NAIST-IS-DD1461208

Doctoral Dissertation

IPv6NET: A Collection of Methodologies for the Evaluation of IPv6 Transition Technologies

Marius Liviu Georgescu

September 16, 2016

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Marius Liviu Georgescu

Thesis Committee:

Professor Tsukasa Ogasawara	(Supervisor)
Professor Keiichi Yasumoto	(Co-supervisor)
Professor Kazutoshi Fujikawa	(Co-supervisor)
Associate Professor Youki Kadobayashi	(Co-supervisor)
Associate Professor Gábor Lencse	(Co-supervisor)

Abstract

Years from now, IPv4 will be remembered as an important part of the Internet's history, but for now it remains the dominant Internet Protocol and a pending danger to the Internet's expansion. In 1998, IPv6 was introduced to solve the address shortage created by IPv4. However, the transition period which should have brought the end of the IPv4 era has no clear end in sight. With about 6% worldwide deployment rate, IPv6 still looks like a promise and the IPv6 transition like an ongoing struggle.

Among the many challenges introduced by this transition process to the Internet community, one of the most difficult was presented to the network operators. All of the existing production networks were forced to reconsider their inner architecture to move towards IPv6. To support network operators in this challenge, the IETF has proposed multiple IPv6 transition and coexistence technologies. Following the standardized specifications, various implementations have been introduced as well.

Considering the internal policies of each network operator, one or more technologies could be considered suitable to complete the transition to IPv6 and offer coexistence support to legacy nodes. In this context, a problem remains open: which one of these transition technologies is more suitable than the rest? Moreover, different implementations of the same technology can have different capabilities, further complicating the problem.

To support network operators solve this problem, we are proposing a collection of practical evaluation methodologies, exploring four feasibility dimensions of transition technologies: network performance, scalability, security and operational capability. The methodologies were associated with a heterogeneous IPv4 and IPv6 network testbed, which we called the IPv6 Network Evaluation Testbed (IPv6NET).

In order to validate these methodologies, we have used them to analyze the feasibility of two open source transition implementations, covering multiple transition technologies. The feasibility analysis was based on practical means, employing existing running code and empirical measurements. To that end, we are showing how network performance, scalability, security and operational capability data can be obtained, analyzed and compared. As a mean to refine the methodology and consider the input of various interested operators and vendors, we have worked on standardizing parts of the proposal in the IETF, within the BMWG and OPSEC working groups.

Keywords: IPv6 Transition, IPv6NET, Benchmarking Methodology, Scalability, Security, Asamap, Tiny-map-e, MAP-E, MAP-T, DSLite, 464XLAT*

*Doctoral Disertation, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1461208, September 16, 2016.

Acknowledgments

This work would not have been possible without the help and support of several people. I would like to start by thanking my former supervisor, Professor Suguru Yamaguchi, to whom I owe my deepest gratitude for giving me the chance to study here, and for offering his invaluable support and guidance.

I am very grateful to Professor Tsukasa Ogasawara, Professor Keiichi Yasumoto, Professor Kazutoshi Fujikawa and Associate Professor Gábor Lencse for kindly agreeing to be members in my thesis committee. Their thorough review and constructive suggestions have considerably improved the quality of my work.

I would like to express my particular appreciation and thanks to Associate Professor Youki Kadobayashi for being a constant source of invaluable feedback and guidance. His generous support and encouragement towards attending the IETF meetings has greatly impacted my work in general, and this work in particular.

I owe a great deal of gratitude to Associate Professor Hiroaki Hazeyama for inspiring, patiently guiding, and generously supporting each of my research steps.

I would like to extend my thank you to Associate Professor Takeshi Okuda and Assistant Professor Shigeru Kashihara for offering their priceless advice and encouragement.

I owe my appreciation and gratitude to Assistant Prof. Doudou Fall, for being a good senior, an insipring colleague and a great friend.

Insightful discussions with my colleagues Christopher Michael Yap and Ady Wahyudi Paundu have been a constant source of inspiration, for which I am very grateful. Christopher's generous efforts to improve my English are also worthy of acknowledgment and led to better quality text in most of my publications.

The Internet Engineering Laboratory is not just a great place to conduct research, it also represents a fertile environment for cooperation and friendship. For that, I would like to extend my gratitude to all the members of this laboratory.

From the IETF community, I would like to start by thanking Scott Bradner for providing a wealth of inspiration with his work and for generously offering great advice and reviews of my work. The gratitude should be extended to Al Morton, Fred Baker and Fernando Gont for their detailed review and useful comments. I would also like to thank my mentor at the IETF, Dan Romașcanu and the Mentoring Program, led by Nalini Elkins, for guiding my steps in the IETF community. Among other IETF participants, I would like to thank Bhuvaneswaran Vengainathan, Kaname Nishizuka, Yasuhiro Ohara, Masataka Mawatari, and Kostas Pentikousis for their useful suggestions. The IETF thank you list would not be complete without mentioning the GPC crew: Maddy Conner, Naveen Khan and Joe Parrott. The IETF days wouldn't have been as fun without you.

I would like to also thank Mr. Masakazu Asama and Mr. Yukito Ueno, the developers of *vyatta asamap* and *tiny-map-e* respectively, open source implementations, upon which the experimental networks were built. Special thanks should be extended to the teams at NICT StarBED, WIDE and NECOMA for their continuous support.

I wish to also thank the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) for providing me with the the opportunity to study in Japan and for continuously supporting my research.

Last but not least, I would like to thank my family and friends for their love and support throughout the entire process.

To the memory of Professor Suguru Yamaguchi.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *Wonder is the feeling of a philosopher, and philosophy begins in wonder.*
>
> Plato

Does the Internet really need IPv6? This is the question I asked myself when starting the research on IPv6 in 2009. Having worked as an engineer for a small size enterprise network, the answer was not obvious at the time. Carrier Grade NAT seemed to have the upper-hand as the IPv6 deployment was still in its infancy. Reading more on the subject, I started seeing IPv6 as a knight in shining armor, coming to save the Internet from the exhaustion dragon. I still remember a passionate discussion about the future of IPv6 with a friend and fellow network engineer, who was skeptical about its adoption even years after. My points were clear: better numbers, better routing aggregation, better security, better everything. Now, I see things quite differently. The following subsections present the context of that change in perspective.

1.1 IPv6 Transition: A Question of When, not a Question of If

Although the aura of IPv6 has dimmed in my eyes, a reality stands: the Internet needs IPv6. *Why ?* you might ask. The answer is pretty simple, and it comes from the bits reserved for the IPv4 address field ($32 \rightarrow 2^{32}$ unique addresses). Approximately 7 billion potential Internet users cannot be serviced by roughly 4 billion addresses. If we consider the Internet of Things (IoT) as well, the Internet expansion rates are simply not sustainable with Carrier Grade NAT and IPv4.

Foreseeing this simple reality, the main standards body behind the Internet, the Internet Engineering Task Force (IETF) started developing IPv6 as early as 1995[1], with a more stable release in 1998[2]. IPv6 uses a 128 bit address, extending the address space to $2^{128} \approx$

$3.4 \cdot 10^{38}$ unique IP addresses, which should last us for many years to come. However, the appeal of IPv6 has diminished in the mean time. The main reason is its lack of backwards compatibility. In other words, IPv6 is unable to communicate directly with its predecessor, IPv4.

This introduced the Internet community to a great challenge, namely the transition to IPv6. This transition can be simply defined as the period that the Internet will have to undergo until IPv6 will completely replace IPv4.

1.2 Motivation and Problem Statement

Given the complexity of the current IPv4-dominated Internet, the transition to IPv6 will likely be a long and complex process. So far, only a small percentage of production networks are IPv6-capable. The APNIC Labs IPv6 deployment report[3] shows that only about 6% of the worldwide Internet users have IPv6 connectivity.

The IETF Next Generation Transition (ngtrans) working group and its successor IPv6 Maintenance (6man) have made many efforts to propose and analyze viable transition technologies. As surveyed by Leng et al. in [4] and by P. Wu et al. in [5], all transition technologies have advantages and disadvantages considering a certain transition scenario, but no transition mechanism can be considered most feasible for all the scenarios.

The question of which one of these transition technologies is most feasible for a particular scenario remains open. Given the complexity and the diversity of transition technologies, this leads to a great challenge for network operators faced with the IPv6 transition. Transition implementations, covering one or multiple transition technologies have been proposed as well, further complicating the problem.

Among the many technical IPv6 transition challenges identified in [4] and [6], we consider the following as most important.

Network Performance: is one of the cornerstones of modern computer networks. The overhead created by the IPv6 transition technologies in heterogeneous environments can affect their quality of service. Important network characteristics, such as latency, throughput and packet loss can be greatly impacted by running both protocols stacks, encapsulating packets or translating packet headers.

Scalability: is one of the biggest concerns for network operators, as the topology of production networks is usually dynamic. Among scalability aspects, we believe the most important is load scalability, as it can affect small-scale transition deployments as well as large ones. Considering the finite addressing schemes and protocol header restrictions, another scalability dimension needs to be considered in the context of IPv6 transition technologies: structural scalability.

Security: is arguably the biggest concern for network operators. Aside from the larger address space, IPv6 has, in theory, a number of advantages over its predecessor in terms of design: a more efficient and extensible datagram, stateless autoconfiguration and better security. Over the years, however, many of these new features have proved to be challenges for enforcing security (e.g. extension headers, stateless auto-

configuration), or not-feasible (e.g. widespread deployment of IPv6 with IPsec). The IPv6 transition has further aggravated these problems, as transition technologies are generally exposed to the threats associated with both IP versions and hybrid blends, depending on the subcomponents. More concretely, a heterogeneous IPv4 and IPv6 environment greatly increases the attack surface.

Operational capability: shows how a certain technology fits in with the existing environment or how it manages to solve operational problems. Considering the simplicity of the two IP versions, the transition technologies evolved into complex systems which are difficult to grasp and manage. For network operators, this can translate to redefining internal procedures or retraining staff, depending on the complexity of the chosen transition technology.

1.3 Contributions

To help network operators solve problem of which of the IPv6 transition technologies is more suitable for their scenario, we are proposing a suite of methodologies associated with the IPv6 Network Evaluation Testbed (IPv6NET), an experimental environment dedicated to the evaluation of IPv6 transition mechanisms in a series of practical scenario-based network tests.

The main contributions of this work are the evaluation methodologies associated with IPv6NET, which cover four technical feasibility dimensions: network performance, scalability, security and operational capability. The details were organized per feasibility dimension as follows.

Network performance

In terms of evaluating network performance, we have proposed the use of well-established metrics such as: round-trip delay, packet delay variation, throughput and frame loss. While the metrics and procedures for employing them have been based on existing work, the performance analysis steps needed to be tailored to the context of IPv6 transition technologies. To clarify, we have proposed associated test procedures and considered the overhead created by the transition technologies in the measurement process.

Scalability

Scalability has often been discussed in the context of network devices, and by extension in the context of IPv6 transition technologies. To the best of our knowledge, however, a formal definition or a measurement method has not been proposed before our work. We have defined scalability as the ability of each transition technology to accommodate network growth. In terms of measurement procedure, we have proposed measuring load scalability by analyzing the performance degradation associated with the network growth. In terms of structural scalability, we have analyzed the structural limits imposed by some of the IPv6 transition technologies addressing schemes.

Security

In terms of security, there are many articles discussing the security considerations of employing IPv6 and IPv6 transition technologies. However, our proposed threat model was the first to offer a structured and scenario-oriented approach in dealing with the security threats of IPv6 transition technologies. Starting with the generic STRIDE approach, we have described the steps needed to identify, classify and prioritize the security threats associated with IPv6 Transition Technologies.

Operational capability

A methodology to quantify the operational capabilities of IPv6 transition technologies was, as well, the first in current literature. To that end, we have defined three evaluation dimensions: configuration capability, troubleshooting capability and applications capability. Using a non-exhaustive approach, we have proposed a list of configuration, troubleshooting and applications tasks to evaluate the conformance of IPv6 transition implementations.

The proposed methodologies have made use of practical means, such as existing implementations and empirical measurements. Aside from acting as validation tool for the proposed methodology, the associated empirical results can be considered as a collateral contribution in itself. The empirical analysis revealed performance trends and unexpected behaviors which could have been been overlooked if simulators or analytical tools would have been employed. Moreover, these results pointed out implementation caveats and practices which should be avoided.

There are two main beneficiaries of our proposed methodologies. First, the resulting empirical data can serve as a direct guideline to network operators faced with a similar transition scenario. The guideline can help network operators understand the impact of the transition on the current service. It can also help avoid implementations which are below an expected standard of feasibility. Considering service level agreements, the empirical analysis can facilitate the construction of an IPv6 transition plan and ultimately, to a faster transition process for the network operator in question.

Second, this can be valuable feedback for transition implementation developers. The empirical analysis can help vendors understand if certain versions of an implementation are up to the mark, which can lead to further improvement of their products. Testing the performance limits can also reveal certain caveats and bad practices in terms of performance or scalability.

By looking at this work from the industry perspective, we have addressed mainly the technological challenges of the IPv6 transition. However, we contend our work can help overcome some other challenges identified in Section 2.2, such as *costs of the adoption*, *availability of IPv6-ready products* and *lack of trained staff*. Regarding the costs of the adoption, the basic network templates can help operators and decision makers alike understand the minimum number of transition devices needed to start the IPv6 transition. In turn, the number can express a baseline investment cost. The benchmarking scores of a specific transition implementation can offer insights about its IPv6-readiness. Ad-

ditionally, more awareness about the feasibility of open-source implementations may fuel the development of other transition implementations. In terms of the lack of IPv6 trained staff, we contend that the detailed operational capability surveys can help operators better understand the essential operational features of transition implementations.

1.4 Thesis Structure

The remainder of the thesis is structured as follows:

Chapter 2 : offers background information, familiarizing the reader with the landscape of the IPv6 transition and the challenges it introduced the Internet community with. The structural details of some of the transition technologies are provided, as well as the means to classify most of the existing technologies. The chapter also gives an overview of literature related to the evaluation of IPv6 transition technologies.

Chapter 3 : presents details about the IPv6NET concept and how the methodology integrates with the rest of the components. The chapter introduces the terminology used throughout the thesis and provides an overview of the proposed methodologies and the associated dimensions.

Chapter 4 : details the methodologies for quantifying network performance and operational capability, opposing the two measurement approaches: closed and open environment testing. The chapter also includes empirical data for the two feasibility dimensions.

Chapter 5 : introduces our approach for quantifying scalability. The methodology to cover both load and structural scalability are discussed in detail. Moreover, an empirical of the two dimensions is included.

Chapter 6 : displays our attempt to approach a security quantification method by first building a supporting structure for the potential security threats associated with IPv6 transition technologies. The chapter also contains a detailed threat analysis of the generic classes of IPv6 transition technologies.

Chapter 7 : discusses the validity and the future of this research project. In addition, the chapter attempts to predict the future of the IPv6 transition technologies in the context of future core Internet technologies.

Chapter 8 : concludes the thesis with a summary of the proposed methodologies in the context of potential future directions and key developments in Internet core technologies.

Chapter 2

The Premises of the IPv6 Transition

> *Life is pleasant. Death is peaceful. It's the transition that's troublesome.*
>
> Isaac Asimov

The social and business environment represented by the Internet today is close to the biggest turning point in its history. The widely deployed Internet Protocol Version 4 (IPv4) is showing its limitations. The IPv4 address space, which has $2^{32} \approx 4.3$ billion unique IP addresses. This number is not sustainable considering roughly 7 billion potential Internet users. The natural consequence is the exhaustion of the IPv4 address space and an imminent threat to the expansion of the Internet.

2.1 IPv6 Transition Overview

On February 3rd 2011 the Internet Assigned Numbers Authority (IANA) announced the allocation of the last blocks of IPv4 addresses [7]. From the five Regional Internet Registries (RIR), four have entered the last stage of IPv4 Exhaustion. The Asia Pacific Network Information Centre (APNIC), the RIR for the Asia Pacific region was the first RIR to announce entering the last stage of IPv4 Exhaustion (the final /8 address block) [8] on April 15th 2012. Soon after, RIPE NCC (September 12), LACNIC (June 2014) and ARIN (September 2015) also announced the last stage of IPv4 exhaustion. AFRINIC is the only RIR not there yet, but it is estimated to reach that stage by March 2018. A detailed report on the IPv4 exhaustion is presented by APNIC's G. Huston in [9].

The answer to the IPv4 addresses exhaustion problem is the deployment of the next generation Internet Protocol, the Internet Protocol Version 6 (IPv6), presented by the Internet Engineering Task Force (IETF) in 1998[2]. IPv6 uses a 128 bit address, extending the address space to $2^{128} \approx 3.4 \cdot 10^{38}$ unique IP addresses, a significant improvement considering the IPv4 address space. However, the appeal of IPv6 has diminished since 1998, mainly because it is not able to communicate directly with its predecessor, IPv4. This introduced the Internet community with a great challenge, the transition to IPv6.

The transition is an ongoing process and is represented by the stages the Internet will have to undergo until IPv6 will completely replace IPv4.

Given the complexity of the current IPv4-dominated Internet, the transition to IPv6 is expected to be a long and complex process. Until now, only a percentage of production networks are IPv6-capable. The highest is for APNIC, with about 20.9 % [10]. As for a global view, the APNIC Labs IPv6 deployment report[3] shows that only about 6% of the worldwide users have IPv6 connectivity.

2.2 IPv6 transition challenges

From the industry perspective, the book *Global IPv6 strategies*[6] explores some of the obstacles preventing Internet companies from adopting IPv6 so far.

Lack of apparent use can be defined as the lack of a killer application to drive the IPv6 adoption.

Costs of the adoption were considered unjustified. Investments in IPv6 were considered unnecessary, as the return of investment (ROI) was hard to predict.

Technology challenges such as the network performance, scalability and security of IPv6 implementations was questioned.

Availability of IPv6-ready products was limited. The lack of commercial IPv6-ready products prohibited transition interested companies from starting the transition process.

Lack of trained staff is still an issue. Many network operation teams lack IPv6 knowledge.

With time, many of these obstacles have been overcome. As shown by the World IPv6 Launch infographic [10] the industry has understood that IPv6 adoption is not a question of if, but a question of when. As the biggest standards community behind the Internet, the IETF has made many efforts to formalize the IPv6 transition, by introducing typical network transition scenarios and proposing transition technologies. However, the very low worldwide adoption rate indicates that there are still open problems.

From the academic perspective, the IPv6 transition presented many opportunities for research. As surveyed by X. Leng et al. in [4], and by P. Wu et al. in [5], deciding which transition technology is the most feasible for a specific network scenario, remains one of the biggest open problems. Among the many technical challenges identified in [4] and [6], network performance, scalability, security and operational capability can summarize most of the technical feasibility spectrum of IPv6 transition technologies. Therefore, these four dimensions have been the priority of our research.

2.3 IPv6 Transition Technologies

IPv6 was not designed to be backwards compatible. In other words IPv6-only nodes cannot directly communicate with IPv4-only nodes. Consequently, coexistence and tran-

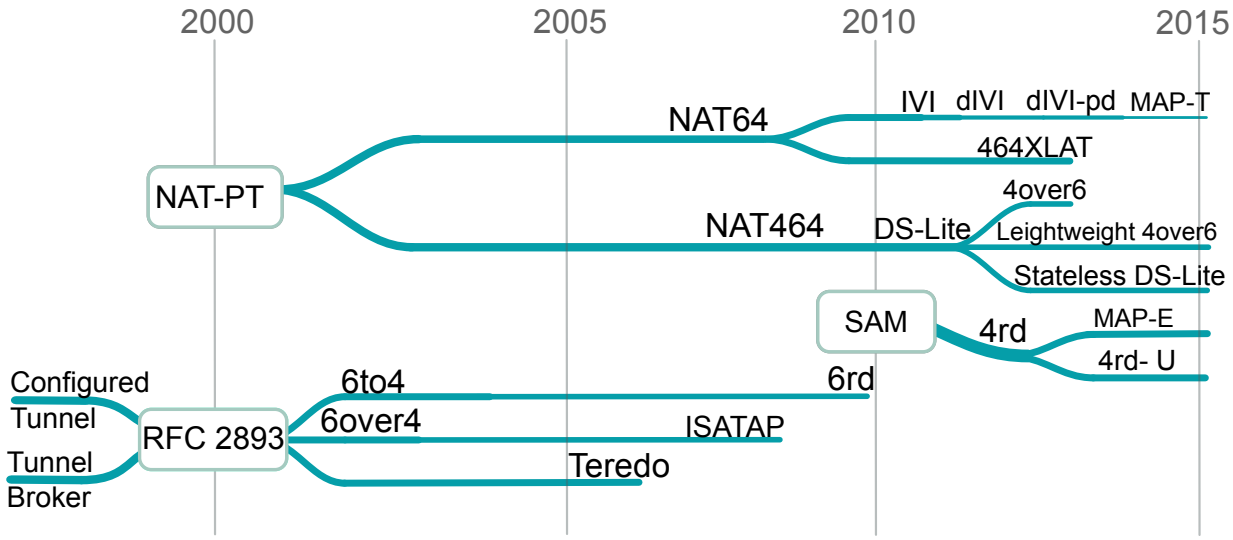
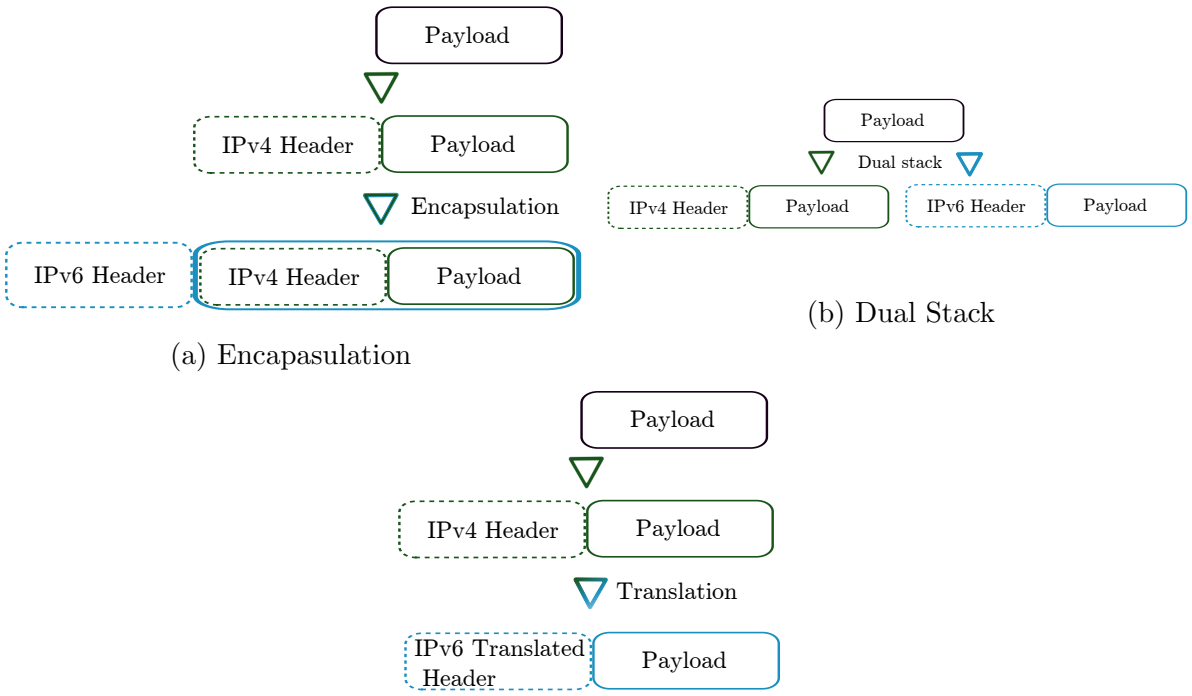


Figure 2.1: Evolution of transition technologies in the IETF



(a) Encapasulation

(b) Dual Stack

(c) Translation

Figure 2.2: Basic IPv6 transition technologies

sition technologies need to be employed. Initially, three basic transition mechanisms were proposed: dual-stack, translation and tunneling. The associated implementation standards are presented in RFC4213 [11] and RFC6144 [12]. An abstraction of the three is shown in Figure 2.2. Over the years many other transition technologies have been introduced by

the ngtrans Working Group of the IETF. Figure 2.1 depicts a road-map of the evolution of some of the transition technologies proposed in the IETF.

For dual-stack abstracted in Figure 2.2b, both IPv4 and IPv6 are implemented on the same node. This method is mostly used in host-side nodes and edge nodes. The main challenge introduced by dual-stack is overhead, as it needs two routing tables and routing processes.

Translation displayed in Figure 2.2c is the only method which achieves direct communication between IPv4 and IPv6, by translating the information and message format between different versions of Internet Protocol. Usually translators are employed at the border between an IPv4-only and an IPv6-only site. The main problem with translation is that it breaks something which IPv6 was supposed to bring back: the end-to-end characteristic of the Internet. Aside from that, translation can affect the functionality of secure protocols, such as IPSec or DNSSEC. Modern translation technologies can be classified as stateless technologies (e.g. IVI[13], dIVI[14])and stateful technologies (e.g. NAT64 [15], DSLite [16]). Stateless translation technologies achieve a one-to-one address mapping, translating only the IP and ICMP headers. On the other hand, stateful translation builds a one-to-many IPv4 address mapping, by using the IPv4 address resources as a pool on the translating device, and allocating them at per port granularity. Stateful translators require a great deal of per-state flow maintenance, in other words every incoming packet has to be classified to its corresponding queue, increasing the overhead on the network devices involved. However, stateless translators need one IPv4 address for every IPv6 host, which negates the primary advantage of IPv6, which is the increase in address space.

Tunneling or encapsulation presented in Figure 2.2a is employed to traverse heterogeneous network environments, by encapsulating the IPvX packets into the payload of IPvY packets, were $X, Y \in \{4, 6\}$. At the border of the IPvY and IPvX networks the packets are decapsulated back into IPvX by an edge router. Tunneling technologies, initially introduced in RFC1933 [17], can be classified in three categories: static tunnels, semi-automatic and automatic tunnels. The static tunnels require manual configuration at both ends. Their main advantage is the simplicity of deployment, which makes them cost effective and attractive for some Internet Service providers (e.g. Nippon Telegraph and Telephone Corporation). Semi-automatic tunnels also need manual configuration, but only on the host side, as the provider side is auto-configured. For automatic tunneling (e.g. 6to4 [18], ISATAP [19], TEREDO [20]) the tunnels are created on-demand. Tunneling mechanisms are confronted with fragmentation and MTU problems because of encapsulation. The encapsulation/decapsulation process will also induce considerable overhead in the network devices involved in the process. Security considerations have to be taken into account as well. Tunnels are especially vulnerable to spoofed encapsulated packet attacks, which can target a normal node or a tunnel end-node. In automatic tunneling mechanisms the security threat can increase by targeting the spoofed packets at the broadcast/multicast address of relay routers.

Trying to compensate for the design simplicity of the two IP versions, IPv6 transition technologies grew ever more complex and subsequently hard to grasp and implement. Many of the modern transition technologies use one or more basic transition technologies. For example Dual-Stack Lite (DSLite)[16] employs dual-stack and translation at the edge nodes

and encapsulation in the core.

Another classification of transition technologies can be achieved by the phases of the IPv6 transition they can be associated with. In RFC6144 [12], three important phases have been identified:

Preparation phase in which IPv6 services are scarce, and few production networks have working IPv6-enabled cores. In this phase, mostly IPv6 over IPv4 technologies (e.g. 6to4 [18], 6rd [21]) are needed.

Transition phase in which IPv6 presence is increasing, hence dual stack support and services should be provided. Although IPv6 use is still very low, many large Internet companies started offering services also over IPv6. This can be considered the current ongoing phase. This phase is expected to increase the number of dual stack and IPv4 over IPv6 technologies (e.g. DSLite[16], MAP-E [22]).

post-Transition phase, the last stage of the transition, in which IPv6 will be the dominant protocol. This phase should offer support to IPv4-only islands over IPv6-only infrastructures, and IPv4 over IPv6 technologies will become dominant.

The classifications mentioned previously can be useful for determining either the subcomponents of a particular transition technology, or for learning the appropriate technology in a certain transition phase. However, in order to reuse some of the evaluation criteria and associated test environments, we have proposed an alternate generic classification of IPv6 transition technologies.

2.3.1 A generic classification of IPv6 Transition Technologies

We start with the assumption that a production network undergoing the IPv6 transition is constructed using the following IP domains:

- Domain A: IPvX specific domain
- Core domain: which may be IPvY specific or dual-stack(IPvX and IPvY)
- Domain B: IPvX specific domain
  $X, Y \in \{4, 6\}$

Considering this production network design, the technologies can be categorized according to the technology used for the core domain traversal as follows:

1. **Single Translation**: the production network is assumed to have only two domains, Domain A and the Core domain. The core domain is assumed to be IPvY specific. IPvX packets are translated to IPvY at the edge between Domain A and the Core domain.
2. **Dual-stack**: the core domain devices use both IP protocols
3. **Encapsulation**: the production network is composed of all three domains. Domains A and B are IPvX specific, while the core domain is IPvY specific. The IPvX packets are encapsulated to IPvY packets at the edge between Domain A and the Core

Table 2.1: IPv6 Transition Technologies Association

	Generic category	IPv6 Transition Technology
1	Dual-stack	Dual IP Layer Operations[11]
2	Single Translation	NAT64[15], SIIT-DC[23], SA46T-AT[24], IVI[13]
3	Double Translation	464XLAT[25], MAP-T[26], dIVI[14]
4	Encapsulation	DSLite[16], Lightweight 4over6[27], MAP-E[22]

domain. Subsequently, the IPvY packets are decapsulated at the edge between the Core domain and Domain B.

4. **Double Translation**: The production network is assumed to have all three domains, Domains A and B are IPvX specific, while the core domain is IPvY specific. A translation mechanism is employed for the traversal of the core network. The IPvX packets are translated to IPvY packets at the edge between Domain A and the Core domain. Subsequently, the IPvY packets are translated back to IPvX at the edge between the Core domain and Domain B.

Table 2.1 shows how some of the existing IPv6 transition technologies can fit into the generic categories.

As a result of categorizing the transition technologies, similar test setups can be used for analyzing the performance of potentially competing technologies or implementations.

2.4 A Detailed Perspective on the Analyzed IPv6 Transition Technologies

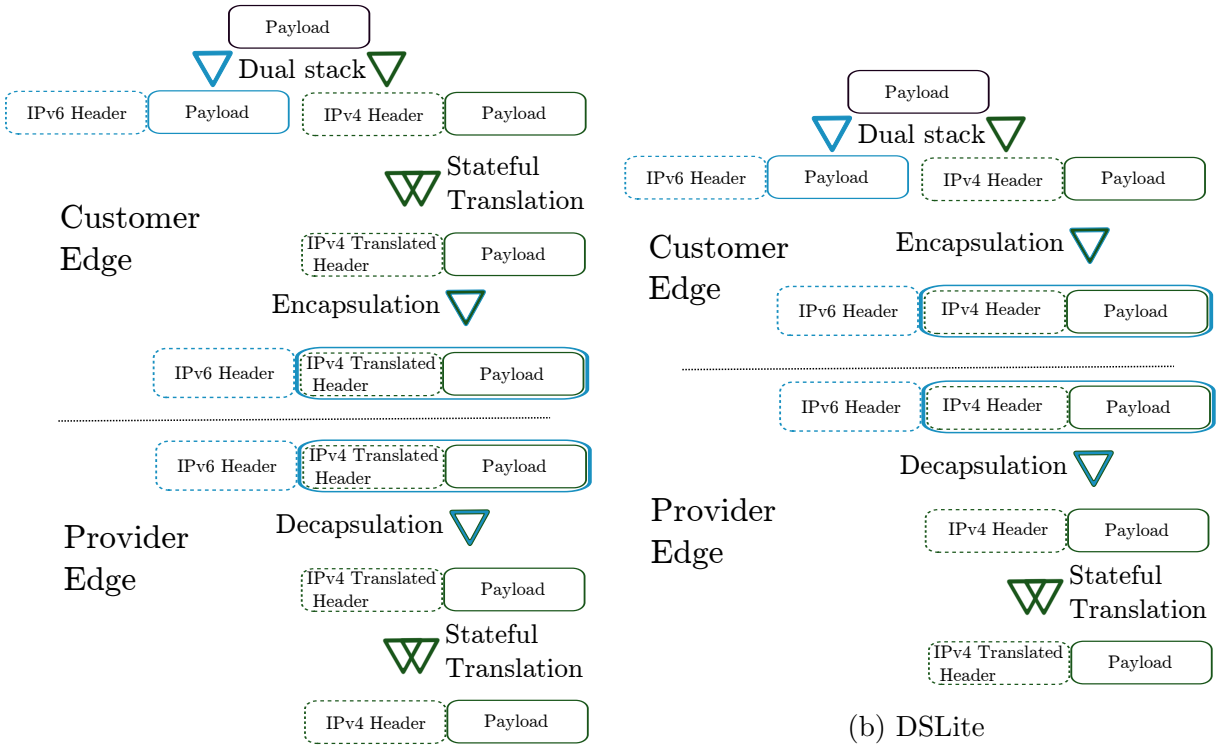
To prove the validity of the proposed methodologies we have used as study case two open source transition implementations: Asamap [28] and Tiny-map-e[29]. Asamap covers four standardized transition technologies: MAP-E [22], MAP-T[26], DSLite [16] and 464XLAT[25]. Tiny-map-e, on the other hand, covers only MAP-E. For a better understanding of the four technologies, here is a detailed analysis of their functionality and sub-components. The four have been structured according to the generic category they would fit into.

2.4.1 Encapsulation Technologies

For encapsulation transition technologies, the core domain is traversed using an encapsulation mechanism. The operations associated with the MAP-E[22] and DSLite[16] standards, are depicted in Figure 2.3.

MAP-E is an automatic tunneling transition mechanism. It allows the transportation of IPv4 packets over an IPv6 backbone network, using IP encapsulation and a mapping mechanism between IPv6 addresses and IPv4 addresses with transport layer ports.

The MAP-E environment needs the following building blocks:



(a) MAP-E

(b) DSLite

Figure 2.3: Encapsulation IPv6 Transition Technologies

- MAP domain: the IPv6 network which interconnects the MAP components. In the same IPv6 networks multiple MAP domains can be employed.
- MAP Border Relay (BR): a MAP-enabled router with at least one IPv6 interface and one IPv4 interface, connected to the native IPv4 network.
- MAP Customer Edge (CE): a customer edge router which serves as a residential site with one IPv6 enabled WAN interface and one or multiple LAN interfaces. It is important to note that the CE router also performs a Network Address Translation (NAT) function.
- MAP Rule: a set of mapping parameters characteristic to a specific MAP domain. For a MAP rule a prefix for both IPv4 and IPv6, and an exact number of Embedded Addresses (EA) bits is required. Additionally, for each customer site, an IPv6 sub-prefix is assigned. Using the EA bits and the customer sub-prefix, the shared IPv4 prefix/IPv4 address and the Port Set Identifier (PSID) are calculated.

One of the advantages of MAP-E can be the CE element architecture. The CE is handling the NAT function, relieving the core network of that responsibility. This also eludes the danger of a single point of failure, characteristic to Carrier Grade NAT (CGNAT) [30] architectures.

Perhaps one of the biggest disadvantages of MAP-E is represented by the mapping rule, which is complex and can introduce operational issues, when configuring or troubleshooting. To that end, a very useful tool is the MAP simulation tool [31] created by Arthur Lacoste of Cisco Systems. The addressing rule can also create problems, but only for large scale production networks (e.g. ISP Networks) with a low public IPv4 address pool.

DSLite is a stateful tunneling mechanism that relies on an IPv6 backbone network. It employs IPv4-in-IPv6 tunnels to cross the IPv6 network and reach a carrier-grade IPv4-IPv4 Network Address Translation (CGNAT) [30] device, allowing customers to share IPv4 addresses. A DSLite environment is based on the following components:

- Basic Bridging Broad Band (B4) component: represents a function implemented in a dual-stack node, either integrated into a CPE or directly connected, which creates an IPv4-in-IPv6 tunnel to an AFTR.
- Address Family Transition Router (AFTR) component: represents a device which is connected to the native IPv4 network and represents the end-point of the IPv4-in-IPv6 tunnels. The AFTR integrates a carrier-grade NAT function which allows B4 enabled CPEs to share the same IPv4 address pool.
- Shared IPv4 address pool: a public IPv4 prefix/IPv4 address shared among multiple CPEs.

In contrast to MAP-E, the provider edge element includes a CGNAT function, which requires per-flow maintenance, increasing the operational complexity. It is also susceptible to the single point of failure issue. However, this can be avoided with a redundant design.

One of the biggest advantages of DSLite is represented by interoperability, as many production networks are already using CGNAT machines.

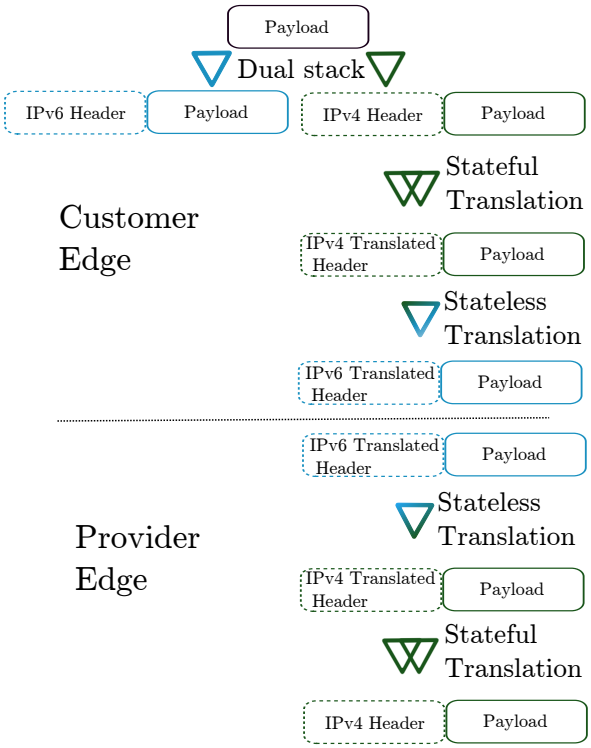
2.4.2 Double Translation Technologies

In the case of double translation IPv6 transition technologies, translation represents the mechanism for traversing the IPvY-only core domain. We are analyzing two double translation technologies: MAP-T [26] and 464XLAT[25]. An abstract model of the operation model of the two technologies is presented in Figure 2.4.

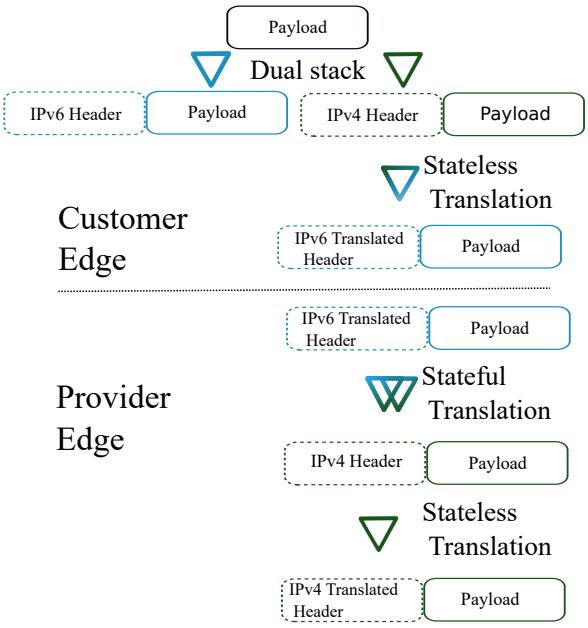
MAP-T is an IPv6-IPv4 Network Address Translation solution which provides shared or non-shared IPv4 address connectivity over an an IPv6-only core network.

Similarly to MAP-E, the MAP-T environment needs the following building blocks:

- MAP domain: the IPv6 core which interconnects the other MAP components. Multiple MAP domains can be employed in the same IPv6 network.
- MAP Border Relay (BR): a MAP-enabled machine connected to the native IPv4 network, at the edge of the MAP domain.
- MAP Customer Edge (CE): a customer edge router used as a residential site. The CE router is performing the Network Address Translation (NAT) function.



(a) MAP-T



(b) 464XLAT

Figure 2.4: Double Translation IPv6 Transition Technologies

- MAP Rule: the mapping parameters specific to a certain MAP domain. Each MAP rule needs an IPv6 prefix, an IPv4 prefix and a specific number of Embedded Address (EA) bits. An IPv6 sub-prefix is assigned for each customer site. From the EA bits and the customer sub-prefix, the shared IPv4 prefix/IPv4 address, and the Port Set Identifier (PSID) are calculated.

The disadvantages of MAP-E stand for MAP-T as well. The mapping rule can increase the operational complexity for both configuring and troubleshooting. The addressing rule can create problems as well, but mainly for large scale production networks with low a public IPv4 address pool.

The CE element architecture can be one of the main advantages of MAP-T as well. By handling the NAT function, the CE relieves the core network of that responsibility. This also can avoid the danger of a single point of failure, characteristic to Carrier Grade NAT (CGNAT) architectures.

464XLAT combines stateful protocol translation with stateless protocol translation to provide IPv4 connectivity across an IPv6-only network. A 464XLAT environment needs the following components:

- PLAT: provider-side translator, which employs stateful translation, N to 1 global IPv6 addresses to global IPv4 addresses, and vice versa.

- CLAT: customer-side translator, employing stateless translation to map 1 to 1 private IPv4 addresses to global IPv6 addresses, and vice versa.

464XLAT also uses a shared IPv4 public address and the stateful translation is realized in the core network. This means it inherits the core network overhead, and single point of failure issues. By combining stateless and stateful translation, 464XLAT is considered easy to deploy and efficient from the public IPv4 pool stand-point. It is also considered suitable for 3GPP transition networks.

All of the four analyzed technologies can be suitable candidates for an operator running and IPv6-only backbone, and although there are structural reasons for choosing one or the other, we contend that a thorough empirical feasibility analysis is needed in order to confirm performance trends or identify interoperability issues and potential pitfalls. To the best of our knowledge, there is no similar initiative, which brings further motivation to our cause.

2.5 Related work

There are a variety of articles dedicated to IPv6 transition experimental environments in current literature. These environments can result in either reasonable and fine-grain performance data or quantitative operational data. Both approaches are relevant for the proposed methodologies. More details can be found in the next two subsections. The other two subsections have been dedicated to literature covering the complementary feasibility dimensions: scalability and security.

2.5.1 Closed Environments

Closed environments are usually local environments, which are isolated from production networks or the Internet. For example, I. Raicu et al. have analyzed the performance of two 6-over-4, and IPv6 in IPv4 tunneling implementations in comparison with a homogeneous IPv6-only network in [32]. S. Narayan et al. evaluates the performance of Linux operating systems in relation to an IPv4-v6 configured Tunnel and a 6to4 Tunnel in [33]. Four workstations were employed to build the testbed. S. Sasanus et. al. measures the differences in bandwidth requirements for common network applications like remote login, web browsing, voice communication and database transactions over 3 types of networks: IPv4-only, IPv6-only and a 6to4 tunneling mechanism in [34]. The environment was built using the OPNET simulator, which also served as the basis for the testbed presented by P. Grayeli et. al in [35], which was dedicated to the performance analysis of transition mechanisms over a MPLS backbone. In [36], G. Lencse et al. evaluate the performance of DNS64 implementations, BIND9 and TOTD running on Linux, OpenBSD and FreeBSD. Furthermore, the research team has analyzed the performance and stability of open NAT64 implementations in [37] and [38].

A common trait of the above mentioned closed environments is the thorough performance analysis, which resulted in quantifiable (hard) data, such as CPU and memory utilization, throughput, end-to-end delay, jitter and execution time.

However, as P. Wu et al. in [39] have underlined, before transition mechanisms are applied in a large scale environment, a systematic and quantitative performance analysis should be performed. This gets us to the second group of experimental environments, namely open environments.

2.5.2 Open Environments

Open environments can be defined as experimental networks connected to a large scale production network or to the Internet. While both types of methodologies can be considered practical, as they usually employ existing implementations, open environments are especially practical as they explore other aspects, less formalized than network performance, such as operational efficiency and interoperability.

In [40], R. Hiromi et al. have identified poor implementation and erroneous operations in a dual-stack environment. A hotel Internet service is presented as a case study. Operational issues such as lack of path/peering, Bad TCP reaction or misbehaving DNS resolution are identified.

H. Babiker et al. in [41] describe the lessons learned from deploying IPv6 in Google's heterogeneous corporate network. The report presents numerous operational troubles: the lack of dual-stack support of the customer-premises equipments (CPE), or the immature IPv6 support of operating systems and applications. One of their conclusions was that the IPv6 transition can affect every operational aspect in a production environment, hence operational considerations have to be made.

In [42], J. Arkko et al. presented experiences with IPv6-only Networks. NAT64 and DNS64 technologies are tested in two open environments: an office and a home environment. Common applications such as web browsing, streaming, instant messaging, VoIP, online gaming, file storage and home control were tested. Application issues in relation to the NAT64/DNS64 technology are identified, for example Skype's limitation to connect to IPv6 destinations, or the lack of network operational diagnostics for certain standalone games.

In [43], Répás et al. have analyzed the application compatibility of open source NAT64 implementations with common protocols such as HTTP, HTPS, SIP, P2P or FTP. The article has successfully identified some of the compatibility issues associated with NAT64, such as VoIP, P2P or active FTP applications.

Experiences with IPv6-only Networks have been also presented by Hazeyama et al. in [44]. A great deal of meaningful interoperability data was presented, such as the IPv6 capability of OSes, applications and network devices. Also many operational issues have been identified. Some examples are long fall-back routine, the low DHCPv6 capability of certain OSes, the lack of IPv6 support in some network devices, DNS64 overload, inappropriate AAAA replies or inappropriate selection of DNS resolvers.

Considering these examples, we can conclude that open environments have the potential of exposing interoperability issues, which can otherwise get overlooked. Combining the advantages of the two evaluation methodologies can lead to a complete feasibility analysis. Consequently, it represents one of the goal for our methodology.

2.5.3 Scalability

To the best of our knowledge, there are no related articles dedicated to benchmarking the scalability of IPv6 transition technologies in current literature. However, scalability benchmarking is approached in other computer science research. In [45], Bondi et al. introduce a general framework for analyzing the scalability of data processing systems. The article identifies four different types of scalability: load scalability, space scalability, space-time scalability and structural scalability. On a more specific note, the load scalability of Ethernet and Token Ring technologies is analyzed. Some measures on how load scalability may be improved are presented as well.

The paper manages to create a solid classification base for the aspects which can affect a system's scalability. Stephens et al. analyze the scalability of enterprise network architectures in [46]. The study is mostly dedicated to the scalability of Ethernet in enterprise networks. The scalability of simulated Ethernet switches under different routing schemes is quantified using three specific scalability metrics: control overhead, forwarding table state, and link bandwidth distribution. The paper identifies the factors affecting Ethernet's scalability and offers a methodology for quantifying the scalability of Ethernet or alternative Layer 2 technologies.

In [47], scalability is viewed as the ability of a Web server to support a large number of concurrent users without degradation of performance. An approach to benchmark the scalability of web server clusters is presented. As a specific metric, the maximum number of requests/second is used. Deshane et al. discuss the scalability of virtualization systems in [48]. Similarly, scalability is regarded as a system's ability to run more virtual machines without loss of performance. The scalability of two popular hypervisors, Xen and KVM is graphed and analyzed.

Scalability has been discussed as well in IPv6 transition related literature. In [49], Bi et al. present general guidelines for analyzing the feasibility of IPv6 transition technologies. The article underlines scalability as one of the most important factors in the overall feasibility of IPv6 transition technologies. The lack of scalability of some of the transition technologies is discussed as well. Similarly, in [39], Wu et al. identify structural scalability issues such as the 6to4 prefix inability to aggregate. However, a clear method for quantifying the scalability of IPv6 transition technologies has yet to be proposed. Our work is attempting to approach this void by proposing a method for benchmarking the load scalability. Our approach is to quantify the performance degradation at higher scales and use it as an indicator for load scalability as it was defined in [45]. Moreover, we propose a method for quantifying the structural scalability of some of the IPv6 transition technologies.

2.5.4 Security

Threat models have been proposed and used for the security analysis of online applications for some time. Probably the most popular is the one introduced in [50], more commonly known as the STRIDE approach. At the heart of the proposal is the STRIDE mnemonic, which stands for **S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service and **E**levation of privilege, a set of generic security threats. The mnemonic

offers a simple and comprehensible classification base. Using STRIDE in conjuncture with a good understanding of the system's components, can result in a good overview of the threats and possible mitigation directions. As a measure of its success, the STRIDE categories are used as well in [51], the threat modeling process used by the Open Web Application Security Project (OWASP). Furthermore, in [52], the STRIDE approach has been used to model the threats of industrial control systems.

As for the security of IP-based systems, there are many articles in current literature dedicated to the issues introduced by IPv4 and IPv6. In terms of IPv4 security, we have consulted the following references: [53], [54], [55], [56], [57]. As for IPv6 security, the following documents were very helpful: [58], [59], [60]. These documents have done a fine job of documenting existing threats for the two protocols and basic transition mechanisms. However, a threat model for IPv6 transition technologies has not emerged so far.

Threat modeling has proved useful for understanding the security of intricate systems. The main reason is its structured approach, which allows one to discover, categorize and classify the threats according to their potential impact on the system. Considering the complicated nature of IPv6 transition technologies, threat modeling makes a good candidate for better understanding their security implications. The proposed model aims to open this path, which could lead as well to a better understanding of the inner-workings of IPv6 transition technologies.

Chapter 3

IPv6NET: the Concept Behind the Methodologies

> *Thoughts without content are empty, intuitions without concepts are blind.*
>
> Immanuel Kant

The methodologies we are proposing are associated with a heterogeneous IPv6-IPv4 testbed, which we named the IPv6 Network Evaluation Testbed (IPv6NET). The IPv6 Network Evaluation Testbed (IPv6NET), introduced in [61], is dedicated to the evaluation of IPv6 transition implementations in relation to specific network scenarios.

3.1 The IPv6NET Concept

As presented in Figure 3.1, conceptually IPv6NET has four important components:

- **The evaluation methodologies:** dictate the coordinates of the conducted network tests.
- **The network template:** associated to a specific network scenario.
- **The network environment:** needed as base for the experimental networks.
- **The transition tuple:** represented by at least one transition implementation covering one transition technology.

By combining the four components we will obtain feasibility scores, which can be organized as an evaluation report. The report can, in turn, become the baseline of a transition plan for network operators. The empirical scores can offer an estimation of the impact of the transition on the current service requirements. For instance, a transitioning Senegalese network operator called *Senecloud* might need to guarantee throughput T for its subscribers. The network performance methodology can help the operator estimate the

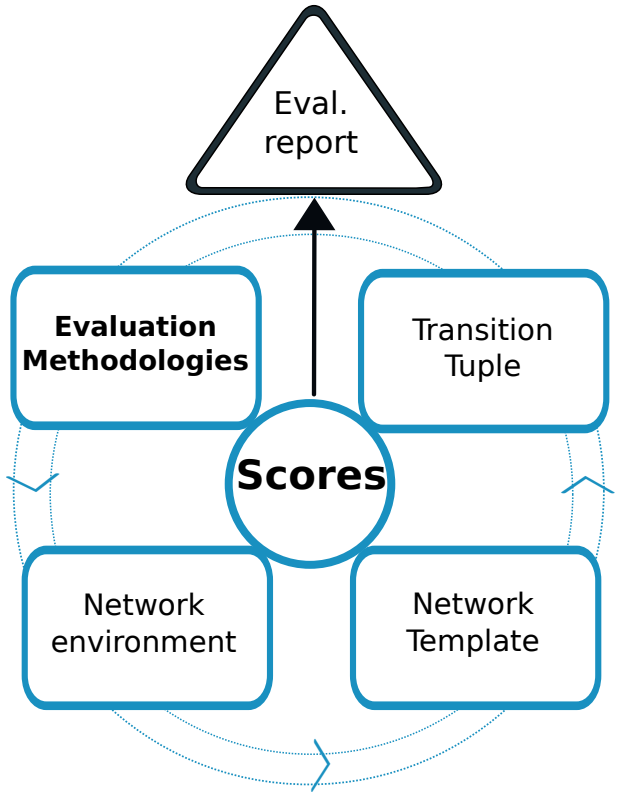


Figure 3.1: IPv6NET Concept

maximum achievable throughput for multiple transition implementations and chose the one with the best performance, or best throughput/cost ratio, assuming their throughput is higher than T.

The feasibility result can be valuable feedback for transition implementation vendors as well. The empirical data can provide feedback about the implementations in development, or improved versions of stable release. For example, hypothetical vendor *Junisco* has released a new version of the MAP-E ([22]) standardized transition technology. By comparing the general feasibility report of the previous version with the report of the current, the vendor can estimate the potential improvement.

As a somewhat collateral contribution, the generic network templates can help both operators and decision makers understand the minimum number of transition devices needed to start the IPv6 transition. In turn, the number can express a baseline investment cost and possibly estimate the return of investment.

3.2 The Overview of the Evaluation Methodologies

This section presents an overview of the proposed methodologies, as well as some clarifications regarding the terminology used throughout this thesis. Figure 3.2 shows a taxonomy of the feasibility dimensions we are proposing as evaluation criteria for IPv6 transition

technologies. The color code in Figure 3.2 reveals the level of maturity of each of the proposed metrics. Some have been long debated and included into existing standards, while others are our recent proposals.

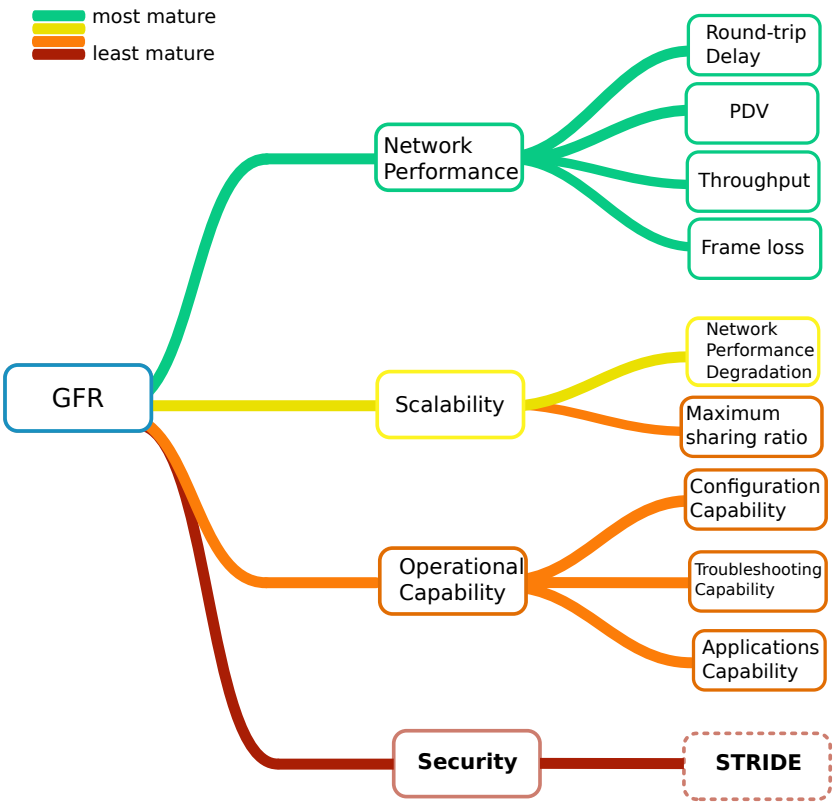


Figure 3.2: Taxonomy of Proposed Feasibility Dimensions

The General Feasibility Report (GFR) is envisioned as a consolidated evaluation report, associated with a transition tuple or a number of competing tuples. The analyzed feasibility dimensions are detailed in the following.

Network Performance: measures the efficiency of each technology in relation to existing computer network standards.

- *Round-trip delay*: follows the latency guidelines defined in RFC2544 [62]
- *Packet Delay Variation (PDV)*: follows the definition and measurement procedure we have defined in [63]
- *Throughput*: follows the guidelines defined in RFC2544 [62]
- *Frame loss*: follows the guidelines defined in RFC2544 [62]

Round-trip delay, packet delay variation, throughput and frame loss have been long discussed in benchmarking communities, such as the BMWG in IETF. This is why we have labeled them as most mature. As a consequence, the resulted network performance data, can be as well considered more trust worhy. More about the

network performance methodology is presented in Section 4.1.

Scalability: is regarded as the ability of each transition tuple to accommodate topology growth.

- *Network Performance Degradation (NPD)*: measures load scalability by computing the network performance degradation when the topology of the transition network grows.
- *Maximum sharing ratio*: quantifies the structural scalability limits imposed by some of the IPv6 transition technologies by calculating the maximum number of subscribers which can be serviced by one public IPv4 address.

Although, still a recent proposal, NPD has been part of the efforts to standardize the proposed methodologies [63]. The methodology has evolved through the reviews of BMWG members. Therefore, it can be considered more mature than maximum sharing ratio, which has not been the subject of external review. More on the methodology dedicated to scalability can be found in Chapter 5.

Operational Capability: shows how a certain technology fits in with the existing environment or how it manages to solve operational problems.

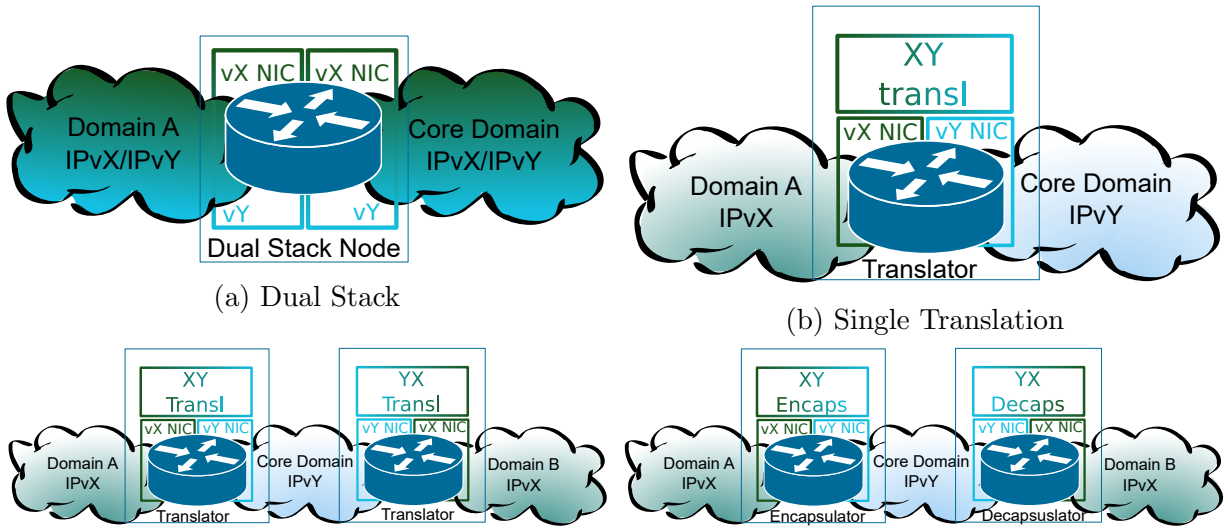
- *configuration capability*: measures how capable a network implementation is in terms of contextual configuration or reconfiguration.
- *troubleshooting capability*: measures how capable a network implementation is at isolating and identifying faults.
- *applications capability*: measures how capable a device is at ensuring compatibility with common user-side protocols.

Operational capability has received external review. However, this was just from the academic community. Since this is another possible standardization, we intend to discuss it in conformance-oriented programs, such as the IPv6 ready consortium. For now, we can consider this methodology not-yet-mature. The detailed methodology for operational capability is presented in Section 4.3.

Security: should quantify the security implications of employing IPv6 transition technologies.

- *STRIDE threat model*: represents the first step towards a security quantification method for IPv6 transition technologies. It is used to identify the threats associated with using IPv6 transition technologies.

Although, we have made efforts to standardize the proposed STRIDE threat model[64], a security quantification method per se was not approached yet. This makes it the least mature of the four proposed methodologies. More about the proposed threat model can be found in Section 6.2.



(a) Dual Stack

(b) Single Translation

(c) Double Translation

(d) Encapsulation

Figure 3.3: IPv6 Transition Technologies Generic Templates

3.3 Network templates

Considering the generic categories defined in Section 2.3.1, we have depicted in Figure 3.3 four generic templates covering the basic use cases and subcomponents of IPv6 transition technologies.

For dual stack IPv6 transition technologies (Figure 3.3a), the basic template would need at least one dual stack node which would act as edge router between Domain A and the Core Domain. The node would integrate at least one Network Interface Card (NIC), which should implement both IP stacks.

As the basic template, single translation technologies (Figure 3.3b) employ one edge router acting as translater between two different version domain: Domain A (IPvX specific) and the Core domain (IPvY specific). As subcomponents, the translator would need to integrate two NICs associated with the two different IP stacks and one virtual interface to handle the XY translation process.

In order to achieve the heterogeneous traversal of the Core domain (IPvY specific), double translation technologies would need at least two edge nodes, one to achieve the XY translation and the other to manage the YX translation. Both nodes would employ as subcomponents two IP specific NICs and one virtual translation interface.

Similar to double translation, encapsulation technologies require two network nodes. One is encapsulating IPvX datagrams into IPvY datagrams, while the other handles the decapsulation process. As subcomponents, each node requires two IP specific NICs and a virtual interface for the encapsulation/decapsulation process.

The network templates described in Figure 3.3 can be the starting point for building test setups for the evaluation of IPv6 transition technologies. For instance, the implementations used for the study case cover four types of IPv6 transition technologies. Two of them

are double translation technologies: MAP-T[26] and 464XLAT[25]. The other two are encapsulation technologies: MAP-E[22] and DSLite[16].

Based on the abstracted templates of double translation and encapsulation, we can deduce that the test setups need at least two devices: an Edge Node (EN) which encapsulates/translates the IPv4 packets in IPv6 packets, and an Edge Node (EN) to handle the decapsulation/translation from IPv6 back to IPv4. The IPv4-only backbone is used for forwarding the IPv4 traffic. The IPv6 traffic would be directly forwarded by the IPv6 backbone. The resulting template is presented in Figure 3.4.

3.4 Network environment

For a comprehensive evaluation of IPv6 transition technologies, we are targeting both closed and open network environments. On one hand, the isolated network environments are suitable for a fine-grain performance analysis. On the other hand, the production network environments are necessary for a better analysis of operational characteristics.

3.4.1 Closed Network Environment

The closed experimental setup presented in Figure 3.4, follows the basic network templates defined for double translation and encapsulation technologies. It includes two edge nodes, which in the context of scalability benchmarking we are calling 1×1 test template.

For the underlying infrastructure of the closed experiments we have used StarBED [65], a large scale general purpose network testbed, administered by the National Institute of Information and Communications Technology (NICT) of Japan. Four computers were used for the 1×1 scale: two for the devices under test (DUT), EN1 and EN2, and two for the benchmarking platform. The benchmarking platform computers have used Ubuntu 12.04.3 server as base operating system.

The traffic was generated using the Distributed Internet Traffic Generator (D-ITG)[66]. One of the computers performed the ITGSend function, generating the traffic, while the other ran the ITGRecv function, receiving the generated traffic and redirecting it back to the ITGSend machine. The ITGSender was also responsible for reporting the network performance of the traffic flow.

In order to test the scalability of the transition tuples in the context of topology growth, we have considered two more topology steps: 10×1 and 30×1. The larger number is the number of client nodes connected to the same server node. As an example, the 10×1 setup is presented in Figure 3.5.

One of the parameters that can limit the scale of the experimental environment is the mapping rule employed with MAP-E or MAP-T. For the MAP-E/MAP-T tests we have used the following mapping rule. The mapping rule uses the IP address blocks reserved by IANA for this purpose [67].

- IPv4 prefix: 198.18.1.0/24
- IPv6 prefix: 2001:2::/48

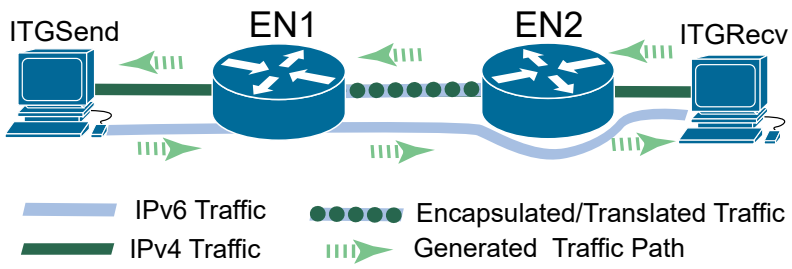


Figure 3.4: 1×1 Test Template

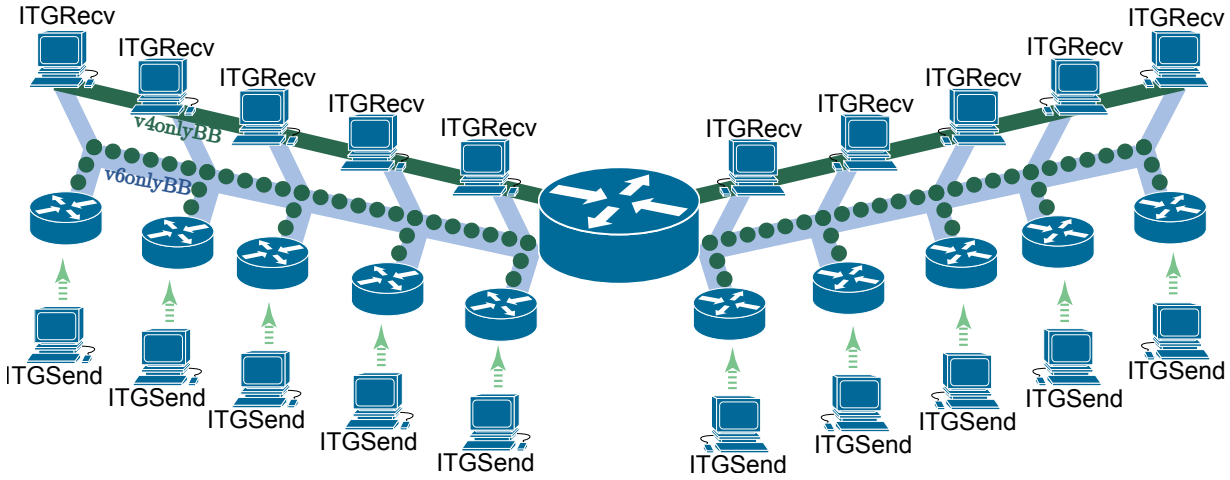


Figure 3.5: 10×1 Test Template

- Embedded Address (EA) bits: 16

From the mapping rule the following limitations can be calculated :

Used IPv4 addresses: 256
This can be derived from the IPv4 prefix's lenghth: $2^{32-[IPv4\ prefix\ length]} = 2^8$.

IPv4 address sharing ratio 1 to 256 Each public IPv4 address is shared between 256 subscriber machines. This is given by the Port Set ID (PSID) length, calculated as: *[port-bits] - [a-reserved-system-port-bits] - [m-contiguous-port-range-bits].*
For the reserved ports [22]recommends $a = 6$.
Also, the $PSID$ cannot be larger than *[port-bits]-[IPv4-suffix]* $\rightarrow 16 - 8 = 8$
In this context, we deduce *a=6* and *m=2*.

Port sharing ratio: 1 to 256
Each subscriber disposes of 252 ports split in 63 ranges of four ports each. This is calculated as: $(2^a - 1) * 2^m = (2^6 - 1) * 2^2 = 63*4$

Maximum number of supported users: 65536
Calculated from the EA bits: $2^{[EA\ bits]} = 2^{16}$

In the context of the isolated experimental environment, these limitations can be adjusted by changing the mapping rule to accommodate the needed network scale. For the

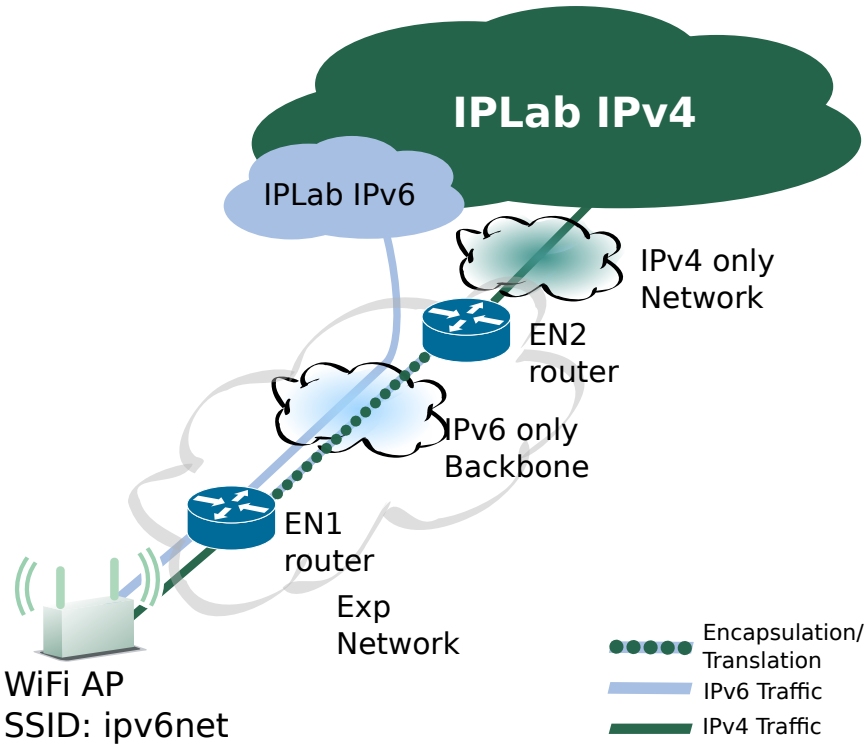


Figure 3.6: Open Network Environment

targeted scales 10×1 and 30×1, this rule was more than sufficient.

In large network testbeds the network flows can be isolated using Virtual LAN (VLAN) technology. Although it is not mandatory, we have considered a separate VLAN for each of the client edge machines. This should help reduce the probability of background traffic affecting the experimental results. In this context, the available number of VLANs in the underlying infrastructure can limit the experimental scale as well. As standardized by [68], the theoretical maximum of usable VLANs is 4096, but in practice at least 2 are reserved. As this is not a requirement, in the event of an insufficient number of available VLANs, the available ones should be reassigned accordingly.

3.4.2 Open Network Environment

The open experiment topology, presented in Figure 3.6 also follows the basic network templates described in Figure 3.3. The major difference is that the benchmarking platform is replaced by open up-link and down-link connections. We have built this type of environment as part of a bigger experimental network, which supplies Internet access to the members of the Internet Engineering Laboratory.

The network consisted of two virtual machines, one for each edge nodes. The two machines have ran on a virtual environment running Citrix Barebone XenServer 6.0 as hypervisor. Previous experiences with building and analyzing a similar open environment are presented in [69]. On the up-link, the IPv4 and IPv6 traffic was routed by a dual-stack

core router. The survey participants were able to connect to the environments through a single SSID, *ipv6net*, handled by a WiFi access point.

3.4.3 Environment Considerations

This subsection discusses some of the software and hardware requirements that need to be considered in order to ensure the accuracy and repeatability of the evaluation data.

Closed Network Environment

For the closed network environment, we recommend considering the following aspects.

Hardware: The hardware characteristics of the tester should allow it to have better observable performance than the device under test (DUT). In other words, the tester should make sure it is not the performance bottleneck of the system. To verify that, we have proposed the use of a direct connection (DC) throughput test, in which the tester sends traffic data back to itself, or the sender device sends data directly to the receiving device. A simple representation of that is presented in Figure 3.7. The test is then rerun with one or multiple DUTs. If the minimum throughput result of the tester is not 10% higher than the maximum result for the DUT, a tester with better hardware characteristics needs to be employed. As a reference, the percent decrease should be calculated with the relative change formula 5.2. This only applies if the measured performance is less than the theoretically possible maximum rate for the respective media. Otherwise the capacity of the media should be increased, as it is the bottleneck of the system.

Software: The accuracy of the traffic generator should be considered to make sure that the tester is not the one generating unstable results. The DC test mentioned can be used as an indicator here as well. A throughput test should be repeated at least 20 times and the variation should be quantified. The relative standard deviation should be calculated. The formula in 3.1 can be used as a reference. The result for the 20 measurements should be lower than 10% .

$$\begin{aligned} &\%rsd = \tfrac{\sigma}{\bar{x}} \times 100 \\ &\sigma - standard\ deviation, \bar{x} - mean \end{aligned} \tag{3.1}$$

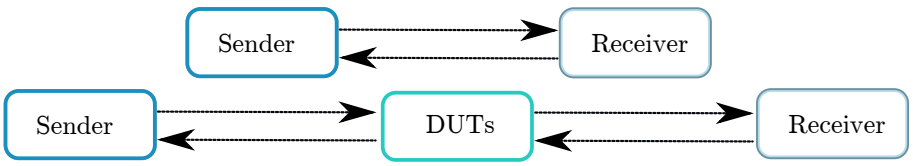


Figure 3.7: Direct connection setup

Open Network Environment

In the context of open environments and operational testing, the following requirements should be met.

Hardware: To make sure that the hardware has no impact on the success rate of the tasks, multiple machines should be employed. Ideally, the machines should have different hardware characteristics. As for the uplink and downlink connections, the network path should be monitored for failures which may affect the success rate of the task. A simple *traceroute*-like application can be employed for this purpose.

Software: In order to isolate the impact of the platform used to run the operational capability tests, multiple platforms should be used. Preferably, at least one mobile platform should be considered in the evaluation process. Moreover, the software used for the applications capability tests should be compliant with the latest RFC requirements.

Penetration Testing Environment

For the penetration test environment, the following aspects should be taken into account.

Hardware: In order to validate the security threats, the penetration testbed hardware should have little impact in the process. To ensure that, multiple machines with different hardware characteristics should be employed. Since this is not a matter of performance, virtual machines are ideal candidates for the task.

Software: The level of compliance of the software used for the system under attack can impact the validity of the penetration test. To that end, the system under attack should implement the latest RFC requirements for the tested protocols.

3.5 Transition Tuples

Given that one transition implementation can cover multiple transition technologies, we are using the term *transition tuple* to describe the set of one transition technology and one transition implementation. We have used for our study case two open source transition implementations: Asamap [28] and Tiny-map-e[29]. On one hand, Asmap covers four transition technologies: MAP-E [22], MAP-T[26], DSLite[16] and 464XLAT[25]. On the other hand, Tiny-map-e includes only MAP-E. The resulting transition tuples are presented in 3.1, which also includes the host Operating System (OS) details.

Table 3.1: Analyzed Transition Tuples

Tuple	Transition technology	Transition implementation	OS
amape	MAP-E	Asamap	Vyatta
amapt	MAP-T	Asamap	Vyatta
amapdslite	DSLite	Asamap	Vyatta
amap464xlat	464XLAT	Asamap	Vyatta
tinymape	MAP-E	Tiny-map-e	Ubuntu server

In terms of implementation details, we would add that Asamap supports a type of fragmentation defined in [70] as inner-fragmentation. This proved to be an interesting implementation parameter. The impact of implementing this type of fragmentation is presented in Section 5.4.3.

Chapter 4

Network Performance and Operational Capability

> *When you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be.*
>
> William Thomson

The service quality of a network performance is critical characteristic of modern computer networks. The overhead created by the IPv6 transition technologies can potentially affect the quality of service in heterogeneous network environments. Expressing that impact with a number facilitates the further understanding of the problem and subsequent improvements. Similarly, quantifying the operational overhead on the network support teams can complement the fine grain performance scores.

4.1 Benchmarking Network Performance

The network performance methodology follows a black-box style performance analysis. In other words, the two tested implementations have been treated as black-boxes, with none of the source code being instrumented to improve the performance. The methodology is, at its core, empirical and consists of a series of benchmarking tests. The performance analysis framework was inspired by the generic guidelines presented by Jain in [71]. The steps we deemed necessary for benchmarking network performance are the following.

Define the goal of the study and the system boundaries:
The main goal of the study was to validate the proposed methodology by measuring

the network performance of the five IPv6 transition tuples presented in Table 3.1. The evaluation study was conducted so that outside components had a minimum effect on the outcome of the evaluation.

Choose the metrics:

We have used as network performance metrics: round-trip delay, packet delay variation and throughput. Both latency metrics were measured in time with sufficiently fine units to distinguish the difference between two events. Considering current network speeds, the proposed measurement unit is in millisecond (ms). The proposed unit of measurement for throughput is in Megabit per second (Mbps).

Define the system parameters and factors:

The parameters that affect the network performance of the system are the software and hardware characteristics of the environmental setup, the **workload traffic**, the **IP version**, the **upper layer protocols**, the **IPv6 transition technology**, the **IPv6 transition implementation** and **the topology scale**. From these parameters, we have considered as factors the ones marked in bold font. In other words, we have maintained constant only the software and hardware characteristics of the environmental setup.

Decide the experimental design:

A full factorial design was employed, hence $F1 \times F2 \times F3 \times F4 \times F5 \times F6 = N$ experiments were conducted. $F1, F2, F3, F4, F5, F6$ represent the values of each of the above mentioned factors. As suggested in RFC2544 [62], the duration of each experiment was 60 seconds. Results for shorter durations proved to be very inconsistent.

Define the evaluation technique:

Since our goal is to benchmark working implementations, a series of empirical measurements were employed for the evaluation. To ensure the consistency of the results, we have repeated each test instance 20 times.

Describe the workload:

For the performed benchmarking tests, the experimental workload was the amount of traffic inserted into the experimental network. In order to avoid exceeding the capacity of the underlying media, the workloads have to consider the frame size overhead introduced by either translation or encapsulation.

The encapsulation method used by MAP-E and DSLite produces a 40 bytes frame overhead, while the translation algorithm performed by MAP-T and 464XLAT creates a 20 bytes overhead. The 40 bytes overhead is created by adding the IPv6 header though the encapsulation process. The 20 bytes overhead result from the difference in size between the IPv6 and IPv4 headers.

Formula 4.1 shows how to calculate the maximum frame rates while accounting for the frame size.

Table 4.1: Maximum Frame Rates

Frame size	Frame rate MAP-T, 464XLAT	Frame rate MAP-E, DSLite
64	1201923	1008065
128	744048	664894
256	422297	395570
512	226449	218531
1024	117481	115314
1280	94697	93284
1518	80231	79214
1522	80026	79014
2048	59866	59298
4096	30222	30077
8192	15185	15148
9216	13505	13476

$$MAX_{FR} = \frac{LineRate(bps)}{(8bits/byte)*(\mathbf{X}+\mathbf{O}+20)bytes/frame}$$
Where:
X - the frame size in bytes (4.1)
O - the overhead in bytes
20 bytes = 8 bytes (preamble) + 12 bytes (inter-frame gap)

For example, for 40 bytes overhead, 1Gbs Ethernet and 64byte frames, the result is the following.

$$\frac{1{,}000{,}000{,}000(bps)}{(8bits/byte)*(\mathbf{64}+\mathbf{40}+20)bytes/frame} = 1,201,923\ fps \quad (4.2)$$

Continuing the calculation, the resulting maximum frame rates for the used media (1Gbps) are presented in Table 4.1.

The traffic was generated using the Distributed Internet Traffic Generator (D-ITG)[66], respecting the characteristics described in 4.1. For the Transport Layer protocol we chose to use UDP traffic because it represents a more reasonable benchmarking base.

Describe the test setup:

Figure 3.4 presents the proposed test setup. The setup follows the recommendations of RFC5180 with a bi-directional traffic exchange between a sender and a receiver element. The devices under test (DUTs) acted as forwarding components.

Choose the data summarizing and variation functions:

To account for the repeatability of the tests, 20 test iterations were performed for each measurement. In order to compile the 20 different empirical results into one single score, we followed the recommendations of [71] and decided between mode, mean and median. According to [71], the simple flow chart shown in Figure 4.1 can be used to decide the most suitable function.

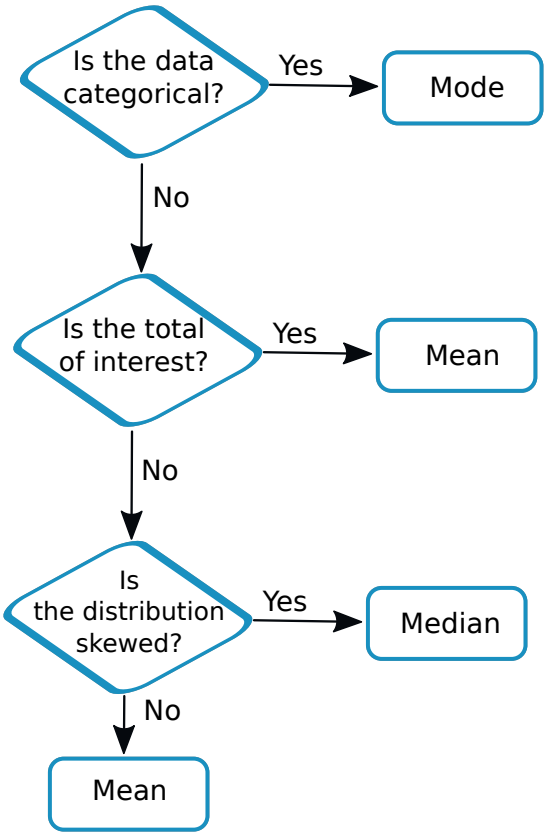
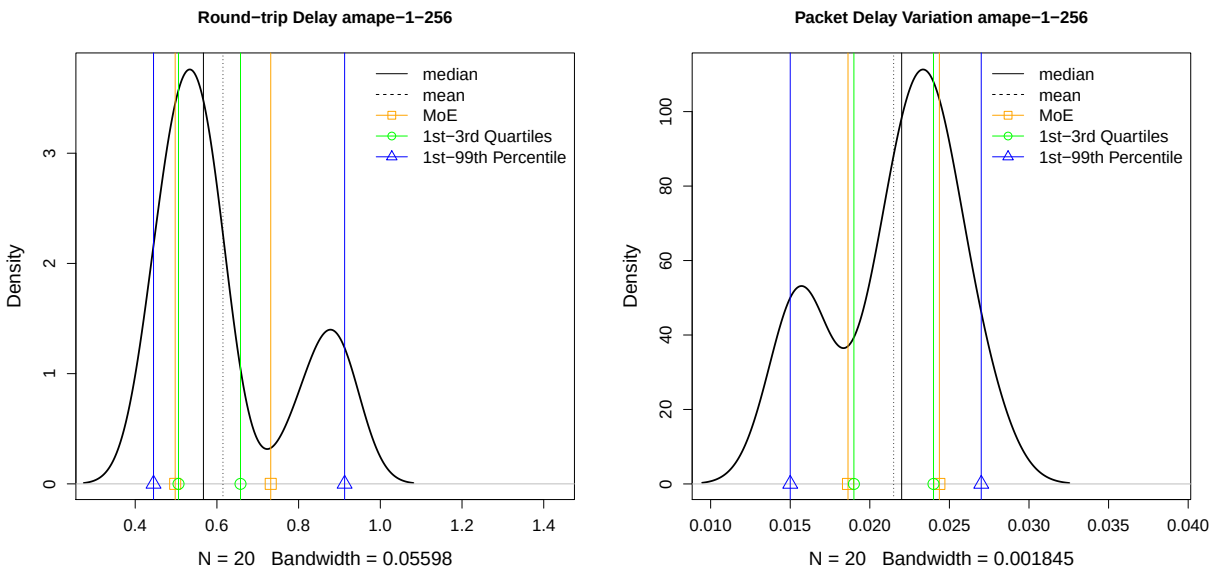


Figure 4.1: Mean, mode and Median flow chart

Considering we did not have any type of categorical data, mode was out of question. Consequently, we decided between mean and median. Since the total of the observations was not of interest, we had to analyze the probability distribution of the data. Subsequently, the distribution indicated the most appropriate of the two. The function to account for the variation in the dataset was decided considering the data distribution as well. In order to summarize across the 12 tested frame size, we needed to summarize once more. Following the same rationale, the distribution of the data was analyzed and the most appropriate function was chosen.

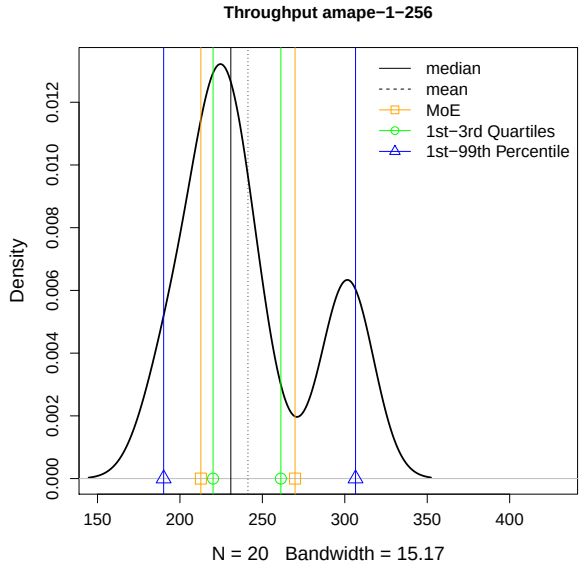
Define the data presentation:

For a fine grain analysis, the final results were presented as a function of frame size in a table. The table included the summarized values as well as the variation of the data. For better visualization of the performance of the tuples, the results were plotted as a function of frame size as well. For an overview of the results across the 12 tested frame sizes, the associated data was compiled in a table according to the central-tendency function.



(a) Round-trip Delay

(b) Packet Delay Variation



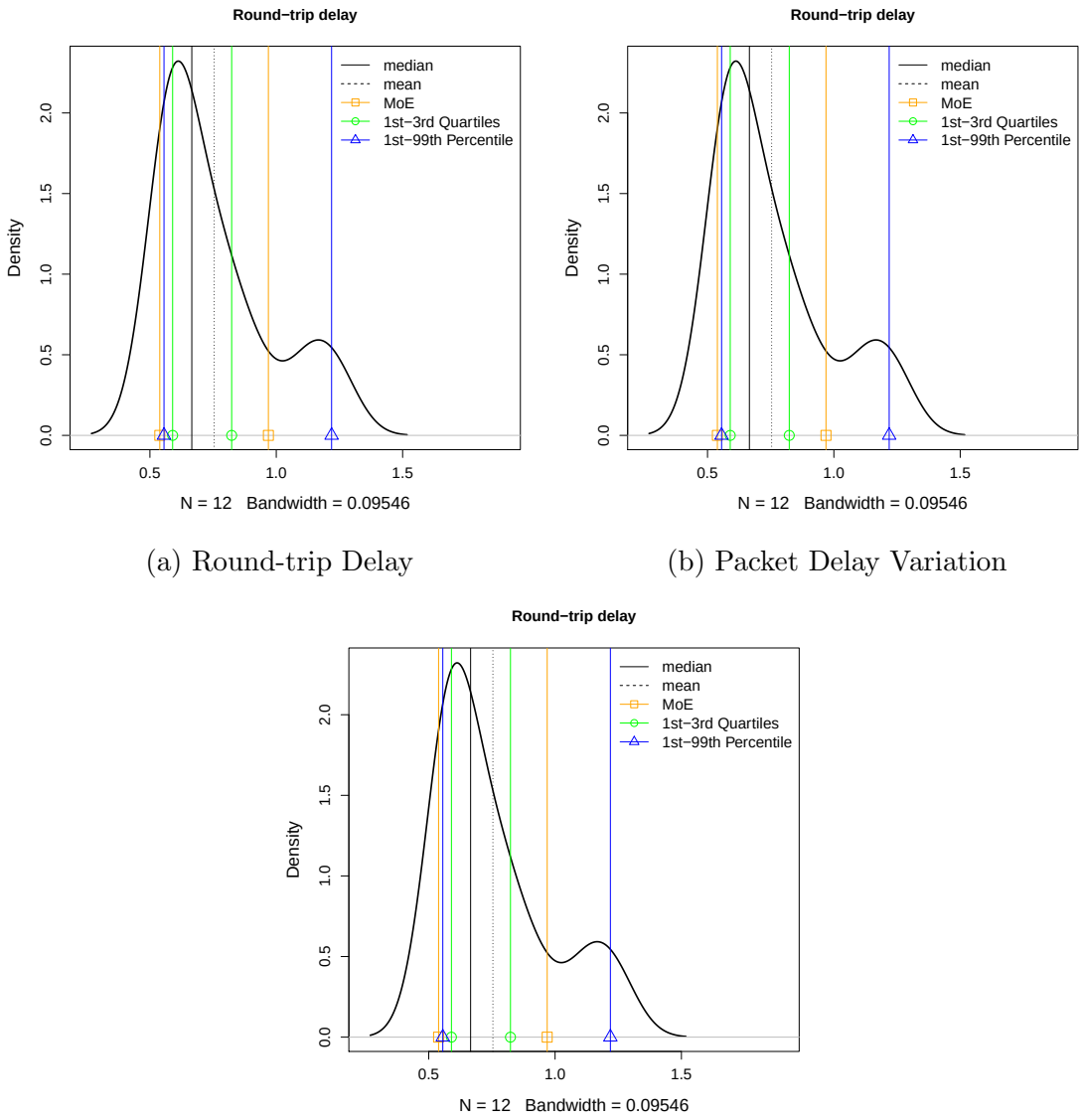
(c) Throughput

Figure 4.2: Probability density functions for the 20 repetitions

4.2 Empirical Network Performance Data

4.2.1 Summarizing and Variation

Figure 4.2 presents the probability distribution of the round-trip delay, packet delay variation for the amape tuple at 256 bytes frames. The density function is calculated with respect to the kernel density estimation guidelines presented in [72].



(a) Round-trip Delay

(b) Packet Delay Variation

(c) Throughput

Figure 4.3: Probability density functions for the 12 frame sizes

The distribution for all three metrics is skewed and has bimodal tendencies. In terms of summarization, the median proved to be more representative than the mean. As for variation, the 1st/99th percentiles captured the most important part of the distribution. The plots in Figure 4.2 revealed as well that the 1st/99th percentiles are more representative than the margin of error (at 99% level of confidence) and the 1st/3rd quartiles. Moreover, across the 12 frame sizes as shown in Figure 4.3, the distribution had an overall skewness as well. The rest of the datasets contain similarly distributed results. Consequently, the median was used for summarizing the data and the 1st/99th percentiles as measures of variation.

Table 4.2: Round-trip Delay Summarized Data

		64	128	256	512	1,024	1,280	1,518	1,522	2,048	4,096	8,192	9,216
amape	**RTD**	**0.562**	**0.555**	**0.567**	**0.640**	**0.598**	**0.605**	**0.793**	**0.773**	**0.692**	**0.915**	**1.117**	**1.232**
	1st	0.437	0.448	0.445	0.509	0.537	0.535	0.575	0.655	0.665	0.890	1.103	1.202
	99th	0.655	0.729	0.913	0.635	0.707	0.645	0.811	0.801	0.714	0.937	1.172	1.264
amapt	**RTD**	**0.511**	**0.538**	**0.578**	**0.685**	**0.516**	**0.566**	**0.658**	**0.645**	**0.618**	**0.888**	**1.141**	**1.203**
	1st	0.451	0.369	0.404	0.417	0.494	0.537	0.626	0.623	0.592	0.865	1.120	1.164
	99th	0.613	0.852	0.900	0.944	0.545	0.592	0.678	0.668	0.648	0.918	1.156	1.234
amapdslite	**RTD**	**0.788**	**0.729**	**0.756**	**0.735**	**0.679**	**0.707**	**0.799**	**0.798**	**0.703**	**0.913**	**1.114**	**1.251**
	1st	0.706	0.617	0.656	0.686	0.638	0.633	0.712	0.700	0.586	0.871	1.097	1.208
	99th	0.877	0.928	0.932	0.956	0.836	0.838	0.825	0.831	0.718	0.936	1.195	1.316
amap464xlat	**RTD**	**0.603**	**0.554**	**0.530**	**0.420**	**0.525**	**0.566**	**0.678**	**0.699**	**0.705**	**0.902**	**0.876**	**0.876**
	1st	0.474	0.416	0.438	0.375	0.437	0.466	0.563	0.546	0.677	0.876	0.876	0.876
	99th	0.779	0.799	0.781	0.549	0.604	0.691	0.716	0.733	0.735	0.934	0.876	0.876
tinymape	**RTD**	**1.288**	**1.229**	**1.156**	**1.135**	**1.179**	**1.357**	**1.299**	**1.298**	**1.403**	**1.613**	**1.814**	**1.951**
	1st	1.103	1.032	0.976	0.995	1.097	1.256	1.178	1.135	1.224	1.489	1.693	1.778
	99th	1.395	1.349	1.572	1.590	1.356	1.634	1.467	1.896	1.938	2.227	2.176	2.376

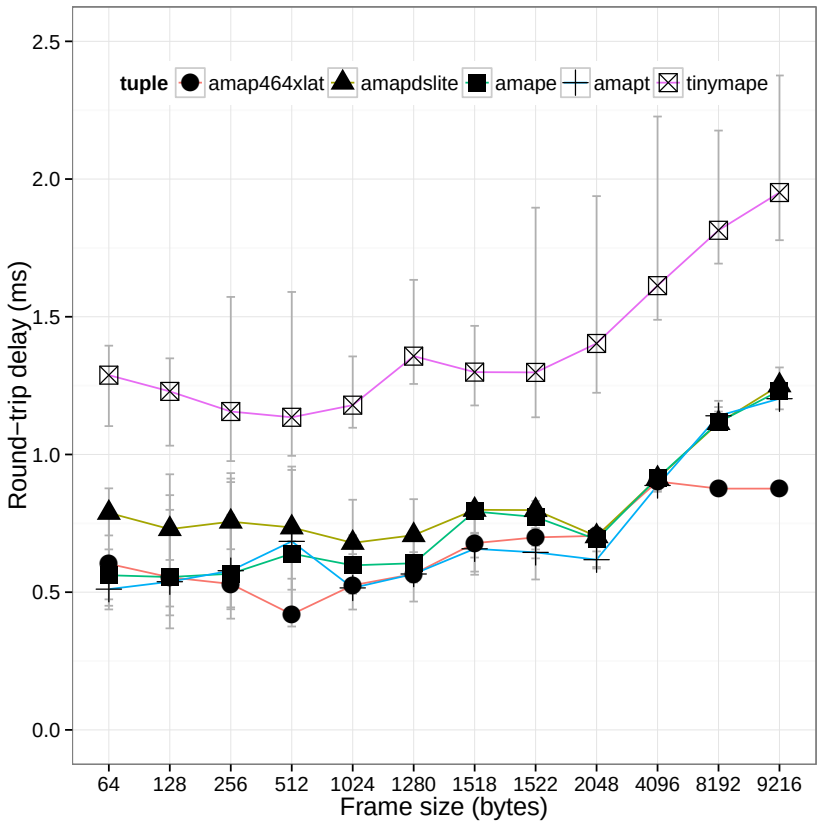


Figure 4.4: Round-trip Delay Comparative Results

4.2.2 Comparative Network Performance Data

The results for the 20 round-trip delay iterations have been summarized in Table 4.2. The table allows us to do a fine grain assessment and comparison of the five tuples in terms of delay. For example, for the 512 frame size we can tell that the amap464xlat tuple

Table 4.3: Packet Delay Variation Summarized Data

		64	128	256	512	1,024	1,280	1,518	1,522	2,048	4,096	8,192	9,216
amape	**PDV**	**0.020**	**0.021**	**0.022**	**0.022**	**0.041**	**0.036**	**0.074**	**0.074**	**0.072**	**0.140**	**0.200**	**0.201**
	1st	0.015	0.016	0.015	0.019	0.031	0.030	0.070	0.069	0.069	0.135	0.192	0.190
	99th	0.028	0.028	0.027	0.027	0.047	0.036	0.075	0.079	0.073	0.148	0.205	0.220
amapt	**PDV**	**0.020**	**0.024**	**0.024**	**0.040**	**0.042**	**0.036**	**0.077**	**0.071**	**0.078**	**0.141**	**0.143**	**0.143**
	1st	0.015	0.015	0.015	0.016	0.029	0.034	0.071	0.065	0.071	0.135	0.143	0.143
	99th	0.025	0.027	0.026	0.049	0.047	0.037	0.079	0.081	0.082	0.145	0.143	0.143
amapdslite	**PDV**	**0.028**	**0.023**	**0.023**	**0.023**	**0.058**	**0.069**	**0.082**	**0.084**	**0.102**	**0.163**	**0.205**	**0.228**
	1st	0.019	0.018	0.018	0.018	0.052	0.060	0.078	0.081	0.094	0.156	0.197	0.223
	99th	0.032	0.036	0.033	0.034	0.061	0.081	0.089	0.095	0.115	0.170	0.211	0.238
amap464xlat	**PDV**	**0.018**	**0.019**	**0.019**	**0.018**	**0.034**	**0.042**	**0.071**	**0.073**	**0.077**	**0.139**	**0.200**	**0.197**
	1st	0.015	0.016	0.015	0.015	0.029	0.036	0.066	0.067	0.069	0.133	0.192	0.189
	99th	0.026	0.021	0.029	0.025	0.046	0.045	0.077	0.077	0.082	0.149	0.208	0.211
tinymape	**PDV**	**0.058**	**0.126**	**0.155**	**0.204**	**0.355**	**0.401**	**0.445**	**0.437**	**0.528**	**0.859**	**1.835**	**2.027**
	1st	0.044	0.118	0.148	0.186	0.267	0.378	0.397	0.397	0.483	0.769	1.395	1.671
	99th	0.073	0.147	0.179	0.211	0.397	0.490	0.506	0.510	0.593	0.985	1.987	2.138

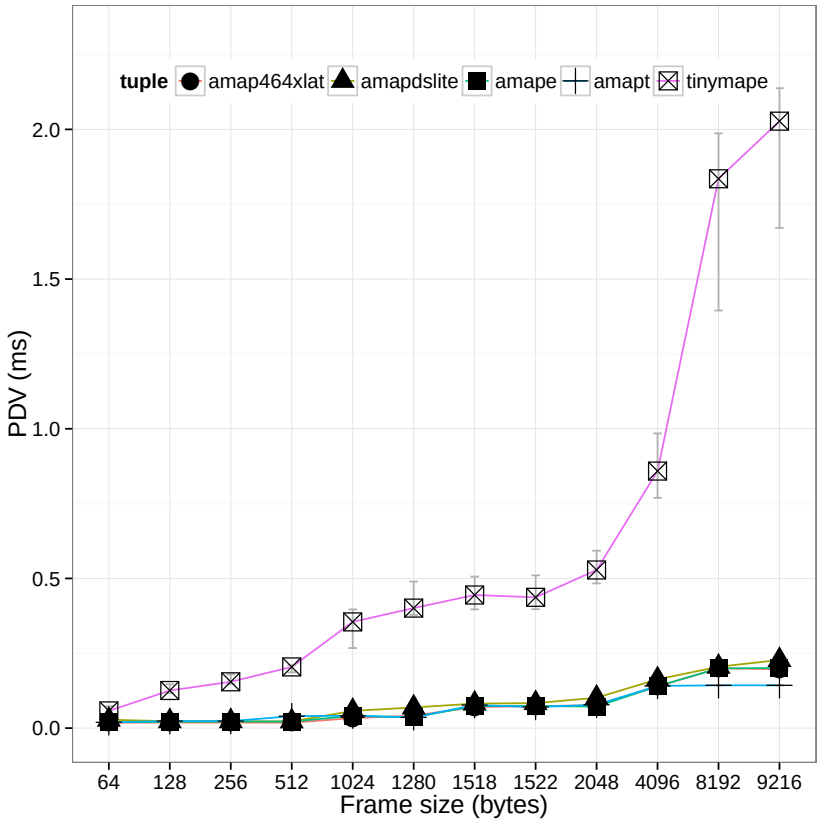


Figure 4.5: Packet Delay Variation Comparative Results

had better performance than amape, amapdslite or tinymape. The summarized score for amap464xlat was 0.420 ms with a variation of [0.375, 0.549]. However, we cannot be sure that amap464xlat had a better delay than amapt since the variation for this tuple was [0.404, 0.900], which overlaps with the variation of amap464xlat.

Table 4.4: Throughput Summarized Data

		64	128	256	512	1,024	1,280	1,518	1,522	2,048	4,096	8,192	9,216
amape	**RTD**	**62.4**	**119.8**	**230.9**	**459.6**	**530.9**	**754.7**	**438.3**	**436.3**	**569.6**	**554.6**	**728.8**	**807.9**
	1st	47.4	95.2	190.1	396.0	461.1	683.7	428.9	411.9	558.9	535.2	720.8	775.4
	99th	77.6	147.6	306.5	526.2	688.5	865.1	456.6	456.8	585.2	570.0	735.1	837.0
amapt	**RTD**	**48.4**	**116.7**	**231.9**	**473.2**	**379.3**	**368.2**	**399.3**	**395.6**	**406.6**	**476.1**	**712.8**	**731.9**
	1st	43.9	77.6	161.5	319.8	363.7	310.0	372.2	356.9	357.8	455.8	705.2	720.7
	99th	67.0	135.8	268.8	540.9	421.0	415.9	411.1	405.4	436.6	488.7	724.3	738.7
amapdslite	**RTD**	**125.6**	**149.6**	**281.0**	**553.4**	**638.9**	**590.9**	**459.1**	**448.2**	**546.6**	**559.3**	**728.4**	**816.1**
	1st	93.4	135.6	195.3	428.8	468.7	563.2	426.7	419.2	520.5	531.9	719.8	793.5
	99th	143.0	170.4	325.7	634.0	743.7	715.0	487.8	488.2	589.5	574.8	736.4	838.5
amap464xlat	**RTD**	**117.1**	**124.1**	**231.1**	**282.4**	**531.4**	**682.5**	**428.2**	**466.2**	**541.1**	**555.1**	**557.1**	**556.7**
	1st	95.4	112.3	211.0	237.1	459.7	676.3	420.7	409.6	513.6	546.7	557.1	556.7
	99th	144.7	174.4	323.6	602.5	740.7	797.9	460.6	497.8	580.8	570.9	557.1	556.7
tinymape	**RTD**	**27.5**	**43.8**	**87.4**	**144.0**	**171.6**	**191.8**	**203.1**	**206.4**	**222.3**	**265.9**	**286.4**	**290.9**
	1st	25.8	41.3	79.4	133.0	158.5	177.1	187.6	190.6	205.4	245.6	264.5	268.7
	99th	32.3	49.8	93.0	160.5	187.9	210.0	242.7	238.3	247.9	299.1	350.7	357.2

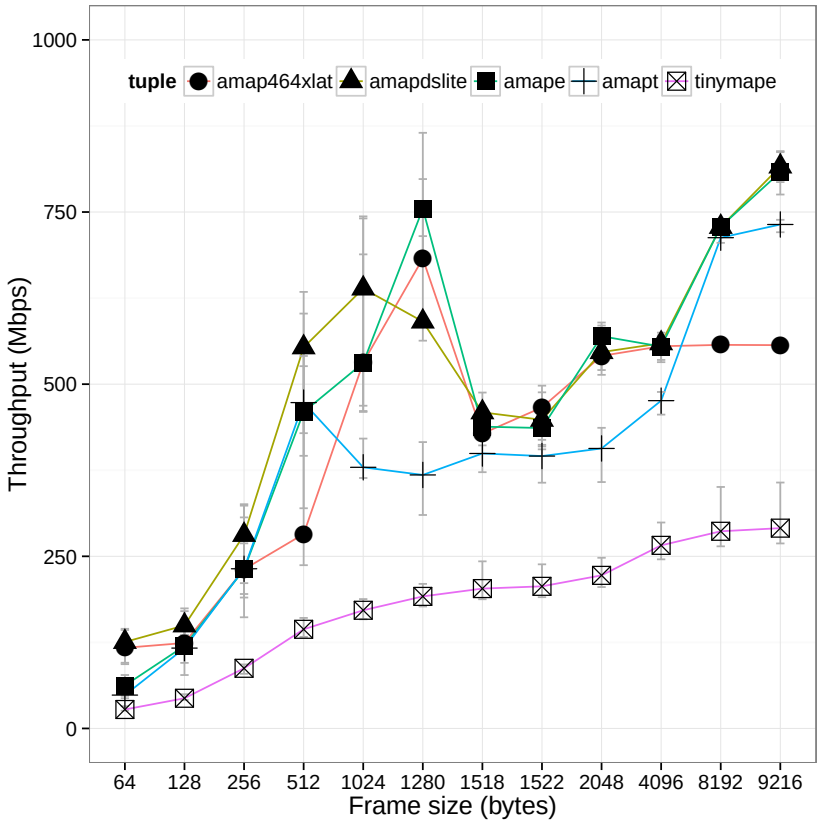


Figure 4.6: Throughput Comparative Results

Figure 4.4 shows a graphical interpretation of Table 4.2. While it is easier to notice from the plot that amap464xlat had a performance peak, it is harder to do a fine grain analysis of the results. Similarly, the packet delay variation and throughput data have been presented in Table 4.3, Figure 4.5 and Table 4.4, Figure 4.6 respectively.

As an overview, within the Asama tuples an overall conclusion is hard to draw because of the variation of the datasets. However, for some of the frame sizes, previously discovered [73] performance trends still stand: tuples which employ translation (amapt, amap464xlat) have better delay and packet delay variation, while tuples which employ encapsulation (MAP-E, DSLite) have better throughput. In terms of the two implementations, for all metrics the tinymape tuple had considerably worse performance. This also confirmed our previous results, published in [74]. It was a somewhat expected result, since the Tinymap-e implementation [29] is still in development and has not been updated in over two years.

4.2.3 Summarized Network Performance Data

With the risk of over-summarization, an easier performance comparison can be drawn from the data summarized across the 12 frame sizes, presented in Table 4.5.

Table 4.5: Summarized Network Performance Data

	RT Delay (ms)	**PDV (ms)**	**Throughput (Mbps)**
amape	0.640	0.057	495.2
amapt	0.618	0.056	397.4
amapdslite	0.756	0.076	550.0
amap464xlat	0.603	0.056	488.7
tinymape	1.298	0.419	197.4

The performance trends are easier to notice now, with amapt and amap464xlat having better delay and delay variation results. On the other hand, amape and amadslite show better performance in terms of throughput. As for the two implementations, it is clearer now that tinymape has the worst performance, leaving a lot of room for improvement.

4.3 Operational Capability Methodology

We have measured the operational capability of the two implementations in the open environment described in Figure 3.6.

The three proposed metrics: configuration capability, troubleshooting capability and applications capability have been characterized using a task-based non-exhaustive approach. For configuration and troubleshooting capability we have proposed 10 tasks each. These configuration and troubleshooting tasks were designed to verify the existence of basic configuration and troubleshooting means, which should not be missing from any such implementation.

The proposed method for measurement was assisted survey, in which the participants were asked to confirm the level of success of each task. This method isolated the human factor, and potential usability problems as much as possible. The score was calculated as a percentage of successful tasks over the total number of tasks (e.g. $7/10 = 70\%$) The data collected was organized in Higher is Better (HB) score tables. In the case of applications capability, we have proposed a list of 20 common user-side applications to be tested in relation with the transition tuples.

For configuration capability, we have considered a number of configuration tasks, inspired by the abstracted guidelines presented in [75]. The tasks can be organized in three generic groups, *initial setup*, *reconfiguration* and *confirmation*. For ease of reference, we have associated each task with a task code, in accordance with the respective group.

1. IinitialSetup1: configure an encapsulation/translation virtual interface using a command line interface or a graphical user interface
2. IinitialSetup2: Save the current temporary configuration commands in a file which can be loaded at start-up
3. IinitialSetup3: Self configuration according to contextual configuration details
4. InitialSetup4: Display warnings in the case of misconfiguration and reject the misconfigured command
5. InitialSetup5: Display warnings in the case of missing command and reject saving the temporary configuration
6. InitialSetup6: Display contextual configuration commands help
7. Reconfiguration1: Convert current configuration settings to configuration commands
8. Reconfiguration2: Back-up and restore the current configuration
9. Confirmation1: Show the current configuration
10. Confirmation2: Show abstracted details for the 464 virtual interface

The configuration capability result were expressed as a percentage of successfully completed configuration tasks, from the total number of tasks.

Similarly, for troubleshooting capability, we have proposed a number of troubleshooting tasks. The tasks follow the fault isolation, fault determination and root cause analysis (RCA) guidelines presented in [75]. Consequently, the tasks can be organized into the three generic categories: *fault isolation*, *fault determination* and *root cause analysis (RCA)*. For

ease of reference, these tasks were associated as well with group codes.

1. FaultIsolation1: Capture and analyze IPv4 and IPv6 packets
2. FaultIsolation2: Send and receive contextual ICMP messages
3. FaultDetermination1: Identify a misconfigured contextual route
4. FaultDetermination2: Identify a misconfigured contextual line in the virtual 464 interface configuration
5. FaultDetermination3: Perform self-check troubleshooting sequence
6. RCA1: Log warning and error messages
7. RCA2: Display log
8. RCA3: Display in the user console the critical messages with contextual details
9. RCA4: Log statistical network interface information
10. RCA5: Display detailed statistical network interface information

The troubleshooting capability result were also expressed as a percentage of successful tasks of the total number of troubleshooting tasks.

Inspired by the efforts presented in [42], we are testing a non-exhaustive list of common user applications in relation with the 464 transition technologies in order to measure applications capability. The applications are organized into the following categories: browsing, E-mail, Instant Messaging (IM) and Voice over IP (VoIP), Virtual Private Networks (VPN), File Transfer Protocol (FTP), Cloud and Troubleshooting. For now the list includes 20 popular applications, which are presented in Table 4.7. The measurement result is presented as a percentage of successfully-tested applications from the total number of applications.

4.4 Operational Capability Results

The assisted survey results for configuration and troubleshooting capability have been summarized in Table 4.6. Since the operational tasks are implementation-oriented, the results have been organized according to the transition implementation.

Regarding the configuration capability, most of the tasks have been completed successfully for the asamap implementation. However, a self-configuration setup sequence is not yet available for the asamap implementation. Given the complexity of the transition technologies, a guided self-configuring setup would be a beneficial feature. Regarding the tinymape implementation, 3 other tasks have failed. This can be explained by the *in-development* status of the implementation. In the case of asamap, most of the the troubleshooting tasks have been completed successfully. Two of the troubleshooting tasks could not be completed: FaultDetermination3 (Displaying critical messages with associated details) and RCA3 (self-check sequence). Regarding the first one, some critical messages were displayed in the user console. However, these are hard to interpret and understand. We believe this feature needs improvement. As for the second failed task, a self-check sequence is not available yet. This would represent a substantial improvement of the troubleshoot-

Table 4.6: Configuration and Troubleshooting Capability Results

Operational Capability		Asamap	Tiny-map-e
Configuration Capability	IinitialSetup1	Pass	Pass
	IinitialSetup2	Pass	Pass
	IinitialSetup3	Fail	Fail
	IinitialSetup4	Pass	Pass
	IinitialSetup5	Pass	Pass
	InitialSetup6	Pass	Fail
	Reconfiguration1	Pass	Fail
	Reconfiguration2	Pass	Pass
	Confirmation1	Pass	Fail
	Confirmation2	Pass	Fail
Configuration capability result		9/10 = 90%	5/10 = 50%
Troubleshooting Capability	FaultIsolation1	Pass	Pass
	FaultIsolation2	Pass	Pass
	FaultDetermination1	Pass	Pass
	FaultDetermination2	Pass	Fail
	FaultDetermination3	Fail	Fail
	RCA1	Pass	Pass
	RCA2	Pass	Pass
	RCA3	Fail	Fail
	RCA4	Pass	Pass
	RCA5	Pass	Pass
Troubleshooting capability result		8/10 = 80%	7/10 = 70%

ing capability. For tinymape RCA3, FaultDetermination2 and FaultDetermination3 failed, one more than asamap. This confirms the lower operational capability of tinymape and makes asamap the first choice from this standpoint.

In terms of applications capability, we tested a non-exhaustive list of common applications, in accordance with [42]. The full list of applications and the results are presented in Table 4.7. The applications were tested using two machines, one running Windows 7 and the other, a mobile device, running Android 4.2. Both devices were connected to the experimental environment through the prepared WiFi SSID: *ipv6net*. To summarize, we did not encounter any application troubles with any of the two implementations.

Table 4.7: Applications Capability Results

Applications			Asamap	Tiny-map-e
Win 7 / Android 4.2	Browsing	Chrome	Pass	Pass
		Firefox	Pass	Pass
		Dolphin	Pass	Pass
	E-mail	Outlook	Pass	Pass
		Thunderbird	Pass	Pass
		Aquamail	Pass	Pass
	IM and VoIP	Skype	Pass	Pass
		Facebook	Pass	Pass
		Google+	Pass	Pass
		VoIP Buster	Pass	Pass
		Viber	Pass	Pass
		DigiOriunde	Pass	Pass
	VPN	Hideman VPN	Pass	Pass
		Spotflux	Pass	Pass
	Cloud	Dropbox	Pass	Pass
		GDrive	Pass	Pass
	FTP	Filezilla	Pass	Pass
	Troubleshooting	puTTY	Pass	Pass
		WinSCP	Pass	Pass
		ConnectBot	Pass	Pass
Applications capability result			20/20 = 100%	20/20 = 100%

4.5 Analysis of the Test System

The following subsections provide a systematic feasibility analysis of the proposed methodologies.

4.5.1 Network Performance

As described in section 3.4.3 the hardware and software characteristics of the testing environment can have an impact on the accuracy of the test results. To that end, we have analyzed the throughput of the direct connection (DC) between the sender and the receiver devices. The results showed a consistently higher throughput for all the tested frame sizes. In terms of repeatability, the relative standard deviation was in average lower than 7%. In conclusion, we considered the testing environment to have had little impact on the outcome of the testing.

In terms of time efficiency, the testing had three important stages: the deployment, the effective testing and post-testing data analysis. The time efficiency of the three phases is presented in the following.

Deployment: The deployment time depends heavily on the skill and experience of the human operator handling the deployment. Consequently, these numbers are solely for

exemplification purposes. For the latest experiment, it took about 1 hour to deploy the 4 machines needed for testing. Another hour was spent on the preparation and troubleshooting of the underlaying StarBed nodes.

Testing time: Each test iteration consumed about 70 seconds (60 seconds of testing and 10 seconds of sleep between iterations). Considering the chosen full-factorial design, for the latest experiments we had 12 ($frame\ sizes$) $\times$ 20 ($repetitions$) $\times$ 1 ($L4\ workload$) $\times$ 5 ($transition\ tuples$) = 1200 iterations. That amounted to 84,000 seconds or ~23 hours.

Post-processing time: For each iteration there was an average of 20 seconds to post-process the test data. Considering the 1200 test iterations mentioned above, the total post processing time was 24,000 seconds or ~7 hours.

4.5.2 Operational Capability

To isolate the hardware and software platforms the operational capability tasks have been tested on multiple platforms: a notebook running Windows 7 and a smartphone running Android 4.2. For configuration capability and troubleshooting capability the tasks were executed over a SSH connection using a terminal application. When testing applications capability, applications dedicated to the two respective platforms were used.

The human factor was isolated by using a survey in which the participant only confirmed the task' success rate. The tasks consumed roughly 2 hours for each of the participants for both tested implementations. That resulted in an effective testing time of 4 hours.

4.6 Summary and Outlook

By using the proposed prosed network performance and operational capability methodologies we were able to indicate and compare the feasibility of the tested IPv6 transition tuples. In terms of network performance, we found that one of the implementation (Asamap) and the associated transition tuples (amape, amapt, amapdslite and amap464xlat) had a better overall performance. Within the Asamap implementation, we were able to identify some performance trends: the translation-based tuples (464XLAT, MAP-T) had a better delay and delay variation. For throughput, the results were in favor of encapsulation-based technologies (MAP-E, DSLite). However, as demonstrated by the Tiny-map-e data, the results are highly dependent on the quality of the used implementation. Consequently, this results should be interpreted in association with the Asamap implementation. The operational capability results indicated as well the Asamap implementation as a better choice. Asamap had better results for both configuration and troubleshooting capability. Considering the network performance data as well, this confirms the in-development status of Tiny-map-e.

As outlook for the network performance methodology, we plan to continue our pursuit of standardization with [63], and expand the number of tested transition tuples. As for operational capability, we have proposed a limited number of tasks for configuration and troubleshooting capability, which test the existence of core features in the subject

implementations. We would like to increase this number in the future, and with it raise the complexity of the survey. In terms of applications capability, we have only targeted 20 common applications. However, this number is far from being realistic. We plan to increase this number, to better fit a realistic scenario. Furthermore, we would like to take into account other operational variables, such as *operator's skill* or *usability of the implementation*. Quantifying these variables should also lead to a more realistic approach of operational capability. In terms of standardization efforts, we intend to propose and discuss the methodology in a conformance testing forum, such as the IPv6 ready consortium.

Chapter 5

Quantifying Scalability

> *Management means measurement, and a failure to measure is a failure to manage,*
>
> Martin L. Abbott

In the development of an IPv6 transition plan, scalability is one of the biggest concerns for network operators, as the topology of production networks is usually dynamic. Consequently, quantifying scalability represented one of the step stones for our research, and has brought our previous efforts one step closer to a complete benchmarking methodology dedicated to IPv6 transition technologies.

5.1 Scalability Dimensions

Among scalability aspects, we believe the most important is load scalability, as it can affect small-scale transition deployments as well as larger ones. To that end, the first priority has been a method for benchmarking the load scalability of IPv6 transition technologies. To achieve that, we have approached scalability from the perspective of the performance degradation it produces. In other words, we are measuring the scalability of transition technologies by analyzing their network performance degradation at higher scales, which generally translates into higher workloads.

Some of the subcomponents of IPv6 transition technologies (e.g. MAP-E [22] or MAP-T[26] mapping algorithm) can impose limits on the potential growth of the network. Consequently, as a complement, we have investigated the structural scalability of these technologies.

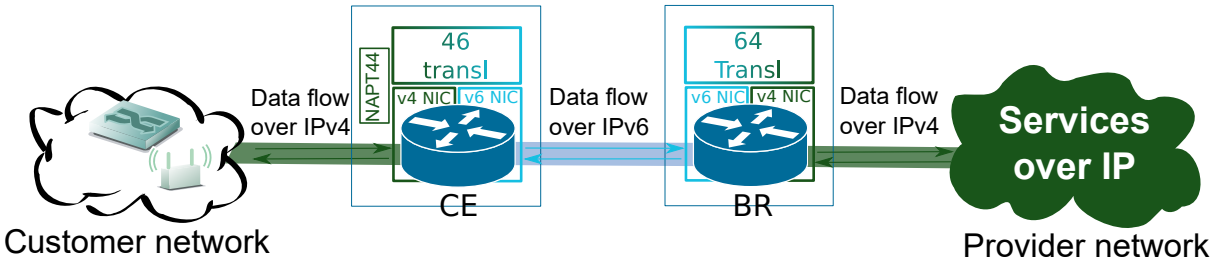


Figure 5.1: MAP-T abstract model

5.2 Benchmarking Network Performance Degradation

As its name suggests, the benchmarking of network performance degradation was largely based on benchmarking the network performance at first. To achieve that, we have followed the benchmarking steps defined in Section 4.1. In order to benchmark network performance degradation, the following complementary steps were needed.

Choosing the metrics

For quantifying network performance, we have used well-established metrics such as round-trip delay, packet delay variation and throughput. Network performance degradation is the metric we have proposed for quantifying load scalability. As measurement procedure, network performance experiments at different network scales need to be conducted. To quantify the load scalability, we have proposed the relative change between the results at the different scales. The proposed unit of measurement is the percentage of relative change for the three network performance metrics. The scores are calculated using the Formula 5.1 for metrics which have a Lower is Better (HB) tendency. Formula 5.2 should be used for metrics with a Higher is Better (HB) tendency.

$$Performance\ degradation(M) = \frac{S_n - S_1}{S_1} \times 100 \quad (5.1)$$

M-specific performance metric (e.g. round-trip delay)

S_n - score obtained at scale n

$$Performance\ degradation(M) = \frac{S_1 - S_n}{S_1} \times 100 \quad (5.2)$$

M-specific performance metric (e.g. throughput)

Define the system parameters and factors

In order to better understand which configurable parameters can have further implications on the scalability, we have built an abstract model of the tested IPv6 transition technologies. As an example, the model for MAP-T is depicted in Figure 5.1

From the subcomponents of MAP-T, both the Customer Edge (CE) node and the Boarder Relay (BR) node include the following elements.

- IPv4 network interface
- IPv6 network interface
- virtual interface which achieves the mapping of the IPv4 and IPv6 prefixes as well as the translation from $4 \rightarrow 6$ or $6 \rightarrow 4$.

One of the common configurable parameters for the three interfaces is the Maximum Transmission Unit (MTU). In order to observe the impact of the packet fragmentation behavior on performance and scalability, we have considered two values: 1280 (the minimum IPv6 MTU) and 1500 (the typical Ethernet MTU).

Moreover, since Asamap also implemented a method called inner-packet fragmentation, proposed in [70], we have tested the two configurable options available: inner-fragmentation enabled or inner-fragmentation disabled. We note that this parameter was not tested on the tinymape tuple, since this is not an available option.

Another configurable parameter which was considered to affect the performance and scalability of the virtual interface, is the number of configured mapping rules. This parameter is particular to the IPv6 transition technologies which employ the mapping with address and port method, namely MAP-E and MAP-T.

5.3 Quantifying Structural Scalability

Considering the finite addressing schemes and protocol header restrictions, another scalability dimension needs to be considered in the context of IPv6 transition technologies, namely structural scalability. The current, structural limit imposed by the IPv6 addressing scheme is a very generous one for today's standards, a total of $2^{128} \approx 3.4 \cdot 10^{38}$ unique IP addresses.

To support the IPv4-only networks, IPv6 transition technologies, typically encompass a method to map private IPv4 to public IPv4 or IPv6 addresses. All four analyzed transition technologies require a form of Network Address Translation. In other words, they need to maintain a state table containing bindings between IP addresses or {IP addresses, transport protocol, port number} tuples. These bindings can be a hindrance for performance, as any concurrent states occupy memory space. In terms of structural scalability, however, considering the 16 bits reserved for the port number, they can be less of a threat than the IPv6 addressing scheme.

Nevertheless, there are instances when the mapping schemes become the structural scalability bottleneck. For MAP-E and MAP-T, the mapped IPv4 prefix and port set is embedded in the IPv6 prefix. This mapping rule can limit the number of CE machines and, by extension, the number of subscribers connected to a MAP domain. Moreover, as discussed in [76] and [77] The A+P method has serious drawbacks in terms of port assignments.

To quantify the scalability limits imposed by this type of mapping rule, we can use the maximum number of subscribers which can be served by a single IPv4 address. The metric can be simply referred as *maximum sharing ratio*.

If the number of available public IPv4 addresses is limited to 1, the sharing ratio

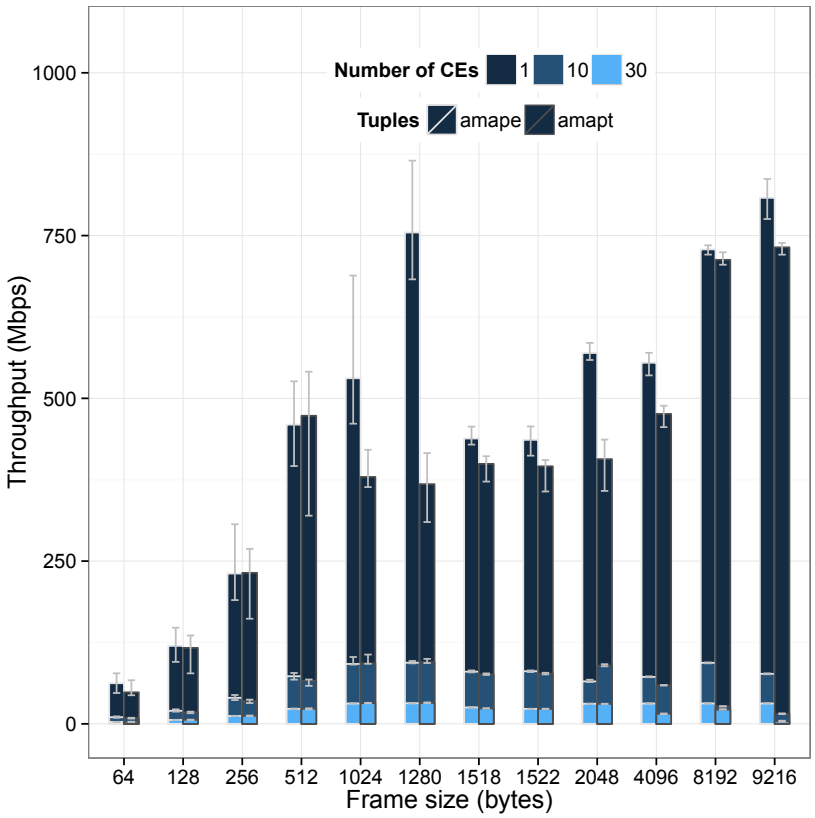


Figure 5.2: Throughput degradation for amape and ampat

described in [22] depends on the number of reserved system ports and the number of ports assigned to each CE machine. In this context, the maximum sharing ratio can be calculated using Formula 5.3.

$$
\begin{aligned}
&R_{max} = MAX(2^{16-a-m}) \\
&\text{Where:} \\
&\text{a - number of bits reserved for system ports} \\
&\text{m - number of bits reserved for contiguous} \\
&\text{port ranges assigned to each CE}
\end{aligned}
\tag{5.3}
$$

We should note that the lower a is, the higher the risk of causing port conflicts between applications running outside the range defined by a. The recommended value in the MAP-E standard[22] is $a = 6$. The maximum sharing ratio can be expressed as $1 : n$ and has a higher is better (HB) tendency. More details on how to obtain the maximum sharing ratio are presented in Section 5.5.

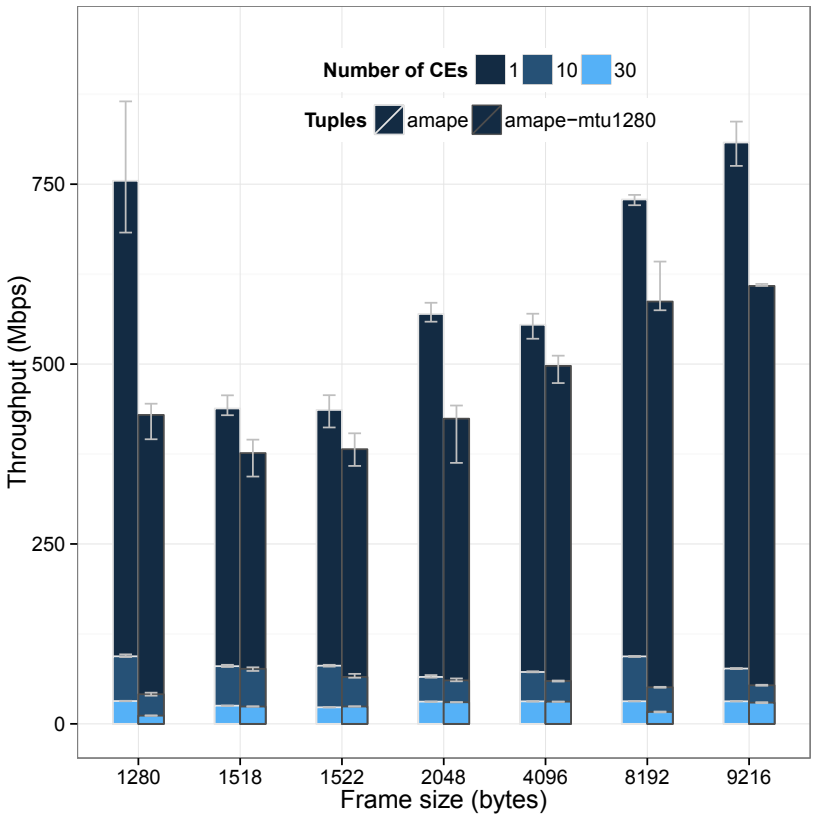


Figure 5.3: Throughput degradation with MTU

5.4 Empirical Analysis of Load Scalability

5.4.1 The Visualization of Network Performance Degradation

Partial comparative results at the three network scales (1×1, 10×1 and 30×1) have been plotted in Figure 5.2. The error bars represent the 1st and 99th percentiles. The plot in Figure 5.2 shows the network performance at the different scales for the amape and amapt tuples, while attempting to give a visual dimension to the associated performance degradation.

The results indicate the amape tuple as performing better for most frame sizes and scales. A detailed analysis of the amape tuple as well as an overall assessment of network performance degradation is presented in Section 5.1.6.

5.4.2 The Impact of MTU on Network Performance Degradation

The throughput results for two MTU values (1500 and 1280) for amape, have been plotted in 5.3. Considering the 40 bytes overhead created by encapsulation, the biggest performance difference was achieved for the 1280 frame size. the results confirmed the expectations, since there is no need to fragment the frame for the 1500 MTU size.

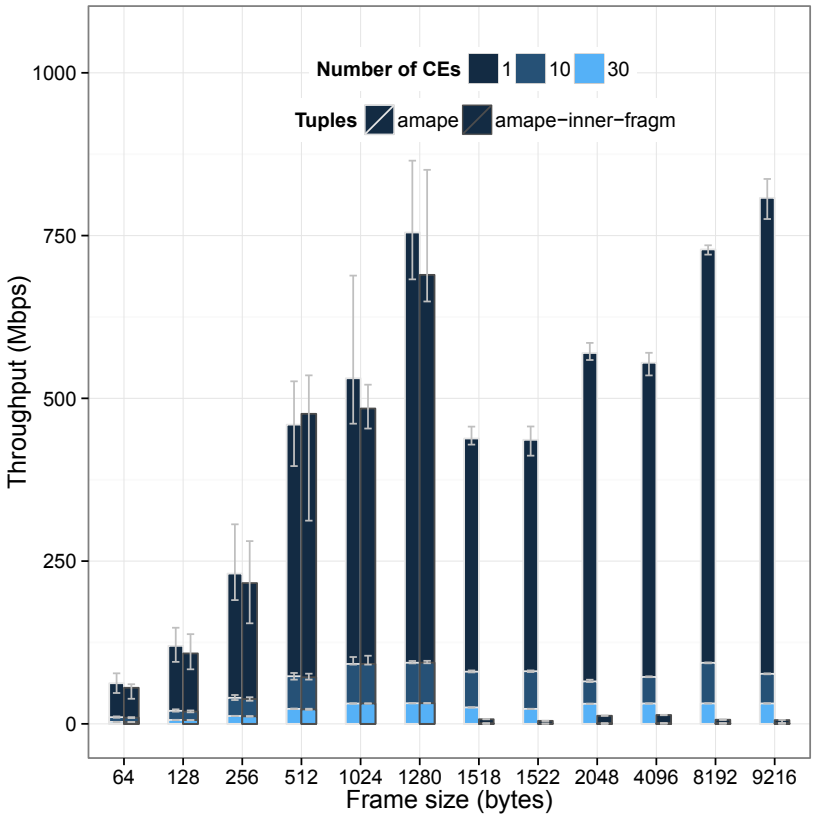


Figure 5.4: Throughput degradation with inner-fragmentation

The need for fragmentation at the 1280 MTU size affected the performance degradation as well, with as much as 91% for the 10×1 scenario and 97% for the 30×1 scenario. Since fragmentation was needed at both MTU sizes for the rest of the jumbo frames, the impact on performance and performance degradation was consistent but lower.

5.4.3 The Impact of Inner Fragmentation on Network Performance Degradation

Figure 5.4 depicts the comparative throughput results for amape with and without using inner fragmentation. From the plot, it is easy to observe the huge drop in performance for the jumbo frames. In numbers, the performance gap is as much as 99% for the 9,216 frame size. After further analysis of the issue, we found that the performance gap was created by the fragmentation and reassembly behavior of the virtual map interface. As anticipated in [70], the virtual interface had trouble with the constant wrapping of the identification field.

At higher scales, the impact on the performance was considerable as well, peaking for the 2,048 frame size at 92% for the 10×1 scale and 97% for the 30×1 scale. Considering the huge impact on performance in general, we would recommend against using this type

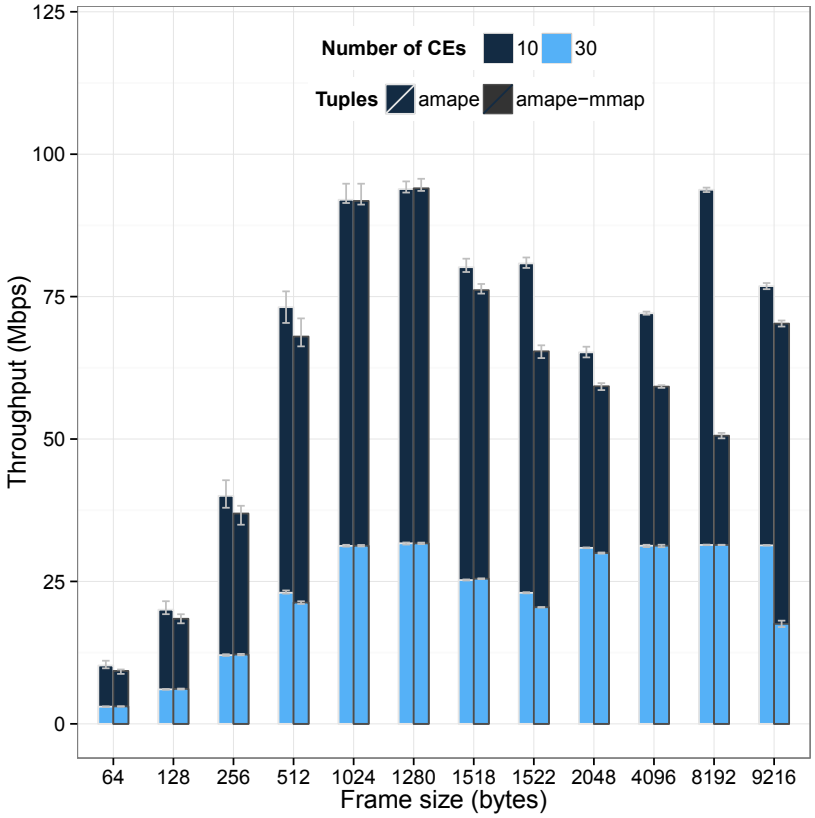


Figure 5.5: Throughput degradation with multiple map rules

Figure 5.6: Throughput results

of fragmentation.

5.4.4 The Impact of Multiple Mapping Rules on Network Performance Degradation

We have plotted in Figure 5.5 the results for amape using one mapping rule versus amape using 10 mapping rules for the 10×1 scenario and 30 mapping rules respectively. The results show a consistent degradation trend for the jumbo frames. A possible cause for this trend could be the conjuncture of mapping and fragmentation. However, we need to further investigate the root cause for this behavior.

5.4.5 Summarized Network Performance Degradation Data

While it is easy to roughly assess the network performance degradation from the presented plots, expressing the summarized data can prove more useful for a detailed analysis. Table 5.1 contains a summarized assessment network performance degradation of amape for all measured metrics: round-trip delay, packet delay variation (PDV) and throughput.

The detailed summarization revealed a non-linear degradation between the 10×1 and 30×1 network scales, as well as degradation peaks for each of the metrics. For example, for 10 CE machines, the degradation of throughput peaked at 951% for the 9,216 frame size, while the worst delay degraded was for the 1,522 frame size with 196%. In terms of packet delay variation, the peak was reached at 891%, again for 9,216 bytes frames.

With the risk of over-summarization, an overall performance evaluation can prove an easy and useful comparison tool. The network performance degradation results are summarized in Table 5.2. When reading the results we must keep in mind that network performance degradation (NPD) is a Lower is Better (LB) metric.

The empirical network performance data presented in [61] showed that within the same implementation, namely asamap, MAPe had an overall better performance. Although this is not the main objective of this manuscript, we want to note that the additional scalability empirical data confirms the amape tuple (MAPe implemented in asamap), as being the less impacted by the growth in scale. The overall results lead us to believe that encapsulation-based mechanisms are a better choice in terms of load scalability.

In addition to [61], empirical data for another transition implementation is presented, only for the 10×1 network scale. The tinymape tuple had the worst performance in terms of scalability of the five tuples. This result confirms the expectation set by the network performance results. Moreover, the tiny-map-e data confirmed that the performance of a transition technology is highly dependent on the implementation, and the data should be collected on an implementation basis.

Table 5.1: Network Peformance Degradation for amape

	RT Delay (%)		**PDV (%)**		**Throughput (%)**	
FS	**1x10**	**1x30**	**1x10**	**1x30**	**1x10**	**1x30**
64	138	334	495	1,860	84	95
128	195	341	476	1,762	83	95
256	186	330	450	1,691	83	95
512	113	281	500	1,782	84	95
1,024	158	333	395	1,380	83	94
1,280	111	237	592	1,994	88	96
1,518	127	359	388	1,321	82	94
1,522	196	364	383	1,261	81	95
2,048	113	208	689	1,571	89	95
4,096	129	188	626	1,514	87	94
8,192	119	177	690	1,298	87	96
9,216	97	175	891	1,441	90	96

Table 5.2: Network Performance Degradation (NPD) results

	RT Delay (%)		PDV (%)		Throughput (%)	
	1x10	**1x30**	**1x10**	**1x30**	**1x10**	**1x30**
amape	163	322	437	1,458	84	95
tinymape	471	-	571	-	94	-
amapt	155	264	129	424	86	94
amapdslite	172	352	443	924	84	94
amap464xlat	170	254	458	712	87	95

5.5 Empirical Analysis of Structural Scalability

As described in Section 4.3, the structural scalability of IPv6 transition technologies employing the mapping with address and port mechanism (MAP-E[22], MAP-T[26]), can be calculated with formula 5.3.

To clarify how the formula is supposed to be used, let us take a practical example. We start with the assumption that the following prefixes can be used within our network.

- IPv4 prefix: 198.51.1.1/32
- IPv6 prefix: 2001:db8::/48
- Embedded Address (EA) bits: 16

In other words, we have only one public IPv4 address to be shared among all our subscribers. In order to find out what would be maximum number of subscribers we could service in a map domain for this rule, we need to minimize a (the bits reserved for system ports) and m (the bits reserved for contiguous port ranges).

As recommended in [22], the safe value for a is 6 , so let us use it. In terms of m, we can chose to assign contiguous ranges of only 1 port to maximize the sharing ratio. Consequently, m would be 0. In this context, the resulted maximum sharing ratio would be the following.

$$R_{max} = 2^{16-6-0} = 1024 \tag{5.4}$$

Under the rules defined in [22], each subscriber would dispose of 63 ports split in 63 ranges of 1 port., while the 0-1024 range would be reserved for system ports. Various less safe combinations of a and m can be achieved, but with consequences of service availability for different network applications running on different CEs within the same map domain.

5.6 Evaluation of the Scalability Test System

Similar to the network performance test environment, we tested the direct connection between the 60 machines that were used as sending/receiving testers. The throughput data showed stable numbers with a relative standard deviation of under 7%.

In terms of time efficiency, the times were detailed according to the three major categories in the following.

Deployment: As underlined earlier, the deployment time depends heavily on the skill and experience of the human operator handling the deployment. Consequently, these

numbers are solely for exemplification purposes. For the latest experiment, it took about 10 hours to deploy the 91 machines needed for testing. Another 16 hours were spent on the preparation and troubleshooting of the underlaying StarBed nodes.

Testing time: Each test iteration needed about 70 seconds (60 seconds of testing and 10 seconds of sleep between iterations). Considering the chosen full-factorial design, for the latest experiments we had 12 ($frame\ sizes$) × 20 ($repetitions$) × 1 ($L4\ workload$)×5 ($transition\ tuples$)×2 ($network\ scales$)×3 ($degradation\ patterns$) = 7200 iterations. That needed 504,000 seconds or ~140 hours.

Post-processing time: For each of the iterations 20 seconds were needed to post-process the test data. Considering the 7200 test iterations, the total post processing time was 144,000 seconds or ~40 hours.

5.7 Summary and Outlook

This was a first attempt at benchmarking the load scalability of IPv6 transition technologies. The methodology included a metric called *network performance degradation*, which measures the percentile degradation at higher scales of network performance aspects, explicitly round-trip delay, jitter, throughput and frame loss. We consider this important as it affects most IPv6 transition scenarios, regardless of their size.

Moreover, we have analyzed the potential structural scalability issues which can arise from using IPv6 transition technologies. This should mainly affect the scalability of very large networks. Nevertheless, it should be quantified, as we anticipate a resourceful, commercial or national entity can follow our methodology to conduct such studies. As quantifying method, we have proposed the maximum sharing ratio, and have demonstrated its use.

In terms of load scalability, we were able to benchmark two open-source IPv6 transition implementations: *Asamap* and *Tiny-map-e*, covering four IPv6 transition technologies MAP-E, MAP-T, DSLite and 464XLAT. Overall, the empirical data indicates the amape tuple (MAP-E implemented in Asamap) as being the less impacted by the growth in scale. Moreover, our benchmarking methodology confirmed the a trend for load scalability within the same implementation. It seems like encapsulation-based technologies (MAP-E, DSLite) are a better choice than translation-based technologies (MAP-T, 464XLAT).

In addition, employing an abstract model of the analyzed IPv6 transition technologies, we were able to identify configurable parameters such as the MTU, the inner fragmentation and multiple map rules. These parameters proved to have a consistent impact on the network performance degradation of the tested tuples.

In terms for outlook for scalability, we plan to continue our standardization efforts, materialized in [63]. The approach has the advantage of being applicable to a wider range of transition solutions, such as commercial implementation, which mostly do not allow any source code instrumentation. However, a white-box approach would offer more insights into the performance limits by tweaking the run-time parameters. To address this, we plan to complement the current work with a white-box analysis of the same implementations.

The conducted scalability tests were within the available hardware resources. For now, we have used bare-metal servers, but for further topology growth tests we will need to use virtual technology, which can affect the quality of the data, and in turn the trust value of the scalability scores. However, virtualization is an ever-increasing phenomenon in today's production networks. The capability of implementations to function within a virtual environment, may also influence their scalability. All things considered, we plan to include virtual technology in the benchmarking process.

Chapter 6

Towards Security Quantification

> *Not everything that can be measured counts, and not everything that counts can be measured.*
>
> Al Morton

Aside from the larger address space, IPv6 has, in theory, a number of advantages over its predecessor in terms of design: a more efficient and extensible datagram, improved routing with easier route computation and aggregation, stateless auto-configuration and mandatory security. Over the years, however, some of these new features have proved to be either challenges for enforcing security (e.g. extension headers, stateless auto-configuration), or not-feasible (e.g. widespread deployment of IPv6 with IPsec). The IPv6 transition has further aggravated these challenges as transition technologies are generally exposed to the threats associated with both IP versions and hybrid blends, depending on the subcomponents.

6.1 Perspective on the Security of IPv6 Transition Technologies

When building an IPv6 transition plan, security is likely the biggest concern for network operators, as a heterogeneous IPv4 and IPv6 environment expands the attack surface for potential threats. To that end, building a threat model for IPv6 transition technologies can help clarify and categorize the associated security threats. In turn, this should facilitate the search for mitigation techniques and can lead to a security quantification method for IPv6 transition implementations. Considering all of the above, we present in this section a threat model built around the well established STRIDE approach described in [50]. The STRIDE mnemonic was used to classify the documented threats. The correlations between elements of the Data Flow Diagrams (DFD) and the STRIDE threat categories are used for the initial basic assessment of the threats for each of the sub-elements.

The generic STRIDE approach represents only the base of our proposal. The main contribution lies in identifying the threat modeling steps necessary for IPv6 transition technologies in particular. To prove the validity of the proposal, the model was used to define and categorize existing and new threats for the four generic types of IPv6 transition technologies defined in Section 2.3.1. The resulting non-exhaustive threat analysis and penetration test data can be considered another important part of the contribution of this work.

6.2 Building a Holistic Threat Model

The proposed threat model involves a series of steps which were inspired by the STRIDE modeling approach presented in [78] and the Open Web Application Security Project (OWASP) foundation's application threat modeling guidelines [51]. In the context of IPv6 transition technologies we recommend the following steps.

(1) Establish the function: the function of the IPv6 transition technology needs to be clearly documented. Depending on the context, the technology can incorporate multiple services, which need to be clearly identified in order to perform an effective threat analysis.

(2) Identify the IPv6 transition technology category: the category should be identified considering the generic classification defined in Section 2.3.1. This step will help build the data flow diagram (DFD) used in the following steps.

(3) Decompose the technology: build a data flow diagram (DFD) and highlight the entry points, protected resources and trust boundaries. The entry points should be assigned a level of trust considering the trust boundaries. The external entities, process, data store and data flow elements should be depicted in the same diagram as defined in [50]. The IP protocol suite and the protocols used for the designated function should be identified as well. This can narrow down the attack surface.

(4) Identify the threats: The STRIDE model associates the six categories of threats to each of the elements described in the DFD. Based on this association, we get an initial assessment of the threats as shown in Table 6.1. To clarify, a data flow, for example, is more susceptible to tampering, information disclosure and denial of service threats. The initial threat assessment must be followed by a detailed analysis which should consider the protocols used in conjuncture with the transition technology.

Considering the level of trust assigned to each entry point, an associated likelihood of attack from that entry point can be deduced. For example, if the entry point is considered trusted, we can assume the likelihood of an attack is low. By extrapolating, the six categories of STRIDE attacks could be assigned a likelihood, considering that their association with the DFD elements that are entry points. For instance, if we have an untrusted entry point which is also an external entity, for which we can have spoofing and repudiation as potential threats, the two types of attacks can be considered to have a high likelihood, as they would be exploited from an untrusted entry. Using this logic, by associating the detailed threats with the STRIDE model and the DFD elements they could be applied to, we can assign each threat a likelihood value. This can represent a base for prioritizing mitigation solutions. Each threat should be documented using the following format.

Table 6.1: STRIDE Threats per Element

	Spoofing	Tampering	Repudiation	Info Disclosure	Denial of Service	Elevation of Privilege
External	✓		✓			
Process	✓	✓	✓	✓	✓	✓
Data Store		✓	†	✓	✓	
Data Flow		✓		✓	✓	

Field Name	Description
Threat_ID	A code associated with each identified threat
Description	A summarized description of the threat
STRIDE	The association with the STRIDE categories
Mitigation	Details about existing mitigation solutions
Likelihood	Likelihood of the threat being exploited
Validation	Empirical validation data

For an easy reference in future publications the [**Threat_ID**] should follow the format: *[Code]-[First Author Initials]-[Publication Submission Year]-[Serial number]*. For an author named John Doe, who is submitting a publication proposal in 2015, an example ThreatID is: *IPv6-JD-2015-1*. The serial number is incremented with each threat found for a particular protocol or technology. A list of codes for the basic transition technologies and generic transition technologies can be found in Appendix A 2. . For the well-known TCP/IP protocol suite we have used the usual acronyms as codes. As the subcomponents interact and the used protocols stack, the threats can fuse and result in convoluted threats with a higher likelihood of exploitation. Depending on the list of discovered threats, the possibility of a fusion between threats should be analyzed.

(6) Review, repeat and validate: steps 1 and 3 have to be reviewed in the context of potential changes in the technology function and associated protocols. Step 4 should be repeated periodically, as threats may have been overlooked, or the context set by steps 1 and 3 may have changed. If the transition technologies have existing implementations, the analysis should be confirmed with empirical data. To that end, penetration testing can be used.

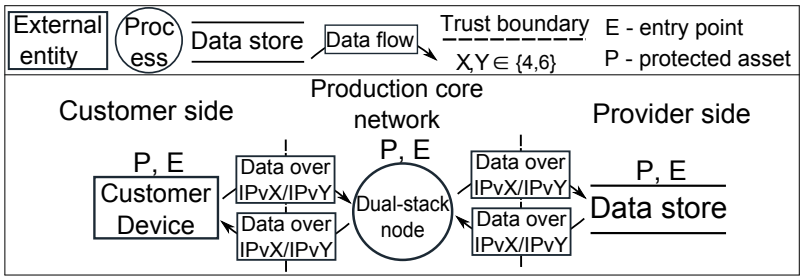
6.3 Applying the Threat Model

In the following subsections, the proposed threat modeling technique was used for the four generic IPv6 transition technologies categories defined in Section 2.3.1. The threat models can be viewed as a starting point for IPv6 transition technologies associated with each category.

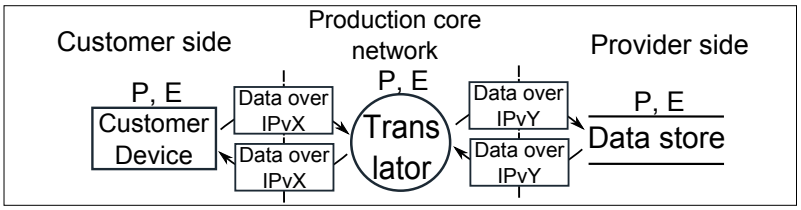
6.3.1 Dual-stack IPv6 Transition Technologies

Establish the function

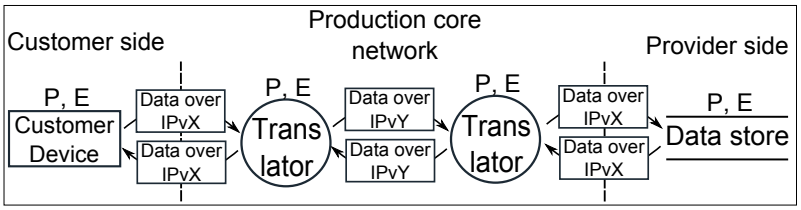
The function for dual-stack transition technologies is to ensure a safe data exchange over a dual-stack infrastructure. In other words, the data can be transferred over both IPv4



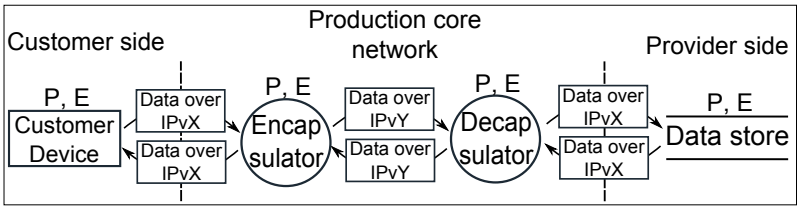
(a) Dual-stack



(b) Single translation



(c) Double translation



(d) Encapsulation

Figure 6.1: Data Flow Diagrams for the generic IPv6 transition technologies categories

and IPv6. From a network service perspective, the main function is data forwarding. This includes interior gateway routing solutions. We start with the assumption that services such as address provision, DNS resolution or exterior gateway routing are performed by other nodes within the core network. This assumption in common for all the four generic categories of IPv6 transition technologies.

Identify the IPv6 transition technology category

This step is meant for more specific transition technologies. Since we are targeting the generic category itself, the step is unnecessary here. This stands for the other three categories as well.

Table 6.2: Generic IPv6 Transition Technologies Convoluted Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation	Validation
Generic IPv6 Transition Technologies										
1	DS -MG-2015-1	ARP-MG-2015-1 + ND -MG-2015-10	H	H	H	H	H		Ensure DoS mitigation for IPv4 suite protocols; use SEND	✓
2	DS-MG-2015-2	ARP-MG-2015-1 + OSPFv3-MG-1	H	H	H	H	H	H	Use OSPv3 with IPsec	✓
3	1transl -MG-2015-1	IP/ICMP -MG-2015-3 + ND-MG-2015-5	H		H	H	H		No widely accepted mitigation	X
4	1transl -MG-2015-2	TCP-MG-2015-1 + ND-MG-2015-10	H	H	H	H	H		Block packets with non-internal addresses; use SEND	✓
5	2transl -MG-2015-1	IP/ICMP -MG-2015-4 + ND -MG-2015-4	H	H	H	H	H		No widely accepted mitigation	X
6	2transl -MG-2015-2	IP/ICMP-MG-2015-4 + OSPFv3-MG-2015-1	H	H	H	H	H	H	OSPFv3 with IPsec	✓
7	encaps -MG-2015-1	IPv6-MG-2015-1 + 4encaps -MG-2015-1				H	H		Use IPv4 firewall before decapsulating	

Legend

L	Associated with low likelihood of exploitation	H	Associated with high likelihood of exploitation

Decompose the technology

A DFD for dual-stack transition technologies is presented in Figure 6.1a. The diagram represents the basic use case and includes a minimal set of elements. On the customer side, we have a Customer Device which initiates the data exchange. It represents one of the entry points of the system and contains important data, which should be regarded as an asset and protected. The Customer Device is regarded as an external element because it is outside the control zone of the production network. The data request is transmitted over IPv4 or IPv6 to a Dual-stack node. The Dual-stack node is another entry point and contains valuable topology information which should to be protected as well. The Dual-stack node forwards in turn the data request to the provider data store. The Data store is another entry point in the system and it is assumed to contain valuable data. The data reply is forwarded back and makes its return on the same path.

The only trusted entry point in the system is the Dual-stack node. The other two entry points are considered untrusted, since they are outside the control of the production network. That means they can be exploited with a higher likelihood by an attacker. Considering the data can be transferred over both IPv4 and IPv6, we need to consider IP protocol suites. Furthermore, the possibility of using security and routing protocols should be considered.

Identify the threats

By analyzing the DFD in association with the STRIDE threats per element chart, we can observe that the Customer Device can be subject to spoofing and repudiation attacks. It being an untrusted entry point, it means there is a high likelihood of an attack. The Dual-stack node can be subject to all six types of attacks. However, the likelihood of that happening is low, considering it is a trusted entry point. The Data flow is vulnerable to tampering, information disclosure and denial of service. Considering it traverses untrusted parts of the system, the level of likelihood of an attack on the data flow is high. Lastly, the Data store could potentially be targeted by tampering, repudiation, information disclosure and denial of service attacks. The likelihood for these to happen is high as well, the data store being an untrusted entry point.

Tables 6.3, 6.5, 6.6, 6.4, 6.7 and 6.8 contain a non-exhaustive collection of existing

threats, which have been collected by surveying a part of existing literature on this subject. For further documentation, each threat has been provided with a reference in the first column. For reuse purposes, the threats are organized according to the categories of protocols which would be necessary for accomplishing the function of the IPv6 transition technologies.

For dual-stack transition technologies the protocol threats associated with the IPv4 suite (Table 6.3), IPv6 suite (Table 6.5), routing (Table 6.4) and switching (Table 6.8) could potentially be exploited from the 3 entries of the system: the **untrusted - High likelihood of exploitation** Customer device (External entity), **trusted - Low likelihood of exploitation** Dual-stack node (Process) and **untrusted - High likelihood of exploitation** Provider Data store (data store).

The IPv4 suite, transport layer and most of the IPv6 suite protocols are associated with all the elements of the DFD. By extrapolation, their threats have a high likelihood of occurrence. Some of the IPv6 protocol threats (Table 6.5), namely ND-MG-2015-3 to ND-MG-2015-6 and the Layer 2 technologies' threats (Table 6.8) can only be associated with routers or switches. In this context, they could only be associated with the Dual-stack node. That means they have a low likelihood of occurrence. Similarly, the routing protocols can only be associated with the Dual-stack node. By association, they also have a low likelihood of being exploited.

By analyzing the interaction between the three elements of the DFD (Figure 6.1a) and the protocols used by Dual stack transition technologies, we can uncover other threats. For example, if the ARP-MG-2015-1 (ARP cache poisoning) is used to perform a Denial of Service attack on the Dual-stack node from the Customer device, the likelihood of exploitation rises for the ND-MG-2015-10 (ND Replay Attacks) threats. Table 6.5. ARP-MG-2015-1 could be replaced by any other DoS threat associated with the IPv4 suite protocols. This complex threat could only be prevented by ensuring that the IPv4 suite DoS threats are properly mitigated. Examples of convoluted threats for the four generic IPv6 transition technologies are presented in Table 6.2.

Another convoluted threat can result from exploiting IPv4 or IPv6 spoofing threats to increase the likelihood of an attack on routing protocols with simple authentication, such as or OSPFv3-MG-2015-1, OSPFv2-MG-2015-1 or RIPv2-MG-2015-1. Since the attack could be performed from an untrusted entry point (Customer device or Data store), the likelihood of the threat being exploited rises to High. This type of attack can be mitigated by using cryptographic authentication for the routing protocols.

The list of threats can help technology implementors and network operators alike prioritize the threats and mitigate accordingly. The protocols which have threats with no widely accepted mitigation techniques have been highlighted and should be treated as first priority.

Table 6.3: IPv4 Suite Protocols Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
IPv4 Suite Protocols			#		#				
			O	O	O	O	O	O	
				=	=	=	=		
				>		>	>		
1	IPv4-MG-2015-1 [53]	IP source address spoofing	H	H	H	H			Apply ACLs and filter source address routed traffic
2	IPv4-MG-2015-2 [54]	Malformed version field		H					Version field must be checked to be 4
3	IPv4-MG-2015-3 [54]	Packets with a forged DSCP field	H				H		Filter packets with unrecognized DSCP
4	IPv4-MG-2015-4 [54]	Buffer overflow with IP fragmentation					H		IP module should implement measures to avoid illegitimate reassembly
5	ICMP-MG-2015-1 [53]	Ping o'death					H		Patch software to not accept oversized ICMP messages
6	ICMP-MG-2015-2 [79]	ICMP redirects	H	H	H	H	H		routing tables should not be modified in response to ICMP Redirect messages
7	ICMP-MG-2015-3 [55]	ICMP sweep for recon				H			Selective filtering of ICMP messages
8	ICMP-MG-2015-4 [55]	ICMP traceroute				H			Selective filtering of ICMP messages
9	ICMP-MG-2015-5 [55]	ICMP firewalk				H			Selective filtering of ICMP messages
10	ICMP-MG-2015-6 [55]	ICMP flooding					H		Selective filtering of ICMP messages
11	ARP-MG-2015-1 [56]	ARP cache poisoning	H	H	H	H	H		Static ARP entries, arpwatch
12	ARP-MG-2015-2 [54]	ARP cache overrun					H		Selectively drop packets
13	IGMP-MG-2015-1 [57]	IGMP flooding					H		selective filtering of IGMP messages, multicast group authentication

Legend

H	associated with High likelihood	**# external, O process**	#	Untrusted element with High likelihood of being exploited
L	associated with Low likelihood	**>data flow, = data store**	O	Trusted element with Low likelihood of being exploited

Table 6.4: Routing Protocols Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
Routing Protocols			#		#				
			O	O	O	O	O	O	
				=	=	=	=		
				>		>	>		
1	RIPv2-MG-2015-1 [80]	RIPv2 simple password authentication issues	L	L	L	L	L	L	Use cryptographic authentication
2	RIPv2-MG-2015-2 [80]	RIPv2 Security Association expiration				L			Let RIPv2 routing fail when the last key expires
3	RIPv2-MG-2015-3 [80]	RIPv2 Security Association					L		The receiver should not try all RIPv2 Security Associations
4	OSPFv2-MG-2015-1 [81]	OSPFv2 simple password authentication	L	L	L	L	L	L	Use cryptographic authentication
5	OSPFv2-MG-2015-2 [81]	OSPFv2 cryptographic authentication sequence number prediction	L	L	L	L	L	L	Use cryptographic sequence number
6	OSPFv3-MG-2015-1 [82]	OSPFv3 using the same manual key	L	L	L	L	L	L	avoid using manual keys

Legend

H	associaced with High likelihood	**# external, O process**	#	Untrusted element with High likelihood of being exploited
L	associaced with Low likelihood	**>data flow, = data store**	O	Trusted element with Low likelihood of being exploited

Table 6.5: IPv6 suite protocols threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
IPv6 Suite Protocols			# O	O	# O	O	O	O	
				= >	=	= >	= >		
1	IPv6-MG-2015-1 [58]	Routing header can be used to evade access controls	H				H		Access controls based on destination addresses
2	IPv6-MG-2015-2 [58]	Site-scope multicast addresses for reconnaissance				H			Drop packets with site-scope destination addresses
3	IPv6-MG-2015-3 [58]	Anycast traffic identification for reconnaissance				H			Restrict the use of outside anycast services
4	IPv6-MG-2015-4 [58]	Extension headers excessive hop-by-hop options					H		Drop packets with unknown options
5	IPv6-MG-2015-5 [58]	Overuse of IPv6 router alert Option					H		Filter externally generated Router Alert packets
6	IPv6-MG-2015-6 [58]	IPv6 fragmentation that would potentially overload the reconstruction buffers					H		Mandating the size of packet fragments; drop non-final fragments smaller than 640 octets
7	IPv6-MG-2015-7 [58]	IPv4-Mapped IPv6 Addresses	H				H		Avoid using IPv4-mapped IPv6 addesses
8	ICMPv6-MG-2015-1 [83]	ICMPv6 message spoofing	H				H		Use IPAuth
9	ICMPv6-MG-2015-2 [83]	ICMPv6 Redirects	H		H	H			Use IPAuth or ESP
10	ICMPv6-MG-2015-3 [83]	Back-to-back erroneous IP packets					H		Implement correctely ICMP error rate limiting mechanism
11	ICMPv6-MG-2015-4 [83]	Send ICMP Parameter Problem Message to the multicast source				H	H		Secure multicast traffic
12	ICMPv6-MG-2015-5 [83]	ICMP messages passed to the upper-layers					H		Use IPSec
13	ICMPv6-MG-2015-6 [60]	ICMPv6 echo request for reconnaissance				H			Deny inbound ICMPv6 echo request
14	SLAAC-MG-2015-1 [58]	Address Privacy Extensions Interact with DDoS Defenses					H		Tune the change rate of the node address
15	ND-MG-2015-1 [84]	Neighbor Solicitation/Advertisement Spoofing	H				H		Use SEND
16	ND-MG-2015-2 [84]	Neighbor Unreachability Detection (NUD) failure					H		Use SEND
17	ND-MG-2015-3 [84]	Malicious Last Hop Router	L	L	L	L	L		Use SEND
18	ND-MG-2015-4 [84]	Default router is 'killed'	L	L	L	L	L		No widely accepted mitigation technique
19	ND-MG-2015-5 [84]	Good Router Goes Bad	L	L	L	L	L		No widely accepted mitigation technique
20	ND-MG-2015-6 [84]	Spoofed Redirect Message	L	L	L	L	L		Use SEND; Still an issue for the ad-hoc case
21	ND-MG-2015-7 [84]	Bogus On-Link Prefix					L		Use SEND
22	ND-MG-2015-8 [84]	Bogus Address Configuration Prefix					L		Use SEND; Still an issue for the ad-hoc case
23	ND-MG-2015-9 [84]	Parameter Spoofing	L		L	L			Use SEND; Still an issue for the ad-hoc case
24	ND-MG-2015-10 [84]	ND Replay attacks	H			H			Use roughly synchronized clocks and timestamps; Use SEND
25	ND-MG-2015-11 [84]	Neighbor Discovery DoS threat					H		Rate limit Neighbor Solicitations
26	DAD-MG-2015-1 [84]	Duplicate Address Detection DoS					H		Use SEND
27	SEND-MG-2015-1 [85]	The Authorization Delegation Discovery process may be vulnerable to DoS					H		Cache discovered information and limit the number of discovery processes
28	MIPv6-MG-2015-1 [58]	Obsolete Home Address Option in Mobile IPv6	H						Secure Binding Update messages

Legend

H	associaced with High likelihood	**# external, O process**	#	Untrusted element with High likelihood of being exploited
L	associated with High likelihood	**>data flow, = data store**	O	Trusted element with Low likelihood of being exploited

Table 6.6: Layer4 Protocols Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
Transport Layer Protocols			# O = >	O	# O =	O = >	O = >	O	
1	TCP-MG-2015-1 [53]	SYN flood					H		Block packets with non-internal addresses from leaving the network
2	TCP-MG-2015-2 [53]	SYN/ACK flood	H		H		H		L3/L4 Packet Filtering
3	TCP-MG-2015-3 [53]	ACK or ACK-PUSH Flood	H		H		H		L3/L4 Packet Filtering
4	TCP-MG-2015-4 [53]	Fragmented ACK Flood					H		L3/L4 Packet Filtering
5	TCP-MG-2015-5 [53]	TCP Spoofing based on sequence number prediction	H						Block packets with non-internal addresses from leaving the network
6	TCP-MG-2015-6 [53]	TCP session hijacking based on sequence number prediction	H	H	H	H	H	H	Block packets with non-internal addresses from leaving the network
7	TCP-MG-2015-7 [53]	RST and FIN DoS					H		L3/L4 Packet Filtering; Stateful Flow Awareness
8	UDP-MG-2015-8 [86]	UDP flood					H		QoS regulation; L3/L4 Packet Filtering
6	NAT44-MG-2015-9 [87]	Port set exaustion					H		Address-Dependent Filtering

Table 6.7: Basic IPv6 Transition Technologies Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
Routing Protocols			# O = >	O	# O = >	O = >	O = >	O	
1	IP/ICMP-MG-2015-1 [88]	IPv4 address spoofing with IPv4-embedded IPv6	L						Implement reverse path checks to verify that packets are coming from an authorized location.
2	IP/ICMP-MG-2015-2 [89]	transport mode ESP will fail with IPv6-to-IPv4 translation				L			Use checksum-neutral addresses
3	IP/ICMP-MG-2015-3 [89]	Authentication Headers cannot be used across an IPv6-to-IPv4				L			No widely accepted mitigation
4	IP/ICMP-MG-2015-4 [89]	Stateful translators can run out of resources					L		No widely accepted mitigation
5	4encaps-MG-2015-1 [58]	Tunneling IPv6 through IPv4 networks could break IPv4 Network's security assumptions				L			route the encapsulated through an IPv4 firewall before decapsulating them

Table 6.8: L2 Technologies Threats

	ThreatID	Description	S	T	R	I	D	E	Mitigation
L2 Technologies			# O = >	O = >	# O	O = >	O = >	O	
1	VLAN-MG-2015-1 [87]	Exhaust a forwarding information base (FIB) of an L2switch					L		IEEE 802.1x authentication
2	VLAN-MG-2015-2 [90]	Content Addressable Memory (CAM) Overflow					L		Use the port-security features
3	VLAN-MG-2015-3 [90]	Basic VLAN Hopping	L						Software update
4	VLAN-MG-2015-4 [90]	Double encapsulation VLAN Hopping	L					L	Disable Auto-trunking
5	VLAN-MG-2015-5 [90]	Spanning Tree Attack				L	L		Disable STP, Use BPDU Guard

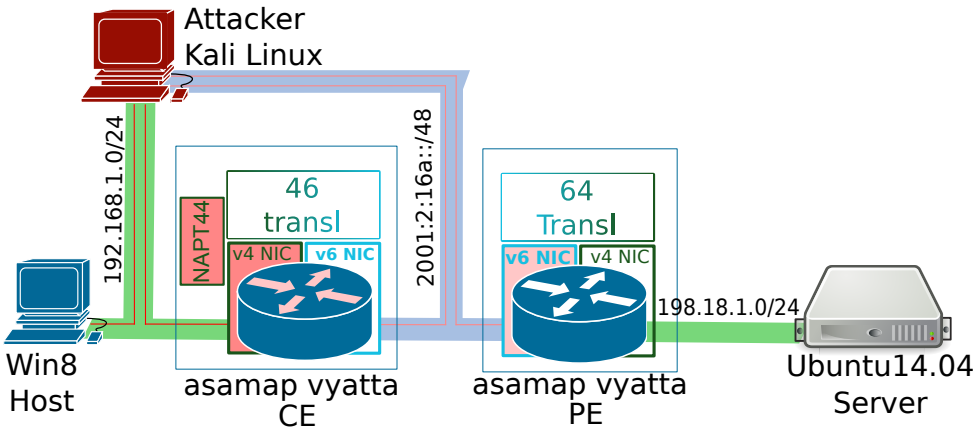


Figure 6.2: MAP-T Penetration Testbed

Review, repeat and validate

This step is necessary if the technology analyzed or associated protocols change. For example if the routing system were to be only OSPFv3, then the threats associated with other routing protocols could be ignored. Also, the detailed analysis of threats is far from exhaustive. In terms of convoluted new threats, only a few are presented as an example. If this was to be an updated database of threats, it would need constant update. To further validate the presented threats, a simple penetration testbed was built. The details of the testbed are presented in Figure 6.2. MAP-T [26] was used as transition technology. Asamap [28], a transition implementation developed in Japan, was used as the base for MAP-T. The threats which were successfully emulated, have been marked accordingly in the last column of Table 6.2. In the case of the convoluted threats identified for Dual-stack transition technologies, both threats were emulated successfully by performing ARP Cache Spoofing, Neighbor Advertisement (NA) flooding and simple traffic analysis.

6.3.2 Single Translation Transition Technologies

To avoid redundant information, the following three subsections will only mark the differences with the threat modeling process presented for Dual-stack transition technologies.

One of the fundamental differences is that the single translation technologies would require a node to algorithmically translate the IPvX packets to IPvY, as shown in Figure 6.1b. For both translation directions ($4 \rightarrow 6$ and $6 \rightarrow 4$) the threats for the IPv4 suite (Table 6.3), IPv6 suite (Table 6.5), routing (Table 6.4) and switching (Table 6.8) should be considered.

There are technologies that use stateful mapping algorithms e.g. Stateful NAT64 [15], which create dynamic correlations between IP addresses or {IP address, transport protocol, transport port number} tuples. Consequently, we need to consider the protocols used at the transport layer (Table 6.6) as part of the attack surface. The threats presented in Table 6.7, associated with the IP/ICMP translation algorithm (IP/ICMP), should be considered as well.

In terms of convoluted threats, one example could be exploiting the IP/ICMP-MG-2015-3 threat (IPAuth does not work with IP/ICMP) which would increase the likelihood of ND-MG-2015-4 (Default router is killed) or ND-MG-2015-5 (Good router goes bad) threats being exploited. Since there is no widely-accepted mitigation for any of the three threats, this convoluted threat is laking a mitigation solution as well. Fortunately, both complex threats could not be validated empirically. An IPsec VPN connection was successfully established using UDP encapsulation between the Windows Host and the Ubuntu Server. Moreover, the ND-MG-2015-4 and ND-MG-2015-5 could not be validated empirically, as Asamap [28] does not accept RA messages when IPv6 forwarding is enabled.

If the TCP-MG-2015-1 threat (SYN flood) is exploited from an untrusted entry point, it increases the likelihood of a ND-MG-2015-10 (ND Replay attacks) threat. This threat can be mitigated by blocking packets with non-internal addresses from leaving the network. Both the SYN flood attack and the Neighbor Advertisement (NA) flooding attacks were staged successfully.

6.3.3 Double Translation Transition Technologies

The main difference between the Single translation case and the double translation case is the need for an extra translation device as part of the core network (Figure 6.1c). Another important difference would be that in the untrusted zone, the Customer device and Data store would employ the same IP suite. Hence, the considered threats for the untrusted elements would be either the IPv4 suite (Table 6.3) or the IPv6 suite (Table 6.5) protocol threats. Similar to the single translation technologies, the routing (Table 6.4), switching (Table 6.8), transport layer (Table 6.6) and IP/ICMP (Table 6.7) threats should be analyzed as well.

The use of stateful translation mechanisms can expose a double translation technology to the IP/ICMP-MG-2015-4 threat (DoS by exhaustion of resources). A convoluted threat can result by exploiting this threat on one of the translators and the ND-MG-2015-4 or ND-MG-2015-5 threats on the other translator. This threat would have a higher likelihood of exploitation since it is associated with an untrusted entry point. In terms of mitigation, further investigation is needed, as there are no widely accepted mitigation techniques. Although the IP/ICMP-MG-2015-4 threat was replicated with success, the ND-MG-2015-10 or ND-MG-2015-5 could not be emulated because of a simple built-in mitigation mechanism implemented by Asamap [28]. Router advertisement (RA) messages are not accepted while in IPv6 forwarding mode.

The IP/ICMP-MG-2015-4 threat can also fuse with the simple authentication threats such as OSPFv3-MG-2015-1 , OSPFv2-MG-2015-1 or RIPv2-MG-2015-1 to affect both translating nodes. The likelihood of the threats become higher by fusing them, since the flooding attack can be performed from an untrusted entry point, the customer network. This threat could be mitigated by using cryptographic authentication or implementing reverse path checks. The convoluted threat was validated by flooding the translation table of the first translator and forcing it to crash. OSPFv3 information disclosure was emulated with simple traffic analysis. To validate the other types of threats, a rogue router instance was created using Asamap.

6.3.4 Encapsulation Transition Technologies

Similar to double translation IPv6 transition technologies, encapsulation technologies, the core network traffic is forwarded through at least two devices, an Encapsulator and a Decapsulator (Figure 6.1d). As the main difference, the traffic is encapsulated. This means more overhead but also more support for end-to-end security protocols. Packets are encapsulated either over IPv4 or IPv6. Consequently, for the untrusted domain devices we would consider either the IPv4 suite (Table 6.3) or the IPv6 suite (Table 6.5) threats. In addition the routing (Table 6.4), switching (Table 6.8), transport layer (Table 6.6) and encapsulation-related (Table 6.7) threats should be considered.

Convoluted threats can arise by exploiting the 4encaps-MG-2015-1 threat (avoiding IPv4 network security measures with encapsulation). This threat can facilitate IPv6 suite DoS threats on the Decapsulator device. This convoluted threat would increase the likelihood of a successful DoS attack from the Customer Device. The threat could be mitigated by making use of an IPv4 firewall before decapsulating the packets.

6.4 Evaluation of the Threat Analysis

In order to isolate the impact of the hardware platform, the security threats have been emulated using two different hardware configurations for the virtual machines employed in the penetration testbed.

In terms of time efficiency, the threat analysis steps had two largely time consuming stages: identifying the threats and validating the threats.

Identifying the threats: was mainly literature review. The time efficiency of the process is largely depending on the experience and knowledge of the security analyst. Therefore, the number is solely for exemplification purposes. The non-exhaustive threat analysis needed roughly 40 hours to complete.

Validating the threats: the simple list of validated threats were validated in about 32 hours. That includes the review for suitable candidates for attack tools.

6.5 Summary and Outlook

As a starting point for quantifying the security of IPv6 transition technologies, we have proposed an associated wide-ranging threat mode. To prove the adequacy of the proposed threat model, we have used it to analyze the security threats of the four generic categories of IPv6 transition technologies defined in Section 2.3.1.

As part of the threat modeling process, for each of the four categories we have defined a common use case, deconstructed the system using Data Flow Diagrams (DFDs), and obtained an initial overview of the security threats by association with the STRIDE approach. Subsequently, we have documented existing threats and mitigation solutions for the IP protocol suites and basic transition technologies representing dependencies of the system. The documented threats have been mapped with the STRIDE elements identified in the DFD, to obtain a rough likelihood for the threats to be exploited. We have

shown how existing threats can lead to new threats by the interaction between subcomponents and various protocols. Lastly, we have empirically validated some of the analytically discovered threats by building a simple penetration testbed.

As a summary, the proposed, holistic threat model has revealed that the concerns related to the security of IPv6 transition technologies are well-endowed. Although it is too early to say that certain technologies are more secure than others, we contend that the proposed method can represent the basis for establishing a methodology that can lead us there. As a general observation for double translation and encapsulation technologies, the lack of shared secrets between the CE and PE devices can have serious consequences on the core network exploit-ability.

As shown, the threat model can be used to classify and prioritize already documented threats. Moreover the threat model can help discover new threats and indicate their level of mitigation. As a secondary contribution , this proposal contains a non-exhaustive database of documented threats associated with IP enabled devices. Moreover, preliminary penetration test data was introduced for one of the existing IPv6 transition implementations.

We contend that this approach can be the starting point for analyzing the threats of specific IPv6 transition technologies. Moreover, we intend to extend this work by proposing a risk quantification technique, which should lead to a security quantification method for IPv6 transition technologies. The first steps in this direction were taken by proposing penetration testing as a validation technique. Although the presented data is preliminary, we aim to continue this effort in future work. In turn, the data can be used as base for a risk quantification method, which should lead to the end goal: a security quantification metric. In terms of standardization, we also plan to continue developing the current draft [64] in the IETF OPSEC working group.

Chapter 7

Discussion and Future Work

> *The future of the Internet has always been easier to experience than to predict. I see no reason for this unpredictability not to continue to be a feature of the Internet going forward.*
>
> Scott Bradner

IPv6 transition scenarios and IPv6 transition technologies have already been known for some time to the Internet community. However, the worldwide deployment rate of IPv6 is still very low. Given the complexity and the diversity of transition technologies, one of the biggest challenges for network operators is understanding which technology to use in a certain network scenario. Various implementations of transition technologies have been introduced, further complicating this problem.

7.1 Validity and the Pursuit of Standardization

This thesis proposes a solution to that problem in the form of practical and novel evaluation methodologies associated with a heterogeneous IP4-IPv6 testbed, called IPv6NET. The basis for the feasibility analysis of IPv6 transition implementations is represented by practical means, such as real implementations and empirical measurements. To prove the validity of these methodologies, we have used them to analyze the feasibility of two suitable transition implementations, covering multiple transition technologies. By analyzing the empirical results, we were able to reach our goal and point out which of the transition implementations was more suitable.

Furthermore, we have identified possible network performance and load scalability trends in IPv6 transition technologies benchmarking. For example, encapsulation-based technologies proved to have better throughput performance, while translation-based technologies had a better latency performance. In terms of scalability, encapsulation was

confirmed to be more feasible than translation, while MAP-based algorithms (MAP-E, MAP-T) proved to be more affected by structural scalability limitations. We were also able to point out some unexpected behaviors, which could have been overlooked if simulators or analytical tools were employed. This underlines the need for a testbed and gives us motivation for a further root cause analysis.

In the great scheme of things, however, empirical proof is only one piece of the puzzle. Often times, the best methods get overlooked or are ditched into the past because of lack of exposure. Besides the exposure, the pursuit of standardization can lead to a refined result, which can withstand the test of time. Open standards offer a great opportunity in this direction, and the IETF is the most productive standards body behind the Internet.

All considering, we started the standardization efforts in the IETF, which over time materialized in [63]. The document, which evolved though contributions of the Benchmarking Working Group (BMWG), covers only two of the proposed methodologies: network performance and scalability. The draft has passed one of the validation steps within the BMWG, by being officially adopted as a working group item. We aim to continue this effort and improve the two methodologies further.

Recently, we have started the standardization effort for another feasibility dimension: security. The proposed threat model has been included in [64]. The draft has been presented in the last Operational Security Capabilities for IP Network Infrastructure (OPSEC) working group, and we hope to continue developing this work in OPSEC. As for the fourth dimension, operational capability, we intend to start the collaboration with a conformance-oriented forum, such as the IPv6 ready consortium.

In the pursuit for standardization, there are, however, potential threats to the technical integrity of the methodologies. For example, in the IETF a document cannot progress in a certain direction, unless rough consensus is reached. Although, in most cases, the rough consensus is based on the technical technical prowess of the arguments, there are times when religion (to be understood as certain mindset) wins the day. A quote from Scott Bradner, one of the most prominent figures of the IETF, comes to mind: *"If you have a strong opinion on X, and Tim doesn't share it, well, it may be a little tricky to get a standard out, Tim is the key to ensuring a consistent architecture, to keeping things from fragmenting."* While this is a normal attitude from the interoperability perspective, it can lead to a diluted technical quality in some contexts.

Let us assume that a methodology similar to what we have developed in [63] is proposed in BMWG. This alternative proposal, however, is only targeting transition technologies which employ translation. Assuming there is sufficient support for this new proposal, the working group could end up discussing the overlap between the two documents and converge towards a single contiguous proposed standard. Depending on the overlap in scope, it is also possible that the two work items are kept separate and the scope is divided in translation and non-translation transition technologies. Both alternate endings can lead to either better or worse technical quality proposals. Nevertheless, there is the risk that a politically motivated decision could end up affecting the technical quality of the document.

7.2 Future work

Counting as one of the core Internet technologies, the Internet Protocol and its transition period will likely shape the future of most associated technologies. Throughout the thesis, we have mentioned various future work directions, which have the potential to become viable research projects themselves. Hereafter, we have highlighted the most important ones in conjunction with other Internet technologies developments.

7.2.1 A Virtualized IPv6NET

Virtual technology is nowadays ubiquitous. However, the shared resources make it very hard to isolate parameters, and by extension obtain reasonable and insightful benchmarking results. Preliminary tests with fully virtualized testers and devices under tests showed as much as 70% performance degradation by comparison with their bare-metal counterparts. This was the reason for using only physical machines in our benchmarking experiments so far. Nevertheless, we expect virtual capable implementations to become the vast majority in the future. That would naturally encompass the world of IPv6 transition technologies as well. Considerations for benchmarking in a virtualized environment are in development in the BMWG working group. Among the relevant documents, we would mention: Considerations for Benchmarking Virtual Network Functions and Their Infrastructure[91], Benchmarking Methodology for SDN Controller Performance[92] and Benchmarking Virtual Switches in OPNFV[93].

Technologies such as the software defined networking standard OpenFlow are already IPv6 capable, and we anticipate that software defined IPv6 transition services will become the norm in the near future. From the operational perspective, virtualization could also become a conformance criteria. In an environment where virtual-capable implementations become the norm, the lack of support for virtualization can limit the integration capabilities of an implementation. In terms of scalability, the virtualization process would facilitate the realism of load scalability tests, by allowing more generous experimental scales. In this context, a virtualized heterogeneous IPv4-IPv6 testbed would be a very useful extension of our work. However, keeping in mind the challenges mentioned above, this will not be however a straightforward task. We envision a period of 1-2 years for the virtual testing specifications to be widely embraced and implemented by the community.

Cloud technology, such as Infrastructure as a service (Iaas) can become an enabling technology of the virtualized IPv6NET. In this future vision, interested researchers can collaborate and evaluate various feasibility aspect of IPv6 and IPv6 transition technologies. The project can become as well a consolidated database of feasibility results and scenario-oriented guidelines.

7.2.2 P3S: Protocol Security Score System

A fundamental hindrance of the proposed threat model approach is represented by the lack of an associated risk quantification step. However, we believe this step to have deeper implications, particularly implementation-specific details which should be considered as

well. This motivates us to continue this work by associating a risk quantification technique to the current threat model.

We envision this technique to be based on a protocol-oriented vulnerability assessment of each transition tuple. To that end, the proposed technique of threat validation through penetration testing can be the starting point. The following impact oriented assessment would, however, require the collaboration of a community of security experts, much like the Common Vulnerability Scoring System was initiated by Shiffman et alia in [94], but is maintained and continuously improved through the efforts of the Forum of Incident Response and Security Teams (FIRST). Consequently, to better refine the concept of the Protocol Security Scoring (P3S), we aim to discuss it in a IPv6 Security oriented group, such as the IPv6 Hackers forum.

The challenges we envision in this context are very similar to the ones that CVSS has been facing, some of which have been publicized in an open letter to FIRST[95] by Eiram et al. Among the ones we consider relevant for P3S, we would mention: the consistency and correctness of the scores, compliance with other scoring systems, the consistency with alternative scoring databases maintained by vendors or other interested parties.

7.2.3 IPv6NET-ready Applications

In the context of operational capability, we have tested the transition tuples' capability to support legacy applications. However, the roles can be reversed: we can test the capability of applications to support certain IPv6 transition technologies.

In the context of a conformance testing framework, part of IPv6NET, we could have an open environment in which various applications can be tested for conformity with certain IPv6 transition technologies (e.g. NAT64/DNS64). This can spark the collaboration with other conformance oriented forums, like the IPv6 ready consortium. Subsequently, we envision a maintained database of IPv6NET-ready applications, with details about the type and number of supported IPv6 transition technologies.

This type of work can influence the development of Internet of things applications, which are looking at IPv6 as main core technology support. We also anticipate that other future technologies, such as information centric networks can be the beneficiary of this envisioned project.

7.2.4 IoT Benchmarks

The Internet of Things (IoT) is the frenetically discussed future vision of the Internet, where IPv6 is seen as one of the biggest enablers. There are efforts to increase the power efficiency of IPv6 datagram exchanges through header compression [96]. If we stretch the concept, we can consider the IP header optimizations, as IPv6 transition technologies. Instead of discussing the impact of translation/encapsulation overhead, we would discuss the compression processing overhead on middle-boxes.

In the emerging context of IoT, we believe that the need for performance evaluation can lead to another extension of our work, IoT tailored benchmarks. One example can be the network performance degradation associated with header compression algorithms.

7.2.5 Back to the Basics: Variable Length Addressing. What if?

The IPv6 transition could have been avoided, if only the developers of IPv4 would have had enough vision to foresee its potential. Let us imagine a parallel universe, where IPv4 was designed to integrate an arbitrary-length address field as defined in [97]. In this universe, our efforts to untangle and evaluate complex transition technologies, could have been redirected instead towards evaluating delay tolerant protocols and associated implementations. We imagine the project name would have been Delay Tolerant Networks Evaluation Testbed (DTNET) and could have resulted in a collection of methodologies and a DTN oriented test framework.

Returning to our fixed length IPv4 address universe, we contend that some of the work could have been considered relevant. For instance, the header processing overhead of variable length addresses can be associated with the overhead introduced by translation or encapsulation. Similarly, the load scalability could have been quantified with the performance degradation at different address lengths.

We do not believe that IPv6 will require a successor any time soon, but we do anticipate that variable length addressing will be the choice in the future Internets. If not for their optimization potential, at least to avoid transition periods like the current one.

Chapter 8

Conclusion

In this thesis we have presented a collection of methodologies for analyzing multiple feasibility dimensions of IPv6 transition technologies. To prove the validity of the proposed methodologies, we have followed them to analyze two transition implementations. The empirical analysis has concentrated on two open source transition implementations, covering multiple transition technologies, Asamap covering MAP-E, MAP-T, DSLite and 464XLAT, and Tiny-map-e, covering MAP-E only. The analysis targeted four major feasibility dimensions of transition technologies: network performance, scalability, security and operational capability. Analyzing the empirical results, we were able to point out Asamap as a more feasible transition technology. Additionally, we were able to identify some performance and load scalability trends within the same implementation, such as the better latency of translation-based tuples (amap464xlat, amapt) or the better throughput of encapsulation-based tuples (amape, amapdslite).

Standardization in the IETF has been a complementary method to validate the proposed methodologies. To that end, our efforts to standardize the network performance and scalability methodologies have materialized in a working group draft [63], developed in the Benchmarking Working Group (BMWG). In an effort to expand our current network performance and scalability methodologies, we intend to consider the ever-increasing trend of virtualization in current production networks. To that end, we plan to build a virtualized testbed which can accommodate software defined transition implementations as well as higher network scales.

As only an individual submission for now, the stride towards standardization for the proposed threat model has began with [64]. As future development of this proposal, we envision a protocol oriented security quantification project (P3S), developed in collaboration with IP security expert groups, such as the IPv6 hackers forum.

In terms of operational capability, however, the methodology has only received review in the academic community. Nevertheless, we intend to expand the project in a collaboration with conformance oriented entities, such as the IPv6 consortium. This project, which we anticipate will be called the *IPv6NET-ready project*, could integrate our current proposals: configuration, troubleshooting and applications capability, as well as new developments. One of these can be an IPv6NET-ready applications, in which applications are tested for conformance in relation to certain IPv6 transition technologies. This can, in turn, have

further uses for IoT and information centric networking (ICN) applications.

As core Internet technologies evolve, we imagine further uses for our proposals. For instance, the current network performance and load scalability methodology can be extended to benchmark low power IoT devices, as well as DTN implementations.

Ultimately we hope that our proposals will contribute to a smoother and faster IPv6 transition for the Internet community. We also expect that our efforts will stand as proof that fixed length addressing schemes are a bad idea for structural scalability. Furthermore, we hope that more visionaries will get involved in leading the development of one of mankind's greatest engineering achievements, the Internet.

Appendix

A 1. List of Publications

1. Journal Articles
   1.1. **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, 2 2015. - Chapter 4
2. Conference Papers
   2.1. **M. Georgescu**, H. Hazeyama, T. Okuda, Y. Kadobayashi, and S. Yamaguchi, "The STRIDE towards IPv6: A Comprehensive Threat Model for IPv6 Transition Technologies," in 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016), February 2016. - Chapter 6
   2.2. **M. Georgescu**, H. Hazeyama, T. Okuda, Y. Kadobayashi, and S. Yamaguchi, "Benchmarking the Load Scalability of IPv6 Transition Technologies: A Black-box Analysis," in Computers and Communication (ISCC), 2015 IEEE Symposium on, pp. 329–334, July 2015. - Chapter 5
   2.3. **M. Georgescu**, H. Hazeyama, Y. Kadobayashi, and S. Yamaguchi. "Empirical Analysis of IPv6 Transition Technologies Using the IPv6 Network Evaluation Testbed," in Testbeds and Research Infrastructure: Development of Networks and Communities, Vol. 137 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 216-228. Springer International Publishing, 2014. - Chapter 4
3. Internet Drafts
   3.1. **M. Georgescu** and G. Lencse, "Benchmarking Methodology for IPv6 Transition Technologies." draft-ietf-bmwg-ipv6-tran-tech-benchmarking-02, July 2016. - Chapter 4, Chapter 5
   3.2. **M. Georgescu**, "The STRIDE towards IPv6: A Threat Model for IPv6 Transition Technologies." draft-georgescu-opsec-ipv6-trans-tech-threat-model-01, July 2016. - Chapter 6
4. Workshop Papers
   4.1. M. Ashar, **M. Georgescu**, T. Sahara, H. Izumikawa, Y. Onogi, M. Tamai

and S. Kashihara, "Comparison of the Current Routing Protocols Availability in DTN2 and IBR-DTN," in Technical Report of IEICE, Vol.113, No.398, MoNA2013-66, pp. 95–96, January 2014.

4.2. **M. Georgescu**, T. Sahara, M. Ashar, H. Izumikawa, Y. Onogi, M. Tamai, S. Kashihara, "Performance Analysis of File Transmission in DTN2 and IBR-DTN," in IEICE Technical Report, vol. 113, No. 398, MoNA2013-49, pp. 1–5, January 2014.

4.3. **M. Georgescu**, H. Hazeyama, Y. Kadobayashi, S. Yamaguchi, "An Empirical Study of IPv6 Transition in An Open Environment - Experiences from WIDE Camp' s Life with IPv6 Workshop," in The Fourteenth Workshop on Internet Technology, June 2013.

A 2. Protocol Codes

Protocol	RFC	Code
IPv4 suite protocols		
Internet Protocol version 4	RFC791	IPv4
Internet Control Message Protocol	RFC792	ICMP
Ethernet Address Resolution Protocol	RFC826	ARP
Reverse Address Resolution Protocol	RFC902	RARP
IP Mobility Support for IPv4	RFC5944	MIPv4
Internet Group Management Protocol	RFC3376	IGMP
IPv6 suite protocols		
Internet Protocol version 6	RFC2460	IPv6
Internet Control Message Protocol for IPv6	RFC4443	ICMPv6
Neighbor Discovery for IP version 6	RFC4861	ND
Optimistic Duplicate Address Detection for IPv6	RFC4429	DAD
IPv6 Stateless Address Autoconfiguration	RFC4862	SLAAC
Multicast Listener Discovery Version 2 for IPv6	RFC3810	MLDv2
Mobility Support in IPv6	RFC3775	MIPv6
Security protocols		
Security Architecture for the Internet Protocol	RFC4301	IPSec
IP Authentication Header	RFC4302	IPAuth
IP Encapsulating Security Payload	RFC4303	ESP
SEcure Neighbor Discovery	RFC3971	SEND
Transport layer protocols		
Transmission Control Protocol	RFC793	TCP
User Datagram Protocol	RFC768	UDP
Basic IPv6 transition technologies		
Dual IP Layer Operation	RFC4213	DS
IP/ICMP Translation Algorithm	RFC6145	IP/ICMP
Encapsulation of IPv6 in IPv4	RFC4213	4encaps
Generic Packet Tunneling in IPv6,Specification	RFC2473	6encaps
Routing protocols		
RIP Version 2	RFC2453	RIPv2
OSPF Version 2	RFC2328	OSPFv2
OSPF for IPv6	RFC5340	OSPFv3
Encapsulation of IPv6 in IPv4	RFC4213	4encaps
Generic IPv6 transition technologies		
Dual Stack	-	DS
Single Translation	-	1transl
Double Translation	-	2transl
Encapsulation	-	encaps

Bibliography

[1] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification." RFC 1883 (Proposed Standard), Dec. 1995. Obsoleted by RFC 2460. 1.1

[2] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification." RFC 2460 (Draft Standard), Dec. 1998. Updated by RFCs 5095, 5722, 5871, 6437, 6564, 6935, 6946. 1.1, 2.1

[3] APNIC, "IPv6 measurements for The World [Online]. Available: "http://labs.apnic.net/ipv6-measurement/Regions/001%20World/" ." 1.2, 2.1

[4] X. Leng, J. Bi, and M. Zhang, "Study on high performance ipv4/ipv6 transition and access service," in *Proceedings of the 4th International Conference on Parallel and Distributed Processing and Applications*, ISPA'06, (Berlin, Heidelberg), pp. 183–194, Springer-Verlag, 2006. 1.2, 2.2

[5] P. Wu, Y. Cui, J. Wu, J. Liu, and C. Metz, "Transition from ipv4 to ipv6: A state-of-the-art survey," *Communications Surveys Tutorials, IEEE*, vol. 15, pp. 1407–1424, Third 2013. 1.2, 2.2

[6] P. Grossetete, C. Popoviciu, and F. Wettling, *Global Ipv6 Strategies: From Business Analysis to Operational Planning*. Cisco Press, first ed., 2008. 1.2, 2.2

[7] NRO, "Free Pool of IPv4 Address Space Depleted [Online]. Available: "http://www.nro.net/news/ipv4-free-pool-depleted"," July 2014. 2.1

[8] APNIC, "Free Pool of IPv4 Address Space Depleted [Online]. Available: "http://www.apnic.net/publications/news/2011/final-8"," July 2014. 2.1

[9] G. Huston, "IPv4 Address Report [Online] Available: http://www.potaroo.net/tools/ipv4/ ," Apr. 2016. 2.1

[10] WorldIPv6Launch, "Launching the future [Online]. Available: "http://www.worldipv6launch.org/infographic/"," July 2014. 2.1, 2.2

[11] E. Nordmark and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers." RFC 4213 (Proposed Standard), Oct. 2005. 2.3, 2.1

[12] F. Baker, X. Li, C. Bao, and K. Yin, "Framework for IPv4/IPv6 Translation." RFC 6144 (Informational), Apr. 2011. 2.3

[13] X. Li, C. Bao, M. Chen, H. Zhang, and J. Wu, "The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Co-existence and Transition." RFC 6219 (Informational), May 2011. 2.3, 2.1

[14] C. Bao, X. Li, Y. Zhai, and W. Shang, "dIVI: Dual-Stateless IPv4/IPv6 Translation." draft-xli-behave-divi-06, Jan. 2014. 2.3, 2.1

[15] M. Bagnulo, P. Matthews, and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers." RFC 6146 (Proposed Standard), Apr. 2011. 2.3, 2.1, 6.3.2

[16] A. Durand, R. Droms, J. Woodyatt, and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion." RFC 6333 (Proposed Standard), Aug. 2011. 2.3, 2.1, 2.4, 2.4.1, 3.3, 3.5

[17] R. Gilligan and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers." RFC 1933 (Proposed Standard), Apr. 1996. Obsoleted by RFC 2893. 2.3

[18] B. Carpenter and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds." RFC 3056 (Proposed Standard), Feb. 2001. 2.3

[19] F. Templin, T. Gleeson, M. Talwar, and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)." RFC 4214 (Experimental), Oct. 2005. Obsoleted by RFC 5214. 2.3

[20] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)." RFC 4380 (Proposed Standard), Feb. 2006. Updated by RFCs 5991, 6081. 2.3

[21] R. Despres, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)." RFC 5569 (Informational), Jan. 2010. 2.3

[22] O. Troan, W. Dec, X. Li, C. Bao, S. Matsushima, T. Murakami, and T. Taylor, "Mapping of Address and Port with Encapsulation (MAP-E)." RFC 7597 (Proposed Standard), July 2015. 2.3, 2.1, 2.4, 2.4.1, 3.1, 3.3, 3.4.1, 3.5, 5.1, 5.3, 5.3, 5.5, 5.5

[23] T. Anderson, "SIIT-DC: Stateless IP/ICMP Translation for IPv6 Data Centre Environments." draft-ietf-v6ops-siit-dc-01, June 2015. 2.1

[24] N. Matsuhira, "SA46T Address Translator." draft-matsuhira-sa46t-at-05, Aug. 2015. 2.1

[25] M. Mawatari, M. Kawashima, and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation." RFC 6877, Apr. 2013. 2.1, 2.4, 2.4.2, 3.3, 3.5

[26] X. Li, C. Bao, W. Dec, O. Troan, , S. Matsushima, T. Murakami, and T. Taylor, "Mapping of Address and Port using Translation (MAP-T)." RFC 7599 (Proposed Standard), July 2015. 2.1, 2.4, 2.4.2, 3.3, 3.5, 5.1, 5.5, 6.3.1

[27] T. Tsou, Y. Cui, M. Boucadair, I. Farrer, and Y. Lee, "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture." RFC 7596 (Proposed Standard), July 2015. 2.1

[28] M. Asama, "MAP supported Vyatta [Online]. Available: "http://enog.jp/ masakazu/vyatta/map/"," July 2014. 2.4, 3.5, 6.3.1, 6.3.2, 6.3.3

[29] Y. Ueno, "Tiny-map-e implementation [Online]. Available: "https://github.com/edenden/tiny-map-e"," July 2014. 2.4, 3.5, 4.2.2

[30] J. Weil, V. Kuarsingh, C. Donley, C. Liljenstolpe, and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space." RFC 6598 (Best Current Practice), Apr. 2012. 2.4.1

[31] A. Lacoste, "MAP simulation tool [Online]. Available: "http://6lab.cisco.com/map/MAP.php" ," July 2014. 2.4.1

[32] I. Raicu and S. Zeadally, "Evaluating ipv4 to ipv6 transition mechanisms," *IEEE International Conference on Telecommunications 2003*, 2003. 2.5.1

[33] S. Narayan, P. Shang, and N. Fan, "Network performance evaluation of internet protocols ipv4 and ipv6 on operating systems," in *Proceedings of the Sixth international conference on Wireless and Optical Communications Networks*, WOCN'09, (Piscataway, NJ, USA), pp. 242–246, IEEE Press, 2009. 2.5.1

[34] S. Sasanus and K. Kaemarungsi, "Differences in bandwidth requirements of various applications due to ipv6 migration," in *Proceedings of the The International Conference on Information Network 2012*, ICOIN '12, (Washington, DC, USA), pp. 462–467, IEEE Computer Society, 2012. 2.5.1

[35] P. Grayeli, S. Sarkani, and T. Mazzuchi, "Performance analysis of ipv6 transition mechanisms over mpls," *International Journal of Communication Networks and Information Security*, vol. 4, no. 2, 2012. 2.5.1

[36] G. Lencse and S. Repas, "Performance analysis and comparison of different dns64 implementations for linux, openbsd and freebsd," in *Proceedings of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications*, AINA '13, (Washington, DC, USA), pp. 877–884, IEEE Computer Society, 2013. 2.5.1

[37] G. Lencse and S. Répás, "Performance analysis and comparison of the tayga and of the pf nat64 implementations," in *Telecommunications and Signal Processing (TSP), 2013 36th International Conference on*, pp. 71–76, IEEE, 2013. 2.5.1

[38] S. Répás, P. Farnadi, and G. Lencse, "Performance and stability analysis of free nat64 implementations with different protocols," *Acta Technica Jaurinensis*, vol. 7, no. 4, pp. 404–427, 2014. 2.5.1

[39] P. Wu, Y. Cui, J. Wu, J. Liu, and C. Metz, "Transition from ipv4 to ipv6: A state-of-the-art survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1407–1424, 2013. 2.5.1, 2.5.3

[40] R. Hiromi and H. Yoshifuji, "Problems on ipv4-ipv6 network transition," in *Proceedings of the International Symposium on Applications on Internet Workshops*, SAINT-W '06, (Washington, DC, USA), pp. 38–42, IEEE Computer Society, 2006. 2.5.2

[41] H. Babiker, I. Nikolova, and K. K. Chittimaneni, "Deploying ipv6 in the google enterprise network lessons learned," in *Proceedings of the 25th international conference on Large Installation System Administration*, LISA'11, (Berkeley, CA, USA), pp. 10–10, USENIX Association, 2011. 2.5.2

[42] J. Arkko and A. Keranen, "Experiences from an IPv6-Only Network." RFC 6586 (Informational), Apr. 2012. 2.5.2, 4.3, 4.4

[43] S. Répás, T. Hajas, and G. Lencse, "Application compatibility of the nat64 ipv6 transition technology," in *Telecommunications and Signal Processing (TSP), 2015 38th International Conference on*, pp. 1–7, IEEE, 2015. 2.5.2

[44] H. Hazeyama, R. Hiromi, T. Ishihara, and O. Nakamura, *Experiences from IPv6-Only Networks with Transition Technologies in the WIDE Camp Spring 2012*, Mar 2012. draft-hazeyama-widecamp-ipv6-only-experience-01.txt. 2.5.2

[45] A. B. Bondi, "Characteristics of scalability and their impact on performance," in *Proceedings of the 2Nd International Workshop on Software and Performance*, WOSP '00, (New York, NY, USA), pp. 195–203, ACM, 2000. 2.5.3

[46] B. Stephens, A. Cox, S. Rixner, and T. Ng, "A scalability study of enterprise network architectures," in *Architectures for Networking and Communications Systems (ANCS), 2011 Seventh ACM/IEEE Symposium on*, pp. 111–121, Oct 2011. 2.5.3

[47] I. Haddad and G. Butler, "Experimental studies of scalability in clustered web systems," in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, p. 185, IEEE, 2004. 2.5.3

[48] T. Deshane, Z. Shepherd, J. Matthews, M. Ben-Yehuda, A. Shah, and B. Rao, "Quantitative comparison of xen and kvm," *Xen Summit, Boston, MA, USA*, pp. 1–2, 2008. 2.5.3

[49] J. Bi, J. Wu, and X. Leng, "Ipv4/ipv6 transition technologies and univer6 architecture," *International Journal of Computer Science and Network Security*, vol. 7, no. 1, pp. 232–242, 2007. 2.5.3

[50] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Threat modeling-uncover security design flaws using the stride approach," *MSDN Magazine-Louisville*, pp. 68–75, 2006. 2.5.4, 6.1, 6.2

[51] OWASP, "Application Threat Modeling." OWASP Foundation, Aug. 2015. 2.5.4, 6.2

[52] A. Gervais, "Security analysis of industrial control systems," *Aalto University-KTH Stockholm, Jun*, vol. 29, 2012. 2.5.4

[53] B. Harris and R. Hunt, "Tcp/ip security threats and attack methods," *Computer Communications*, vol. 22, no. 10, pp. 885–897, 1999. 2.5.4, 6.3, 6.6

[54] F. Gont, "Security assessment of the internet protocol version 4." RFC 6274 (Informational), July 2011. 2.5.4, 6.3

[55] C. Low, "Icmp attacks illustrated," *SANS Institute URL: http://rr. sans. org/threats/ICMP_attacks. php (12/11/2001)*, 2001. 2.5.4, 6.3

[56] C. L. Abad, R. Bonilla, *et al.*, "An analysis on the schemes for detecting and preventing arp cache poisoning attacks," in *Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference on*, pp. 60–60, IEEE, 2007. 2.5.4, 6.3

[57] Z. Khallouf, V. Roca, R. Moignard, and S. Loye, "A Filtering Approach for an IGMP Flooding Resilient Infrastrcuture." 4th Conference on Security and Network Architectures(SAR'05), Batz sur Mer, France, 2005. 2.5.4, 6.3

[58] E. Davies, S. Krishnan, and P. Savola, "IPv6 Transition/Co-existence Security Considerations." RFC 4942 (Informational), Sept. 2007. 2.5.4, 6.5, 6.7

[59] S. Convery and D. Miller, "Ipv6 and ipv4 threat comparison and best-practice evaluation," 2004. 2.5.4

[60] A. Pilihanto, "A complete guide on ipv6 attack and defense," *Sans. org [online]*, 2011. 2.5.4, 6.5

[61] M. Georgescu, H. Hazeyama, Y. Kadobayashi, and S. Yamaguchi', "Empirical analysis of ipv6 transition technologies using the ipv6 network evaluation testbed," in *9th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, 2014. 3, 5.4.5

[62] S. Bradner and J. McQuaid, "Benchmarking methodology for network interconnect devices." RFC 4057 (Informational), 1999. 3.2, 4.1

[63] M. Georgescu and G. Lencse, "Benchmarking methodology for ipv6 transition technologies." draft-georgescu-bmwg-ipv6-tran-tech-benchmarking-01, March 2016. 3.2, 4.6, 5.7, 7.1, 8

[64] M. Georgescu, "The stride towards ipv6: A threat model for ipv6 transition technologies." draft-georgescu-opsec-ipv6-trans-tech-threat-model-00, March 2016. 3.2, 6.5, 7.1, 8

[65] T. Miyachi, K.-i. Chinen, and Y. Shinoda, "Starbed and springos: large-scale general purpose network testbed and supporting software," in *Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, valuetools '06, (New York, NY, USA), ACM, 2006. 3.4.1

[66] A. Botta, A. Dainotti, and A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios," 2012. 3.4.1, 4.1

[67] C. Popoviciu, A. Hamza, G. V. de Velde, and D. Dugatkin, "Ipv6 benchmarking methodology for network interconnect devices," 2008. 3.4.1

[68] "Ieee standard for local and metropolitan area networks–media access control (mac) bridges and virtual bridged local area networks–corrigendum 2: Technical and editorial corrections," *IEEE Std 802.1Q-2011/Cor 2-2012 (Corrigendum to IEEE Std 802.1Q-2011)*, pp. 1–96, Nov 2012. 3.4.1

[69] M. Georgescu, H. Hazeyama, Y. Kadobayashi, S. Yamaguchi, "An empirical study of IPv6 transition in an open environment - experiences from WIDE camp's Life with IPv6 Workshop," in *The Fourteenth Workshop on Internet Technology*, June 2013. 3.4.2

[70] P. Savola, "Mtu and fragmentation issues with in-the-network tunneling." RFC 4459, 2006. 3.5, 5.2, 5.4.3

[71] R. Jain, *Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling*. John Wiley & Sons, May 1991. 4.1, 4.1

[72] M. P. Wand and M. C. Jones, *Kernel smoothing*, vol. 60. Crc Press, 1994. 4.2.1

[73] 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, 2 2015. 4.2.2

[74] M. Georgescu, H. Hazeyama, T. Okuda, Y. Kadobayashi, and S. Yamaguchi, "Benchmarking the load scalability of ipv6 transition technologies: A black-box analysis," in *Computers and Communication (ISCC), 2015 IEEE Symposium on*, pp. 329–334, July 2015. 4.2.2

[75] D. Harrington, "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions." RFC 5706 (Informational), Nov. 2009. 4.3, 4.3

[76] S. Miyakawa, "Ipv4 to ipv6 transformation schemes," *IEICE transactions on communications*, vol. 93, no. 5, pp. 1078–1084, 2010. 5.3

[77] G. Lencse, "Estimation of the port number consumption of web browsing," *IEICE Transactions on Communications*, vol. 98, no. 8, pp. 1580–1588, 2015. 5.3

[78] R. McRee, "IT Infrastructure Threat Modeling Guide." Microsoft Technet, June 2009. 6.2

[79] S. M. Bellovin, "Security problems in the tcp/ip protocol suite," *SIGCOMM Comput. Commun. Rev.*, vol. 19, pp. 32–48, Apr. 1989. 6.3

[80] R. Atkinson and M. Fanto, "RIPv2 Cryptographic Authentication." RFC 4822 (Proposed Standard), Feb. 2007. 6.4

[81] J. Moy, "OSPF Version 2." RFC 2328 (INTERNET STANDARD), Apr. 1998. Updated by RFCs 5709, 6549, 6845, 6860. 6.4

[82] M. Gupta and N. Melam, "Authentication/Confidentiality for OSPFv3." RFC 4552 (Proposed Standard), June 2006. 6.4

[83] A. Conta and M. Gupta, "Internet control message protocol (icmpv6) for the internet protocol version 6 (ipv6) specification." RFC 4443 (Proposed standard), Feb. 2006. 6.5

[84] P. Nikander, J. Kempf, and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats." RFC 3756 (Informational), May 2004. 6.5

[85] J. Arkko, J. Kempf, B. Zill, and P. Nikander, "SEcure Neighbor Discovery (SEND)." RFC 3971 (Proposed Standard), Mar. 2005. Updated by RFCs 6494, 6495. 6.5

[86] A. Garg and A. N. Reddy, "Mitigation of dos attacks through qos regulation," *Microprocessors and Microsystems*, vol. 28, no. 10, pp. 521–530, 2004. 6.6

[87] ITU-T, "ITU-T Rec. X.1037 (10/2013) IPv6 technical security guidelines." Recommendation X.1037, Oct. 2013. 6.6, 6.8

[88] C. Bao, C. Huitema, M. Bagnulo, M. Boucadair, and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators." RFC 6052 (Proposed Standard), Oct. 2010. 6.7

[89] X. Li, C. Bao, and F. Baker, "IP/ICMP Translation Algorithm." RFC 6145 (Proposed Standard), Apr. 2011. Updated by RFC 6791. 6.7

[90] S. A. Rouiller, "Virtual lan security: weaknesses and countermeasures," *available at uploads.askapache.com/2006/12/vlan-security-3. pdf*, 2003. 6.8

[91] A. Morton, "Considerations for Benchmarking Virtual Network Functions and Their Infrastructure," Internet-Draft draft-ietf-bmwg-virtual-net-03, Internet Engineering Task Force, June 2016. Work in Progress. 7.2.1

[92] V. Manral, M. Tassinari, B. Vengainathan, A. Basil, and S. Banks, "Benchmarking Methodology for SDN Controller Performance," Internet-Draft draft-ietf-bmwg-sdn-controller-benchmark-meth-02, Internet Engineering Task Force, July 2016. Work in Progress. 7.2.1

[93] M. Tahhan, B. Mahony, and A. Morton, "Benchmarking Virtual Switches in OPNFV," Internet-Draft draft-ietf-bmwg-vswitch-opnfv-00, Internet Engineering Task Force, July 2016. Work in Progress. 7.2.1

[94] M. Schiffman and C. Cisco, "A complete guide to the common vulnerability scoring system (cvss)," in *Forum Incident Response and Security Teams (http://www. first. org/)*, 2005. 7.2.2

[95] C. Eiram and B. Martin, "The cvssv2 shortcomings, faults, and failures formulation," Technical report, Forum of Incident Response and Security Teams (FIRST), 2013. 7.2.2

[96] J. Hui and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks." RFC 6282 (Proposed Standard), Sept. 2011. 7.2.4

[97] W. Eddy and E. Davies, "Using Self-Delimiting Numeric Values in Protocols." RFC 6256 (Informational), May 2011. 7.2.5