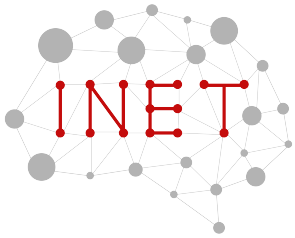

Bachelorthesis

Evaluating Dual-Stack against NAT64 deployment schemes with DNS64 and CLAT

Wodke, Daniel Jin

21. Mai 2025

Gutachter: Prof. Dr. Stefan Schmid
Prof. Dr. Stefan Tai
Betreuerin: Max Franke and Dr. Philipp Tiesel
Matrikelnr.: 456675

Technische Universität Berlin
Faklutät IV - Elektrotechnik und Informatik
Institut für Telekommunikationssysteme
Fachgebiet Intelligent Networks

Eidesstattliche Versicherung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den Date

_____

Wodke, Daniel Jin

Zusammenfassung

Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte. Abgeschieden wohnen Sie in Buchstabhausen an der Küste des Semantik, eines großen Sprachozeans. Ein kleines Bächlein namens Duden fließt durch ihren Ort und versorgt sie mit den nötigen Regelialien. Es ist ein paradiesmatisches Land, in dem einem gebratene Satzteile in den Mund fliegen. Nicht einmal von der allmächtigen Interpunktion werden die Blindtexte beherrscht – ein geradezu unorthographisches Leben. Eines Tages aber beschloß eine kleine Zeile Blindtext, ihr Name war Lorem Ipsum, hinaus zu gehen in die weite Grammatik. Der große Oxmox riet ihr davon ab, da es dort wimmele von bösen Kommata, wilden Fragezeichen und hinterhältigen Semikoli, doch das Blindtextchen ließ sich nicht beirren. Es packte seine sieben Versalien, schob sich sein Initial in den Gürtel und machte sich auf den Weg. Als es die ersten Hügel des Kursivgebirges erklommen hatte, warf es einen letzten Blick zurück auf die Skyline seiner Heimatstadt Buchstabhausen, die Headline von Alphabetdorf und die Subline seiner eigenen Straße, der Zeilengasse. Wehmütig lief ihm eine rethorische Frage über die Wange, dann setzte es seinen Weg fort. Unterwegs traf es eine Copy. Die Copy warnte das Blindtextchen, da, wo sie herkäme wäre sie zigmal umgeschrieben worden und alles, was von ihrem Ursprung noch übrig wäre, sei das Wort "und" und das Blindtextchen solle umkehren und wieder in sein eigenes, sicheres Land zurückkehren.

Abstract

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Contents

Chapter 1

Introduction

The transition from IPv4 to IPv6 has shifted from a far concern to an immediate challenge. With IPv4 address exhaustion and rising costs for public IPv4 addresses in cloud environments, organizations face increasing pressure to adopt IPv6. However, complete migration remains complicated by the reality that many applications and services still depend on IPv4 connectivity, whether due to legacy dependencies or third-party integrations. Two primary approaches have emerged. Dual-Stack maintains both IPv4 and IPv6 connectivity, offering wide compatibility but increasing operational overhead and continuous IPv4 allocation. In contrast, IPv6-only with NAT64/DNS64 and CLAT (464XLAT) enables a smaller IPv4 footprint while maintaining reachability to IPv4 services via translation. While this has worked well in mobile networks, the practical trade-offs for enterprise and cloud workloads remain a subject of debate. Industry collaboration This thesis was conducted in close cooperation with SAP SE and was supervised from the industry side by Dr. Philip Tiesel. This collaboration ensured that the research questions addressed are directly aligned with the real-world challenges faced by large-scale platform engineering teams. The primary motivation comes from the architectural decisions required by projects like SAP Gardener, an open-source initiative for managing Kubernetes clusters at scale. Within SAP's Gardeners ecosystem, teams decide between Dual-Stack clusters and IPv6-only clusters. The choice has implications for cost, operational complexity, and performance.

1.1 Problem Statement

While the conceptual trade-offs between Dual-Stack and IPv6-only with 464XLAT are understood, there is a lack of publicly available, empirical data that quantifies the performance overhead of common, open-source translation implementations. Network architects and platform operators are forced to make long-term architec-

tural decisions based on assumptions or data from differing environments, such as mobile networks. The central problem this thesis addresses is: Does a CLAT implementation create performance disadvantages compared to native Dual-Stack connectivity that would justify the continued reliance on IPv4 infrastructure? The problem is increased by the fact that performance data from mobile networks, where 464XLAT has seen widespread adoption, may not be directly applicable to data center or cloud environments. Mobile networks operate under different constraints, with specialized hardware and traffic patterns that differ significantly from the high-throughput, low-latency requirements common in enterprise applications. Moreover, the choice between translation mechanisms is not only about raw performance numbers. Different implementations operate at different layers of the network stack—some in kernel space, others in userspace—each with implications for CPU utilization and integration complexity. Without concrete performance data, teams cannot make informed decisions about which approach best fits their specific use case. The practical impact of this knowledge gap extends beyond technical considerations. In cloud environments where IPv4 addresses cause direct costs, the performance penalty of translation mechanisms directly influences the economic viability of IPv6-only deployments. If the overhead is minimal, organizations can confidently transition to IPv6-only architectures and reduce their IPv4 footprint. However, if translation introduces significant performance degradation, the operational costs may outweigh the savings from reduced IPv4 usage.

1.2 Objectives and Scope

The objective of this thesis is to produce empirical evidence on the performance trade-offs between native Dual-Stack connectivity and IPv6-only deployments that rely on NAT64/DNS64 with a client-side translator (CLAT) as defined by 464XLAT. In practical terms, the work aims to answer whether a CLAT-based path introduces a measurable performance disadvantage that would argue against replacing Dual-Stack in enterprise and cloud contexts. To address this question, the thesis focuses on quantifying the overhead of translation relative to a Dual-Stack baseline using two fundamental network metrics: throughput and round-trip time. Throughput is measured with iperf, and RTT is measured with ping, providing a straightforward and reproducible basis for comparison. The evaluation concentrates on open-source software components that reflect different implementation strategies in the network stack: Jool as a stateful kernel-space NAT64, Tayga as a stateless user-space NAT64, and Tundra as a stateless user-space NAT64. The measurements are carried out across three environments: an AWS cloud setup, a single Ubuntu machine where translators run in separate Linux network namespaces on one client host, and a two-host Ubuntu setup connected via Ethernet with the iperf

server on the second machine. The scope of the work is intentionally narrow to keep the measurements focused and comparable. The metrics are limited to TCP throughput and ICMP RTT. Tail latency, per-packet loss under sustained load, and application-level behaviors are not evaluated. Likewise, the study does not include a systematic analysis of CPU utilization, power consumption, or a cost model. DNS resolution performance itself is not measured. The translators under test are software components typical of general Linux systems. Other transition mechanisms such as SIIT-DC are not considered, since the goal is to compare Dual-Stack with the specific IPv6-only approach built around NAT64/DNS64 and CLAT. Baselines and roles are defined as follows. The Dual-Stack baseline represents native IPv4/IPv6 connectivity without any translation. The CLAT-based path represents an IPv6-only client that uses a local NAT46 function (CLAT). All translators are placed in separate network namespaces to ensure isolation and to make it possible to observe effects to the corresponding implementation. This layout is kept consistent across the three environments. Several limitations follow from this design. Because only iperf and ping are used, the results primarily capture bulk data throughput and a basic latency profile. They do not fully characterize behavior under bursty traffic patterns, short flows, or high packet rate scenarios. The choice of software-only translators on Linux narrows the conclusions to deployments that resemble this setup; organizations using hardware offload may see different results. Moreover, the study does not attempt to evaluate security aspects such as application-level gateways or resilience under failures. While these topics are important in practice, including them would shift the focus from the central performance question. A discussion of limitations and their implications is provided in Chapter 6 - Conclusion. The work is designed to be reproducible. Configurations, scripts, and environment details are documented and collected in the appendix so that others can repeat the measurements or extend them with additional metrics. Chapter 4 - Experiment Design details the setups and methodology, including the namespace configuration on the single-host setup and the role of the iperf server on the second host in the dual Ubuntu environment.

1.3 Thesis Structure

This thesis proceeds from motivation and context to reproducible measurement and evidence-based conclusions in a linear fashion. Following the introduction, which frames the problem and outlines the industry motivation in cooperation with SAP, Chapter 2 provides the technical background required to interpret the results. It revisits IPv4 exhaustion and the ongoing transition to IPv6, contrasts Dual-Stack with IPv6-only strategies, and introduces the principles of NAT64/DNS64 together with the role of client-side translation in 464XLAT to explain how IPv4-

only endpoints remain reachable from IPv6-only clients. The chapter closes with a description of the software components used in the experiments—Jool, Tayga and Tundra. Chapter 3 presents related work and identifies the specific gap this study addresses by reviewing key studies on IPv6 transition mechanisms, summarizing reported performance characteristics, before pinpointing the lack of publicly available empirical data for the selected open-source implementations in the environments considered here. Chapter 4 then explains the experiment design by introducing the three test environments—an AWS setup, a single Ubuntu host with translators running in separate network namespaces, and a dual-host Ubuntu setup connected via Ethernet with the iperf server on the second machine—documenting how Tayga, Tundra, and Jool are set up, how namespaces and routing are configured, and how measurements are taken with iperf for throughput and ping for RTT, and discussing practical challenges encountered during setup, such as clocksource differences and their effect on timing, together with the steps taken to mitigate them. Chapter 5 presents the test results and evaluation, showing throughput and RTT outcomes per environment, analyzing the translators against the Dual-Stack baseline, synthesizing findings across environments and providing a summary comparison of Jool, Tayga, and Tundra. Finally, Chapter 6 concludes the thesis by summarizing the main findings with respect to the central question, outlining practical implications engineering teams considering IPv6-only with NAT64/DNS64 and CLAT, reflecting on limitations arising from the chosen metrics, software focus, and environments, and suggesting directions for future work, while the document ends with references and an appendix containing configuration files and scripts, and result tables to support reproducibility.

Chapter 2

Background

Internet connectivity is based on a straightforward idea: endpoints exchange packets using globally reachable addresses. For most of the Internets history, those addresses have been IPv4, and the surrounding ecosystem: routing practices, DNS operations, firewall policies, and application behavior: grew up around that assumption. As networks expanded and diversified, administrators turned to private addressing and NAT to stretch limited IPv4 space. Over time these workarounds stopped being exceptional and became part of the baseline, with many applications implicitly depending on NAT and other middleboxes rather than a clean end-to-end model. The limits of this approach are structural. IPv4 offers a 32-bit address space of about 4.3 billion addresses, and a noticeable amount is reserved for private use, special purposes, or infrastructure. To keep growth possible, operators widely adopted NAT44 and eventually carrier-grade NAT, centralizing state and rewriting addresses to multiply supply. These techniques do conserve addresses, but they introduce edge cases at the protocol level and make debugging or policy enforcement more complex. It is worth addressing a common question: what happened to "IPv5"? The name informally refers to the experimental Internet Stream Protocol (ST and later ST2), which used protocol number 5 in the IP header. It was designed for connection-oriented streams and quality-of-service experiments, not as a general replacement for IPv4. Importantly, it retained the 32-bit addressing model and never achieved wide deployment on the public Internet. When the IETF set out to design the next general-purpose Internet Protocol, it skipped over the "5" label already associated with ST and standardized IPv6 as the successor to IPv4. IPv6 was designed to remove IPv4's constraints rather than to extend them incrementally. Its 128-bit address space (2^{128} addresses, or roughly 340 undecillion) exceeds IPv4s 32-bit space, an increase by a factor of 2^{96}. The protocol revises addressing and neighbor discovery and in typical deployments avoids the need for NAT. Even so, the shift cannot happen overnight. The global free pools of IPv4 addresses were exhausted at different times—IANA in 2011, ARIN in 2015, and RIPE NCC

in 2019—and IPv6 enablement has progressed unevenly across networks, content providers, and regions. As a result, the Internet has been living in a coexistence phase. Many networks and services speak IPv6, yet IPv4-only systems are still common enough that compatibility must be maintained. The real-world effects come down to two things: IPv4 addresses are running out, and the shift to IPv6 is happening slowly.

2.1 IPv4 exhaustion and transition to IPv6

Internet use has expanded to critical infrastructure across consumer, enterprise, and public sectors. Services running at any time of the day and the increase of connected devices have driven steady growth in traffic and endpoints, making address management a central concern[1], [2]. IP addresses perform two fundamental roles: identifying endpoints and enabling packet delivery across networks, and uniqueness at global scale is essential[2]. IPv4, standardized in 1981-1983, provides a 32-bit address space of roughly 4.3 billion addresses[3].In practice, not all addresses are usable on the public Internet due to special-use and private allocations, and early design choices further reduced the effectively usable pool[1], [2], [4]. Management techniques such as CIDR and NAT slowed the pace of consumption but could not eliminate the finite limit[1]. Exhaustion means that the pool of unused IPv4 addresses has run out, not that IPv4 connections themselves have stopped working. The process started at the global level and then moved downward: IANA allocated its final IPv4 blocks on 3 February 2011, after that, each Regional Internet Registry (RIR) moved into its final phase: APNIC in April 2011, RIPE NCC in September 2012, LACNIC in June 2014 while AFRINIC held on the longest[2]. A global policy passed on 6 May 2012 established mechanisms for the recovery and redistribution of returned IPv4 address space, yet scarcity has continued to persist[1]. The impact has been uneven across regions because historical allocations left some operators with far fewer addresses per user than others[2]. The shortage was unavoidable because demand kept rising: by 2014, about 2.9 billion people were online, with more than 200 million new users joining each year after 2010. Internet use jumped from less than 1

IPv6 was standardized in 1995 as the long-term successor, expanding addressing to 128 bits—on the order of 3.4×10^{38} unique addresses—and introducing protocol-level improvements aimed at routing scalability, mobility support, and operational security[1], [2], [5]. The allocation data highlights how abundant IPv6 is compared to IPv4, even when large IPv6 distributions are considered[1]. However, IPv4 and IPv6 don't work together on their own, so bridging mechanisms are needed while both remain in use[2]. Even though the technical benefits of IPv6 were clear, its adoption lagged behind the urgency created by IPv4 shortages. IPv6 allocations

reported by the RIRs lagged behind user growth, with especially low adoption in some regions, like Africa. At the same time, user-side measurements, such as those from Google, showed less than 10Operators and policymakers have followed three main approaches: making better use of IPv4, allocating the remaining IPv4 space more efficiently and transitioning to IPv6[2]. On the technical side, NAT, especially carrier-grade NAT, conserves public IPv4 addresses by multiplexing many private hosts behind a smaller set of public addresses[6]. By around 2014, a measurable fraction of users were estimated to traverse CGN[7]. While NAT is effective at saving IP addresses, it can make end-to-end connectivity more complicated and add to operational complexity[8]. Dual stack (running IPv4 and IPv6 in parallel) is broadly available and will remain common for years, but it does not address the decline of IPv4[2]. Policy measures like smaller allocations and efforts to reclaim unused addresses have helped ease IP address scarcity somewhat, but they don't eliminate the need for IPv6[2]. Strategically this means that IPv4 and IPv6 will continue to coexist for a long time, connected through integration methods[1], [2]. This situation drives the focus of this thesis: as network operators decide whether to keep dual-stack setups or move to IPv6-only access with translation technologies (like NAT64/DNS64, often paired with CLAT on the client side), it's important to understand the performance trade-offs involved. The next section takes a look at the main transition mechanisms that form the basis of this comparison[1], [2].

2.2 IPv6 transition mechanisms

2.3 Software implementations of NAT64

Chapter 3

Related Work

Chapter 4

Implementation

4.1 Test environments

4.2 Implementation of Translation Technologies

4.3 Networking Namespace Configuration

4.4 Measurement Methodology

4.5 Challenges and Solutions

Chapter 5

Results

5.1 Throughput

5.2 RTT

5.3 Discussion

Chapter 6

Evaluation

6.1 RTT

6.2 Throughput

6.3 TSC vs HPET Clocksource Difference

Chapter 7

Conclusion

Chapter 8

References

Appendix

Bibliography

[1] J. Beeharry and B. Nowbutsing, "Forecasting ipv4 exhaustion and ipv6 migration," in *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)*, 2016, pp. 336–340. DOI: 10.1109/EmergiTech.2016.7737362.

[2] S. L. Levin and S. Schmidt, "Ipv4 to ipv6: Challenges, solutions, and lessons," *Telecommunications Policy*, vol. 38, no. 11, pp. 1059–1068, 2014, ISSN: 0308-5961. DOI: https://doi.org/10.1016/j.telpol.2014.06.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0308596114001128.

[3] *Internet Protocol*, RFC 791, Sep. 1981. DOI: 10.17487/RFC0791. [Online]. Available: https://www.rfc-editor.org/info/rfc791.

[4] R. Moskowitz, D. Karrenberg, Y. Rekhter, E. Lear, and G. J. de Groot, *Address Allocation for Private Internets*, RFC 1918, Feb. 1996. DOI: 10.17487/RFC1918. [Online]. Available: https://www.rfc-editor.org/info/rfc1918.

[5] D. S. E. Deering and B. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 1883, Dec. 1995. DOI: 10.17487/RFC1883. [Online]. Available: https://www.rfc-editor.org/info/rfc1883.

[6] M. Holdrege and P. Srisuresh, *IP Network Address Translator (NAT) Terminology and Considerations*, RFC 2663, Aug. 1999. DOI: 10.17487/RFC2663. [Online]. Available: https://www.rfc-editor.org/info/rfc2663.

[7] I. Livadariu, K. Benson, A. Elmokashfi, A. Dhamdhere, and A. Dainotti, "Inferring carrier-grade nat deployment in the wild," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, IEEE, 2018, pp. 2249–2257.

[8] T. L. Hain, *Architectural Implications of NAT*, RFC 2993, Nov. 2000. DOI: 10.17487/RFC2993. [Online]. Available: https://www.rfc-editor.org/info/rfc2993.

List of Figures

List of Listings