Acta Technica Jaurinensis

Vol. 13, No. 2, pp. 85-98, 2020
DOI: 10.14513/actatechjaur.v13.n2.530

Towards the Security Analysis of the Five Most Prominent IPv4aaS Technologies

A. Al-Azzawi*

Department of Networked Systems and Services, Budapest University of Technology and Economics, Magyar tudósok körútja 2, 1117 Budapest, Hungary
***E-mail: alazzawi@hit.bme.hu**

Abstract: This paper surveys the five most important technologies for IPv4aaS (IPv4-as-a-Service), namely 464XLAT, DS-Lite (Dual-Stack Lite), lw4o6 (Lightweight 4over6), MAP-E and MAP-T. The aim of our effort is to identify the potential security issues within these technologies. We plan to perform their security analysis following the STRIDE approach, which stands for spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege. We give a short introduction for the method. Within the five IPv4aaS technologies, we focus on 464XLAT, its architecture and operation. We construct a DFD diagram suitable for its security analysis according to the STRIDE methodology, thus making the first steps towards finding its potential vulnerabilities and seeking for their mitigations.

Keywords: *IPv6 transition technologies; DNS64; NAT64; security analysis; STRIDE; 464XLAT*

1. Introduction

This paper is an extended version of our former conference paper [1].

After the wide spread of the Internet usage all over the globe, IPv4 public addresses started to be liable to depletion[2], which actually happened in 2011. The next version of the Internet Protocol, IPv6 was defined in a draft standard RFC 2460 in 1998 [3], while the actual deployment of it took way longer than it should have, because of the technical issue it faced with the old equipment and its compatibility with the new proposed IPv6. A lot of technologies were proposed to overcome the

depletion of IPv4 and to start the transition process to IPv6, such as dual-stack, Tunnelling and Natting. The transition operation itself is expected to be on the run for three decades [4]. The high number of technologies intended to facilitate the transition from IPv4 to IPv6 have different security vulnerabilities and they are sensitive to different kinds of attacks [5].

A very informative book by Adam Shostack [6] presents us the STRIDE methodology, which stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege, those are the terms that can be applied on each possible potential security threat and can be analysed accordingly.

So, this paper's goal is to tackle some network security concerns with the idea designing a systematic method to identify the possible security issues of the five covered IPv4aaS IPv6 transition technologies using the STRIDE approach with understating their data flow diagrams.

The remainder of this paper is organized as follows. Section 2 is an overview of the IPv6 transition technologies. Section 3 is a short summary of the STRIDE method. Section 4 is the application of SRIDE to the 464XLAT transition technology. Section 5 is a short summary of our plans for future research. Section 6 concludes our paper.

2. Transition Technologies

There were a high number of IPv6 transition technologies defined, which can be applied for different communication scenarios. For example, [7] mentions 40 of them. RFC 8219 [8] has classified the high number of IPv6 transition technologies into a small number of categories: dual-stack, single translation, double translation and encapsulation technologies. We follow these categories, but not address all the before mentioned 40 technologies, only the most important ones.

2.1. Dual-stack

According to [9], the main idea of dual-stack is to have a certain node where both IPv4 & IPv6 are usable. If one would like to move forward with IPv6, then a priority may be given to IPv6.

The issue with dual stack is the time delay that comes with it. For instance, as explained in Fig. 1 below:

1. Dual-stack host wants to communicate with the remote server, so it sends a request to the local DNS server asking for all available addresses for the remote server.

2. DNS server replies with both A & AAAA records.
3. The host will choose one of the records. However, the host will normally choose AAAA record (IPv6 address) as its configured that way by default for communication, then a TCP session will be established with the remote sever.

There is an issue, when the remote server has an AAAA record (IPv6), but IPv6 communication is not successful with it, which means that the client has to repeat the communication cycle with IPv4 this time, which results in time delay and communication latency.

In RFC 8305 [10], a technology called "Happy Eyeballs" is presented to choose the better IP version for the packet delivery. This solution improves the user experience by staring the two sessions concurrently and thus decreasing the worst-case delay.

However, dual-stack does not handle the shortage of IPv4 addresses and it is not a cost-effective solution, as it requires the maintenance of both protocol stacks.

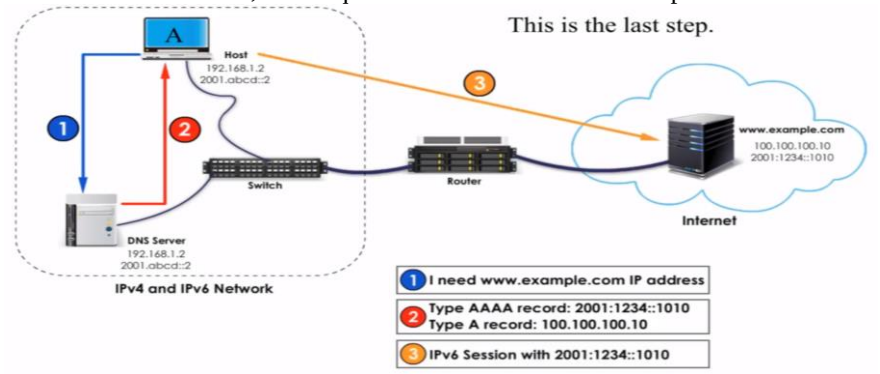

Figure 1. Dual-Stack Communication Process [11]

2.2. Single Translation Type Technologies

There are several single translation technologies, who caught the attention of many researchers in the field.

In [7], there is a brief classification for some of those technologies as their priority goes such as DNS64, NAT64, SIIT, etc.

2.2.1. DNS64 + NAT64

In [12], the process of NAT64 + DNS64 was carefully explained in detail. As shown in Fig. 2, the process consists of 10 steps. It starts with the IPv6 only client sending a request (AAAA query) to DNS64 asking for an address of IPv4 only server and ends up with the IPv6 only client contacting the IPv4 only server and receives an acknowledgment as well.

The solution is good enough but not perfect as another distinctive drawback emerged on the surface. Some IPv4 only applications don't work with this solution like Skype, Netflix, etc. For a detailed examination of the compatibility of the most important applications with the NAT64+DNS64 technology, please refer to [13].

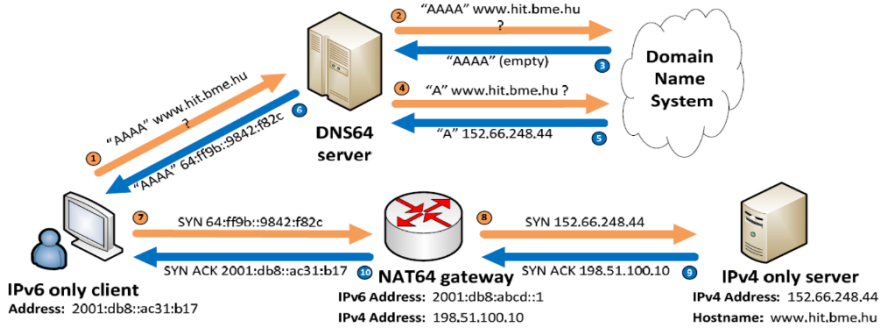

Figure 2. DNS64+NAT64 Process, IPv6 Only Client contacting IPv4 Only Server [12]

2.2.2. SIIT

SIIT is the well-known stateless IP/ICMP Packet header translator as defined by RFC 7915. The idea of SIIT being stateless does put some restrictions on it, while facing with public IPv4 pool depletion [1]. However, it proved to be a sufficient solution coupling with other technologies such as 464XLAT [14].

2.2.3. SIIT-DC

This technology focuses on Data centres and its servers' compatibility with IPv6 in terms of serving IPv4 only clients [15].

2.3. Double Translation Type Technologies

This type of translation involves translating an IPv4 packet, when it reaches boundaries of an IPv6 network into IPv6 packet and then translates it back to IPv4,

when the packet leaves IPv6 network and gets back to an IPv4 network. There is one limitation though, and it is that an IPv6 packet cannot be carried in an IPv4 packet because an IPv6 address cannot be stored in the place of an IPv4 address [7]. There are several Double-Translation methods such as 464XLAT, MAP-T, 4rd, dIVI, etc. In this paper, we are focusing on some of them due to their importance.

2.3.1. 464XLAT

To go around the problem that NAT64 +DNS64 couldn't solve by their own, a new solution emerged, and it's called 464XLAT [16], the technology was adopted by an American company called T-Mobile. This new technology allows IPv6 only services to run on IPv4 only devices and applications.

RFC 6877 [16] described 464XLAT as single/dual translation technology. 464XLAT allows ISPs to use only IPv6 in their infrastructure.

In fact, the double translation process isn't being conducted on high scale, only a small percentage of the network traffic is actually covered by double translation process. So, 464XLAT is not a pure double translation technique, it's rather a single translation with a possibility to be used as a double translation mechanism.

According to [16], CLAT device always placed at the client edge, which performs stateless NAT46 and it normally translates the packets of IPv4 only network into IPv6. It also translates back the returned IPv6 packets to IPv4 so it can be readable within IPv4 only network applications. So, CLAT performs SIIT (Stateless IP/IPCMP Translation algorithm).

Meanwhile, in the ISP side, there is another device called PLAT which performs Stateful NAT64.

In conclusion, IPv6-only clients do get IPv6 addresses and they can reach IPv6 servers while they need DNS64+NAT64 to reach IPv4-only servers without the need for double translation process. However, IPv4-only application need CLAT double translation process to perform the full 464XLAT operation.

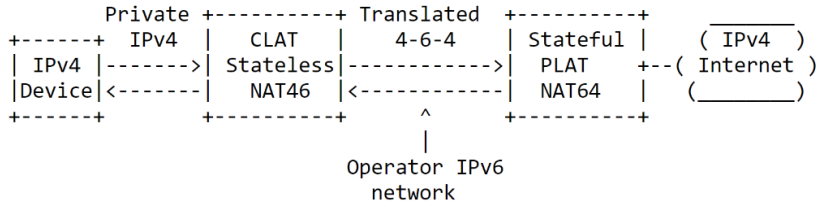

Figure 3. Overview Architecture of 464XLAT [14]

According to [5], no previous testing process was conducted on CLAT on its different platforms because CLAT run on the customer edge and it could be anything (like iPhone, Android, etc.).

Fig. 3 presents the overview architecture of 464XLAT, it shows the chain of communication and translation steps that being conducted in order to ensure that IPv6 is being used on every platform no matter what even within IPv4-only applications.

464XLAT comes with different scenarios in terms of communication source and destination as shown in Table 1.

Table 1. 464XLAT Traffic Treatment Scenarios [16]

Host	***Server***	***Traffic Treatment***	***Location of Translation***
IPv6	IPv6	End-to-End IPv6	None
IPv6	IPv4	Stateful Translation	PLAT
IPv4	IPv4	464XLAT	CLAT +PLAT

2.3.2. MAP-T

MAP-T [17] was adopted by IETF RFC 7599 and its high-level operation is similar to that of 464XLAT, but MAP-T is much more complicated. It uses different set of terminology and equipment such as:

- CE (Customer-Edge): according to [17], MAP-T performs NAT44 in order to maintain a low users port number of TCP /UDP, then it conducts a stateless translation from IPv4 to IPv6 encoding IPv4 source address & port number into the IPv6 source address. The process has two scenarios, the first one is that the packet will be destined to another user within the IPv4 network, where another CE process will occur, but if the packet is heading toward a destination which is outside the IPv4 network, then BR will come in handy as explained below.
- BR (Border-Relay): performs the transformation, where a packet has been translated by CE and wants to go out from the IPv6 network towards the IPv4 Internet.

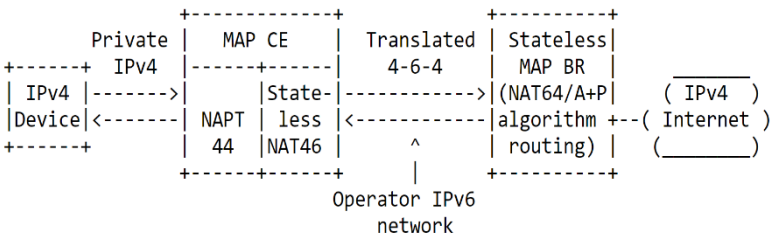

Figure 4. Overview Architecture of MAP-T [14]

MAP-T Security Consideration

According to [7], MAP-T is very complex solution, and being complex makes it liable to many security holes, besides firewalls will have an issue interpreting IPv6 traffic containing packets translated from IPv4 packets. Potential threats are "Spoofing", "Denial of Service" and "Routing Loop" attacks.

2.4. Encapsulation Type Technologies

2.4.1. The Simplest Encapsulation-based Solutions

Due to the unfinished infrastructure of IPv6, 6in4 tunnelling was used to carry IPv6 address through IPv4 tunnel to connect IPv6 islands. It normally uses static preconfigured tunnels between two ports [9]. This solution is not used widespread by the average Internet users, but can serve as a building block for more complex solutions.

Similarly, 4in6 encapsulation defined by RFC 2473 [18] is also not a wide spread solution for the average Internet users, but it serves as an essential building block for the following three solutions.

2.4.2. MAP-E

As explained in [14], MAP-E uses stateless algorithm in order to embed part of IPv4 address into the IPv6 prefix, which allows huge number of clients to be provisioned using a single MAP rule. At CE, the router runs a stateful NAPT44 operation in order to translate IPv4 private source address and source port into an address and a port range. Moreover, at the CE router begins the encapsulation of IPv4 packet inside IPv6 packet and send it to another host within the MAP domain or to BR if the destined packet to area not covered by MAP rules. The BR will start the decapsulation procedure from his side (decapsulating the IPv4 packet from the IPv6 carrying it).

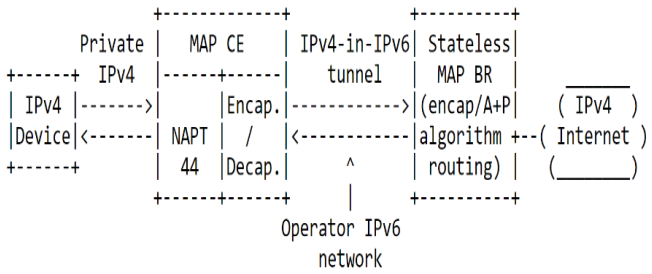

Figure 5. Overview Architecture of MAP-E [14]

MAP-E Security Consideration

Due to similarity between MAP-T & MAP-E, [7] also classified MAP-E as important, but replaceable and preferred other solutions (transition technologies).

2.4.3. Dual-Stack Lite

IETF RFC 6333 defined DS-Lite, the transition process consists of the following steps:

1. At the CE, DS-Lite uses “Basic Broadband Bridging” B4 to encapsulate IPv4 into IPv6 packet, and then send it through IPv6 network.
2. At the other end, AFTR (Address Family Transition Route) performs the encapsulation / decapsulation process of the 4in6 data.
3. After extracting the IPv4 packet, Stateful NAPT function translates the IPv4 packet with private IPv4 addresses into an IPv4 packet with public IPv4 addresses.

However, RFC 6908, presented some issues in DS-Lite in general.

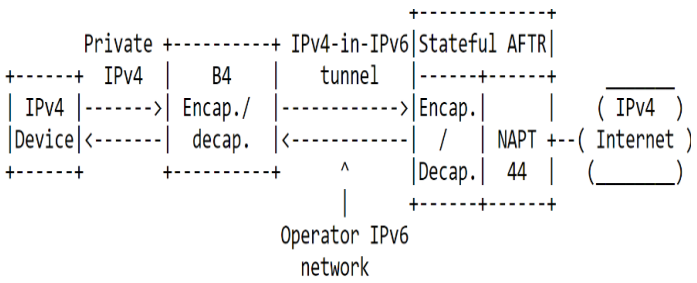

Figure 6. Overview Architecture of DS-Lite [14]

DS-Lite Security Consideration

According to [7], DS-Lite has also been classified as important, but replaceable with other technologies and it preferred to proceed with other solutions in their survey.

2.4.4. Lightweight 4over6

According to [14], lw4o6 is an extension of DS-Lite, but it has some different aspects, where the NAPT function is relocated from AFTR to customer's B4 element which is called "lwB4". The encapsulation process will be as below:

- At lwB4 side, NAPT44 performs private to public IPv4 conversion, then encapsulation of IPv4 packet within IPv6 tunnel occurs.
- lwAFTR builds a "Binding Table" (Customer source address and allocated range of port addresses).
- In this case, lwAFTR conducts A+P (Address + Port number) routing and 4in6 decapsulation process.
- Direct communication between two lwB4s is being done through hair pinning traffic through lwAFTR, RFC 6333.

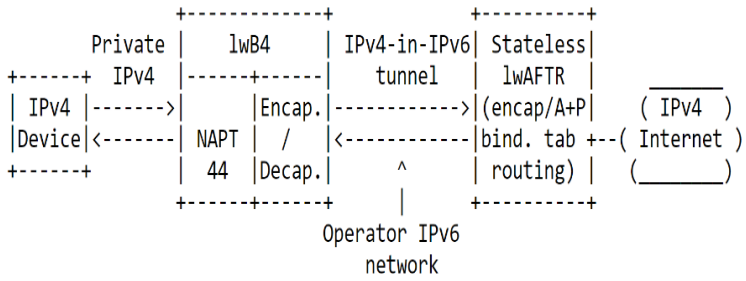

Figure 7. Overview Architecture of Lw4o6 [14]

Lw4o6 Security Consideration

According to [7], Lw4o6 is classified as replaceable and it preferred other solutions.

3. Stride in a Nutshell

Stride stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege and they are general attack types explained below in details:

- Spoofing: in general, it is claiming to be someone that it is not you, or someone pretends to be your website [6]. Concerning computer networks, it often means the usage of the IP address of another device as source IP address to gain unauthorized access to some resources or to hide the real identity of the attacker [5].
- Tampering: the attacker might play and change something in the data flow, while it bounces back and forth between two nodes [6].
- Repudiation: is the claim that a host didn't do something or not responsible for a specific act / behaviour [6].
- Information Disclosure: An attacker getting confidential information that he shouldn't have got it [5], like the TTL value of specific packet within DNS64 server.
- Denial of Service: The attacker floods the targeted network server with huge number of useless requests (queries), which causes preventing the legitimate user from contacting or accessing the designated server [5].
- Elevation of Privilege: incorrectly allowing a user/hacker to access an un-authorized server/service [6].

Table. 2 Vulnerability of different DFD Elements to different Threats [5]

Element	***Spoofing***	***Tampering***	***Repudiation***	***Information Disclosure***	***Denial of Service***	***Elevation of Privilege***
Data Flow		✓		✓	✓	
Data Stores		✓		✓	✓	
Processes	✓	✓	✓	✓	✓	✓
Interactors	✓		✓			

The STRIDE method uses the DFD (Data Flow Diagram) of the examined system for its security analysis. The DFD is build up by four types of elements: Data Flows, Data Stores, Processes and Interactors. Each one of the four elements is susceptible

to some of the mentioned threats but not susceptible to some others. Table. 2 shows the DFD elements and the threats that they are susceptible to, marked with a ✓ sign.

4. Towards the Threat Analysis of 464XLAT

In this chapter, we are focusing on 464XLAT, its security analysis by applying STRIDE theory and analysing its DFD diagram, focusing on possible vulnerable spots and taking in consideration the previous implementation handled by [5].

Fig. 8 shows the DFD for the threat analysis of 464XLAT. By applying the STRIDE method on this DFD, the threats that might face each element within 464XLAT architecture may be discovered.

Our plan to perform STRIDE on DFD in Fig. 8 is to check all possible vulnerabilities (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) at each DFD element (1-14) and prioritize the level of threat at each element then come up with analysis of the most vulnerable element within the DFD and classify the threats based on the number of their possible occurrence in order to tackle them top to bottom.

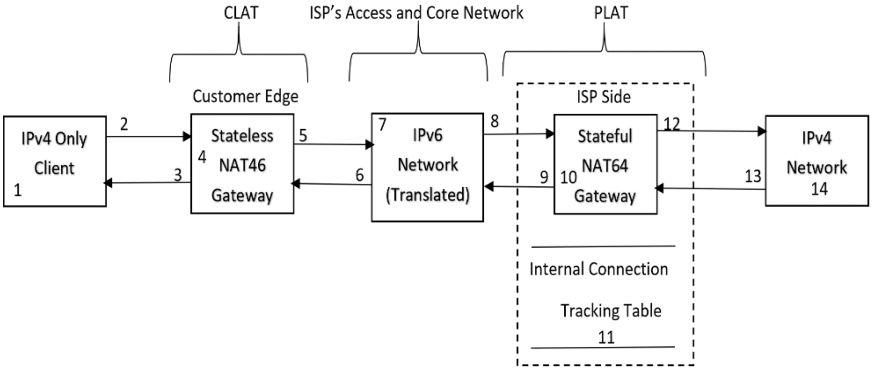

Figure 8. DFD for the threat Analysis of 464XLAT (CLAT + PLAT)

5. Plans for Future Research

Our next step will be the comprehensive security analysis of the 464XLAT technology according to the methodology defined in [5] and for the other 4 technologies as well (DS-Lite, lw4o6, MAP-E and MAP-T). It means that the first phase will be the theoretical analysis of the 464XLAT technology using the DFD shown in Fig. 8, and the second phase will be the practical evaluation of the most

important free software (CLAT and PLAT) implementations, whether they are actually susceptible to the vulnerabilities pointed out in the first phase. Concerning the most important implementations, the results of [19] will be a great help for us.

6. Conclusion

Among all surveyed transition technologies, 464XLAT proved to be a suitable one, while it has some security threats. It solved the problem of IPv4 only applications in an IPv6 environment and the CLAT+PLAT solution is very practical solution for this issue. Further research is required in this area and that includes both the theoretical analysis of the PLAT and CLAT solutions (using the STRIDE methodology), what security vulnerabilities they may have, and also the practical testing of their most important free (open source) software implementations.

Acknowledgement

Thanks to my Supervisor; Dr. Gabor Lencse, Budapest University of Technology and Economics, who helped me all the way through with my research.

References

[1] A. Al-Azzawi, Plans for the security analysis of IPv4aaS technologies, 14th International Symposium on Applied Informatics and Related Areas, University of Óbuda, Székesfehérvár, Hungary (2019) pp. 101–105.

[2] G. Huston, The changing foundation of the internet: Confronting ipv4 address exhaustion, The Internet Protocol Journal 11 (3) (2008) pp. 19–36.

[3] S. Deering, R. Hinden, Internet protocol, version 6 (IPv6) specification, IETF RFC 2460 (1998).
doi: *https://doi.org/10.17487/RFC2460*

[4] M. Nikkhah, R. Guerin, Migrating the Internet to IPv6: An exploration of the when and why, IEEE/ACM Transactions on Networking 24 (4), (2016) pp. 2291-2304.
doi: *https://doi.org/10.1109/TNET.2015.2453338*

[5] 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 77 (1) (2018) pp. 397-411.
doi: https://doi.org/10.1016/j.cose.2018.04.012

[6] A. Shostack, Threat Modeling: Designing for Security, 1st Edition, Wiley, Indaiana, 2014.

[7] Lencse, Y. Kadobayashi, Comprehensive survey of IPv6 transition technologies: A subjective classification for security analysis, IEICE Transactions on Communications E102-B (10) (2019) pp. 2021–2035.
doi: https://doi.org/10.1587/transcom.2018EBR0002

[8] M. Georgescu, L. Pislaru and G. Lencse, Benchmarking methodology for IPv6 transition technologies, IETF RFC 8219 (2017).
doi: https://doi.org/10.17487/RFC8219

[9] E. Nordmark, R. Gilligan, Basic transition mechanisms for IPv6 hosts and routers, IETF RFC 4213 (2005) [cited 2019-11-15].
URL https://tools.ietf.org/html/rfc4213

[10] D. Schinazi, P. Pauly, Happy eyeballs version 2: Better connectivity using concurrency, IETF RFC 8305 (2017) [cited 2019-11-15].
URL https://tools.ietf.org/html/rfc8305

[11] S. Classroom, IPv4 to IPv6 transition – dual stack [cited 2019-11-15].
URL https://www.youtube.com/watch?v=s0TNGC9GP48

[12] G. Lencse, A. G. Soós, Design of a tiny multi-threaded dns64 server, 38th International Conference on Telecommunications and Signal Processing, Prague (2015) pp. 27–32.
doi: https://doi.org/10.1109/TSP.2015.7296218

[13] S. Répás, T. Hajas, G. Lencse, Application compatibility of the NAT64 IPv6 transition technology, 37th International Conference on Telecommunications and Signal Processing, Berlin (2014) pp. 49-55.
doi: https://doi.org/10.1109/TSP.2015.7296383

[14] G. Lencse, J. Palet Martinez, L. Howard, R. Patterson, I. Farrer, Pros and cons of IPv6 transition technologies for IPv4aaS, active Internet Draft, 2020. [cited 2019-11-15]
URL *https://tools.ietf.org/id/draft-lmhp-v6ops-transition-comparison-02.html*

[15] T. Anderson, SIIT-DC: Stateless IP/ICMP translation for IPv6 data center environments, IETF RFC 7755 (2016) [cited 2019-11-15].
URL *https://tools.ietf.org/html/rfc7755*

[16] M. Mawatari, M. Kawashima, C. Byrne, 464XLAT: Combination of stateful and stateless translation, IETF RFC 6877 (2013) [cited 2019-11-15].
URL *https://tools.ietf.org/html/rfc6877*

[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 (2015) [cited 2019-11-15].
URL *https://tools.ietf.org/html/rfc7599*

[18] A. Conta, S. Deering, Generic packet tunneling in IPv6 specification, IETF RFC 2473 (1998) [cited 2019-11-15].
URL *https://tools.ietf.org/html/rfc2473*

[19] O. D'yab, An overview of the most important implementations of IPv4aaS technologies, 14th International Symposium on Applied Informatics and Related Areas, University of Óbuda, Székesfehérvár, Hungary (2019) pp. 143-146.