-study-t-mobile-us-goes-ipv6-only-using-464xlat/

[63] Alcatel – Lucent, “464XLAT in mobile networks: IPv6 migration strategies for mobile networks”, strategic whitepaper, MKT201501979EN, Apr. 2015. [Online]. Available: https://www.apnic.net/wp-content/uploads/2017/01/IPv6_Migra tion_Strategies_for_Mobile_Networks_Whitepaper.pdf

[64] J. Palet, “Using 464XLAT in residential networks”, RIPE 74, Budapest, Hungary, May 8-12, 2017, slides of presentation, [Online]. Available: https://ripe74.ripe.net/presentations/151-ripe-74-ipv6-464xlat-residential-v2.pdf

[65] T. Anderson, “Clatd - a CLAT / SIIT-DC edge relay implementation for Linux”, [Online]. Available:
https://github.com/toreanderson/clatd

[66] 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, Jul. 2015.

[67] S. Miyakawa, “IPv4 to IPv6 transformation schemes”, IEICE Transactions on Communications, vol. E93-B, no. 5, pp. 1078–1084, May 2010. DOI:10.1587/transcom.E93.B.1078

[68] G. Lencse, “Estimation of the port number consumption of web browsing”, IEICE Transactions on Communications, vol. E98-B, no. 8. pp. 1580–1588, Aug. 2015 DOI: 10.1587/transcom.E98.B.1580

[69] R. Despres, S. Jiang (ed), R. Penno, Y. Lee, G. Chen, M. Chen, “IPv4 residual deployment via IPv6 - A stateless solution (4rd)”, IETF RFC 7600, Jul. 2015.

[70] C. Bao, X. Li, Y. Zhai, W. Shang, “dIVI: Dual-stateless IPv4/IPv6 translation”, expired Internet Draft, https://tools.ietf.org/html/draft-xli-behave-divi-07

[71] E. Nordmark, R. Gilligan, “Basic transition mechanisms for IPv6 hosts and routers”, IETF RFC 4213, Oct. 2005.

[72] A. Conta, S. Deering, “Generic packet tunneling in IPv6 specification”, IETF RFC 2473, Dec. 1998.

[73] B. Carpenter, K. Moore, “Connection of IPv6 domains via IPv4 clouds”, IETF RFC 3056, Feb. 2001.

[74] B. Carpenter, “Advisory guidelines for 6to4 deployment”, IETF RFC 6343, Aug. 2011.

[75] E. Aben, “6to4 – How bad is it really?”, RIPE NCC, [Online]. Available: https://labs.ripe.net/Members/emileaben/6to4-how-bad-is-it-really

[76] O. Troan, B. Carpenter (Ed.), “Deprecating the anycast prefix for 6to4 relay routers”, IETF RFC 7526, May 2015.

[77] C. Huitema, “Teredo: Tunneling IPv6 over UDP through network address translations (NATs)”, IETF RFC 4380, Feb. 2006.

[78] W. Townsley, O. Troan, “IPv6 rapid deployment on IPv4 infrastructures (6rd) – Protocol specification”, IETF RFC 5969, Aug. 2010.

[79] V. Kuarsingh (Ed.), Y. Lee, O. Vautrin, “6to4 provider managed tunnels”, IETF RFC 6732, Sep. 2012.

[80] F. Templin, T. Gleeson, D. Thaler, “Intra-site automatic tunnel addressing protocol (ISATAP)”, IETF RFC 5214, Mar. 2008.

[81] J. De Clercq, D. Ooms, S. Prevost, F. Le Faucheur, “Connecting IPv6 islands over IPv4 MPLS using IPv6 provider edge routers (6PE)”, IETF RFC 4798, Feb. 2007.

[82] J. De Clercq, D. Ooms, M. Carugi, F. Le Faucheur, “BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN”, IETF RFC 4659, Sep. 2006.

[83] O. Troan (ed), W. Dec, X. Li, C. Bao, S. Matsushima, T. Murakami, T. Taylor (ed), “Mapping of address and port with encapsulation (MAP-E)”, IETF RFC 7597, Jul. 2015.

[84] A. Durand, R. Droms, J. Woodyatt, Y. Lee, “Dual-stack lite broadband deployments following IPv4 exhaustion”, IETF RFC 6333, Aug. 2011.

[85] Y. Lee, R. Maglione, C. Williams, C. Jacquenet, M. Boucadair, “Deployment considerations for dual-stack lite”, IETF RFC 6908, Mar. 2013.

[86] Y. Cui, J. Wu, P. Wu, O. Vautrin, Y. Lee, “Public IPv4-over-IPv6 access network”, IETF RFC 7040, Nov. 2013.

[87] Y. Cui, Q. Sun, M. Boucadair, T. Tsou, Y. Lee, I. Farrer, “Lightweight 4over6: An extension to the dual-stack lite architecture”, IETF RFC 7596, Jul. 2015.

[88] N. Matsuhira, “Stateless automatic IPv4 over IPv6 encapsulation / decapsulation technology: Specification”, expired Internet Draft, https://tools.ietf.org/html/draft-matsuhira-sa46t-spec-11

[89] A. Durand, P. Fasano, I. Guardini, D. Lento, “IPv6 tunnel broker”, IETF RFC 3053, Jan. 2001.

[90] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, B. Palter, “Layer Two Tunneling Protocol "L2TP"”, IETF RFC 2661, Aug. 1999.

[91] J. Massar, “AYIYA: Anything in anything”, expired Internet Draft, https://tools.ietf.org/html/draft-massar-v6ops-ayiya-02

[92] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, B. Palter, “Layer Two Tunneling Protocol "L2TP"”, IETF RFC 2661, Aug. 1999.

[93] J. Lau (ed), M. Townsley (ed), I. Goyret (ed), “Layer Two Tunneling Protocol - Version 3 (L2TPv3)”, IETF RFC 5641, Mar. 2006.

[94] B. Carpenter, C. Jung, “Transmission of IPv6 over IPv4 domains without explicit tunnels”, IETF RFC 2529, Mar. 1999.

[95] G. Lencse, Sz. Szilágyi, F. Fejes, M. Georgescu, “MPT network layer multipath library”, active Internet Draft, Dec. 2018, https://tools.ietf.org/html/draft-lencse-tsvwg-mpt-03

[96] B. Almási, G. Lencse, Sz. Szilágyi, “Investigating the multipath extension of the GRE in UDP technology”, Computer Communications, vol. 103, no. 1, pp. 29–38, May 1, 2017, DOI: 10.1016/j.comcom.2017.02.002

[97] Á. Kovács, “Comparing the aggregation capability of the MPT communications library and multipath TCP”, in: Proc. 7th IEEE International Conference on Cognitive InfoCommunications (CogInfoCom 2016), Wroclaw, Poland, Oct. 16-18, 2016, pp. 157–162, DOI: 10.1109/CogInfoCom.2016.7804542

[98] F. Fejes, R. Katona, and L. Püsök, “Multipath strategies and solutions in multihomed mobile environments”, in: Proc. 7th IEEE International Conference on Cognitive InfoCommunications (CogInfoCom 2016), Wroclaw, Poland, Oct. 16-18, 2016, pp. 79–84, DOI: 10.1109/CogInfoCom.2016.7804529

[99] F. Fejes, S. Rácz, and G. Szabó, “Application agnostic QoE triggered multipath switching for Android devices”, In: Proc. 2017 IEEE International Conference on Communications (ICC 2017), Paris, France, May 21-25, 2017, pp. 1585–1591. DOI: 10.1109/ICC.2017.7997450

[100] Sz. Szilágyi, F. Fejes, R. Katona, “Throughput performance comparison of MPT-GRE and MPTCP in the Fast Ethernet IPv4/IPv6 environment”, Journal of Telecommunications and Information Technology, vol. 2018. no. 2. pp. 53–59. DOI: 10.26636/jtit.2018.122817

[101] Á. Kovács, “Evaluation of the aggregation capability of the MPT communications library and Multipath TCP”, Acta Polytechnica Hungarica, to be published.

[102] G. Lencse, J. Palet Martinez, L. Howard, R. Patterson, I. Farrer, “Pros and cons of IPv6 transition technologies for IPv4aaS”, active Internet Draft, Jan. 2019, https://tools.ietf.org/html/draft-lmhp-v6ops-transition-comparison-02

Gábor Lencse received his M.Sc. and Ph.D. degrees in computer science from the Budapest University of Technology and Economics, Budapest, Hungary in 1994 and 2001, respectively. He has been working for the Department of Telecommunications, Széchenyi István University, Győr, Hungary since 1997. Now, he is an Associate Professor. He has been working part time for the Department of Networked Systems and Services, Budapest University of Technology and Economics as a Senior Research Fellow since 2005. He was a Guest Researcher at the Laboratory for Cyber Resilience, Nara Institute of Science and Technology, Japan, from June 15 to December 15, 2017, where his research area was the security analysis of IPv6 transition technologies.

Youki Kadobayashi received his Ph.D. degree in computer science from Osaka University, Japan, in 1997. He is currently a Professor in the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. Since 2013, he has also been working as the Rapporteur of ITU-T Q.4/17 for cybersecurity standardization. His research interests include cybersecurity, web security, and distributed systems. Dr. Kadobayashi is a member of IEEE Communications society.