Technische Universität München
TUM School of Computation, Information and Technology

Detecting and Evaluating QUIC Deployments as Part of the Internet Ecosystem

Johannes Christopher Zirngibl

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Jörg Ott

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Georg Carle
2. Prof. Dr. Kimberly C. Claffy
3. Prof. Anja Feldmann, Ph.D.

Die Dissertation wurde am 11.06.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 16.01.2025 angenommen.

Abstract

The Internet is a complex ecosystem due to the interaction of various interdependent protocols and involved parties. Deploying new protocols needs to be done with care and further increases the overall complexity. With the standardization of QUIC, a new, fundamental network protocol was designed, which is expected to drastically influence the Internet ecosystem. QUIC challenges existing transport protocols, *e.g.*, TCP but also unreliable transport. It directly integrates TLS and stream functionality and is designed as a general purpose protocol. This work provides tools to identify and evaluate QUIC. Furthermore, the work highlights properties of QUIC deployments and supported functionality within the Internet ecosystem. It relies on active measurement studies, which help to understand the ecosystem, changes, and future challenges. However, the complexity of the Internet prevents an isolated examination of QUIC deployments without consideration of other protocols. This work follows an active measurement approach and covers QUIC but also the IP layer and Domain Name System (DNS).

In the first part of this thesis, we investigate IP measurements with a focus on IPv6. While full IPv4 address space scans are feasible, IPv6 scans can not easily be done. However, with the depletion of IPv4, the importance of IPv6 increases, and research has to consider it. With the IPv6 Hitlist, a vital tool to scan IPv6 was created. This work cleans the IPv6 Hitlist, analyzes its development, and improves its quality for future use. With the collection of new IP address sources and the application of IPv6 target generation algorithms, the number of responsive addresses within the hitlist was increased from 2.9M to 19.3M.

In the second part of this thesis, we analyze the DNS ecosystem and new DNS resource records, namely `SVCB` and `HTTPS`. From our research perspective, these records are especially of value in the context of QUIC scans because they can be used to indicate within DNS that a service can be reached using QUIC. Furthermore, the impact of domain parking, a mechanism to monetize unused domains, was quantified. We showed that up to 30% of domains are parked. Domain parking affects new generic Top Level Domains (TLDs) but also popular TLDs, *e.g.*, `.com`, and even top lists. Due to their limited value to normal Internet users, they should be treated differently in research and are not necessarily important for higher-layer protocol scans.

In the third part of this thesis, we focus on QUIC. We provide tools to identify and evaluate QUIC deployments. We show widespread deployment of QUIC shortly before the release of RFC9000 in 2021 and an increase in the following years. Furthermore, our tools can identify used parameters and QUIC server libraries. The complexity of the protocol and the variety of implementations impact scans and tools, and proper testing and evaluation are required. We performed Internet-wide scans and identified at least one deployment for 18 QUIC libraries. The developed approach can identify the libraries with 8.0M IPv4 and 2.5M IPv6 addresses. Our approach provides a comprehensive view of the landscape of competing QUIC libraries.

In conclusion, we evaluated three significant parts of the Internet ecosystem, focusing on identifying and evaluating interesting QUIC deployments. However, our findings can also be generalized. They should be applied during scans of new protocols in the future besides QUIC: *(i)* While exhaustive IPv6 scans are still an unsolved task, the IPv6 Hitlist provides a good foundation for scans, and introduced improvements are essential for future research. *(ii)* The DNS ecosystem is an important source of information for the Internet ecosystem and can be used for scans. However, its value depends on the data quality and used domains. *(iii)* Scans for new protocols need to be done carefully, considering involved implementations and their interpretation of standards and thus, impact on scans. A good tool set to identify and analyze deployments is essential.

Zusammenfassung

Das Internet ist aufgrund der Interaktion verschiedener voneinander abhängiger Protokolle und beteiligter Parteien ein komplexes Ökosystem. Die Einführung neuer Protokolle muss mit Bedacht erfolgen und erhöht die Gesamtkomplexität des Internets. Mit der Standardisierung von QUIC wurde ein neues, grundlegendes Netzwerkprotokoll entwickelt, das das Internet-Ökosystem drastisch beeinflussen wird. QUIC stellt bestehende Transportprotokolle in Frage, z. B. TCP, aber auch verbindungslose Protokolle. Diese Arbeit bietet Werkzeuge zur Identifizierung und Bewertung von QUIC. Darüber hinaus werden die Eigenschaften von QUIC und die unterstützten Funktionen innerhalb des Internet-Ökosystems herausgestellt. Diese Arbeit stützt sich auf aktive Messungen, die helfen, das Ökosystem, Veränderungen und zukünftige Herausforderungen zu verstehen. Die Komplexität des Internets verhindert jedoch eine isolierte Untersuchung von QUIC im Internet ohne Berücksichtigung anderer Protokolle. Diese Arbeit deckt neben QUIC auch die IP-Schicht und das DNS ab.

Im ersten Teil dieser Arbeit untersuchen wir IP-Messungen mit Schwerpunkt auf IPv6. Während der vollständige IPv4 Adressraum gescannt werden kann, sind komplette IPv6 Scans nicht möglich. Durch den Mangel an IPv4 Adressen nimmt die Bedeutung von IPv6 jedoch zu und muss von der Forschung berücksichtigt werden. Mit der IPv6-Hitliste wurde ein wichtiges Instrument zum Scannen von IPv6 geschaffen. Diese Arbeit bereinigt die IPv6-Hitliste, analysiert ihre Entwicklung und verbessert ihre Qualität für die zukünftige Nutzung. Mit der Sammlung neuer IP-Adressquellen und der Anwendung von Algorithmen zur Generierung neuer IPv6 Zieladressen konnte die Anzahl der antwortenden Adressen in der Hitliste von 2,9M auf 19,3M erhöht werden.

Im zweiten Teil dieser Arbeit analysieren wir das DNS-Ökosystem und neue DNS-Einträge, namentlich `SVCB` und `HTTPS`. Aus unserer Forschungsperspektive sind diese Einträge vor allem im Zusammenhang mit QUIC-Scans wertvoll, da sie bereits im DNS die Information übermitteln, dass ein Dienst über QUIC erreichbar ist. Darüber hinaus werden die Auswirkungen von Domain-Parking, einem Mechanismus zur Monetarisierung ungenutzter Domains, quantifiziert. Wir konnten zeigen, dass bis zu 30% der Domains geparkt werden. Domain-Parking betrifft neue generische Top Level Domains (TLDs), aber auch populäre TLDs, z. B. `.com`, und sogar Toplisten. Aufgrund ihres begrenzten Wertes für Internetnutzer sollten sie in der Forschung differenziert behandelt werden.

Im dritten Teil dieser Arbeit konzentrieren wir uns auf QUIC. Wir stellen Werkzeuge zur Verfügung, um Server zu identifizieren, die QUIC unterstützen und diese zu bewerten. Wir zeigen einen weit verbreiteten Einsatz von QUIC kurz vor der Veröffentlichung von RFC9000 im Jahr 2021 und einen Anstieg in den folgenden Jahren. Außerdem können unsere Werkzeuge die verwendeten Parameter und QUIC-Server-Bibliotheken identifizieren. Die Komplexität des Protokolls und die Vielfalt der Implementierungen wirken sich auf Scans und Tools aus, und es sind angemessene Tests und Bewertungen erforderlich. Wir haben Internet-weite Scans durchgeführt und mindestens eine Implementierung für 18 QUIC-Bibliotheken identifiziert. Der entwickelte Ansatz identifiziert die verwendete Bibliothek von 8,0M IPv4 und 2,5M IPv6 Adressen. Unser Ansatz bietet einen umfassenden Überblick über die Landschaft der konkurrierenden QUIC-Bibliotheken.

Zusammenfassend lässt sich sagen, dass wir drei wichtige Teile des Internet-Ökosystems bewertet haben, wobei wir uns auf die Identifizierung und Bewertung interessanter QUIC-Server konzentriert haben. Unsere Erkenntnisse können jedoch auch verallgemeinert werden. Dies sollte bei zukünftigen Analysen neuer Protokolle angewandt werden: *(i)* Während komplette IPv6-Scans immer noch eine ungelöste Aufgabe sind, bietet die IPv6-Hitliste eine gute Grundlage für Scans, und eingeführte Verbesserungen sind für zukünftige Forschung unerlässlich. *(ii)* Das DNS-Ökosystem ist eine wichtige Informationsquelle

für das Internet-Ökosystem und muss für weitere Scans verwendet werden. *(iii)* Scans neuer Protokolle müssen sorgfältig durchgeführt werden, unter Berücksichtigung der beteiligten Implementierungen und ihrer Interpretation der Standards. Ein gutes Set an Werkzeugen zur Identifizierung und Analyse von Implementierungen ist unerlässlich.

Contents

Chapter 1

Introduction

The Internet is an ever-growing network representing a fundamental infrastructure of modern society. It provides the basis for modern communication (*e.g.*, mail, messaging, or video communication), entertainment (*e.g.*, streaming or gaming), or automation (*e.g.*, (industrial) Internet of Things). The variety of use cases, the increase of use cases, and the growth of users comes with consequences: a drastically increasing number of connected devices, services and rapidly developing requirements. To accommodate this development, new protocols and extensions are regularly designed and deployments are quickly updated.

A significant development is the newly designed QUIC protocol. It was finally standardized in 2021 [1] after several years of discussion, the development of different implementations and interoperability tests. QUIC tackles different requirements, for example, *(i)* security and authentication by directly including Transport Layer Security (TLS), *(ii)* reduced latency by combining handshakes from different layers, *(iii)* multiplexing within single connections through streams, and *(iv)* rapid deployment through a user space approach. Further extensions to support multipath connections [2], proxies [3], [4], or better support for media services [5] are designed.

However, QUIC can not solve all requirements and challenges and is tightly integrated into the Internet ecosystem. To tackle the growing number of users and connected devices, IPv6 has been adopted by more and more providers. Even though it was already standardized in 1998 [6], it only saw more widespread adoption since the depletion of IPv4. Furthermore, an increased importance of domains and the DNS is introduced by new protocols. For example, within QUIC, there is a need to differentiate services hosted on the same infrastructure. Furthermore, deploying new protocols with similar features in parallel requires service information besides the IP address for an educated protocol selection, *e.g.*, information whether Hypertext Transfer Protocol (HTTP) version 2 or 3 is supported based on DNS [4].

Essential tools for researchers to analyze protocols and their deployments are Internet-wide scans. However, new protocols require new methodologies and tools to scan and evaluate deployments. Furthermore, due to the integration of individual protocols into the complex ecosystem

of the Internet, other fundamental protocols must be understood, identified, and scanned. Each component brings its challenges and can potentially induce its own biases.

1.1 Research Questions and Goals

This thesis aims to provide insights into deployments of the new protocol QUIC. It answers the following research questions:

Q1: What is the state of QUIC deployments?

- Q1.1: How can we identify QUIC deployments?
- Q1.2: Who deploys QUIC?
- Q1.3: Which libraries are used, and how are servers configured?

This work provides insights into QUIC deployments and the means to analyze the protocol development on the Internet. To provide an extensive view on the QUIC ecosystem, this thesis additionally covers dependencies, namely IPv6 and DNS, *e.g.*, new `HTTPS` [7] resource records, and evaluates potential biases, *e.g.*, fully responsive prefixes and domain parking.

Q2: How can the IPv6 ecosystem be scanned?

- Q2.1: What is the current state of IPv6 scans?
- Q2.2: How can we identify new, interesting IPv6 deployments?

Q3: What is the impact of DNS?

- Q3.1: How can DNS help during the detection of deployments?
- Q3.2: Which biases are induced by DNS?

Based on the research questions, the following goals have been derived. Answers will be presented alongside these goals in the remaining work. To answer these questions, we designed approaches and required tools to effectively detect and evaluate QUIC deployments and identify different QUIC libraries.

G1: Methodologies to Identify IPv6 Deployments.
The first goal of this work answers the second research question and focuses on IPv6. Out of the considered dimensions, it is the most fundamental Internet protocol. The goal is to allow the analysis and scans of the IPv6 ecosystem alongside IPv4. The increasing relevance of IPv6 makes it inevitable to cover its ecosystem in research. Therefore, means to identify and scan deployments must be available because full /0 scans are infeasible. The *IPv6 Hitlist* by Gasser *et al.* [8] is an ongoing service that provides others with lists of responsive IPv6 addresses. Still, maintenance, improvements, and thorough evaluations are required to prepare the service for the following years.

G2: Evaluation of the Impact of DNS on Internet-wide Scans.
DNS is an essential source of information not only for users but also for researchers and as an input for higher-layer protocols and scans. Domain resolutions can be used to analyze the

general state of the ecosystem, its centralization, and usage. Furthermore, it is often used to identify targets for higher-layer protocol scans, *e.g.*, focusing on top lists or using full zone files.

Due to servers hosting multiple domains on the same instance, clients have to indicate the expected service during the connection establishment. To reduce the handshake time, this indication is directly integrated into TLS as Server Name Indication (SNI). Regarding QUIC, this results in domains being important information already in the client's first packet.

Therefore, the second goal of this work is to analyze the DNS ecosystem, potential biases, and the impact of new Domain Name System Resource Records (DNS RRs) on scans. It seeks answers to the third research question.

G3: Identification and Evaluation of QUIC Deployments and TLS Properties.
QUIC incorporates TLS and functionality from the TCP. However, due to fundamental differences, existing methodologies to analyze deployments can not easily be used. Furthermore, it is built on top of the User Datagram Protocol (UDP), hampering existing tools (*e.g.*, ZMap) to identify deployments.

Therefore, the third goal of this work is to analyze QUIC deployments on the Internet and answer the first and most important research question of this work. This goal is divided into multiple steps. The first step is to develop methodologies to identify servers supporting QUIC (including the IPv4 and IPv6 ecosystem). Secondly, methodologies to analyze QUIC deployments in detail, including their configurations and used libraries, should be developed and tested. Finally, Internet-wide scans based on the previous goals should be used to evaluate the state of QUIC deployments on the Internet on a large scale. Besides the analysis, this goal includes creating and publishing tools as a foundation for future research and continuous evaluation.

Table 1.1: Structure of this thesis and a mapping of chapters to research goals.

	Part I: IP Measurements	
G1	State of the IPv6 Hitlist	Chapter 4
	Target generation algorithms	Chapter 5
	Part II: DNS Measurements: A General Scan Foundation	
G2	Potential of new Domain Name System Resource Records	Chapter 6
	Influence of domain parking	Chapter 7
	Part III: An Evaluation of QUIC Deployments	
G3	Detection and evaluation of QUIC deployments	Chapter 8
	Identification of used QUIC libraries	Chapter 9

1.2 Structure of this Thesis

The accomplishments towards each research goal and supporting work are split into three larger parts. They are structured along three major protocols relevant to the analysis of current services on the Internet. An overview of the main parts and corresponding objectives can be seen in Table 1.1.

Chapter 2 introduces the Global INternet Observatory (GINO) as the foundation for all conducted scans in this work. Chapter 3 presents related work for the remaining thesis. The thesis covers Internet Protocol measurements and evaluations with a focus on version 6 in Part I to tackle *Goal 1*. Afterward, the thesis evaluates the DNS ecosystem in Part II, focusing on new DNS RRs and biases to tackle *Goal 2*. Finally, Part III covers the detection and analysis of the new protocol QUIC, including TLS, as defined in *Goal 3*. Part IV summarizes and concludes this thesis and presents possible future work.

1.3 Publications in the Context of this Thesis

The following peer-reviewed publications, (co-)authored by the author of this thesis are part of this work:

> J. Zirngibl, P. Buschmann, P. Sattler, B. Jaeger, J. Aulbach, and G. Carle, "It's over 9000: Analyzing early QUIC Deployments with the Standardization on the Horizon," in *Proc. ACM Internet Measurement Conference (IMC)*, 2021. DOI: 10.1145/3487552.3487826

> J. Zirngibl, S. Deusch, P. Sattler, J. Aulbach, G. Carle, and M. Jonker, "Domain Parking: Largely Present, Rarely Considered!" In *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2022

> J. Zirngibl, L. Steger, P. Sattler, O. Gasser, and G. Carle, "Rusty Clusters? Dusting an IPv6 Research Foundation," in *Proc. ACM Internet Measurement Conference (IMC)*, 2022. DOI: 10.1145/3517745.3561440

> J. Zirngibl, P. Sattler, and G. Carle, "A First Look at SVCB and HTTPS DNS Resource Records in the Wild," in *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2023. DOI: 10.1109/EuroSPW59978.2023.00058

> L. Steger, L. Kuang, J. Zirngibl, G. Carle, and O. Gasser, "Target Acquired? Evaluating Target Generation Algorithms for IPv6," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2023. DOI: 10.23919/TMA58422.2023.10199073

P. Sattler, J. Zirngibl, M. Jonker, O. Gasser, G. Carle, and R. Holz, "Packed to the Brim: Investigating the Impact of Highly Responsive Prefixes on Internet-wide Measurement Campaigns," *Proc. ACM Netw.*, 2023. DOI: 10.1145/3629146

J. Zirngibl, F. Gebauer, P. Sattler, M. Sosnowski, and G. Carle, "QUIC Hunter: Finding QUIC Deployments and Identifying Server Libraries Across the Internet," in *Proc. Passive and Active Measurement (PAM)*, 2024. DOI: 10.1007/978-3-031-56252-5_13

The following papers were co-authored, but are not directly part of this work. However, they influenced the work of the author and are related to the presented topic in this work. A list of student theses advised by the author of this thesis can be found in Appendix A.1.

Peer-reviewed

- F. Franzen, L. Steger, J. Zirngibl, and P. Sattler, "Looking for Honey Once Again: Detecting RDP and SMB Honeypots on the Internet," in *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2022. DOI: 10.1109/EuroSPW55150.2022.00033
- P. Sattler, J. Aulbach, J. Zirngibl, and G. Carle, "Towards a Tectonic Traffic Shift? Investigating Apple's New Relay Network," in *Proc. ACM Internet Measurement Conference (IMC)*, 2022. DOI: 10.1145/3517745.3561426
- M. Sosnowski, J. Zirngibl, P. Sattler, G. Carle, C. Grohnfeldt, M. Russo, and D. Sgandurra, "Active TLS Stack Fingerprinting: Characterizing TLS Server Deployments at Scale," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2022
- M. Sosnowski, J. Zirngibl, P. Sattler, and G. Carle, "DissecTLS: A Scalable Active Scanner for TLS Server Configurations, Capabilities, and TLS Fingerprinting," in *Proc. Passive and Active Measurement (PAM)*, 2023. DOI: 10.1007/978-3-031-28486-1_6
- B. Jaeger, J. Zirngibl, M. Kempf, K. Ploch, and G. Carle, "QUIC on the Highway: Evaluating Performance on High-Rate Links," in *IFIP Networking Conference (IFIP Networking)*, 2023. DOI: 10.23919/IFIPNetworking57963.2023.10186365
- J. Naab, P. Sattler, J. Zirngibl, S. Günther, and G. Carle, "Gotta Query 'Em All, Again! Repeatable Name Resolution with Full Dependency Provenance," in *Proceedings of the Applied Networking Research Workshop*, 2023. DOI: 10.1145/3606464.3606478
- S. Bauer, P. Sattler, J. Zirngibl, C. Schwarzenberg, and G. Carle, "Evaluating the Benefits: Quantifying the Effects of TCP Options, QUIC, and CDNs on Throughput," in *Proceedings of the Applied Networking Research Workshop*, 2023. DOI: 10.1145/3606464.3606474
- M. Sosnowski, J. Zirngibl, P. Sattler, G. Carle, C. Grohnfeldt, M. Russo, and D. Sgandurra, "EFACTLS: Effective Active TLS Fingerprinting for Large-scale Server Deployment Characterization," *IEEE Transactions on Network and Service Management*, 2024. DOI: 10.1109/TNSM.2024.3364526

- M. Sosnowski, P. Sattler, J. Zirngibl, T. Betzer, and G. Carle, "Propagating Threat Scores With a TLS Ecosystem Graph Model Derived by Active Measurements," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, May 2024
- M. Kempf, N. Gauder, B. Jaeger, J. Zirngibl, and G. Carle, "A Quantum of QUIC: Dissecting Cryptography with Post-Quantum Insights," in *IFIP Networking Conference (IFIP Networking)*, 2024
- M. Sosnowski, J. Zirngibl, P. Sattler, J. Aulbach, J. Lang, and G. Carle, "An Internet-wide View on HTTPS Certificate Revocations: Observing the Revival of CRLs via Active TLS Scans," in *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2024
- M. Kempf, B. Jaeger, J. Zirngibl, K. Ploch, and G. Carle, "QUIC on the Fast Lane: Extending Performance Evaluations on High-rate Links," *Computer Communications*, vol. 223, 2024. DOI: `https://doi.org/10.1016/j.comcom.2024.04.038`

Non peer-reviewed

- J. Mücke, M. Nawrocki, R. Hiesgen, P. Sattler, J. Zirngibl, G. Carle, T. C. Schmidt, and M. Wählisch, *Waiting for QUIC: On the Opportunities of Passive Measurements to Understand QUIC Deployments*, 2022. [Online]. Available: `https://arxiv.org/abs/2209.00965`

Chapter 2

Global Internet Observatory

A major foundation for this work is large-scale, Internet-wide measurements to identify responsive targets (see Part I), to identify used and interesting domain names (see Part II), and to identify and analyze QUIC deployments (see Part III). These scans require *(i)* scalable and up-to-date tools, *(ii)* the infrastructure to run scans but also to store and analyze results, and *(iii)* the maintenance of regular scans and all components. The latter includes compliance with ethical considerations and interaction with interested or affected network operators (see Section 2.3). The Global INternet Observatory (GINO) is a collaborative initiative to establish these scans as ongoing, well-maintained, and ethical measurements for Internet research. It was founded in 2016 and has since been operated.

Author's Contributions: *The author of this work joined the group in 2019 and was a central member of this group. This includes:*

(i) *Design, implementation and maintenance of scanning tools, for example:*
   - *the ZMapv6 [29] fork that allows large scale IPv6 scans,*
   - *the Goscanner [30] that performs TLS handshakes and extracts server fingerprints*
   - *and the QScanner [9] for QUIC scans and evaluations.*

(ii) *Setup of new and maintenance of existing scans and infrastructure including scan, storage and analysis servers as well as the network infrastructure.*

(iii) *Regular assessments whether ethical considerations are met and adjustments of GINO when necessary.*

2.1 Scan Tools

The group of GINO developed and maintains multiple scanning tools and makes them publicly available for the community to use. During the work for this thesis, the author was actively

involved in the design, development, and maintenance of these tools. The following lists the most important tools directly influencing this work.

ZMapv6: ZMapv6[1] is a fork of the high-performance, stateless ZMap scanner by Durumeric *et al.* [31]. The ZMapv6 fork extends the original functionality to support IPv6 scans and was initially published by Gasser *et al.* [8]. This tool is the foundation for the *IPv6 Hitlist* maintained by GINO (see Section 2.2). During this work, the tool was regularly updated based on the original ZMap and used alongside the *IPv6 Hitlist* as foundation for the results presented in Chapter 4 and Chapter 5. The ZMapv6 fork was further extended with a QUIC module to identify QUIC deployments within the IPv4 and IPv6 ecosystem (see Section 8.3.1).

Goscanner: The Goscanner[2] is a high-performance TLS scanner based on Go. It was first published in 2017 during the work of Amann *et al.* [32]. It allows conducting full TLS handshakes with TLS over TCP servers. The tool extracts TLS properties, X.509 certificates and error messages from the handshake for future analysis. In case of a successful handshake, a HTTP request can be sent.

As part of the work for this thesis, the scanner was actively maintained, updated (*e.g.*, to support TLS 1.3), and enhanced (*e.g.*, to extract further information from the TLS handshake [18], [19], [23]). The Goscanner is used for weekly TLS scans as described in Section 2.2 and was used during this thesis in collaboration with Sosnowski *et al.* [18], [19], [23] to analyze TLS deployments and to compare QUIC and TLS (see Chapter 8).

QScanner: The QScanner[3] is a high-performance QUIC scanner based on quic-go [33] that allows full Internet Engineering Task Force (IETF) conform QUIC handshakes. It extracts QUIC (*e.g.*, transport parameters) and TLS information (*e.g.*, used cipher suites and X.509 certificates). Furthermore, after successful handshakes, HTTP/3 requests can be sent and HTTP headers and settings can be extracted based on the results.

The author of this work provided the idea and design of the scanner. It was implemented by Philippe Buschmann during his Master's Thesis [34] and was further enhanced and maintained by the author of this work and GINO. The scanner provides the foundation for results presented in Chapter 8 and 9, and was used in collaboration with Mücke *et al.* [28], Sattler *et al.* [17] and Bauer *et al.* [22].

[1] `https://github.com/tumi8/zmap`

[2] `https://github.com/tumi8/goscanner`

[3] `https://github.com/tumi8/qscanner`

2.2 Regular Scans

Besides on-demand scans for specific studies, GINO maintains a set of regular measurements (*e.g.*, daily or weekly) targeting different protocols and goals. The following list provides a general overview about conducted scans. More details about specifically used scans, input sources and configurations are provided for each individual study in the later parts of this work.

The* IPv6 Hitlist:* With ZMap [31], complete IPv4 address space scans are possible within hours. In contrast, the IPv6 address space is too large for complete scans and a list of responsive and relevant IPv6 addresses is required. Gasser *et al.* [8], [35] established an ongoing service that collects IPv6 addresses from different sources (*e.g.*, DNS resolutions and traceroutes) as a cumulative list. Regularly, the input list is updated and filtered, and addresses are tested to determine whether they are responsive to different protocols. Part I analyzes the development of this hitlist and extends it to better support future research. It is further used as input for the TLS and QUIC scans. Thus, it is relevant for Part III.

DNS Resolutions*:* Domain names are an essential Internet resource in general and for different scans. *(i)* They can be used to identify targets (*e.g.*, IPv6 capable servers), and *(ii)* they are an important input for higher layer protocol scans. Many TLS and QUIC deployments serve multiple domain names as virtual servers. Therefore, servers require a domain as SNI during the handshake to serve the correct service and certificate. Part III shows the impact of this feature on QUIC scans and the possibilities to identify deployments.

We use MassDNS and a local Unbound to resolve a list of domains. We resolve `A`, `AAAA`, `MX` and `NS` records daily, and `SVCB` and `HTTPS` weekly. See Chapter 6 for a detailed analysis of these new records. The input consists of different sources, for example:

- Ranked domains from top lists, *e.g.*, Majestic Top 1M [36] and Umbrella Top 1M [37].
- A static collection of 98.1M domains from 52 country-code TLDs (ccTLDs) (partial zones, *e.g.*, 22M `.tk` and 13M `.de` domains).
- Domains of full zone files from Centralized Zone Data Service (CZDS) [38], including established zones (*e.g.*, `.com`, `.net` and `.org`) but also new zones (*e.g.*, `.live`).
- Domains extracted from X.509 certificates added to Certificate Transparency Log (CT log).

However, those lists also contain biases and domains of different relevance. For example, parked domains, only displaying advertisements or sales banners, are not interesting for most Internet-users and need to be considered differently during higher layer protocol evaluations (see Chapter 7).

TLS Scans*:* To evaluate the TLS ecosystem and collect used certificates, GINO conducts TLS scans on a weekly basis. The scans focus on the Hypertext Transfer Protocol Secure (HTTPS)

ecosystem and target servers on TCP port 443. They are divided into three stages each done for IPv4 and IPv6 respectively:

(i) ZMap is used to identify targets with an open port 443 (the default for HTTPS). For IPv4, the complete address space is scanned, while for IPv6, the *IPv6 Hitlist* and IPv6 addresses from `AAAA` records are used.

(ii) The Goscanner is used to conduct complete TLS handshakes with identified targets from the previous stage and to extract information.

(iii) All IP addresses with an open port 443 are combined with DNS resolutions to identify domain names that can be used as SNI. Each (IP address, domain)-pair is then scanned with the Goscanner.

***QUIC Scans**:* The general setup of QUIC scans is the same as for TLS scans. While also run on a weekly basis, divided into three stages and conducted for IPv4 and IPv6, it uses a QUIC-specific ZMap module and the QScanner for handshakes. However, the scan additionally relies on `HTTPS` records and HTTP Alternative Service (ALT-SVC) Headers (extracted from the TLS scans) to identify additional targets potentially missed by the ZMap module. For more information, see the detailed explanations and evaluations in Chapter 8.

2.3 Ethical Considerations

All Internet scans conducted during this work are set up based on a set of ethical measures we follow strictly. These are mainly based on informed consent [39] and well-known best practices [40]. Our study relates to users, personally identifiable information, or otherwise privacy-sensitive data. We focus on publicly reachable and available services. While sensitive data may occasionally be present in such sources (*e.g.*, the DNS), this is not in focus.

We test all implemented scanning tools in local environments with self-hosted services to reduce potential impact. Furthermore, to not cause harm to any infrastructure, we apply measures described by Durumeric *et al.* [31]. We limit the rate of our scans and use a collective blocklist. We respond to each inquiry regarding our scans and, if requested, we add IP ranges and domains to our blocklist to be excluded from our scans. We are directly registered as abuse contact for our scan infrastructure and react quickly to all requests. Furthermore, we host websites on all IP addresses used for scanning to inform about our research and provide contact information for further details or scan exclusion.

Besides limiting the rate of each scan, we try to limit the overall volume of our scans to not overload the network. As explained in Section 8.2, *Initial* QUIC packets need to be at least 1200 B. This increases the overall traffic from our QUIC scans compared to most TCP ZMap scans but mainly impacts our own uplink to the Internet. Due to the randomization of scanned targets, we argue that the impact on servers is still small. Furthermore, we limit the number of scanned domains per IP address to reduce the load on hosting services and providers.

Chapter 3

Related Work

This chapter provides an overview about relevant related work to this thesis. This work builds upon general advances in the scope of Internet-wide measurements. Especially the publication of ZMap by Durumeric *et al.* [31] allowed various studies and quick development of Internet-wide scans. The tool allows users to scan the complete IPv4 address space for servers with an open port within minutes. The tool is the de facto standard for Internet-wide port scans and the foundation for many studies. It was used to identify TLS over TCP capable targets in [32], [41], or QUIC capable targets in [42]. Similarly, we use it during our TLS over TCP and IPv6 scans and enhance it for our QUIC scans.

3.1 IPv6 Scans

While ZMap can easily be used for IPv4, scanning the complete IPv6 address space is infeasible. Therefore, so-called hitlists (lists of responsive, interesting IPv6 addresses) were researched and published. Initial attempts to establish IPv6 hitlists have been conducted, *e.g.*, by Gasser *et al.* [35] in 2016 and Fiebig *et al.* [43] in 2017. Gasser *et al.* [35] collected addresses from passive traces and active sources such as traceroutes and DNS resolutions, while Fiebig *et al.*[43] relied on reverse DNS to identify used addresses. In 2016, Foremski *et al.* [44] modeled IPv6 addresses based on their entropy and derived *Entropy/IP*, relying on structural similarities between addresses to generate target candidates based on address seeds. Based on these findings, Murdock *et al.* [45] developed 6Gen in 2017 and derived an IPv6 hitlist containing 55M active addresses. However, they identify 98% of these as aliased.

The most prominent *IPv6 Hitlist* was published by Gasser *et al.* [8], [35] and is still maintained by GINO. It collects addresses from multiple sources, applies filters and tests addresses for responsiveness. During their initial study, Gasser *et al.* [8] found that many addresses are within so-called aliased prefixes which drastically bias the hitlist. Therefore, they proposed a detection mechanism on multiple prefix-levels and implemented a filter into the ongoing service. Further-

more, they evaluate the characteristics of different input sources, their contained addresses, and their stability.

While the *IPv6 Hitlist* is an important tool, it does not include *all* used and responsive IPv6 addresses. Different studies were published that tackle IPv6 scans and try to improve them. Rye *et al.* [46], [47] focused on scans to identify Customer-premises equipment (CPE). They find that while end-user devices normally do not rely on EUI-64 addresses anymore, some CPE manufacturers still rely on them. Therefore, as they are based on the MAC address, they can be used to track devices and users behind them. Furthermore, they can be used to geolocate CPEs. In 2023, Rye *et al.* [48] created a new hitlist based on Network Time Protocol (NTP) clients. Therefore, they added servers to the NTP-pool and tracked client requests. They accumulate more than 7.9 B IPv6 addresses. However, many of them are not responsive or only responsive for a short amount of time. The list mostly consists of clients and data can not be published. Therefore, it is no replacement to the *IPv6 Hitlist* but provides complementary insights.

3.1.1 Target Generation Algorithms

Discovery strategies of unknown IPv6 addresses were already described in RFC7707 [49] based on drafts dating back to 2012 and first approaches published in between 2015 and 2017 [44], [45], [50]. Since the creation of the *IPv6 Hitlist*, different IPv6 address generation algorithms have been published [51]–[59]. All approaches assume that IPv6 addresses contain patterns due to assignment strategies, allowing research to guess new, responsive addresses. They differ based on selected address representations and machine learning approaches. Cui *et al.* [52] use General Adversary Networks, while Liu *et al.* [55] represent addresses as a space tree. The studies mostly rely on the *IPv6 Hitlist* [8] and evaluate their generated lists with active scans. Song *et al.* [57] were able to generate hitlists containing more than one billion candidates and reach hit rates of up to 50%. However, as of January 2024, we only found a snapshot of generated addresses from DET, and other sources could not be verified or reused for further studies. We compared these algorithms in multiple steps and combined them with the *IPv6 Hitlist* to improve its current state for other researchers (see Part I).

Rye *et al.* [46] took a slightly different approach when introducing *edgy* in 2020, focusing on the efficient discovery of the IPv6 periphery, *i.e.*, not servers or clients, but last hop routers. With *edgy* they were able to discover more than 64M active last hop router addresses. One year later, Li *et al.* [60] described a similar approach, discovering more than 50M last hop router addresses by tracerouting non-existent IPv6 addresses in known or suspected customer subnets of Internet Service Providers (ISPs).

3.1.2 Aliased Prefixes

Besides collecting and generating new address candidates, detecting aliased prefixes is important to understand and remove induced biases to IPv6 hitlists. Different alias detection methodologies have been proposed to detect different addresses of the same router [61]–[65] and to detect IPv4 and IPv6 siblings [66]. However, most of these side channels are not available on every target. Therefore, Murdock *et al.* [45] proposed an alias detection for IPv6 hitlists based on the

responsiveness of random addresses within prefixes of size /96. Gasser *et al.* [8] extended this idea to a multi-level aliased prefix detection on different prefix lengths. Furthermore, they combined TCP/80 and ICMP probes with results from previous days to account for probe timeouts. They verified the effectiveness of their approach to identify aliases with TCP fingerprints.

Song *et al.* [57] suggested using the Too Big Trick (TBT) introduced by Beverly *et al.* [62] to evaluate real aliases. They show that some prefixes identified as aliased by the *IPv6 Hitlist* might contain multiple hosts, although they are fully responsive. In this work, we combine detected aliases based on the multi-level detection method with TCP fingerprinting, the TBT, and information about specific Autonomous Systems (ASes) and hosted domains to shed further light on identified aliased prefixes.

3.1.3 Great Firewall of China

While the Great Firewall of China does not necessarily seem related to IPv6 hitlists, we show its relation and impact in Section 4.3, mainly its DNS injection behavior. It has been analyzed in a different context [67]–[69] and has been recently used as a side channel to actively analyze DNS root server performance in China by Zhang *et al.* [70]. Most importantly, these works find that the GFW injects DNS responses for censored domains at the border of Chinese networks. Anonymous *et al.* [69] analyzed the behavior in 2020, observing multiple responses that can be mapped to different injectors. They further find that erroneous responses typically contain generally routed, valid IP addresses. However, these addresses can be mapped to operators unrelated to the requested domain. We identified the GFW to be responsible for a majority of addresses responsive to DNS probes. Results showed similar behavior as reported by related work, highly impacting the quality of the *IPv6 Hitlist*. However, we observed different addresses in all responses.

3.2 The DNS Ecosystem

Besides our DNS scans, different large-scale measurement campaigns for DNS exist. Two prominent campaigns also sharing data on request or during collaborations are OpenINTEL [71] and Rapid7 [72]. Both campaigns use domain lists similar to ours as sources and query similar records on a regular basis. We use data from both campaigns in this work to complement our data with additional results and different vantage points (*e.g.*, Chapter 4 and Chapter 6).

3.2.1 Service Information

`SVCB` and `HTTPS` records have seen little attention from other research as of January 2024. Trevisan *et al.* [73] use alternative service information to identify QUIC deployments but only HTTP ALT-SVC Headers from additional HTTP requests. They implied that HTTP ALT-SVC Headers are widely deployed. We show that fewer `HTTPS` records were deployed in 2021 and 2023, but growth is visible. Ralf Weber [74] reported on the visibility of `HTTPS` queries from a network (Akamai) perspective. While many queries failed with incorrect behavior initially, the correctness of seen responses changes quickly. Additionally, they only observed records for

126.4k domains and no alias mode. Aguilar-Melchor *et al.* [75] evaluate a potential positive effect of `HTTPS` records but do not evaluate its current deployment state.

In 2019, Chai *et al.* [76] evaluated Encrypted SNI, an older version of Encrypted Client Hello (ECH) that relied on TXT DNS RR to distribute key information. They identified more than 100k domains within the Alexa Top 1M. Similar results have been reported by Tsiatsikas *et al.* [77] in 2022. In 2022, Hoang *et al.* [78] found 1.5% to 2.25% domains with a respective TXT record out of 300M domains from TLD zone files. We show that no transition to ECH and `HTTPS` records was visible as of January 2024.

Furthermore, the security and impact of ECH have been analyzed [79], [80] and related work has evaluated the state of DNS over TCP, HTTP, or QUIC [81]–[84], and shows increased deployment and, in general, good performance. Thus, the fundamentals for a successful deployment of `SVCB` and `HTTPS` records are given.

3.2.2 Domain Parking

Domain parking as a service was previously analyzed by Alrwais *et al.* [85], focusing on the customer perspective, analyzing monetization chains and potential malpractice. They used passive DNS data and a hitlist of name server records to find monetization chains. Vissers *et al.* [86] focused on the user perspective. They also inferred the use of parking services based on DNS indicators. Based on the DNS Census data covering 106M registered domains collected throughout two years, they identify 7.5M parked domains for a collection of 15 parking services. They further analyzed the identified domains with respect to typo-squatting and malicious behavior. They propose a set of features derived from identified parking pages that could be used as the basis for a browser-based classifier. We offer an extended and updated list of providers, resulting in a number of parked domains nearly an order of magnitude larger. We also focus on the impact of ignoring the special role of parked domains on research, in particular as it relates to centralization.

Kührer *et al.* [87] identified domain parking as the reason for up to 10% of blocklist entries. Similar to Vissers *et al.* [86], they designed a classifier based on websites to identify parked domain names. However, they focus on blocklists and top lists and train their classifier with a relatively small input set. We focus on a general quantification of domain parking and its impact on further areas.

Domain parking was also observed as part of evaluations of specific (*i.e.*, `.xxx` and `.biz`) and newly introduced (starting in 2013) generic TLDs (gTLDs) [88]–[90], the practice of domain registrars [91], [92], and the analysis of hosting providers [93]. In various works, Halvorson *et al.* [88]–[90] have shown that around 30% of domains from gTLDs are parked, and even 23% of `.com` is parked. We identify a larger fraction for the established TLDs `com`, `net`, `org` (30%) but slightly less parked domains on average for all remaining available gTLDs (25%). Alexander [91] and Lauinger *et al.* [92] identified parking as common practice of registrars as well, promoting their portfolio and collecting expired domains. We include dedicated parking services but also services from registrars into our analysis and extend evaluation further to domains from ccTLDs and all available gTLDs, including traditional gTLDs, *e.g.*, `.com`, but

also newly registered gTLDs. Our DNS scans cover more than 1k gTLDs compared to 502 investigated by Halvorson *et al.* [88]. Zembruzki *et al.* [93] saw domain parking as an influential factor on the analysis of hosting centralization but did not investigate its impact.

3.3 Evaluation of QUIC

The relatively new QUIC protocol has already been widely evaluated before the final RFC was released. Research often focuses on security aspects [94], [95], interoperability [96], [97], the variety of available implementations [98] and performance evaluations including comparisons to TLS and / or TCP [20], [97], [99]–[106].

3.3.1 QUIC Scans

QUIC was initially proposed by Google in 2013 [107] and afterwards transferred to the IETF within its own working group.[1] The first Internet-wide measurement study to identify and analyze QUIC deployments was conducted by Rüth *et al.* [42] in 2018. They focused on Google's QUIC versions in 2016 and 2017. The IETF drafts were only in an early stage and not considered. They discover a steady growth of IP addresses with gQUIC support and an increasing traffic share. It was mainly deployed by Google and Akamai as driving forces. While gQUIC initiated the standardization of QUIC, both variants have large differences, *e.g.*, the former does not rely on TLS but its own security mechanism. With the standardization of QUIC in 2021 [1], a decrease in gQUIC was expected and visible, as shown in this work. Therefore, our analysis of QUIC in Part III solely focuses on IETF QUIC. Furthermore, we included IPv6 and investigated additional sources, namely `HTTPS` DNS RR and HTTP ALT-SVC Header, to discover QUIC capable targets. The inclusion of the latter two sources reveals additional QUIC deployments not found by the implemented ZMap module. Furthermore, our stateful approach allows the analysis of deployment characteristics like QUIC transport parameters and TLS configurations.

A subsequent study from Piraux *et al.* [97] introduces a test suite which scans targets on the Internet and evaluates specification conformity of QUIC implementations. They determined that 10% to 20% of the responses contain errors, which is similar to our SNI stateful scans but did not further present the distribution of error messages. In comparison, we investigate deployments in more detail, analyzing their current state independent of used implementations.

A more recent study from Trevisan *et al.* [73] investigated the HTTP/3 adoption based on the HTTP ALT-SVC Header extracted from the open-source HTTPArchive Dataset but found only 14k websites with HTTP/3 support in December 2020. The remaining study focused on a performance comparison between HTTP versions of found sites. Deploying active scans, we found that the QUIC and HTTP/3 support is larger by multiple orders of magnitudes.

A longitudinal analysis of the general TLS deployment can be found in Kotzias *et al.* [108]. They showed the overall reaction to known high-profile attacks (*e.g.*, Heartbleed) and that

[1] `https://tools.ietf.org/wg/quic/`

the ecosystem can react quickly. The deployment of TLS 1.3 throughout its standardization and early years was analyzed by Holz *et al.* [41]. Similar to QUIC, they find that the driving forces behind the deployment of this new TLS version are few hypergiants, including Cloudflare, Google and Facebook. They solely focus on TLS 1.3 over TCP and do not investigate QUIC. Our QScanner can be used to perform similar analyses for QUIC.

Gigis *et al.* [109] recently reported about hypergiants' off-nets and their development over several years. They used TLS certificates and HTTP header information from Internet-wide scans to identify edge Point of Presences (POPs) of large providers. They report extensive usage of edge POPs by large providers, *e.g.*, Google, and Facebook. Using QUIC transport parameters in combination with HTTP Server header values, we independently identify similar deployments for QUIC.

3.3.2 Library Differences

Differences in implementations and their behavior during communications have been used by Sosnowski *et al.* [18], [19] and Althouse *et al.* [110] to differentiate TLS/TCP deployments or to analyze operating systems and TCP behavior [111], [112]. In contrast, in Chapter 9 we focus on QUIC libraries and specific features independent of the TLS library and deployment configurations used.

QUIC deployments and their behavior have been analyzed at different stages of the standardization phase [42] and since the release of RFC 9000 [113], [114]. Nawrocki *et al.* [113] evaluated the behavior of different deployments with a focus on certificates and their conformance to amplification limits during the handshake. They show differences between QUIC deployments. However, they focus not on specific library differences and their identification but on the interplay between QUIC and certificate (chain) sizes. Mücke *et al.* [28] evaluate QUIC deployments based on different server behavior. However, they focus on specific hypergiants and do not focus on the identification of libraries in general. Furthermore, Marx *et al.* [98] have analyzed libraries and showed the impact of different draft interpretations on the implementations, *e.g.*, due to congestion control mechanisms. Gbur *et al.* [115] have evaluated QUIC libraries focusing on security aspects in local tests.

While most of these works showed differences in QUIC libraries and their impact on specific features, none analyzes the impact of these differences on scans and provides means to identify libraries on the Internet. To the best of our knowledge, there is no published test environment for QUIC scanners, and no methodology exists to identify deployed QUIC libraries.

Part I
IP Measurements

The first part of this thesis focuses on the first research goal (see Section 1.1): *Methodologies to Identify IPv6 Deployments.* Scalable and stateful Internet-wide evaluations of protocol deployments require efficient and fast stateless scans to identify targets offering a specific service. This is often done with port scans to identify IP addresses with an open port used by specific services (*e.g.*, TCP port 443 for HTTPS or UDP port 53 for DNS).

With the publication of ZMap by Durumeric *et al.* [31], this can be done for the complete IPv4 addresses space in less than one hour. However, it can not be used to scan the complete IPv6 address space within feasible time. As a solution, IPv6 hitlists and so-called target generation algorithms are published to support IPv6 research. They offer active IPv6 addresses to be used as input for further scans.

The most commonly used hitlist was initially published by Gasser *et al.* [8], [35] and is maintained by the GINO group as explained in Chapter 2. It is an important foundation for IPv6 research and used in a variety of publications, *e.g.*, [59], [116]–[119]. To accomplish our research goal, we set out to evaluate the *IPv6 Hitlist* development over the last years since its publication. Furthermore, we improved it and its value to further support this work but also other researchers.

Overview of the first part.

	Part I: IP Measurements	
G1	State of the IPv6 Hitlist	Chapter 4
	Target generation algorithms	Chapter 5

Chapter 4 evaluates how the *IPv6 Hitlist* developed and reveals drastic negative effects by the GFW. DNS injections impacting the *IPv6 Hitlist* scans resulted in a drastic error within DNS scans and a bias towards Chinese ASes. Furthermore, we analyze aliased prefixes and show that most of them do not follow the initial definition — a single host using the full prefix as alias — but are fully responsive prefixes served by multiple hosts. We compare these findings to the IPv4 ecosystem.

Chapter 5 sets out to extend the *IPv6 Hitlist* with new addresses from different sources. It combines insights from different studies into hitlist extensions mainly based on target generation

algorithms and their application on the *IPv6 Hitlist*. Besides target generation algorithms, we evaluated new sources and re-scanned previously unresponsive addresses.

Chapter 4

Evaluation of the IPv6 Hitlist

The long-running *IPv6 Hitlist* service is an important foundation for IPv6 measurement studies. It helps to overcome infeasible, complete IPv6 address space scans by collecting valuable, unbiased address candidates and regularly testing their responsiveness. However, the Internet itself is a quickly changing ecosystem that can affect long-running services, potentially inducing biases and obscurities into ongoing data collection means. Frequent analyses but also updates are necessary to enable a valuable service to the community.

In this chapter, we show that the existing hitlist was highly impacted by the Great Firewall of China, and we offer a cleaned view on the development of responsive addresses. While the accumulated input shows an increasing bias towards some networks, the cleaned set of responsive addresses is well distributed and shows a steady increase over time.

Although it is a best practice to remove aliased prefixes from IPv6 hitlists, we show that this also removes major content delivery networks. More than 98% of all IPv6 addresses announced by Fastly were labeled as aliased and Cloudflare prefixes hosting more than 10M domains were excluded. Depending on the hitlist usage, *e.g.*, higher layer protocol scans, inclusion of addresses from these providers can be valuable. We further compare this effect of the IPv6 ecosystem to IPv4 and show that while the address space is feasible to scan completely, scans are impacted by fully responsive prefixes as well.

This chapter is based on the following publications:

> J. Zirngibl, L. Steger, P. Sattler, O. Gasser, and G. Carle, "Rusty Clusters? Dusting an IPv6 Research Foundation," in *Proc. ACM Internet Measurement Conference (IMC)*, 2022. DOI: 10.1145/3517745.3561440 [11]

> P. Sattler, J. Zirngibl, M. Jonker, O. Gasser, G. Carle, and R. Holz, "Packed to the Brim: Investigating the Impact of Highly Responsive Prefixes on Internet-wide Measurement Campaigns," *Proc. ACM Netw.*, 2023. DOI: 10.1145/3629146 [14]

The first publication builds in parts on Lion Steger's Master's thesis [120].

Author's Contributions: *The author of this work led the research presented in [11] by framing the original research question, laying out the methodology and by leading the analysis and synthesis into the publication. The author is one of the primary maintainers of the IPv6 Hitlist used as primary data source for this research. Therefore, he collected most of the used data. Furthermore, he was responsible for the analysis of the historic IPv6 Hitlist data and the analysis of fully responsive prefixes.*

The author influenced the work presented in [14] during discussions, the framing of the research questions and helped with the analysis of data. Insights from the work on fully responsive prefixes in IPv6 directly influenced the presented results. The author helped with the general scan setup, the evaluation of results and was responsible for the comparison to IPv6.

4.1 Motivation

The usage and importance of IPv6 are steadily increasing [121]–[123]. With the IPv4 address depletion of all but one Regional Internet Registry (RIR), the necessity to deploy IPv6 is prevalent for more and more operators and content providers. While this development is generally positive, it imposes fundamental difficulties to research and network analysis. The immense size of the address space, combined with the sparse distribution of used addresses renders active IPv6 measurements difficult. While tools like ZMap effectively scan the complete IPv4 address space, complete scans for IPv6 are impossible.

With the publication of the *IPv6 Hitlist* service in 2018, Gasser *et al.* [8] established an ongoing service that collects IPv6 address candidates, identifies aliased prefixes, and tests the responsiveness in respect to different protocols, namely ICMP, TCP on port 80 (HTTP) and 443 (HTTPS), and UDP on port 53 (DNS) and 443 (QUIC) (see Figure 4.1). The service has been used as de facto standard for IPv6 analysis and scans, *e.g.*, [9], [124]–[128]. However, it has not seen significant updates or evaluations since its initial publication. Changes in the usage of IPv6 and input sources might have influenced the quality of the service since 2018.

Our main contributions in this chapter are:

(i) We evaluated the development of the *IPv6 Hitlist* in between 2018 and 2022 (before extensions presented in Chapter 5) and new biases introduced by the accumulation of new addresses. Our findings allowed us to filter targets incorrectly tested as responsive. We identified 134M addresses falsely reported as responsive to UDP/53 by the *IPv6 Hitlist* since 2018 due to the Great Firewall of China's DNS injection.

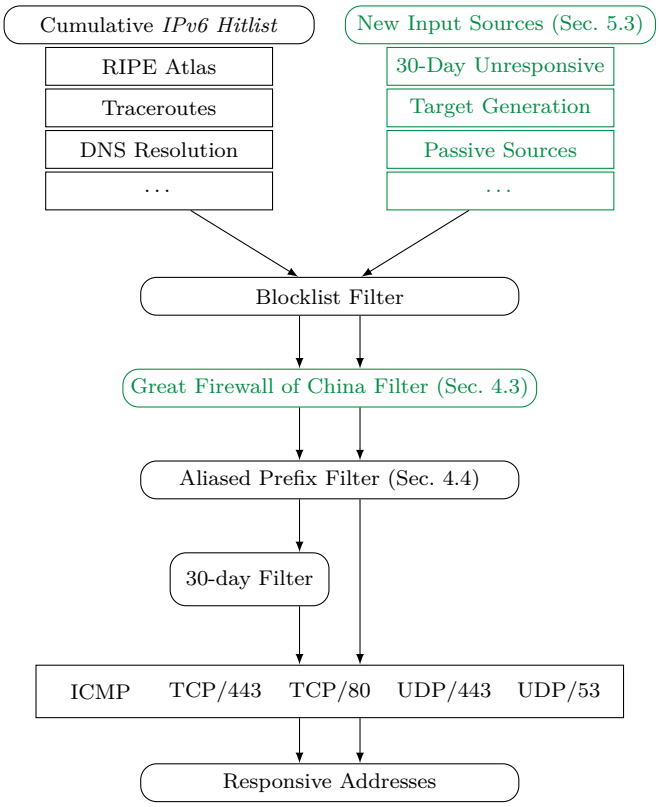

Figure 4.1: *IPv6 Hitlist* pipeline. This work analyzes the existing steps and adds a Great Firewall of China filter (see Sec. 4.3). Furthermore, the filtered, aliased prefixes are analyzed in Sec. 4.4 and new input sources are evaluated in Chapter 5. Newly implemented service components are indicated by green borders.

(ii) We analyzed aliased prefixes in more detail and investigated whether the initial definition of a single host responsive to a complete prefix remains correct or whether a set of addresses needs to be treated differently. We show that aliased prefixes hosted at least 15M domains including ranked domains from different top lists [36], [37], [129]. In combination with additional findings, we suggest users of the hitlist to include subsets of these prefixes in future research.

(iii) We compare our findings regarding fully responsive prefixes to the IPv4 ecosystem. We show that while the full address space can feasibly be scanned, these scans are also impacted by fully responsive prefixes.

(iv) We updated the *IPv6 Hitlist* service [8] to allow future research to use our findings within the established service.

4.2 Data Sources

Our primary source is the *IPv6 Hitlist* initially published by Gasser *et al.* [8] and maintained by GINO (see Chapter 2). The service collects address candidates from multiple sources, including DNS AAAA resolutions, conducted traceroutes, and public sources, *e.g.*, from RIPE Atlas and

CT logs. The service frequently *(i)* updates addresses, *(ii)* uses *all* collected addresses as input, *(iii)* applies multiple filters, and *(iv)* tests the responsiveness of addresses in respect to different protocols. Figure 4.1 shows the service pipeline.

The first filter removes the addresses of operators who requested exclusion of regular scans following ethical considerations described in Section 2.3. The most important existing filter is the aliased prefix detection. The initial definition of aliased prefixes describes a single host responsive for all addresses in a prefix. Each individual aliased prefix may be infeasible to scan, offers limited value to following scans, and introduces a bias. As shown in Section 4.4, aliased prefixes in our data had different lengths between /28 and /120. While earlier work from Murdock *et al.* [45] tests for aliased prefixes with a fixed length of /96, the aliased prefix detection of the *IPv6 Hitlist* tests prefixes of different lengths, including:

- IPv6 prefixes announced in BGP
- all /64 prefixes with at least one address contained in the *IPv6 Hitlist* service input
- and prefixes longer than /64 (in steps of four bits) with at least 100 addresses.

The implemented detection [8] relies on the assumption that it is highly unlikely that multiple randomly selected addresses within an IPv6 prefix are responsive. Therefore, the detection selects one random address within each of the 16 more specific prefixes (0-f) and uses ZMapv6 [29] to test responsiveness. For example, to test whether prefix `2001:db8::/32` is aliased, a single, random address is tested within all subprefixes `2001:db8:[0-f]000::/36`. This address generation distributes the pseudo-random targets evenly across the complete prefix. If all 16 addresses are responsive, the prefix is labeled as aliased. ICMP and TCP/80 are tested, and results are merged across protocols and with the previous three scans. This reduces misclassification of prefixes, *e.g.*, due to random network events or packet loss during individual scans.

The final filter in Figure 4.1 removes all addresses that are unresponsive for at least 30 days. This filter reduces the required scan load drastically. However, these addresses are never tested for responsiveness again after exclusion. We re-scanned these addresses to test whether addresses are responsive after 30 days again as discussed in Section 5.3.

After all filter steps, the service executes traceroutes using Yarrp [130] to all targets to potentially identify new targets. Furthermore, ZMapv6 scans ICMP, TCP port 80 (HTTP) and 443 (HTTPS), and UDP port 53 (DNS) and 443 (QUIC). While scans were executed on a daily basis initially, the growth of the input set increased the overall runtime to several days. The data considered in this chapter covers more than 750 scans between July 2018 and April 2022.

Besides the *IPv6 Hitlist* data, we use a single snapshot of the DNS scans conducted by GINO based on Chapter 2 from April 7, 2022 to analyze aliased prefixes in Section 4.4. Furthermore, we conduct additional scans to analyze the responsiveness of addresses within aliased prefixes to other probes then tested during their detection and deploy the TBT. For all additional scans, we follow ethical considerations as introduced in Section 2.3.

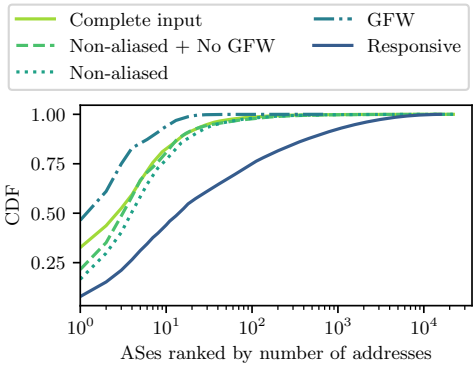

Figure 4.2: Distribution of addresses in the input list across ASes. Additionally, the effect of the alias and GFW filter are displayed. Note the log x-Axis. While the complete input is biased towards some ASes, established and new filters result in a well distributed set of responsive addresses.

4.3 IPv6 Hitlist Development

As explained in Section 4.2, the *IPv6 Hitlist* [8] is implemented as an ongoing service, regularly updating its input, identifying aliased prefixes (analyzed in more detail in Section 4.4), and testing the responsiveness of addresses.

4.3.1 Input Development

Starting with 90M addresses in July 2018, the service has accumulated more than 790M addresses until April 2022. They cover 22 074 ASes compared to 10 866 in 2018 [8]. The 22k ASes cover 76% of ASes announcing at least one IPv6 prefix in BGP as of April 2022. The number of announced prefixes and ASes is based on a routing information base from RIPE RIS collector rrc00 [131].[1] As of April 2022, the input list covered four times more (97k) announced BGP prefixes compared to 2018, 62% of all announced prefixes. The visible growth is similar to the general growth of IPv6 deployments and usage on the Internet and shows that the hitlist is able to adapt to it.

Figure 4.2 shows the cumulative distribution of IPv6 addresses in the hitlist input across ASes. Without any filter, the most prominent AS is Amazon (AS16509), covering 32% of all addresses. However, 99.6% of these are filtered due to the aliased prefix detection. Section 4.3.2 explains the additional plots covering the GFW impact and responsive addresses.

Nevertheless, after filtering aliased prefixes, 80% of the input is still covered by only 10 ASes, mostly from ISPs such as ANTEL (AS6057, 16%) or DTAG (AS3320, 10%). Analyzing the source of these addresses reveals that they are mostly due to changing prefix assignments and rotating addresses accumulated by regular traceroutes, especially from RIPE Atlas. 282M addresses from the input contain a EUI-64 IPv6 Interface ID (IID) including `ff:fe` and are based

[1] `https://data.ris.ripe.net/rrc00/2022.04/bview.20220407.0800.gz`

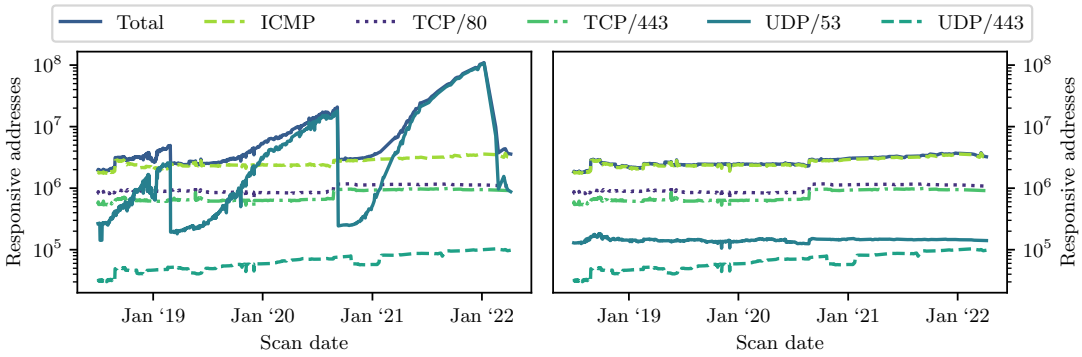

Figure 4.3: Comparison of the *IPv6 Hitlist* before this work and cleaned from GFW injection (right). Without injected responses by the GFW, a steady development of the *IPv6 Hitlist* and each protocol is visible. Note the log y-Axis not starting at 0.

on a MAC address. Extracting the IID reveals that these addresses are only derived from 22.7M distinct MAC addresses. Grouping addresses based on the EUI-64 values shows that 9M occur only within one IPv6 address each, while the remaining are seen in multiple addresses. The most frequent EUI-64 value can be seen in 240k distinct IPv6 addresses. The Organizationally Unique Identifier (OUI) of the MAC address is mapped to a vendor (ZTE) and all addresses are part of the same /32 prefix, but within different subnets. Similarly, Rye *et al.* [47] found a variety of CPE used EUI-64 IIDs and whose ISPs regularly rotated prefixes. They were able to track individual devices within ISPs.

These rotating addresses result in a visible bias of the complete input list towards a small set of ASes due to the ongoing accumulation of addresses. While the filter of unresponsive addresses after 30 days removes most of these addresses from regular scans and the responsive results afterward, the overall input list has to be treated carefully by other research relying on this data.

4.3.2 Address Responsiveness

Figure 4.3 shows the responsiveness of addresses throughout the *IPv6 Hitlist* service lifetime. It depicts the responsiveness for each tested protocol and a total count of addresses responsive to at least one protocol within each scan. The left half represents the state of the *IPv6 Hitlist* before this work. It had clearly visible spikes which will be explained in the following. On the right, a *cleaned* version is presented, the published state since April 2022.

The Prevalence of DNS: The hitlist contained significant spikes in responsive addresses as seen in Figure 4.3. During these events, the number of addresses responsive to DNS probes were larger compared to all other protocols. After each peak, the number of addresses dropped to a similar level as before the event. The last event started in early 2021 with the most drastic growth of addresses, peaking at more than 100M. In comparison, only 3.5M and 1.4M addresses were responsive to ICMP and TCP/80 respectively, and no increase was visible during the

Table 4.1: Top 10 ASes of addresses impacted by the GFW. The total accumulated number of impacted addresses is 134 M.

Name	ASN	# Addresses	%	CDF
China Telecom	4134	62.3M	46.44	46.44
China Telecom	4812	19.5M	14.59	61.03
ChinaNet	134774	18.6M	13.88	74.92
ChinaNet	134773	10.7M	8.04	82.96
China Telecom	140329	3.1M	2.37	85.34
ChinaNet	134772	2.5M	1.93	87.28
China Unicom	4837	2.5M	1.87	89.17
ChinaNet	136200	2.3M	1.76	90.94
China Telecom	140330	2.3M	1.72	92.66
China Telecom	140316	1.6M	1.24	93.91

same periods. The peak dropped in February 2022, after we implemented a filter based on the following findings and updated the service.

To understand the origin of these peaks, we describe the scan configuration and analyze results in detail. The service sends a DNS query requesting a AAAA record for `www.google.com`. All peak events share similarities but differ slightly. During the first two events, a significant fraction of addresses responded with A records only containing an IPv4 address. During the third event, responses carried AAAA records as requested but contained Teredo addresses. Note that Teredo is a deprecated standard embedding an IPv4 into an IPv6 address [132]. Furthermore, ZMap accumulated two or three responses for each scanned address, with up to 440 responses in the worst case.

Analyzing the erroneous IPv4 addresses contained in A records during earlier events and embedded in the returned Teredo addresses in the latter event reveals that none can be associated to the Google ASes but other companies like Facebook, Microsoft or Dropbox. Collecting all IPv6 addresses that responded with a clearly erroneous record (IPv4 or Teredo address) throughout the four years accumulates to more than 134M addresses (17% of the cumulative *IPv6 Hitlist* input on April 7th, 2022).

Querying a different domain shows that these targets are not responsive themselves, but responses are injected and falsely interpreted as success by ZMap. Most addresses with this response behavior are announced by ASes of Chinese networks, *e.g.*, China Telecom Backbone (AS4134) and China Telecom (AS4812) originating 46.44% and 14.59% of impacted addresses respectively. Figure 4.2 compares the AS distribution of IPv6 addresses responding with these incorrect records to the complete input. These addresses cover only 695 ASes. 93% of addresses are located in only 10 Chinese ASes (see Table 4.1). We used MaxMind GeoLite2 [133] as an additional indicator of network location. While we are aware of potential inaccuracies especially on a city level [134], [135], it mapped a majority of impacted IPv6 addresses to China. Given these indicators a strong relation of these addresses to Chinese networks can be seen.

The overall behavior has been described similarly in related work [67]–[69]. We see multiple responses to a single query indicating multiple injectors, responses are mostly in relation to addresses from China and `www.google.com` is a blocked domain. Querying different blocked domains from these addresses shows similar behavior. In contrast, a domain owned by ourselves,

Table 4.2: Development of responsive IPv6 addresses and covered ASes over four years. Results are based on cleaned data, removing GFW injected responses. For each year, a snapshot representing a single scan is used. The cumulative result covers all scans since the start of the *IPv6 Hitlist*.

Year	ICMP		TCP/443		TCP/80		UDP/443		UDP/53		Total	
	Addr.	ASes	Addr.	ASes	Addr.	ASes	Addr.	ASes	Addr.	ASes	Addr.	ASes
July 1, 2018	1.7M	10.1k	550.6k	5.8k	832.1k	6.2k	31.0k	0.9k	129.1k	5.1k	1.8M	10.3k
April 1, 2019	2.4M	11.0k	645.8k	6.2k	919.2k	6.6k	50.4k	1.0k	145.4k	5.2k	2.5M	11.2k
April 1, 2020	2.3M	11.7k	632.8k	6.6k	836.2k	6.9k	67.7k	1.3k	148.4k	5.1k	2.4M	11.9k
April 2, 2021	3.0M	13.7k	954.8k	7.4k	1.1M	7.7k	83.0k	1.3k	148.0k	6.0k	3.1M	13.9k
April 7, 2022	3.1M	15.4k	910.8k	7.9k	1.0M	8.2k	98.1k	2.0k	140.7k	6.0k	3.2M	15.7k
Cumulative	45.3M		6.7M		8.6M		2.5M		200k		46.8M	

most likely not blocked, results in no response at all, not even a DNS error. A difference to existing related work is that the injected responses at this point in time carried a Teredo address not explicitly reported in previous findings [67]–[69]. However, contained IPv4 addresses show similar behavior (cf. [69]) and can be mapped to previously identified incorrect networks.

The source of these addresses is the regularly conducted traceroutes by the *IPv6 Hitlist* service using Yarrp. Traceroute captures regularly changing addresses mostly with randomized IIDs, and they are visible as the last responsive hop. The targeted address is not responsive itself. Scanning these addresses in the following with ZMap triggers a DNS injection by the GFW, but for a majority no other protocol is responsive. In some cases, these targets are actually responsive for other probed protocols. Thus, invalid DNS responses should be filtered, but individual addresses should remain in the *IPv6 Hitlist* if responsive to other protocols.

Reducing the Great Firewall of China Impact: We initially implemented a filter removing 134M IPv6 addresses to reduce the GFW impact. The *IPv6 Hitlist* service saw at least one DNS injection for these addresses during July 2018 and April 2022 but no response to any other protocol. This filter immediately reduced scan duration and impact on the Internet.

To reduce the future impact on the ongoing *IPv6 Hitlist* service, we filtered the DNS/53 results directly after the scan. Therefore, the results correctly reflected the responsiveness of newly scanned input addresses. Thus, if addresses were responsive to any other protocol, they remain in the scan input. Otherwise, they were filtered by the 30-day filter (see Figure 4.1).

While changing the queried domain could prevent injection, *e.g.*, to `example.com`, we initially decided to operate the *IPv6 Hitlist* service with its current configuration. This consistency in the service behavior increases the comparability of results over time. With a consistent scan, different results indicate changes in the target behavior and are not induced by the scan itself.

However, by the time of the study by Steger *et al.* [13], we received DNS responses pointing to addresses from Facebook's network instead of Teredo addresses. We expect that the GFW can change its behavior in unpredictable ways. Therefore, we chose not to adapt the filter of the *IPv6 Hitlist* service to this new type of injection, and instead changed the domain name which is used in the regular probes. The new domain name is not censored by the GFW and does not trigger any injections. The domain is owned by us and not used for any other service besides

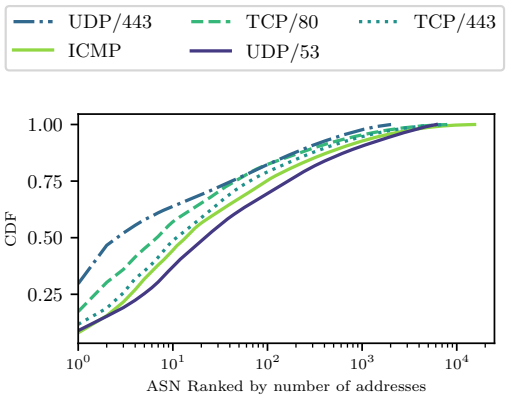

Figure 4.4: AS distribution of addresses responsive to each protocol on April 7, 2022.

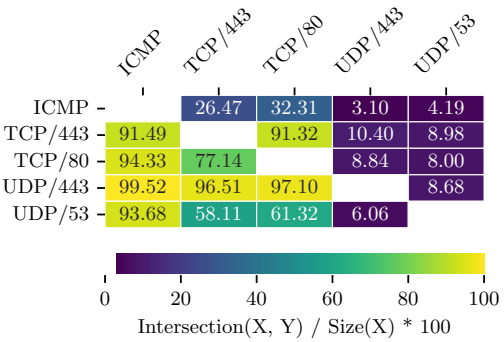

Figure 4.5: Overlap of addresses responsive to each protocol in the *IPv6 Hitlist* on April 7, 2022.

these scans. We argue that this yields more stable and usable results for researchers using the *IPv6 Hitlist*.

Evaluation of Remaining IPv6 Addresses Supporting DNS: After removing GFW injected DNS responses, 140k addresses responsive to DNS probes remained. We evaluated the quality of the remaining IPv6 addresses with an experiment including a domain under our control. To analyze the behavior of each scanned target, we queried for a subdomain including a unique hash instead of a static domain. The name server returned a valid AAAA record for the requested domain. This approach allowed us to map outgoing probes towards individual addresses to incoming requests at our name server. Within this filtered set of targets, 131.8k (93.8%) of probes resulted in valid DNS responses with status codes indicating errors, because these targets are either name servers or resolvers unwilling to resolve the requested domain recursively. 6.5k (4.6%) return regular responses with the correct AAAA record and according requests from the same IPv6 addresses are visible at our name server. 593 targets respond with a referral to name servers of the root or our domain's parent zone. Only 15 IPv6 addresses return a correct record, but the source addresses of incoming requests on our name server do not match the probed targets, *e.g.*, due to proxies or the usage of another interface.

The remaining 1.1% of targets respond incorrectly but an analysis of results reveals nothing similar to the GFW injection. Responses contain for example incorrect status codes or referrals to *localhost*.

***Cleaned* IPv6 Hitlist:** We cleaned the historical data from GFW injected responses resulting in the timeline shown in the right part of Figure 4.3. This results in a relatively stable number of responsive addresses for all protocols respectively, each with a slight increase throughout the four-year period. This development is in line with statistics reported by RIPE NCC regarding the general growth of IPv6 assignments [121]. Most addresses responsive to at least one protocol are responsive to ICMP, followed by TCP/80 and TCP/443. Figure 4.5 shows the exact overlap between protocols. For ICMP, the responsiveness increased from 1.78M addresses covering

10.1k ASes to 3.15M addresses in 15.5k ASes. Results for the remaining protocols can be seen in Table 4.2.

While UDP/443 (QUIC) increased the most by a factor of three, it still shows the worst response rate below UDP/53, even though QUIC was finally standardized in May 2021 [1]. Interestingly, we show in Chapter 8 that 210k IPv6 addresses supported QUIC in 2021, more than twice as many addresses as reported by the *IPv6 Hitlist*. However, the respective study included addresses from DNS resolutions many of which are located in aliased prefixes (*e.g.*, from Cloudflare, Fastly, and Amazon) and thus filtered by the hitlist service. We analyze this effect in more detail in Section 4.4.

While an overall growth is visible throughout the four years, the number of responsive IPv6 addresses to ICMP, TCP/80 and TCP/443 slightly decreases in between the analyzed scans from 2019 and 2020 (see Table 4.2). This is primarily due to input sources only added once, *e.g.*, rDNS data. A fraction of these addresses are not responsive anymore after several scans. However, without updates of the input, newly responsive addresses are not discovered.

Figure 4.2 shows the distribution of responsive addresses across ASes. Compared to the overall input, the distribution is flatter. The top AS, Linode (AS63949), covers only 7.9%, followed by China Telecom (AS4812). 50% of responsive addresses are covered by 14 ASes. A distribution of addresses responsive to individual protocols can be seen in Figure 4.4.

Whereas the cardinality of the responsive address set remained relatively stable, with a small increase over time, the set of actually responsive addresses per scan changes more drastically. Figure 4.6 indicates the stability of the responsive address set over time. It shows how many addresses are newly responsive or unresponsive for each scan compared to the previous scan. In case of new IPv6 addresses, the figure differentiates between addresses that are completely new or were already responsive in a previous scan but not the last. Generally, a frequent churn of 200k to 500k IPv6 addresses can be seen between two consecutive scans within one to five days. However, unresponsive addresses are frequently recurring afterward. Nevertheless, completely new IPv6 addresses are regularly visible. As explained in Section 4.2 the *IPv6 Hitlist* service is not executed daily anymore but takes multiple days resulting in the increased churn in between scans towards the end of the analyzed period as seen in Figure 4.6.

More significant changes are either due to issues with the aliased prefix detection, if a drop is directly visible after a large increase or due to missing sources for a scan if the drop is before the rise. Increases without a close drop are primarily due to added sources, *e.g.*, an addition of IPv6 addresses from rDNS scans as used by Fiebig *et al.* [43].

In total since July 2018, 46.8M non-aliased addresses have been responsive at least once and to one protocol. This is primarily due to ICMP with 45.3M (96.8%) addresses followed by TCP/80 with 8.6M (18.4%). This shows that frequent change is visible, and an up-to-date service is necessary to provide a high-quality service for further research.

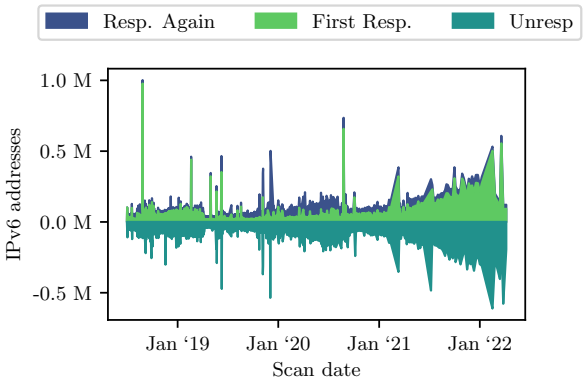

Figure 4.6: Development of responsive addresses over time. Later scans have a larger runtime covering up to 7 days, increasing the seen churn in between scans.

4.3.3 Key Take-Aways and Suggestions

The analysis of the hitlist state shows that a regularly updated, ongoing service is required to provide an up-to-date view and adapt to regular changes within the Internet. While 176.6k addresses were responsive throughout the entire period (5.4% out of 3.2M on April 7, 2022), regular churn is visible across all protocols. On the other side, the ongoing accumulation of addresses leads to an overall input list containing unresponsive addresses. Especially regular traceroutes identify a large set of addresses derived from a fraction of EUI-64 IIDs but located within regularly changing prefixes (see Section 4.3.1).

Furthermore, ongoing services require regular monitoring to understand the impact of network changes on the scanning methodology, e.g., the impact of the GFW that slowly ramped up over time and changed behavior frequently. The GFW injections additionally led to increased problems in the comparability of results. Chinese vantage points are most likely affected by the GFW injection as well but on the complete opposite set of addresses, namely targets outside Chinese networks.

We suggest frequent monitoring of the hitlist in the future by the operators but also the community. As an ongoing service, maintenance is required and a regular assessment of results helps to provide a useful service to the community. Collective efforts, such as GINO, are important to maintain ongoing research services.

4.4 Aliased Prefix Analysis

Besides the difficulty of IPv6 scans due to the inherent problem of the large address space itself, challenges occur due to the fact that individual addresses do not necessarily identify individual targets. With IPv6, servers can be reached using multiple IP addresses or even complete prefixes. They often appear as fully used prefixes with each address responsive, *e.g.*, to ICMP or TCP handshakes. The most commonly assumed reason is aliasing, where a *single* target is reachable using a complete prefix. The *IPv6 Hitlist* tries to identify these prefixes

as described in Section 4.2. Regarding scans, this is mainly a problem with IPv6. However, similar occurrences have also been mentioned in combination with IPv4 address scans, *e.g.*, by Izhikevich *et al.* [136] or Alt *et al.* [137]. We compare our findings to IPv4 in Section 4.5.

The implemented mechanism by the *IPv6 Hitlist* tests for responsiveness. However, identified prefixes do not have to be used by a single endpoint, *i.e.*, as an alias. Other reasons can be for example Content Delivery Networks (CDNs), using complete IP prefixes for multiple servers, or middle-boxes and proxies preemptively terminating connection attempts. Especially in the case of CDNs, (a subset of) these fully responsive prefixes might be a valuable input for advanced scans, *e.g.*, to analyze protocol deployments such as QUIC or TLS 1.3. In the following, we investigate these address regions in more detail to allow a better understanding of the *IPv6 Hitlist* and a better usage of its outcome.

In 2018, the *IPv6 Hitlist* service identified 12k aliased prefixes of different sizes. The number of aliased prefixes increased steadily throughout the year, reaching 42.8k in January 2022 and a sudden increase to 111.5k prefixes afterward. The latter growth by 66.4k prefixes is due to a single AS, namely Trafficforce (AS212144), a Lithuanian network starting to announce a larger number of prefixes in February 2022, solely limited to IPv6. All aliased prefixes were /64, responding to ICMP but not TCP/80. This sudden increase by a single operator stands out. The classification was based on successful ICMP probes and is reproducible.

We analyze whether the increasing number of addresses in the overall hitlist (cf. Section 4.3) results in the remaining increase from 12k to 42.8k labeled prefixes. As explained in Section 4.2, the service only checks for aliased prefixes of size /68 and longer, if more than 100 addresses from this prefix are part of the *IPv6 Hitlist* service input. Thus, aliased prefixes might remain unrecognized if too few addresses were in the input. Therefore, we check the size of aliased prefixes shown in Figure 4.7. It shows the distribution of aliased prefixes across prefix sizes for a single snapshot each year. It has been similar throughout the years with a small percentage of prefixes within /28 and /60. The shortest aliased prefixes are several /28s announced by EpicUp (AS397165) a US-based cloud provider. However, most aliased prefixes constantly have a length of /64 where the hitlist does not require a threshold. For this prefix length, only a single address is required to trigger the detection mechanism. Newly identified prefixes throughout the years were either not aliased in 2018 or did not contain a single address. Therefore, testing /64 prefixes if a single address is known to the *IPv6 Hitlist* proves to be effective to detect a majority of aliased prefixes.

4.4.1 Fingerprinting Aliased Prefixes

We used fingerprinting approaches to analyze aliased prefixes identified by the *IPv6 Hitlist* service in more detail. These methodologies allowed us to evaluate whether aliased prefixes were a single responsive host, or whether some identified aliased prefixes contained multiple hosts. We combined two methods to fingerprint hosts in identified aliased prefixes, namely fingerprinting based on TCP response features as conducted by Gasser *et al.* [8] and the Too Big Trick (TBT) as presented by Beverly *et al.* [62].

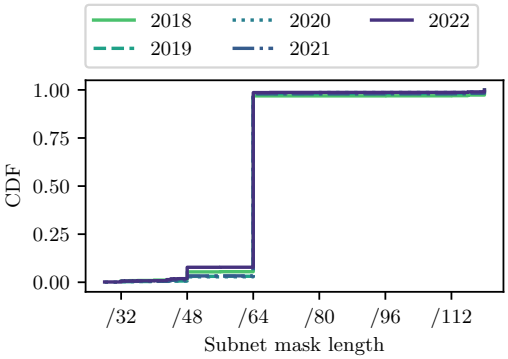

Figure 4.7: Distribution of aliased prefix sizes over time. The plot for 2022 excludes Trafficforce (AS212144) accounting for 60 k (61.6 %) aliased prefixes mostly /64. For each year, a snapshot representing a single scan is used. More than 90 % of aliased prefixes had a length of /64.

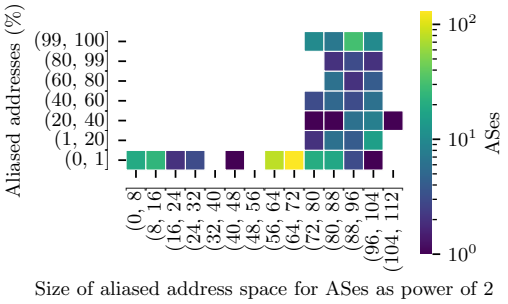

Figure 4.8: Number of aliased addresses (power of two) within ASes in relation to their overall number of announced addresses. If multiple prefixes within an AS are aliased, the number of addresses are summed. For some ASes, more than 99 % of the announced IPv6 addresses were covered by aliased prefixes. Results are based on the aliased prefix detection of April 7, 2022.

***TCP Fingerprints**:* We used TCP fingerprints extracted from the scans for the aliased prefix detection [8]. They rely on different features from the TCP handshake, namely the *Optionstext*, an order preserving string representation of TCP options, the TCP window size and window scale option, the Maximum Segment Size (MSS) and the iTTL. The iTTL rounds the TTL to the next power of 2 [138], [139] and thus represents the likely selected initial TTL. Therefore, the iTTL reduces inconsistencies based on routing changes or middleboxes. We omit the timestamp analysis as Linux machines using kernel 4.10 or newer randomize the value and are expected to show increased deployment since 2018. In contrast to the TBT, the same values between two IPv6 addresses do not necessarily indicate the same host, but similarly, varying values suggest different hosts.

TCP fingerprints can be derived for 33.5k aliased prefixes. The remaining prefixes were detected based on ICMP scans. For 33.3k (99.5%) of these prefixes, all values match while for 160, differences are visible. The iTTL, MSS, window scale option and *Optionstext* only differ for addresses within up to 13 prefixes. In contrast, the TCP Window Size is different for addresses in 154 aliased prefixes. However, the Window size can change on a single host within different connections and different values are not necessarily a discriminating factor. In comparison to the evaluation by Gasser *et al.* [8] we see similar results. Most fingerprintable prefixes respond with a uniform behavior, while a small fraction shows variable behavior. This supports the initial assumption that detected fully responsive prefixes are often aliases for the same host. However, some fingerprints indicate that some detected address blocks are potentially used differently and by multiple hosts.

***Too Big Trick**:* We used the TBT as an additional indication of prefix usage. It relies on the characteristic of IPv6 that only end hosts are allowed to fragment packets and relies on a shared Path Maximum Transmission Unit (PMTU) between aliased addresses for the same host. In

general, if a router receives an IPv6 packet that is too big, it has to notify the sender using an ICMPv6 Packet Too Big Messages. Afterward, the sender should update its PMTU cache and fragment the respective packet. While this process was initially utilized to identify alias addresses of routers, Song *et al.* [57] proposed it to analyze aliased prefixes. We shortly describe the required steps in the following:

(i) The Too Big Trick verifies that a set of addresses (8) under test within a prefix replies to ICMP Echo Requests of size 1300 B, slightly larger than the minimum required MTU for IPv6 of 1280 B, without fragmentation.

(ii) It sends ICMPv6 Packet Too Big Messages itself to *one* of the addresses and verifies that the next round of ICMP Echo Request messages is in fact fragmented.

(iii) It sends ICMP Echo Requests to the remaining addresses under test *without* the preceding error message. In case all addresses are aliases for the same device and interface, they share the same PMTU cache and should now fragment the response.

The methodology only provides insights if targets respond to the initial ICMP Echo Requests without fragmentation. We used this methodology on the 111k prefixes identified by the *IPv6 Hitlist* service on April 7, 2022 and received successful results for 29.4k. Out of these, for 27.6k (93.75%) all eight responses were fragmented after the initial error message to a single address indicating a shared PMTU cache. Only for 249 (0.85%) prefixes, no request resulted in a fragmented response but each individual address required a new error message.

Interestingly within the remaining prefixes (1592, 5.4%), between two and seven addresses share a PMTU cache but not all. This effect is mostly seen with Akamai and Cloudflare with 1k and 268 prefixes respectively. This supports our assumption that a fraction of identified *aliased* prefixes is not completely in line with the initial definition of a single host with the complete prefix as an alias. Nevertheless, addresses from these fully responsive prefixes are still not assigned to single hosts used within a load balancing setup, *e.g.*, of CDNs.

4.4.2 Characteristics of Aliased Prefixes

Besides these technical fingerprints, we analyzed additional characteristics of aliased prefixes in addition to the originating ASes. We argue that these characteristics allow for a more informed evaluation and usage of the *IPv6 Hitlist* results in the future.

Are Individual Prefixes Fully Responsive or Complete ASes?: To analyze whether aliased prefixes are more likely due to individual network entities or set up on an AS-level, we analyzed the fraction of aliased addresses within each AS in respect to the total number of announced IPv6 addresses by an AS. Figure 4.8 shows this relation for all ASes with at least one aliased prefix. The total number of addresses from aliased prefixes is given as power of two on the x-axis ranging from 2^8 to 2^{112} addresses. This value does not necessarily represent single prefixes but sums all aliased prefixes within each AS. The highest number of IPv6 addresses from aliased prefixes is again due to EpicUp (AS397165) announcing 61 fully responsive /28 prefixes. The y-axis depicts the fraction in respect to all announced IPv6 addresses by the respective AS. The axes are binned for better visibility.

Table 4.3: Responsiveness of aliased prefixes. For each prefix one random address is tested to reduce impact.

Protocol	# Prefixes	# ASes
ICMP	39.0k	270
TCP/443	31.9k	155
TCP/80	32.3k	179
UDP/443	28.8k	41
UDP/53	172	32

While the fraction for many ASes is less than 1‰, for 80 ASes more than 50% of announced addresses are located in aliased prefixes, and for 61 ASes even more than 90% are reached. The most prominent candidates in the latter category are Fastly (AS54113) with 95.3%, but also AS33905 owned by Akamai and AS209242 owned by Cloudflare both aliased to 100%. We argue that in these cases even without exact fingerprinting, it is highly unlikely that all addresses are an alias of a single host because these CDNs serve numerous websites and clients. The complete exclusion of all addresses might almost exclude complete ASes.

Are Domains Hosted in Aliased Prefixes?: To further evaluate the effect of an exclusion of all aliased prefixes, we analyzed whether domains are hosted within these networks. In the case of higher layer protocol evaluation, including TLS, QUIC or HTTP but also the analysis of Internet consolidation, these targets can be highly relevant and should not be excluded completely. We used a single snapshot of our DNS scans introduced in Section 4.2 from April 7, 2022 to analyze whether identified, aliased prefixes host domains and to which extent.

Based on our data 15.0M domains resolved to 5.2k aliased prefixes in total. 133 different ASes announced these prefixes. Note that this is a lower bound, given that we resolved a subset of the DNS namespace and that load balancing might impact our scans. Nevertheless, it shows that a fraction of hosting infrastructure can be missed by research based on the *IPv6 Hitlist*. The most prominent AS with aliased prefixes hosting domains was Cloudflare (AS13335) with 115 prefixes each hosting a mean of 167.0k domains. The highest number of domains even reached 3.94M domains within a single fully responsive /48. Additional CDNs, such as Fastly (AS54113), Amazon (AS16509), and Google (AS15169), were affected.

We investigated how many of these domains were on top lists. The DNS snapshot for this study contained three top lists, all containing 1M domains respectively on April 7, 2022. Domains from all top lists resolved to IPv6 addresses within aliased prefixes: 177.0k Alexa Top 1M; 170.2k Majestic Top 1M; 118.0k Umbrella Top 1M. Considering the Alexa Top 1M, 129 and 22.6k domains were within the Top 1k and Top 100k respectively, including `facebook.com` and `spotify.com`. Affected domains from the Majestic Top 1M were of similar ranks. In contrast, only 53 affected domains from the Umbrella top list were within the Top 1k.

Including addresses out of aliased prefixes contained in AAAA records would include 2.7M distinct addresses, a small fraction of the respective aliased prefixes. We argue that all of these or at least a subset of IPv6 addresses should be considered by researchers in the analysis of protocols on top of IPv6 even though they are identified as aliased prefixes.

Are other Protocols Responsive?: Based on the previous findings, we tested aliased prefixes for their responsiveness to all protocols. We excluded 66.4k aliased prefixes announced by Trafficforce, to reduce the impact as they were only responsive to ICMP probes during the initial multi-level aliased detection. We only selected a single, random address from each prefix to reduce traffic on real aliases and due to our assumption that all addresses behave the same. Scan results for 42.8k aliased prefixes can be seen in Table 4.3. Most probed addresses were responsive to ICMP or TCP/80, and already tested during the multi-level aliased prefix detection. Additionally, TCP/443 and especially UDP/443 were supported by a majority of tested addresses as well. As shown in Table 4.2, UDP/443 accounted for the lowest number of responsive addresses with 98.1k in the *IPv6 Hitlist*. Therefore, using a single address from each aliased prefix increases the amount of responsive addresses by 29.4%. This is in line with our observations that aliased prefixes are frequently seen in combination with CDNs and the findings presented in Chapter 8 that large providers mainly drive the deployment of QUIC. In contrast, only 172 addresses were responsive to UDP/53 probes, *e.g.*, from Cloudflare or Misaka (AS50069), an anycast DNS service.

In this scan, aliased prefixes were responsive to at most four protocols. Only Cloudflare originates at least one prefix responsive to each probe respectively. In no prefix was UDP/443 and UDP/53 seen in combination.

4.4.3 Key Take-Aways and Suggestions

The aliased prefix *detection of the* IPv6 Hitlist *service is an important feature, necessary to allow an ongoing, feasible service and to prevent biases in the set of responsive addresses, e.g., towards Amazon with more than 200M addresses. However, the initial definition as an alias for a single host does not necessarily hold and the number of aliased prefixes is increasing over time. Their complete removal can result in the exclusion of complete ASes, e.g., Fastly, or targets hosting multiple millions of domains, e.g., within Cloudflare. An informed assessment of these fully responsive prefixes depending on the usage of hitlist results is essential and can drastically change results and improve insights.*

We suggest extending information regarding these address regions, e.g., statistics about hosted domains, to allow an informed selection of address candidates out of these. Furthermore, at least a single address out of each aliased prefix can be added to the hitlist. Even if the complete prefix is an alias for a single host, testing its responsiveness with one of these addresses can cover its behavior and offer a valuable foundation for future research. As shown in Table 4.3, random addresses are often responsive to different protocol scans. However, known addresses contained in the input from DNS scans or passive sources should be used if possible because these addresses are either actively announced by operators in DNS or known to be used by network devices (e.g., responsive to RIPE Atlas traceroutes).

4.5 Highly Responsive Prefixes in IPv4

Fully responsive prefixes mostly are a problem in IPv6 due to their size and thus their impact on scans. However, they can be found in IPv4 as well. The work presented by Sattler *et al.* [14]

identifies the impact on large scale across more than 160 ports, scans covering multiple years and global vantage points. This section summarizes the key-findings and their relation to the above analysis of aliased prefixes in IPv6 and the *IPv6 Hitlist*. The published paper provides additional insights regarding the overall methodology, stability over time, spatial stability between vantage points and application layer information.

Using ZMap [31], the full IPv4 address space can be scanned in minutes. However, especially for TCP scans, it just reveals whether the respective port is open, not whether an actual service is offered. Previous work mentioned the existence of fully responsive prefixes in IPv4 [136]. However, to the best of our knowledge, there has been no attempt yet to identify the various categories that together constitute these prefixes.

As described in Section 4.2, the *IPv6 Hitlist* identifies fully responsive prefixes within IPv6 based on 16 random probes and the assumption that it is highly unlikely to randomly select 16 responsive addresses within a prefix. However, the IPv4 address space is already depleted, more widely used and drastically smaller. Therefore, it is feasible to scan the complete address space and use this data to identify fully responsive prefixes. However, Internet-wide scans are often impacted by different factors.

(i) Packet loss can occur, causing a prefix to appear as not 100% responsive, even when it actually is. It is important to note that even a minimal loss rate can substantially impact an Internet-wide IPv4 scan.

(ii) Scalable port scanners such as ZMap sometimes trigger rate limiting on the receiver side. While built-in target randomization helps mitigate the risk of overburdening target systems, it cannot completely eliminate this problem.

Thus, a strict focus on fully responsive prefixes where *all* addresses are responsive might be misleading and so-called highly responsive prefixes should be considered. To identify HRPs, only a subset of addresses above a defined threshold need to be responsive.

4.5.1 Datasets

The following describes a subset of used data relevant to this work. The work presented by Sattler *et al.* [14] used additional data sources (*e.g.*, application layer and DNS scans) to analyze IPv4 prefixes in more detail. However, this work focuses on the evaluation of IPv4 prefixes in general and a comparison to IPv6. For further insights, we refer a reader to the published paper [14].

We conducted different port scans from three vantage points as on demand scans orchestrated by GINO. Two vantage points are located in Germany (Munich and Saarbrücken), while the third is located in Australia (Sydney). The first two vantage points are used for most of the scans. The third was used in sync with Munich to evaluate the visibility of HRPs from two geographically and topologically distant vantage points (see [14] for details about this). We use ZMap and its TCP SYN scan module to perform these scans. The used scan module reports a target as being responsive, if a TCP SYN-ACK packet has been received from that target. We run two types of scans from Munich and Saarbrücken:

(i) Long-running scan campaign for TCP port 443 on the announced IPv4 address space (January 2021 — January 2023). We use the resulting data to analyze the stability of HRPs over time. These scans are part of the weekly TLS scans conducted by GINO and not specifically set up for this work (see Chapter 2).

(ii) A wider-range IPv4 scanning campaign targeting 36 different TCP ports (August 2022 – September 2022). These scans allow us to evaluate HRPs while considering potential artifacts on our scan machines, of the used blocklist, and scanned targets.

To complement our own scans, we used data provided by Rapid7's Project Sonar [140]. Note that common ports (*e.g.*, 80 or 443) are scanned more frequently by Rapid7. We use these frequent scans to determine the stability of HRPs besides our results. Rapid7 also provides application-layer scan data. We rely on Rapid7 data to reduce the need to run our own measurements and to reach a large coverage of different ports and protocols. For our analysis, we used port scans conducted by Rapid7 from February 1, 2021 to the end of 2022.

4.5.2 Highly Responsive Prefixes

To obtain a viable and robust responsiveness threshold to identify HRPs, we analyze the distribution of responsive addresses inside a prefix.

A prefix length of 24 bits is a de facto limit for the most-specific, globally routable prefix length in IPv4 [141]–[143]. Thus, this constitutes the smallest IPv4 prefix that reliably propagates in the Border Gateway Protocol (BGP), representing the most limited subset of addresses we evaluate. The number of responsive addresses within /24 prefixes shows a large variety. This can be seen in Figure 4.9 for two standard ports (TCP/80 for HTTPS and TCP/443 for HTTPS) but also alternative/random ports (TCP/8984 and TCP/60000). For TCP/443 scans, more than 75% of prefixes contain fewer than 15 responsive addresses, and 91% include at most 50 addresses. However, there is a substantial increase in prefixes with 231 or more responsive addresses (*i.e.*, 90% of a `/24` prefix). Although these account for only 2.2% of *visible /24 prefixes*, they account for *30% of responsive addresses*. A similar pattern is visible for TCP/80 in Figure 4.9b. For the remaining ports (Figure 4.9c, Figure 4.9d), the impact is even stronger.

Consequently, this observation holds true not only for specific ports but also extends to all other ports. Except for the SSH port (TCP/22), at least 30% of the responsive addresses are located within HRPs. The SSH port is considered sensitive, which may be why operational configurations treat it differently. Separate evaluations on other ports are available on our website [144]. These supplementary evaluations also exhibit a similar pattern to what is observed for TCP/80, TCP/443, and the aggregated analysis. Well-known ports (*e.g.*, TCP/80, TCP/443) and their alternative ports (TCP/8080 and TCP/8443) have an HRP address share of 30%-40%. Ports for less popular services and other alternative ports generally have a higher share of HRP addresses, which goes up to 80%. In general, the smaller the overall set of responsive addresses and the less known a port is, the higher the fraction of HRP addresses will be. One reason for this is a group of HRPs that are responsive on all scanned ports. 50% of all distinct HRPs are only classified as such based on the responsiveness of a single port. For port TCP/80, we find 59.8k prefixes; for port TCP/443, we find 40.7k; and for port TCP/25, we find 33.8k. 30% of prefixes

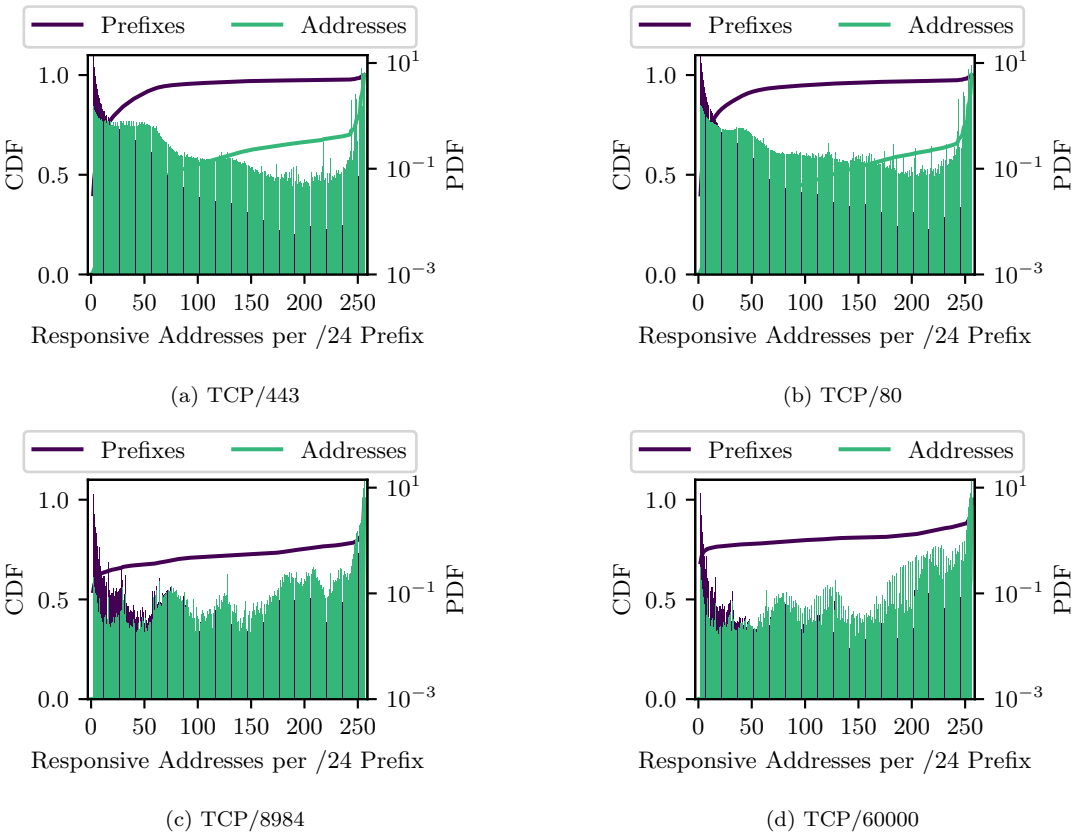

Figure 4.9: Respective (bars – right y-axis) and cumulative (line – left y-axis) probability distribution of responsive addresses inside prefixes. The address data represents the influence of prefixes on the scan results. The prefix indicates that the majority of prefixes do not meet the criteria for HRPs. Note the logarithmic axis for the right Y-axis. More ports can be found on our published website [144]

are classified as HRP based on two responsive ports 80, and 443. However, 4700 (1%) HRPs are responsive on at least 139 ports within our data.

Key take-away: *We find HRPs for all evaluated TCP ports and show that they cover between 30% and up to 80% of all addresses responsive on a given port. While 80% of HRPs are classified as such based on one or two responsive ports, 5% are responsive on more than 120 ports. Due to the nature of UDP scanning, the role of HRPs is less prominent. However, we can still find instances of UDP HRPs.*

4.5.3 Origin ASes

The HRPs we find are announced by 67.2k origin ASes. We determine the origin AS with BGP dumps from RouteViews [145] from the day of the respective scan. We investigate which ASes announce HRPs and to what degree an AS may deploy features that make a prefix an HRP.

Of the ASes that announce HRPs, 42% have HRPs for a single port; 76% of ASes have HRPs with at most five different, responsive ports. 118 ASes announce at least one HRP where the

IP addresses respond on all ports. Table 4.4 shows the top ten ASes with at least one HRP. Cloudflare (AS13335), for example, announces a /16 that is fully responsive on all ports. We identified that Spectrum [146]–[148], a reverse TCP/UDP proxy for DDoS protection, is related to the affected range. We contacted Cloudflare and confirmed with them that the service is built on the principle of binding to all ports. Google, the second largest AS with an HRP for all ports, offers a similar service.

In Table 4.4, we find Akamai at the top of the list with 22k HRPs in AS16625 and 21k in AS20940. The latter AS has nearly all of its prefixes where we find responsive addresses also classified as HRPs. Notable other ASes are Amazon, which has 6k HRPs, but these cover only 4.4% of its reachable prefixes. The ASes with an HRP on all ports are all large network infrastructure providers: Cloudflare, Amazon, Google, Cogent, and TANet[1]. It is not too surprising that CDNs are commonly connected to HRPs: they deploy various techniques to support millions of domains served on limited address space [149]. ASes primarily providing CDN services consist of up to 99% of HRPs compared to their overall announced address space (see HRP share in Table 4.4). However, this impact on scan data has so far received little attention.

We also evaluate the AS distribution per port. For port TCP/443, Akamai has the most HRPs, followed by Google, Cloudflare, Amazon, and Fastly. These five organizations cover 64% of all port 443 HRPs. While Akamai's AS20940 has more port 443 HRPs, AS16625 (also Akamai) has more than double the amount of HRPs on port 80 compared to port 443. This result may reflect different use cases that the prefixes inside these ASes serve.

CDNs affinity to HRPs: We identified several CDN ASes in our results that are nearly entirely populated with HRPs. For instance, Akamai exhibits 97.8% and 85.6% of announced /24 prefixes classified as highly responsive for two different ASes, while Cloudflare demonstrates 98.3%, and Fastly registers 87.6%. Therefore, we set out to uncover the reasons for CDNs using HRPs. For Cloudflare, we find a paper from Fayed *et al.* [149] with an accompanying blog post [150] that explains Cloudflare's *addressing agility* approach. This technique decouples IP addresses from domain names and services. The authoritative name server can select the addresses in the query response from a full prefix. Hence, all addresses inside this prefix have to be responsive, and Cloudflare's approach needs to handle all prefix-assigned services. We verified this assumption by registering a test website with Cloudflare, resolving the domain to an A record, and connecting to several addresses inside the A record's /24 prefix. Each TLS handshake, which included our registered domain name in the server name indication extension to any address inside this prefix, resulted in the same certificate for our domain name.

Key take-away: *In this section we show that the distribution across ASes is dominated by CDNs and other content providing ASes. HRPs make up a substantial part of these ASes. We could confirm that Cloudflare's address agility technique is responsible for Cloudflare's large share of HRPs. We assume that other CDNs deploy similar techniques.*

[1] Amazon and TANet are only missing out on a single port. See [14] for more information.

Table 4.4: Top 10 ASes based on the number of HRPs we detect across all scans in August 2022. The HRP share is the degree to which an AS is filled with HRPs.

AS		Visible /24	HRPs	HRP Share	Visible Ports	Ports with HRPs
Akamai	AS16625	22.9k	22.4k	97.8%	5	3
Akamai	AS20940	24.7k	21.1k	85.6%	136	5
Telin	AS7713	12.5k	6.5k	52.5%	136	4
Amazon	AS16509	134.9k	6.0k	4.4%	136	135
DoD	AS721	4.9k	4.5k	91.3%	136	55
DoD	AS5972	4.5k	4.4k	99.3%	60	54
du	AS15802	4.3k	4.1k	96.0%	136	3
Cloudflare	AS13335	3.1k	3.0k	98.3%	136	136
Cogent	AS174	17.0k	2.8k	16.5%	136	136
TANet	AS1659	8.2k	2.4k	29.8%	136	135

4.5.4 Comparison to IPv6

Our *IPv6 Hitlist* service runs an aliased prefix detection during its regular scans. Note that the detection relies on merged ICMP and TCP/80 scans, thus it might detect aliased prefixes solely based on ICMP, which we do not use during our IPv4 scans. Furthermore, it detects prefixes on different prefix lengths. Comparing involved ASes shows similarities to our IPv4 analysis. Similar to the results shown in Section 4.4, Akamai (AS20940), Amazon (AS16509) and Cloudflare (AS13335) are heavily contributing to the number of HRPs. However, other ASes responsible for most addresses within IPv6 HRPs are not in the top ranks of IPv4 HRPs, *e.g.*, Fastly (AS54113) or Trafficforce (AS212144). We assume this to be due to different address assignment strategies and the general availability of addresses between IPv4 and IPv6, but also due to the available data in IPv6 limited to the hitlist input. A more detailed comparison of HRPs in IPv4 and IPv6 in the future can offer further insights into different assignment strategies and sibling prefixes between both protocol versions.

4.6 Discussion and Summary

***Fully Responsive Prefixes**:* The *IPv6 Hitlist* tries to identify *aliased* prefixes, where a complete prefix is used by a single host. However, we show that identified prefixes are not necessarily used by a single host, but these are sometimes a result of CDN or middlebox deployments. As shown in Section 4.4, multiple CDNs assign large parts of their owned address space to their fleet of servers, *e.g.*, Fastly or Cloudflare. Amazon alone would introduce 200M addresses from fully responsive prefixes. These addresses still do not represent individual hosts each, introduce significant biases, and result in a massively increased scan load. However, these prefixes are used in combination with multiple hosts for load balancing as seen with the Too Big Trick and host multiple millions of domains including highly ranked domains according to top lists but also name servers or mail exchangers. We suggest naming identified prefixes *fully responsive prefixes* in the future and to analyze their characteristics in more detail to allow for better differentiation. Fully responsive prefixes are a superset of aliased prefixes, identifiable by the implemented multi-level aliased prefix detection. However, not all identified prefixes are actual aliases on a single machine. While we will exclude most of these targets from the ongoing scans of the *IPv6 Hitlist*, we argue that a subset of these addresses can be included and should be

used by follow-up research. Especially in the case of higher layer protocol scans and evaluations, such as TLS or QUIC scans, these targets should not be excluded but considered, *e.g.*, based on up-to-date DNS resolutions.

We suggest that for each fully responsive prefix, at least a single address can be tested by research. Even if the prefix is an actual alias for a single host, it is an actual host, is part of the Internet ecosystem, and should thus be represented in the *IPv6 Hitlist* and considered in the future. Regarding fully responsive prefixes related to CDNs, we suggest a use case specific selection of addresses. Single addresses can be enough to identify server setups, *e.g.*, in respect to TLS where a centrally administered configuration within a CDN can be expected. In contrast, multiple addresses might be required to analyze hosted domains, certificates or websites to spread the load, *e.g.*, for Cloudflare with multiple million domains within a single fully responsive prefix.

***Service Maintenance**:* An intrinsic motivation of this research was to improve the foundation for IPv6 research and thus improve a valuable existing building block, namely the *IPv6 Hitlist* service [8]. Ongoing measurement studies that freely publish results and data are valuable research resources. The *IPv6 Hitlist* has been operated for more than five years and has been used by a multitude of research [9], [51]–[53], [55], [56], [124]–[128]. However, its operation and maintenance requires continual effort which can be difficult especially in an academic environment. While different approaches have evaluated the hitlist and proposed new address generation methodologies, they did not establish a new continuously running service. We updated the service and included newly identified addresses into the service to improve the hitlist for future use. To better enable future collaborations to maintain the *IPv6 Hitlist* service and allow reproducible network measurement studies, we additionally share all data used for this work and used analysis scripts [151]. This includes our adaptation to 6Tree, the distance clustering implementation and a tool to filter ZMap output as published by the *IPv6 Hitlist* service from the GFW injection.

***Conclusion**:* In this work, we took steps to dust the *IPv6 Hitlist* service after four years of ongoing operation and prepare it for the following years. Our steps include the analysis of the current state, cleaning the list of addresses caused by the Great Firewall of China, and evaluating the development of the *IPv6 Hitlist* over time. We further analyzed identified aliased prefixes and highlight that their strict omission excludes large CDNs such as Fastly or Cloudflare and thus multiple million domains from research based on the *IPv6 Hitlist*. Due to their load balancing mechanisms, announced prefixes appear to be aliased even though the backing infrastructure can be expected to be more than a single host and is valuable to analyze.

Chapter 5

Extension of the IPv6 Hitlist

The *IPv6 Hitlist* is an important source for IPv6 scans. As shown in the previous chapter, it is a stable source of information after filtering injected DNS responses. However, it contains less responsive addresses compared to scan results from IPv4 scans. Therefore, we set out to improve the quality of the *IPv6 Hitlist* with additional sources and new addresses.

We evaluated different new address candidate sources during multiple studies. We show that a combination of different methodologies is able to identify new, responsive addresses. Figure 5.1 provides an overview of the *IPv6 Hitlist* development. The increases indicated by the label **F**, **G**, **H** and **I** are related to this work and will be explained in the following.

We collected different new sources and analyzed different Target Generation Algorithms (TGAs), which are used to increase the coverage of IPv6 measurements by generating new responsive targets for scans. We evaluate their performance under various conditions and find generated addresses to show vastly differing responsiveness levels for different TGAs.

This chapter is based on the following publication:

> J. Zirngibl, L. Steger, P. Sattler, O. Gasser, and G. Carle, "Rusty Clusters? Dusting an IPv6 Research Foundation," in *Proc. ACM Internet Measurement Conference (IMC)*, 2022. DOI: 10.1145/3517745.3561440 [11]

> L. Steger, L. Kuang, J. Zirngibl, G. Carle, and O. Gasser, "Target Acquired? Evaluating Target Generation Algorithms for IPv6," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2023. DOI: 10.23919/TMA58422.2023.10199073 [13]

The first publication [11] builds in parts on Lion Steger's Master's thesis [120]. The second publication [13] builds in parts on Liming Kuang's Bachelor's thesis [152].

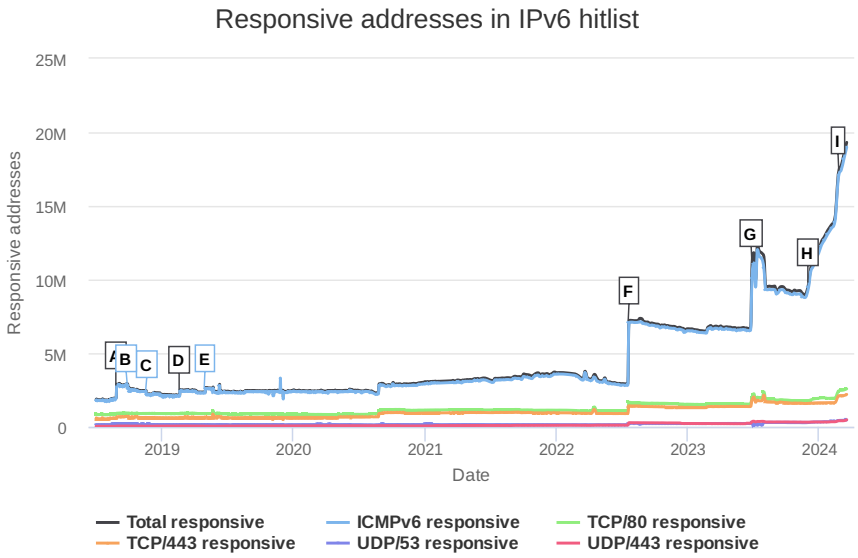

Figure 5.1: Responsive address history of the *IPv6 Hitlist* as of 26 March, 2024.

Author's Contributions: *This chapter combines key findings from two publications. The author had major contributions to both.*

The first work [11] was mainly done by the author, formulating the research questions, setting up required scans, collecting data and evaluating results. The author collected new, passive data sources and worked closely together with Lion Steger to apply target generation algorithms.

The distance clustering algorithm used in Section 5.3 was designed, implemented and applied by Lion Steger. It is mentioned in this work to fully explain changes at label ***F****.*

The second work [13] was mainly influenced by planning and setting up conducted measurements. The author supported this work by maintaining the IPv6 Hitlist service providing the input for Target Generation Algorithms and historic data. Furthermore, the author actively participated during discussions of results and the creation of the publication.

5.1 Motivation

The adoption of IPv6 is continuously increasing, with on average 40% of all Google users connecting via IPv6 in March 2023 [122]. Due to the sheer size and sparse population of the IPv6 address space, exhaustive scans such as in IPv4 [31], [153] are infeasible in the IPv6 Internet. Therefore, Internet measurements targeting IPv6 hosts rely on up-to-date collections of responsive addresses, often known as *Hitlists*. Moreover, the success of these measurements heavily depends on the quality of their input, reliable targets, and high coverage of the active IPv6 Internet.

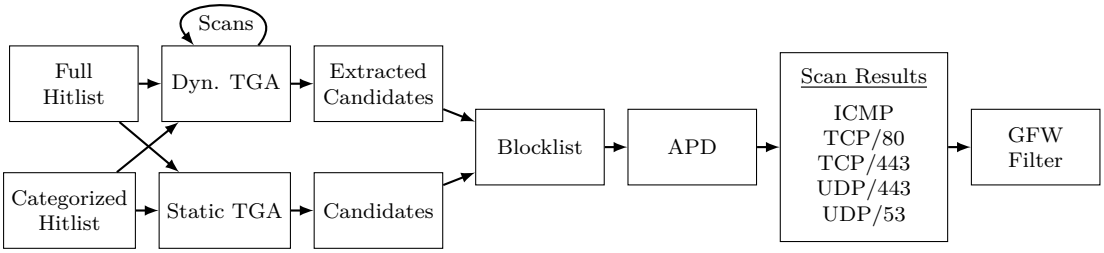

Figure 5.2: Pipeline to analyze TGAs (Table 5.1) and their performance within different IP address categories.

However, no major, new sources have been added to the *IPv6 Hitlist* after its creation in 2018 until the work presented in this chapter and [11], [13]. Besides new data sources from other research, different approaches exist to increase IPv6 address coverage, *e.g.*, by generating new targets. This is often achieved through so-called TGAs, which employ different methods such as machine learning [52], [53] and other pattern recognition techniques [44], [56]. New sources and TGAs have the potential to improve the *IPv6 Hitlist* by discovering new responsive addresses that can be added to the service, probed regularly and published to the community.

This chapter summarizes our efforts to improve the *IPv6 Hitlist* with new addresses in multiple iterations in between 2021 and 2023. This research enables fellow researchers to make better use of the *IPv6 Hitlist* and TGAs. Our contributions in this chapter are:

(i) We added new sources to the *IPv6 Hitlist*, and evaluated their effect on the service.

(ii) We evaluate the effectiveness of different TGAs to identify previously unknown addresses.

(iii) In order for users of the *IPv6 Hitlist* to benefit from our findings, we update the service and include the newly discovered addresses to the established service.

5.2 Data Sources and Target Generation

In the following, we describe our data sources and the application of TGAs introduced in Section 5.2.2 to improve the *IPv6 Hitlist*.

5.2.1 Data Sources

We used different snapshots of the *IPv6 Hitlist* as input for our Target Generation Algorithms. For the first study presented in Section 5.3 we used the full list of responsive addresses from the *IPv6 Hitlist* service from April 7, 2022. For the second study presented in Section 5.4 we used the full list from March 3, 2023. Besides the *IPv6 Hitlist* service, we used additional data throughout this chapter to improve the *IPv6 Hitlist* for future research.

***CAIDA Ark**:* While the *IPv6 Hitlist* contains its own traceroutes and data from RIPE Atlas, traceroutes conducted on a regular basis from the CAIDA Archipelago (Ark) Infrastructure are not included [154]. We used a snapshot of the data from March 2022 to analyze the value

Table 5.1: List of target generation algorithms with publicly available code used in this work. The checkmarks (✓) indicate whether the TGA has been used during the first or second study.

Year	Authors	Name	Scanning	Ref	*(i)*	*(ii)*
2016	Foremski et al.	Entropy/IP	Static	[44]		✓
2019	Liu et al.	6Tree	Dynamic	[55]	✓	✓
2020	Song et al.	DET	Dynamic	[57]		✓
2020	Cui et al.	6GCVAE	Static	[51]		✓
2021	Cui et al.	6VecLM	Static	[53]	✓	✓
2021	Cui et al.	6GAN	Static	[52]	✓	✓
2021	Hou et al.	6Hit	Dynamic	[54]		✓
2022	Yang et al	6Graph	Static	[56]	✓	✓
2022	Yang et al	6Forest	Static	[58]		✓
2023	Hou et al.	6Scan	Dynamic	[59]		✓

additional vantage points can offer to reveal new routers or whether the existing data set covers most contained targets.

***DET**:* Song *et al.* [57] collected IPv6 addresses from different services similar to Gasser *et al.* [8] and additionally used target generation algorithms. They advertise an ongoing service and data publication. However, we were only able to download a single snapshot of responsive addresses and did not receive a reply requesting additional data. Thus, we only used this snapshot as a new source of addresses and evaluated its value.

***IPinfo.io**:* As of December 2023, a new collaboration with *IPinfo.io* [155] was established. The collaboration implies mutual data sharing. Addresses identified by their service were added to the *IPv6 Hitlist* service. Their impact is not described in a previous study but visible in the service starting with the label **H** in Figure 5.1.

***DNS Scans**:* We used DNS scans from GINO. The direct AAAA resolution of domains were already used as input of the *IPv6 Hitlist* service, but the name server (NS records) and mail exchanger (MX records) domains were not explicitly included. We evaluated the quality of these domains as new input in Section 5.3 and included them into the *IPv6 Hitlist* service.

5.2.2 Target Generation Algorithms

Discovery of responsive targets for IPv6 scans is an important task since full address space scans are infeasible. Besides hitlists, combining targets from existing sources, *e.g.*, resolution of domain names, public sources and traceroutes, a variety of so-called Target Generation Algorithms (TGAs) were developed. TGAs take a completely different approach to this problem. They try to identify patterns within existing collections of responsive addresses called the *seed data set*, and generate new targets which are likely to be responsive, called a *candidate set*. These addresses can be used as input for scans and tested for their responsiveness. Some of these algorithms also implement their own dynamic scanning mechanisms, which allows them to adapt their search strategy based on intermediate scanning results, and achieve a higher response rate.

Table 5.1 provides an overview about algorithms we evaluated and used in this work. These were all the algorithms found in related work which provided publicly accessible source code as of April 2023. They extract structural information from IPv6 seed sets and apply different methodologies to improve the quality of generated addresses. They report different response rates which are hardly comparable. While we are not able to reproduce their published hit rates, we show that some algorithms can be used as valuable new sources in addition to the existing *IPv6 Hitlist*.

5.2.3 Target Generation Methodology

Figure 5.2 shows our pipeline to test TGAs on the different input files. In our studies we ran and evaluated TGAs listed in Table 5.1. Four were used during the first iteration (Section 5.3), while all 10 were used during the second iteration (Section 5.4).

We ran the static TGAs without any modifications apart from input files or output hyper-parameters. We modify the hyper-parameters number of epochs for 6GAN and the generation budget for 6GCVAE. To run algorithms in feasible time, we set the total number of epochs of 6GAN to ten and run 6VecLM with only the first of the predefined temperature hyper-parameters. 6GAN offers multiple modes for seed classification, of which we choose the *Entropy Clustering* method since the authors report the highest number of generated addresses for this method, which is the metric we optimize for. Since 6GAN and 6VecLM define the amount of input that is processed via hard-coded values and Entropy/IP is not intended for input greater than 100k addresses, we randomize the input data set with a static, reproducible seed.

During the second study, we set the output budget to 10M whenever possible, since we wanted to keep the size of the candidate sets at a similar scale to the input, *i.e.*, the hitlist. We implement the approach described by the authors of 6Graph to generate candidates from the dense regions identified by their algorithm. For 6Forest, this process is not fully described or implemented. Therefore, we follow the same procedure as for 6Graph, generating distance-based targets and additionally generating full combinations if the number of wildcards, *i.e.*, free dimensions, is smaller than four.

Before running the algorithms with dynamic scanning capabilities, we modified them *(i)* to conform to our scanning parameters and *(ii)* to output not only the addresses responsive to their scans but also the addresses which they probed, *i.e.*, the addresses which they considered to be in the *candidate set*. We did this because the integrated scanning mechanisms of the algorithms only scan with ICMP probes, while we wanted to apply our own scanning mechanisms with a variety of protocol probes. Furthermore, in order to compare the response rate of both dynamic and static algorithms, we scanned the candidate sets for both instead of only the results of the dynamic algorithms. In order to achieve consistency with the static algorithms, we ran each dynamic TGA with the same 10M scanning budget.

5.2.4 Scan Methodology

We used the existing pipeline from the *IPv6 Hitlist* (see Figure 4.1) to conduct scans testing these new IPv6 address sources and target generation methodologies. For both studies, we

Table 5.2: New input sources for IPv6 address candidates from the first study evaluated in this chapter. The covered ASes are set in relation (%) to the total number of ASes (29 k) announcing IPv6 prefixes based on RIPE RIS [131].

			ASes	
Source	Information	Addresses	Total	%
Passive sources	Extracted addresses from, *e.g.*, MX/NS records, CAIDA Ark [154], DET [57]	356.7k	3.6k	12.5
Unresponsive addresses	All addresses excluded from scans due to the 30 day unresponsive filter	638.6M	18.5k	64.9
6Graph [56]	Applied on responsive addresses in December 2021	125.8M	18.9k	65.2
6Tree [55]	Applied on responsive addresses in December 2021	37.6M	15.0k	51.7
6GAN [52]	Applied on responsive addresses in December 2021	3.3M	249	0.8
6VecLM [53]	Applied on responsive addresses in December 2021	70.3k	278	0.9
Distance clustering	Extending clustered address regions	5.3M	7.2k	25.0

combined the respective data sources into one target file (combining passive sources if available and generated addresses). We used ZMapv6 with the same probe modules, configurations and payloads as the *IPv6 Hitlist* service from the same vantage point. We further deploy the multi-level aliased prefix detection from Gasser *et al.* [8] and filter our scans with already known aliased prefixes and blocklists collected by the existing service.

5.3 First Discovery of New Addresses

Our first study to improve the *IPv6 Hitlist* was done during the work published by Zirngibl *et al.* [11]. The *IPv6 Hitlist* was initially created with a variety of sources [8]. However, other sources are available via different protocols (*e.g.*, MX and NS records) or scanning from different vantage points (*e.g.*, CAIDA Ark [154]). Furthermore, Gasser *et al.* [8] showed the potential of target generation algorithms as part of the *IPv6 Hitlist* input collection. The reachability of IPv6 addresses from available vantage points impacts the identification of candidates, *e.g.*, due to location specific load balancing. Additionally, active sources such as DNS scans and traceroutes are biased towards available scan targets. They can only detect IPv6 addresses mapped to a domain or responsive to traceroutes. Target generation algorithms try to mitigate these disadvantages from other sources. An overview of all new sources can be seen in Table 5.2.

Due to changes in the availability of data sources (*e.g.*, the discontinuation of the free availability of Rapid7 data), an ongoing update of the initial data collection efforts is necessary. We used the new set of passive sources explained in Section 4.2, the DNS MX and NS resolutions, CAIDA Ark traceroute efforts and the published list of responsive addresses by Song *et al.* [57]. In total, these sources result in 3.5M candidates. However, 90% of these addresses were already contained in the service, *e.g.*, from its DNS resolutions or traceroutes, and 7.5% were additionally located in aliased prefixes. 369.1k (71%) of the IPv6 addresses related to NS and MX records were located within Amazon, a highly aliased prefix. Therefore, these sources result in only 356.7k new addresses in total out of which 84.9k were not aliased.

As described in Section 4.2, the *IPv6 Hitlist* service stops scanning addresses unresponsive for more than 30 days and never re-evaluates their responsiveness. This list accumulated 787.7M

addresses until April 2022. We applied the blocklist filter and removed all candidates that showed GFW injection, resulting in 638.6M remaining candidates. The explicit inclusion of these addresses into the scan requires deactivating the final filter of the *IPv6 Hitlist* service (see Figure 4.1). We re-scanned these addresses once in 2022 to get insights into whether addresses are responsive again. We followed the same approach in 2024 and scanned all addresses part of the 30-day filter.

Target Generation: We applied different target generation algorithms to the set of responsive addresses of December 2021, including 6Graph [56], 6Tree [55], 6GAN [52] and 6VecLM [53]. We did not try to optimize or tune the algorithms but follow the respective explanations and standard parameters. Furthermore, we generated addresses with a simple approach named distance clustering (DC), extending more densely clustered address regions that show high entropy in the last nibble(s) of the address. Note, these regions were not fully responsive but only densely populated. Therefore, we collected clusters of addresses with at least 10 addresses and a distance of at most 64 between two addresses. Given the vast address space of IPv6, even a few addresses (10) within this comparably small distance are highly likely not assigned randomly but based on active assignment policies. We generated missing addresses within these clusters. We tested for potential new aliased prefixes after extending these clusters before our scans.

Table 5.2 lists how many addresses were generated by each applied method. While the respective publications often limit the number of generated addresses, *e.g.*, to 50k, we invested more computation time to increase the number of generated candidates. However, we did not test different parameters or subsets of our input during the first study but during the second iterations (see Section 5.4). We used published code and the contained default configuration.

In theory, 6Tree actively scans candidates during target generation and uses results to improve detection. To reduce scan impact, it contains functionality to detect aliases. However, it did not detect aliased prefixes effectively in our initial tests, quickly inducing bias towards fully responsive regions. 6Tree generated a set of more than 8.3M IPv6 addresses, part of a single /48 prefix originated by Akamai (AS20940). Most of these addresses were incrementally assigned and responsive but not labeled as aliased by the given 6Tree implementation. However, the aliased prefix detection of the *IPv6 Hitlist* identified these addresses as aliased. Therefore, we prevented active scans, limited 6Tree to target generation only, and used the detection proposed by the *IPv6 Hitlist* service during our scans.

New Responsive Addresses: We used ZMapv6 to scan all previously introduced, non-aliased addresses for ICMP, TCP/443, TCP/80, UDP/443, and UDP/53. We scanned multiple times across four weeks to account for packet loss or network events and aggregated the results afterward. The only source we did not completely scan multiple times is the set of unresponsive addresses due to its size and thus ethical reasons. Here, we only included responsive addresses during the first test into the following scans.

Even though we filtered responses injected by the GFW before generating addresses, some algorithms generated many addresses located within Chinese ASes and thus were affected by the GFW. Considering 6Graph, 18.5M (14.5%) out of the generated 125.8M addresses were

Table 5.3: Responsive addresses for new sources divided by protocol. The top ASes for each source based on the number of responsive addresses indicates potential biases in each data set. ASes are abbreviated as symbols explained in the footer of the table. *IPv6 Hitlist* results are from April 7, 2022.

	Responsive Addresses						ASes					
Source	ICMP	TCP/443	TCP/80	UDP/443	UDP/53	Total	Top 1		Top 2			Total
6Graph	3.8M	428.4k	491.1k	121.9k	78.6k	3.8M	52.1%	◆	5.1%	▼		10.7k
6Tree	2.2M	374.2k	425.5k	116.6k	62.8k	2.2M	41.0%	◆	8.0%	▼		11.5k
Unresponsive addr.	1.2M	274.8k	282.3k	18.6k	51.6k	1.3M	34.4%	▲	6.2%	▼		9.0k
Distance clustering	637.1k	167.7k	193.4k	85.1k	32.4k	651.0k	14.9%	●	10.9%	★		5.5k
Passive sources	21.0k	1.5k	1.9k	358	3012	21.6k	6.7%	●	3.2%	★		2.9k
6GAN	4.3k	27	28	2	2	4.3k	82.8%	◆	12.3%	▲		39
6VecLM	990	103	116	38	22	1.0k	17.1%	◆	14.9%	◆		105
New Sources	5.4M	764.9k	843.4k	164.0k	144.3k	5.6M	26.8%	◆	5.8%	▲		14.6k
IPv6 Hitlist	3.2M	910.8k	1.1M	98.1k	140.7k	3.2M	7.9%	▲	7.4%	▼		15.7k
Total	8.6M	1.7M	1.9M	266.2k	287.4k	8.8M	25.5%	◆	5.5%	▲		17.3k

◆ Free SAS (AS12322), ▲ VNPT (AS45899), ▼ DigitalOcean (AS14061), ● China Mobile (AS9808), ★ Racktech (AS208861),
◆ CERN (AS513) ▲, ARNES (AS2107), ▼ home.pl (AS12824), ● Deutsche Glasfaser (AS60294), ★ Akamai (AS20940),
◆ Level3 (AS3356), ▲ Linode (AS63949), ▼ China Telecom (AS4812)

affected. We filtered these responses for the following analysis. Table 5.3 shows the cleaned number of responsive addresses for each source.

For the new passive sources, 25% of non-aliased addresses were responsive (21k out of 84.9k). For 638.6M previously excluded addresses, unresponsive for at least 30 days, more than 1.2M addresses were responsive again. However, the responsiveness of these addresses decreased after the initial scan to nearly half the number of addresses. Our additional scan of all addresses unresponsive for 30 days in 2024 showed that this improvement should be done on a regular basis. Around 4.0M and 474.4k addresses were responsive to ICMP and TCP/80 again respectively. This scan explains the drastic increase of responsive addresses at label **I** in Figure 5.1.

Regarding target generation algorithms, our naive approach to extend densely assigned address regions achieved better results than more sophisticated approaches, namely 6GAN and 6VecLM. The latter methods only resulted in 1k and 4.4k responsive addresses respectively. The hit rate of both algorithms was low even before filtering GFW injected responses. Our naive distance clustering approach resulted in 651k addresses with a hit rate of nearly 12%.

6Tree and 6Graph resulted in the highest number of responsive addresses with 2.2M and 3.8M, respectively. However, due to their large number of generated addresses, their discovery rate was 6% and 3%, respectively.

In total, we were able to identify and generate 5.4M new or previously removed addresses responsive to at least one protocol. This resulted in nearly twice as many responsive addresses for UDP/53 in total and three times more addresses for UDP/443. Interestingly, slightly fewer addresses were responsive to TCP/80 and TCP/443 compared to the *IPv6 Hitlist*. However, new sources identify 168% more responsive addresses to ICMP probes compared to the *IPv6 Hitlist*. This is mostly due to the bias of target generation algorithms towards specific providers which we discuss next.

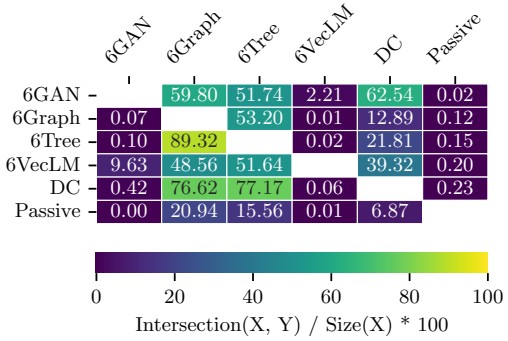

Figure 5.3: Overlap between responsive addresses from new sources in % in respect to the total number of responsive addresses for each row.

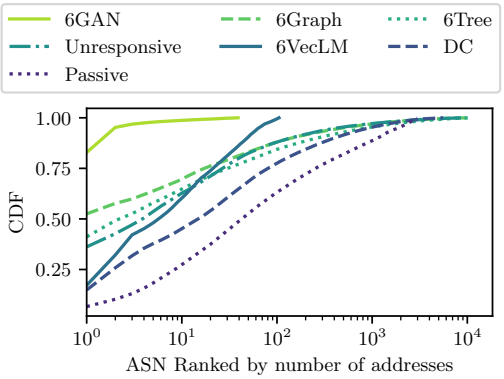

Figure 5.4: AS distribution of responsive addresses from new inputs.

Overlap: Individual sources do not necessarily contribute unique responsive addresses to the *IPv6 Hitlist*, but an overlap between new sources is visible. Figure 5.3 shows the overlap between all new sources in relation to the total number of responsive addresses for each row in percent. Thus, 89.34% of all responsive addresses generated by 6Tree were also generated by 6Graph. All target generation algorithms identified responsive addresses which were also part of passive sources and thus visible in traceroutes or DNS data. Furthermore, all sources provided unique responsive addresses and thus show potential to improve the hitlist in the future. Even though new passive sources analyzed in this work only provide 21.6k addresses (0.7% of to the total of 3.2M on April 7, 2022), they offer new responsive candidates that are also not covered by target generation algorithms. Especially with the end of freely available data sources, *e.g.*, Rapid7 forward DNS data, a frequent evaluation of new sources will be valuable.

Distribution Across ASes: To check whether new addresses impose a new bias to the *IPv6 Hitlist*, we analyzed their AS distribution. Figure 5.4 shows the distribution of responsive addresses across ASes for each new input while the total number and most common ASes are listed in Table 5.3. 6Graph and 6Tree, contributing most new addresses, both show a visible bias towards Free SAS covering up to 52% of the respective candidates. The second most prominent AS, DigitalOcean, is only at around 5% to 8%. We verified the correct classification of these addresses as non-aliased and came to the same conclusion as the automatic detection. While many addresses were responsive, we can identify unresponsive addresses. The existing *IPv6 Hitlist* already contains 149.8k responsive addresses from Free SAS on April 7, 2022.

Other sources provide different AS distributions and top hitters, *e.g.*, many previously unresponsive addresses were from VNPT. The distance clustering approach and new passive sources show the most even distribution. The latter even covers 2.9k ASes with only 21k responsive addresses.

Key take-away: *90% of passively extracted addresses from existing data sets were already contained in the* IPv6 Hitlist *or unique addresses which were however aliased. In contrast,*

Table 5.4: Amount of candidate (cand.) and responsive (resp.) addresses generated by the algorithms when using different categories as well as the full hitlist as seed data set.

	6Forest		6GAN		6GCVAE		6Graph		6Hit	
	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.
Content	2M	174k	487k	13k	3M	14k	35M	443k	10M	231k
ISP	3M	2M	410k	55k	845k	179k	25M	3M	8M	3M
NSP	2M	128k	521k	4k	3M	15k	31M	527k	10M	552k
Educational	1M	19k	316k	3k	700k	585	2M	22k	24M	100k
Non-Profit	711k	39k	125k	9k	284k	3k	296k	15k	20M	3M
Full	2M	494k	486k	41k	2M	111k	106M	5M	18M	3M
	6Scan		6Tree		6VecLM		DET		Entropy	
	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.
Content	9M	491k	11M	417k	78k	4k	9M	361k	6M	8k
ISP	8M	4M	11M	3M	18k	2k	8M	3M	6M	1M
NSP	9M	884k	9M	1M	66k	6k	2M	382k	6M	16k
Educational	10M	38k	11M	107k	84k	1k	1M	745	4M	3k
Non-Profit	10M	946k	8M	2M	0	0	6M	356k	4M	14k
Full	6M	2M	35M	5M	49k	4k	8M	1M	6M	59k

address generation approaches were able to identify new, previously unknown but responsive addresses. While their individual hit rate was below 10%, combining all responsive addresses from passive sources and address generation can more than double the number of responsive addresses reported by the IPv6 Hitlist.

Often, a major difficulty with these techniques is their incomplete documentation. Hence, we were not able to reproduce results of 6GAN [52] but it only generated 4k responsive addresses.

5.4 Second Discovery of New Addresses

Our second improvement of the *IPv6 Hitlist* was done during the work published by Steger *et al.* [13]. The following explains in more detail how TGAs were used during this study and their value to the *IPv6 Hitlist*.

Given the diverse composition of the *IPv6 Hitlist* with respect to covered network categories and the differences in responsiveness and IP stability for these categories, we set out to analyze the properties of TGAs. We applied all algorithms listed in Table 5.1 to the full list of responsive addresses from March 3, 2023. During this second study, we further divided the input into filtered versions, *i.e.*, the addresses inside the active hitlist filtered based on the PeeringDB categories *Content*, *ISP*, *NSP*, *Non-Profit*, and *Educational*. While PeeringDB has more network categories, we excluded all categories with less than 5% representation in the hitlist, additionally including the two categories *Educational* and *Non-Profit* to test the algorithms on smaller seed sets. For more information about the categorization of addresses within the *IPv6 Hitlist* and different behavior within each category, we refer to the work by Steger *et al.* [13].

Generation Rate and Candidate Set Size: Table 5.4 shows that the various algorithms generate vastly different numbers of candidate addresses. Moreover, the generated addresses are

also highly dependent on the used seed set. Therefore, we compare the *generation rate* of the algorithms, *i.e.*, the size of the candidate set relative to the seed set size. Algorithms like 6VecLM and 6GAN have relatively low generation rates, which is due to the fact that these algorithms limit the amount of processed input to a hard coded value (see Section 5.2). Algorithms such as 6Graph can generate more than 100M candidate addresses, which is 1638% of the size of the seed sets. With the Non-Profit seed set, 6VecLM is unable to generate a candidate set, as the seed set is smaller than the predefined input size, which we could not successfully modify. If the scanning budget of a measurement is critical, large candidate sets should either be sampled or another algorithm should be chosen. If large candidate sizes are desired, large inputs or algorithms with high generation rates are best suited.

Ratio of Fully Responsive Prefixes: Fully responsive prefixes (see Section 4.4 for more details), are excluded in our scans as they do not add any valuable information, but instead introduce a bias to the results. It is therefore an important measure of quality for a candidate set to contain few addresses from fully responsive prefixes. As described in Section 5.2.3, we filtered all fully responsive prefixes from the *IPv6 Hitlist* service and conducted a new detection for previously unknown addresses. We compared the unfiltered and filtered sets and found that most algorithms have a negligible rate of addresses from fully responsive prefixes, thereby not impacting the algorithms' candidate set quality when filtered. Two exceptions are 6GCVAE and Entropy/IP, which generated up to almost 50% addresses from fully responsive prefixes for some categorized input as well as the full input. This decreases the usable size of their candidate sets substantially, which should be kept in mind before scanning. The exact rate of fully responsive prefixes can be found in the work by Steger *et al.* [13]. Although the rate of fully responsive prefixes in most candidate sets were relatively low, which means that only little unnecessary scanning overhead would be introduced, we still stress the need for Aliased Prefix Detection (APD).

Response Rates: Internet measurement studies are not only dependent on a scanning budget, but also strive to avoid unnecessary probes which are unlikely to trigger responses. Therefore, it is important to analyze the response rate, *i.e.*, the portion of addresses which responds to at least one protocol, for the different candidate sets. As can be seen in Table 5.4, a larger candidate set does not lead to a higher response rate. Instead response rates are more strongly linked with the input set category as well as the difference between dynamic and static algorithms. Dynamic algorithms, due to their ability to adapt their generation strategy based on the results of their scans, have among the top response rates for all categories, up to 45% for some. On the other hand, static algorithms rarely show response rates over 15%, with 6Forest being one of the few exceptions. Using ISP addresses as input shows the best response rates for almost all algorithms, even better than with uncategorized input. Candidate addresses generated from educational networks, on the contrary, have the lowest response rate at hardly over 1% for any algorithm. Again, for measurements with limited scanning budget, choice of input is shown to be critical for the efficiency of the TGAs and subsequently the scans.

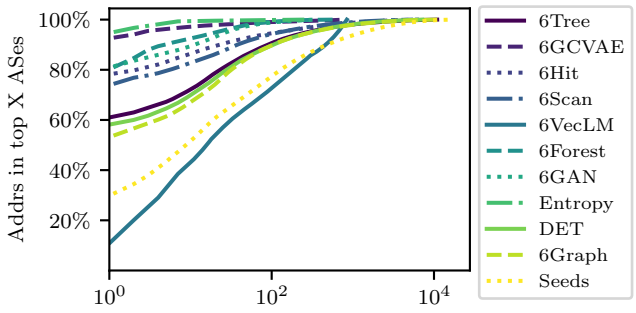

Figure 5.5: Cumulative AS distribution of the responsive candidate sets generated by the algorithms using the full hitlist as input. Note the logarithmic x-axis.

AS Origin Distributions: The cumulative AS distribution of the candidate sets generated from the full hitlist are shown in Figure 5.5. Most candidate sets generated by the TGAs are more biased towards single ASes. The majority of TGAs contain 50–95% addresses from a single AS, whereas the top ten ASes of the seed set cover only around 50% of their addresses. The most popular AS for all but one candidate set is Free SAS (AS12322, an ISP network from France). The only exception to this bias is the candidate set of 6VecLM, which contains only five addresses from Free SAS and is even more evenly distributed among ASes than the seed set dataset. While Free SAS is also the AS with the highest share in the seed data set, it covers only 30% of it. Looking at the structure of the addresses from this AS responding to ICMP, it is visible that over 99% of them have the host part set to `::1`. They are all within the same /39 prefix and only differ in the 10 to 15 nibble of the address. This a very clear structure which has easy to detect patterns, ideal for discovery by TGAs. Addresses from this AS were first added to the hitlist via CT logs and the Bitnodes dataset [8] and their share drastically increased with the first phase of extensions in 2022 (see Section 5.3). The high percentage of addresses from Free SAS in most candidate sets explains their bias towards the ISP network category. This further stresses the need to filter certain categories for use cases where addresses from the respective categories should not be targeted. Furthermore, addresses from specific ASes should be filtered, as their inclusion in the seed set can introduce biases towards those networks far beyond their presence in the seed set.

Even when combined, the candidate sets of all algorithms only cover 75% of the ASes in the seed set. While the combined candidate sets include 684 ASes which have not been covered by the seed set, 4875 ASes from the seed set are not included. It is therefore imperative to use additional address sources in order to achieve a higher AS coverage if measurements should cover a representative subset of the IPv6 Internet. It also raises the question if current TGAs adequately address the need for a balanced candidate set across many ASes.

Protocol Responses: Depending on the use case for the generated addresses, it can be crucial to discover targets with a high response rate to a certain protocol. Figure 5.6 shows the response rate to the different protocols per candidate set. All candidate sets have the highest response rate to ICMP probes, which is to be expected due to the prevalence in the seed set. Moreover,

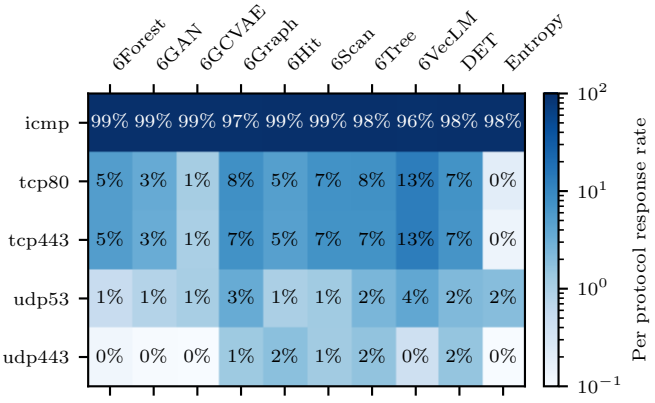

Figure 5.6: Response rates to the different protocols per algorithm generated on full hitlist input. Note the color map log scale.

unlike in IPv4, ICMP in IPv6 can not simply be fully blocked due to its important functionality in stateless address autoconfiguration [156]. Responses to other protocols are much less frequent for all candidate sets. The response rate for HTTP and HTTPS is very similar to the share of non-ISP addresses in the responsive portion of the candidate sets. The responses to the candidate set of 6VecLM have the lowest share of ISP addresses and the highest number of responses to HTTP and HTTPS. Entropy/IP and 6GCVAE on the other hand, have more than 95% ISP addresses in their respective responses and the lowest share of protocol responses other than ICMP. With CDN input addresses, all candidate sets receive between 30% and 65% HTTP and HTTPS responses, whereas with ISP input, no candidate set generates more than 3% response rate for any protocol besides ICMP. This shows that input and algorithm should be chosen carefully depending on the desired protocol responses for the use case.

Key take-away: *In general, given the full* IPv6 Hitlist *as input but also subsets, most TGAs are biased towards ISPs. However, in detail, the collected TGAs perform differently and created address sets have different compositions.*

5.5 Discussion and Summary

As shown in this chapter, updating the *IPv6 Hitlist* with new input sources can be valuable to improve its quality and increase the amount of identified responsive addresses. We collected and successfully applied a set of different target generation methods. While we were not able to reproduce most hit rates published with target generation mechanisms, we can identify new responsive targets, more than doubling the number of responsive addresses in multiple steps. However, a comprehensive and public evaluation is only possible if generated candidates and results are shared. This would further improve the impact of these methodologies in general and on the *IPv6 Hitlist*. We demonstrated that the input has a strong influence on various metrics, such as the number of generated and responsive addresses, protocol responses, and addresses origin. All but one candidate sets generated from uncategorized input show a very strong bias

towards ISP networks, which in turn have a strong bias towards single ASes and generally have a response rate below 10% for any protocol other than ICMP.

While target generation mechanisms sometimes bias towards certain providers, such as Free SAS, generated targets still cover a variety of ASes and show a similar distribution across protocols as the current *IPv6 Hitlist*. Nevertheless, solely relying on those generated candidates shifts the focus towards different providers and deployments and changes the view on the IPv6 ecosystem. In our opinion all sources (existing passive sources, frequent traceroutes, generated candidates) provide individual value, and their contribution to the *IPv6 Hitlist* offers a valuable foundation.

We incorporated new sources into the ongoing *IPv6 Hitlist* service, and plan to update the service with new sources on a regular basis. Future IPv6 Internet measurements are encouraged to use our findings to increase the efficiency of their scans by removing unnecessary scanning overhead and generating targets better suited for their use case. Researchers conducting IPv6 measurements should keep in mind that the current hitlist shows a bias towards ISP addresses. These addresses are only short-lived and should therefore not be used for long-term studies. A proper selection of scan specific targets from the hitlist and a proper application of TGAs on specific seed sets can however improve future scans and reduce unnecessary probing.

Part II
DNS Measurements: A General Scan Foundation

The second part of this thesis focuses on the second research goal (see Section 1.1): *Evaluation of the Impact of DNS on Internet-wide Scans.* Domain resolutions are used to analyze the general state of the Internet ecosystem, its centralization and usage. Furthermore, DNS is often used to identify targets for higher layer protocol scans, *e.g.*, focusing on top lists or using full zone files. Therefore, we take a closer look at DNS as an important input source for Internet measurement studies and focus on two properties of DNS data.

Different domain lists, *e.g.*, top lists or zone files, are the foundation for a variety of research and used throughout this work *e.g.*, in scans or QUIC with SNIs. The author of this work has been key to maintaining and improving an ongoing measurement setup that resolves and analyzes large parts of the DNS ecosystem on a daily basis as described in Chapter 2.

Overview of the second part.

	Part II: DNS Measurements: A General Scan Foundation	
G2	Potential of new Domain Name System Resource Records	Chapter 6
	Influence of domain parking	Chapter 7

In Chapter 6, we analyze two new resource records, namely `SVCB` and `HTTPS` focusing on their general deployment and involved users and operators. These records are specifically designed to provide service information for specific domains already within the DNS ecosystem [4]. We evaluated their deployment status and contained information and used them as valuable assets to analyze protocol deployments in Chapter 8.

While DNS is valuable, domains within the ecosystem are not equally interesting to users or relevant for studies. Top Lists are often used to focus on a small subset, but advances in Internet-wide measurements and their scalability make it easier to resolve a large set of domains and use results for evaluations or followup scans. Examples are our scans as explained in Chapter 2 or the OpenINTEL project [71]. One category of domains with limited value to a majority of users are parked domains. Domain parking typically involves leveraging advertisements to generate

revenue on otherwise inactive domain names. In Chapter 7 we quantify the impact of domain parking on DNS data and its effect on DNS based studies, *e.g.*, related to Internet consolidation.

Besides the work presented in this thesis, collected insights into the DNS ecosystem have also influenced the proposed resolution approach by Naab *et al.* [21] and collected data has been used in different studies, *e.g.*, [14], [17]–[19], [22], [28].

Chapter 6

SVCB and HTTPS: New DNS Resource Records

Even though DNS was initially specified multiple decades ago, the IETF specifies new extensions to the protocol. The IETF has standardized new DNS resource records, namely `SVCB` and `HTTPS`. Both records inform clients about endpoint and service properties such as supported application layer protocols, IP address hints or ECH information. Therefore, they allow clients to reduce required DNS queries and potential retries during connection establishment and thus help to improve the quality of experience and privacy of the client. The latter is achieved by reducing visible meta-data, which is further improved with encrypted DNS and ECH. This extension has the potential to drastically improve Internet-wide scans, reduce scan effort and invalid connection requests.

The standardization was finalized in 2023 and companies announced support, *e.g.*, Cloudflare and Apple. Therefore, we provided the first large-scale overview of actual record deployment by analyzing more than 400M domains. We found 3.96k `SVCB` and 10.5M `HTTPS` records. As of March 2023, Cloudflare hosted and served most domains, and most records only contained Application-Layer Protocol Negotiation (ALPN) and IP address hints. Besides Cloudflare, we saw adoption by a variety of authoritative name servers and hosting providers indicating increased adoption in the near future. Lastly, we can verify the correctness of records for more than 93% of domains based on three application layer scans.

This chapter is based on the following publication:

> J. Zirngibl, P. Sattler, and G. Carle, "A First Look at SVCB and HTTPS DNS Resource Records in the Wild," in *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2023. DOI: 10.1109/EuroSPW59978.2023.00058 [12]

Author's Contributions: *The author of this work led the research, developed the used scan methodology, collected used data and framed the original research question. He laid out the methodology and led the analysis and synthesis into the publication.*

6.1 Motivation

With the ongoing development of the Internet, available protocols and versions, a general requirement is getting more important, namely *information about supported application layer protocols, versions and properties by individual endpoints*. The latter information can be exchanged during a handshake or first communication (*e.g.*, ALT-SVC Headers in HTTP). However, missing knowledge increases the handshake duration and information from existing solutions can only be used in subsequent connections. Each connection attempt and the potential use of insecure protocols reveals further meta-data related to a client and its desired connection, thus impacting its privacy and security.

To circumvent this problem, the IETF works on a new general DNS RR named `SVCB` ("SerViCe Binding) that provides service bindings for a domain [7]. This record accomplishes two major goals, directing a client *(i)* to another alias or *(ii)* to an endpoint including service information. As a first subtype, the `HTTPS` DNS RR was specified with a focus on HTTPS endpoints. The records allow a client to receive all required information, namely supported protocols, used ports and IP addresses, using a *single*, recursive DNS query. Provided information can be used to directly establish a secure communication channel using a protocol both endpoints support. Information about available application protocols and their explicit version can also reduce the risk of on-path or downgrade attacks, *e.g.*, make HTTP Strict Transport Security (HSTS) obsolete. Furthermore, the new `HTTPS` record is supposed to be extended to provide ECH information to the client in the future. Once specified and deployed, ECH [157] further reduces the visibility of connection-related meta-data, *e.g.*, the SNI.

Quick and widespread deployment of these new records can drastically improve the privacy of clients on the Internet. Different operators including Cloudflare [158] and Akamai [159] but also client software, *e.g.*, Apple iOS [160] and Google Chromium [161] have already announced support for the new records.

Therefore, we set out to evaluate actual deployments and availability of the new records based on a large-scale measurement. Our contributions in this chapter are:

(i) We evaluate the support of new records for more than 400M domains. We show that the deployment is mostly driven by Cloudflare. However, other operators show initial deployment.

(ii) We evaluate the properties of received records and their implication for a client and established connections. We show that most domains have records with service information, mainly ALPN values and *ipv4-* and *ipv6hints*. Further parameters are rarely visible.

(iii) We verify the correctness of received information with application layer scans. We were able to connect to 96% of targets extracted from `HTTPS` records.

6.2 Background

Listing 6.1: Example SVCB and HTTPS DNS RR

```
1 coffebike.no.            3600    IN  SVCB        0 barmobile.no.

2 _dns.dnshome.de.         300             IN      SVCB    2 dns.dnshome.de.
      ↪ alpn="dot" port=853 ipv4hint=45.86.125.58,45.86.125.59
      ↪ ipv6hint=2a0c:8901::53,2a0c:8902::53

3 cloudflare.com.          30              IN      HTTPS   1 .
      ↪ alpn="h3,h3-29,h2" ipv4hint=104.16.132.229,104.16.133.229
      ↪ ipv6hint=2606:4700::6810:84e5,2606:4700::6810:85e5
```

The SVCB DNS RR represents a more general record to be used with different service types, while the HTTPS DNS RR is specifically designed to be used with HTTPS. These DNS RRs allow clients to select the correct service properties directly. To indicate the desired service, domains for SVCB records should be prefixed with Attrleaf labels [162] (*e.g.*, *_dns*). Using HTTPS records implies HTTP as service. Records can contain for example ALPN values (*e.g.*, HTTP/2 or HTTP/3), in combination with correct target names, ports and IP addresses. The IETF designed both records to be flexible and expandable. Listing 6.1 shows three example records. The first record (SVCB) is in alias mode, indicated by the priority of *0*, and redirects the domain to another target name. In comparison to canonical name (CNAME) records, this is also possible at the apex of a zone [7].

The second (SVCB) and third (HTTPS) records are in service mode and provide further information about the respective endpoint. In service mode, a target name can be set to indicate another name (*e.g.*, record 2). The target name is "." if the actual domain should be used (*e.g.*, record 3). Additional record data is organized as key-value data, so-called *SvcParams*. Each parameter has to have a specified format to allow interoperability. As of March 2023, the draft specified six different parameter keys and their value format. These values have all been adapted in the published RFC [7]. By default, an HTTPS record indicates HTTP/1.1 support. The *alpn* parameter can indicate additional protocols. If an endpoint does not support HTTP/1.1 but other ALPNs the *no-default-alpn* parameter has to be added. The *port* parameter allows indicating alternative ports, while *ipv4-* and *ipv6hint* allow informing about IP addresses. Finally, the *mandatory* parameter can be used to indicate a set of parameters that must be used for the service to function correctly.

The initially drafted but now reserved *ech* parameter relies on a different draft [157]. However, it lacks deployment (see Section 6.4) and its final publication is delayed as of March 2024. Therefore, after a discussion [163], the parameter and references were removed from the SVCB and HTTPS draft to allow an RFC publication. We evaluated the presence of this parameter in Section 6.4.

For SVCB records prefixed with *_dns*, the RFC additionally adds the *dohpath* parameter that allows to specify a Uniform Resource Identifier (URI) template for DNS over HTTPS [164]. This study was initially done in March 2023 and the respective RFC was still a draft but contained the same parameter [165].

6.3 Data Collection

This work relies on our DNS scans (see Chapter 2). To verify the usefulness of collected records, additional HTTP scans were conducted. This section provides relevant information about specifically used scans. The majority of this study is based on scans conducted between February 22 and March 9, 2023 as published in [12]. Additional information about the development of records (see Section 6.4.4) is based on scans between January 10 and 15, 2024.

DNS Scans: We used `SVCB` and `HTTPS`, but also A and NS resolutions of more than 400M domains. We further resolved the name server domains from the latter NS records to their respective A records. This allows us to analyze who serves the new records and which operators are involved. We used the following sources as input for this study:

- Names on the Majestic [36], Alexa[1] [129], and Umbrella [37] Top 1M lists;
- More than 1k available zone files from the CZDS, *e.g.*, *.com*, *.net* and *.org*;
- A static collection of 98M domains from 52 country-code TLDs;
- *www.* domains extracted from Certificate Transparency (CT) logs.

We additionally prefixed domains with the Attrleaf label *_dns* [162]. As of March 2023, it was the only available label based on an IETF draft [165]. As of January 2024, no additional label was added to the registry. We exclude www. domains for this measurement but included domains from NS record names.

Protocol Scans: For each domain with an `HTTPS` record in service mode, we extracted the supported ALPNs, port and IP addresses from the *ipv4hint* in the records. If no *ipv4hint* was available, we relied on each domain's additionally requested A records. We used these tuples of domain, IP address, port, and ALPN to seed our protocol scans. We used the QScanner (see Chapter 8) and the Goscanner [30] to test whether received ALPN information is valid for the given domain. These scans were only conducted during the initial study because no major development of these records was visible afterwards. Thus, no major changes were expected.

6.4 Analysis

We analyzed the deployment of `SVCB` and `HTTPS` records as of February 2023 based on our measurements described in Section 6.3. Resolving more than 400M domains, we received `SVCB` records for 3.96k domains but `HTTPS` records for 10.56M domains. `SVCB` should be available for domains with Attrleaf labels [162]. Therefore, we additionally resolved domains prefixed with the first specified label (*_dns*) but only received records for 27 domains.

[1] We use the last published list before deprecation from February 1, 2023. `https://toplists.net.in.tum.de/archive/alexa/`

Table 6.1: Number of domains with `SVCB` and `HTTPS` DNS resource records as of March 2023.

		Mode	
Record	Total	Alias	Service
SVCB	3.96k	3.9k	62
HTTPS	10.56M	2.6k	10.55M
SVCB + *_dns*	27	0	27

Table 6.2: Number of domains with each property and parameter in their `SVCB` and `HTTPS` DNS resource records. The `Mandatory` and `No Default` key were never used.

		Keys					
Record	Total	ALPN	Port	ECH	IPv4 Hint	IPv6 Hint	DoH Path
SVCB	3.96k	53	2	0	25	15	-
HTTPS	10.56M	10.55M	13	20	10.55M	10.23M	-
SVCB + *_dns*	27	26	12	0	1	1	1

6.4.1 General Record Analysis

Table 6.1 shows which modes (alias vs service) were used and Table 6.2 shows which keys were commonly present in available records. Regarding `SVCB` records, 3.9k (98.4%) domains used the record for alias mode, aliasing the service to a different domain. Only 62 domains used the service mode and mostly advertised ALPN values or IPv4 and IPv6 addresses as hints. 27 domains prefixed with *_dns* resulted in `SVCB` records. All records were in service mode, advertising different ALPN values (4× *h2* for DNS over HTTPS and 26× *dot* for DNS over TLS). The DoH path advertised by a single domain was `/dns-query?dns`. The `SVCB` record in both scenarios was only deployed by a few domains, and we focus on `HTTPS` records for the remainder of this chapter.

Regarding `HTTPS` records, only 2.6k (0.02%) domains used the alias mode, while a majority advertised endpoint information using the service mode. Similarly, most domains advertised ALPN values and IPv4 and IPv6 addresses as hints. The `HTTPS` record implies support of HTTP/1.1 by default if the *no-default-alpn* parameter is not present. In our results, no domain with an `HTTPS` record in service mode had the flag set. Table 6.3 shows the Top-10 advertised ALPN parameters. A majority of domains advertised HTTP version 2 but also 3 indicating

Table 6.3: Top 10 advertised ALPN sets/values in `HTTPS` DNS resource records. Note that `HTTPS` records imply the support of HTTP/1.1 by default [7].

ALPN set	Domains	ALPN values	Domains
h3,h3-29,h2	9.7M	h2	10.6M
h2	834.4k	h3	9.7M
	3.2k	h3-29	9.7M
h3,h3-29	866	http/1.1	15
h2,h3	242	h2c	10
h3,h2	123	h1	7
h2,h2c	10	h3-32	1
h3	9	h3-31	1
http/1.1,h2	4	h3-30	1
h1,h2	4	h3-28	1

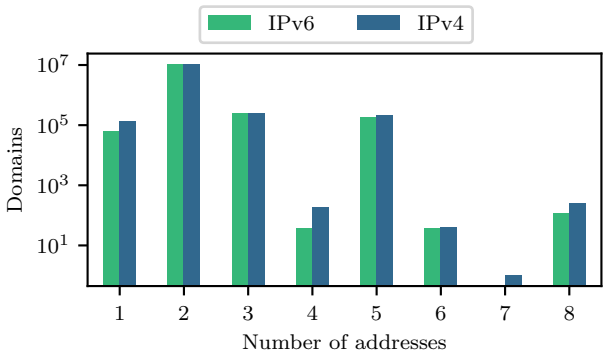

Figure 6.1: Addresses in *ipv4*- and *ipv6hints*. Note the logarithmic y-axis.

QUIC support, while 834.4k only advertise HTTP/2. 3.2k domains did not advertise additional ALPN values but only relied on the default. A client can still use record information and only establish a connection if it supports HTTP/1.1.

While for 10.55M (99.9%) domains IPv4 hints were available, 10.23M (96.9%) additionally advertised IPv6 addresses. Most hints contained two addresses respectively but up to eight different addresses were visible as shown in Figure 6.1. This allows a client to select from a set of different addresses and fallback to alternatives if necessary. All other keys were only visible with a few domains. The advertised ports in `HTTPS` records were 80 (2×), 443 (10×) and 8920 (1×). Furthermore, we only received 20 ECH configurations. This supports the discussion that the respective ECH draft [157] still lacked deployment while the DNS RRs were already deployed for many domains and both drafts should be decoupled [163]. 146.5k domains from the Alexa [129], 169k from Majestic [36] and 80.8k domains from the Umbrella [37] Top 1M lists had an `HTTPS` record. The most prominent candidates were `google.com` with a service mode record and an ALPN parameter *h2,h3* and `youtube.com` with a service mode record without additional data.

Key take-away: *The `SVCB` records in both scenarios were only deployed by a few domains. In contrast, more than 10M domains made use of `HTTPS` records, mostly serving address hints and ALPN values. The alias mode or remaining parameters were rarely used and should be reevaluated in the future.*

6.4.2 Involved Operators

For the following analysis, we focused on domains with `HTTPS` records in service mode (10.55M) due to their advanced deployment. To get a better understanding of involved operators, we analyzed where domains are hosted and which name servers are used. If available, we used *ipv4hints* and mapped addresses to the AS announcing the respective prefix. For all domains without this parameter, we used queried A records for IPv4 addresses.

Domains with `HTTPS` records were hosted in 2.3k ASes. However, Table 6.4 shows that a majority of domains (98.8%) resolved to ASes operated by Cloudflare (AS13335 and AS209242). Domen-

Table 6.4: Top 5 web hosters (out of 2.3k) and name server providers (out of 661) of domains with `HTTPS` records. Domenshop uses three different name servers for most domains located in three different ASes (AS1921, AS12996, AS208045)

Hosting			Name server		
ASN	Name	#Doms	ASN	Name	#Doms
13335	Cloudflare	10.4M	13335	Cloudflare	7.7M
12996	Domenshop	61.6k	12996	Domenshop	24.0k
209242	Cloudflare	49.7k	16509	Amazon	3.2k
397273	Render	4.9k	397226	Neustar	3.1k
14061	DigitalOcean	4.6k	44273	GoDaddy	2.5k

shop, a Norwegian web hoster, hosted the second-highest number of domains and accounted for a large share of domains indicating support for HTTP/2 but not HTTP/3. Following the Top 3 a more even distribution of the remaining 72k domains across 2.3k ASes was visible.

To analyze responsible name servers, we relied on NS records for domains exactly matching domains in our input. We did not follow CNAME records or extracted information from SOA records. During our scan, we received NS records for 7.8M domains with an `HTTPS` record. Domains without NS records in our data were either resulting in SOAs only (mostly *www.* domains) or resolved to canonical names and would require further resolution steps. In general, we were able to identify name servers supporting `HTTPS` records hosted in 661 different ASes. This shows a widespread deployment of name servers that support the new record in general.

Similar to web hosting, most `HTTPS` records were served by name servers hosted within Cloudflare followed by Domenshop. The latter appeared as three different ASes (AS1921, AS12996, AS208045). Each AS hosted a name server authoritative for a similar amount of domains respectively. Most domains had one NS record for each of the three name servers for resilience.

Key take-away: *Domains with `HTTPS` records were hosted in more than 2.3k ASes and name servers serving the records were in more than 1.6k ASes. However, most records were hosted in and served by Cloudflare (98%).*

6.4.3 Validity of Records

We conducted HTTP scans to check the validity of collected records and whether clients can use the received information for an HTTP request. The general scan approach is described in Section 6.3. We focused on HTTP/1.1, HTTP/2 and HTTP/3, and selected targets for each scan based on the ALPN and IP address hints. Table 6.5 provides an overview about results. TLS/TCP handshakes were successful for more than 96.6% of evaluated targets for each HTTP version respectively while QUIC handshakes were successful for more than 93.6% of HTTP/3 targets. For 90%, we were further able to conduct an HTTP HEAD request. Most unsuccessful connection attempts either resulted in a timeout (1.1: 6.4k, 2: 9.1k, 3: 50.8k) or a generic TLS handshake failure (1.1: 708.9k, 2: 692.1k. 3: 1.2M).

Successful scans for HTTP/1.1 and HTTP/2 still covered 1.8k ASes while HTTP/3 and thus QUIC scans only covered 416 ASes out of 2.3k candidates. Analyzing failed scans reveals that a major origin of errors during QUIC scans and for timeouts during the HTTP request was an

Table 6.5: Protocol scan results based on `HTTPS` records. Targets are a combination of domain and IP address.

		Successful			
HTTP	Targets	TLS Handshake		HTTP Requests	
1.1	21.44M	20.72M	96.63%	19.48M	90.84%
2	21.43M	20.73M	96.69%	19.47M	90.84%
3	19.59M	18.34M	93.64%	17.04M	87.01%

attack prevention mechanism by Cloudflare [166]. It is an automated challenge mechanism that delays the page load which results in errors with both the Goscanner and QScanner.

Furthermore, we found 23.0k domains with `HTTPS` records served by Cloudflare name servers but hosted in different ASes that only resulted in timeouts at least during QUIC scans. For those domains, scan results (timeouts) were reproducible. Interestingly, those domains were hosted in more than 1.3k ASes and no relation was visible besides the Cloudflare name servers. Furthermore, all `HTTPS` records contain the same ALPN set (*h3, h3-29, h2*). We assume a misconfiguration and informed Cloudflare.

Key take-away: *A majority of available `HTTPS` records contains valid, usable information especially if used by clients able to pass Cloudflare's attack prevention. However, we identified a set of records with incorrect ALPN values. For those domains requests for some announced ALPN values time out consistently (mostly HTTP/3).*

6.4.4 Development after the Initial Study

We followed the development of the records after the initial study at the end of February 2023 based on weekly scans of the same input sources. However, no major changes can be seen. As of January 2024, 11.89M domains have an `HTTPS` DNS RR, an increase by 12.6%. Most records still only inform about *alpn* values, *ipv4-* and *ipv6hints*. A single domain indicates mandatory keys, namely *ipv4-* and *ipv6hints* with the respective keys. Similarly, a slight increase in `SVCB` is visible.

Interestingly, in September 2023, Cloudflare activated ECH for all free zones by default [167]. This was also visible within `HTTPS` DNS RRs because Cloudflare is the main organization deploying those records (see Table 6.4). Therefore, more than 9M records immediately contained an *ech* configuration to be used. However, due to issues with the implementation, Cloudflare deactivated ECH [168] and did not re-activate it until the end of this analysis in January 2024.

6.5 Discussion and Summary

In this chapter, we provided the first large-scale overview of the deployment of new `SVCB` and `HTTPS` DNS resource records. While we found only very few domains with `SVCB` records (3.96k without and 26 with an Attrleaf label), we show that more than 10M domains already resolved to `HTTPS` records. These records mainly provided ALPN values and *ipv4-* and *ipv6hints*. We found only 20 domains with an ECH parameter which indicates lacking deployment. However, we

show that most domains were hosted within Cloudflare, and Cloudflare operated name servers were authoritative. This remained relatively stable after the publication of the final RFC [7].

Nevertheless, information contained in most available records was correct, and handshakes followed by HTTP requests with indicated versions were possible. Therefore, clients already querying the records (*e.g.*, Apple devices [160]) can effectively make use of `HTTPS` records for more than 10M domains and reduce DNS requests and visible meta-data during connection establishments while reducing handshake cost.

Chapter 7

The Influence of Domain Parking on DNS Data

Domains are an important input for many studies and Internet-wide measurements, *e.g.*, to identify interesting targets for the IPv6 Hitlist (see Part I) or as Server Name Indication during TLS and QUIC scans (see Part III). However, besides a small set of domains from top lists, domains are often treated as a set of equally relevant elements without further consideration of their actual use.

While many domains are used for relevant services, *e.g.*, websites, domains can also be parked. Domain parking typically involves leveraging advertisements to generate revenue on otherwise inactive domain names. Their content is rarely of real value to users and tends to be highly similar across parked domains. They have commonalities beyond content alone: parked domains can share hosting and DNS infrastructure. Parking rarely receives special treatment in existing studies (*e.g.*, content analyses or infrastructure concentration studies). While the presence and possible bias introduced by parked pages is sometimes acknowledged in studies, the studies still treat parked domains as any other, either because differentiation is infeasible, or because doing so is considered out-of-scope.

We argue that the impact of parked domains on analyses regarding the current state and future development of the Internet should not be overlooked. In this chapter, we motivate this argument through quantification, and take steps towards helping other researchers identify parked domains.

We systematically collected a list of 82 parking services and developed DNS-based indicators to help identify parked domains. We next quantified the presence of parked domains, using large-scale DNS data containing hundreds of millions of registered domain names, representative for a significant part of the global DNS namespace. Overall, we pinpointed 60M parked domains, which is a significant percentage of all names under consideration (23%) and identified up to 4% of domains from top lists to be parked. These findings demonstrate that the effect of parked

pages is potentially pronounced. We also break down into the various parking services and DNS zones. This helps us demonstrate and further discuss the effect that domain parking can have on research and Internet consolidation.

This chapter provides insights into this ecosystem, quantifies parked domains and discusses the impact of parking on relevant studies. It is based on the following publication:

> J. Zirngibl, S. Deusch, P. Sattler, J. Aulbach, G. Carle, and M. Jonker, "Domain Parking: Largely Present, Rarely Considered!" In *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2022 [10]

The publication builds in parts on Steffen Deusch's Bachelor's thesis [169].

Author's Contributions: *The author of this work led the research by formulating the original research question, specifying the methodology and by leading the analysis and synthesis into the publication. Furthermore, the author is mainly involved in the collection and aggregation of relevant data (see Chapter 2).*

The collection of parking services and specific indicators was mainly done by Steffen Deusch based on the methodology proposed by the author of this work and his guidance.

7.1 Motivation

The Internet is becoming increasingly centralized. Over the past years, the development towards centralization and consolidation has emerged as an important subject of discussion among research and network operator communities.

The upsurge of CDNs has added to Internet centralization. A number of recent studies evaluate properties of CDNs and consolidation — related in particular to the web and DNS ecosystems — and offer a basis on which to discuss and evaluate its impacts [93], [109], [170]–[173]. To evaluate trends and possible impacts, studies often rely on domain names and an analysis of the infrastructure with which names are associated. In such studies, domain names are usually treated the same, even though not all names are equal. In particular, domains can be parked. Domain parking is a concept to generate revenue from registered but otherwise unused domains, for example via advertisements. So-called domain parking providers offer the means for such monetization.

Parking pages differ from user-centric web content. Their content is of little use and importance to users, yet similar across parked domains. Hosting and DNS infrastructure can also be in common. As a result, parked domains can introduce *bias* in centralization studies. While some studies do mention this limitation, differentiating parked domains is then either considered infeasible or out of scope.

We argue that parked domains require consideration when evaluating and discussing Internet consolidation. To this end, we quantify the prevalence of parked domains at scale across multiple TLDs but also top lists. We further evaluate the impact of domain parking on DNS and hosting

providers, *e.g.*, CDNs, and reason about effects of lack of consideration in related studies. To the best of our knowledge, we are the first to offer this perspective. This work does not evaluate the monetization schemes of domain parking services, individually hosted domains or potential wrongdoing (*e.g.*, typo-squatting) and vulnerabilities.

Our contributions are:

(i) We systematically collected 82 parking, marketplace and placeholder services. We analyzed their modus operandi and developed DNS-based indicators that enable identifying parked domains.

(ii) We study the prevalence of domain parking using multiple sources of large-scale DNS data. These sources are representative for a sizable part of the global DNS namespace, containing hundreds of millions of registered domain names from well over 1k TLDs, including ccTLDs, legacy gTLDs such as `.com`, and newer gTLDs such as `.tokyo`;

(iii) With our new-found insights into parked domain name prevalence, we discuss findings from existing research regarding the development and consolidation of the Internet and of CDNs.

7.2 Domain Parking Services

Monetizing unused domains through advertisements or sales is referred to as domain parking. This concept has been generalized and professionalized by dedicated domain parking services that administer and monetize registered but unused domains. These services host websites on parked domains including advertisements and sales banners, manage visibility and cash flows. Domain owners can register their domains with these services and park their assets using DNS in one of several ways. First, domains can be parked by delegating authority for a domain name to the name servers of a parking service. Under this approach the name server delegation (*i.e.*, NS records) will point to the service-specific name servers. Second, domain name owners can use their own name servers but configure an IP address record (*i.e.*, A or AAAA) or canonical name record (CNAME) and point it to the infrastructure of the parking service. Both approaches can be inferred from DNS data, for example by actively querying for records.

We systematically collected a list of 82 parking services and established if they require delegation or involve canonical names or IP address records. These DNS-based indicators are input to our methodology to identify parked domains similarly to Vissers *et al.* [86]. Our list contains providers of varying size. We started our collection with a web search for prominent parking services and published configurations, *e.g.*, GoDaddy (Free Parking) [174]. Based on these insights, we searched for indicators of parking in domains of our data, *e.g.*, ParkingCrew uses the name servers `nsX.parkingcrew.net`. Lastly, we rely on our DNS data to analyze frequently used name servers and IP addresses to identify the most impactful services. This follows our assumption that parking services rely on a few name servers or IP addresses for many domains. We selected a random set of parked domains for each service and used visual confirmation to verify parking services (cf Section 7.4.3). We exclude services without a clear identification possibility based on DNS indicators, *e.g.*, Namecheap mixes services on the same infrastructure

and relies on HTTP redirects. For a more detailed description of our collection process we refer to [169]. We did not investigate infrastructure associated with less than 10k domains and might miss specialized services for TLDs not in our data.

Depending on the specific parking service, different types of indicators allow for parked domains to be identified. For example, GoDaddy (Free Parking) uses specific IP addresses (`34.98.99.30` and `34.102.136.180`), while AfterNic relies on a set of name servers (`ns*.afternic.com.`). We identified the reference points that are explicitly used for parking and use these values to identify parked domains using DNS measurement data (see Section 7.4). To lower the barriers for other researchers to identify and consider parked domains separately, we published our list of services including the reference points [175].

We divide the 82 identified services into four categories based on the content they display on parked pages.

Advertisements These services use domains to host a web page focusing on advertisements which generate money if a client accesses the page. Monetization is either click based (Pay-Per-Click (PPC)) or based on redirects (Pay-Per-Redirect (PPR)). We identified 25 services in this category including well known companies in the segment of hosting like GoDaddy.

Domain marketplaces These services mainly sell domain names that are considered valuable. To advertise the domain itself, they host a plain web page with a banner indicating the domain is for sale. These services are not necessarily open for use by others and can try to only sell their own domain name assets. This applies, for example, to HugeDomains.com. While not strictly acting as domain parking services as mentioned by Halvorson *et al.* [88], they show similar behavior to parking services, namely centralized infrastructure, similar content, and limited importance to most users. Out of the 82 identified services, 30 are of this category. They are marked in the published list of services accordingly.

Placeholders The third category has some similarities with the previous two, but notably comes without apparent monetization attempts. This category involves landing pages of the "this domain is taken but not yet in use" kind. These domains can also share dedicated infrastructure and the hosted pages thus only display placeholders. In some cases, these services are operated by registrars or CDNs. Our list includes 23 services of this category.

Mixed Four identified services use the same infrastructure for advertisement focused parking but also domain name sales or simple placeholders, *e.g.*, Sedo and Uniregistry. These services are still parking providers but can not be mapped to one of the introduced categories. We add them to the list of services but indicate them as mixed category.

7.3 DNS Data Sources

To quantify the prevalence of parked domains, we rely on three different DNS data sources. All three data sources involve independent collection efforts. One collection effort is implemented by GINO (see Chapter 2), labeled as TUM in the following. The other two involve independent projects that collect and share DNS data with researchers on a request basis.

TUM scans We used weekly DNS scans from GINO (see Chapter 2), targeting more than 325M domain names each run. The total input of domains remained relatively even with a slight increase by 3% throughout one year. Each scan takes 24 h to 48 h to complete, responsibly distributing measurement load and impact on name servers. We excluded domains from CT log during this study for comparability with the other sources. Therefore, this measurement was seeded with domains from:

- Well over 1k available zone files from the CZDS, which includes legacy gTLDs and newer gTLDs[1];
- Names on the Alexa [129], Majestic [36] and Umbrella [37] Top 1M lists;
- A static collection of 98.1M domains from 52 ccTLDs (partial zones, *e.g.*, 22M `.tk` and 13M `.de` domains).

The country-code domains are from a static list obtained before our measurement period. However, the rate of resolving domains remained mostly stable throughout our measurement period. The data set we used includes scans performed between January 1, 2021 and January 28, 2022. As such, we cover roughly 13 months.[2] We resolve A and AAAA records during the complete period and started to explicitly resolve and collect NS records in May 2021. We cover the ethical considerations for this measurement in Section 2.3.

OpenINTEL We used data from the OpenINTEL project [71]. This project collects, among others, the A, AAAA and NS records of domain names, through active querying. OpenINTEL primarily seeds its measurement on the basis of full TLD zone files and covers approximately 65% of the global DNS namespace. The OpenINTEL measurement is seeded with domains from:

- Well over 1k available zone files from the CZDS;
- Names under 16 ccTLDs (full zone);
- Names on the Alexa [129] and Umbrella [37] Top 1M lists;

We used aggregate data: statistics regarding parked services for one scan each week. The data from OpenINTEL allows us to analyze equally longitudinal and complementary data from the same time period as the TUM scans, but from another vantage point.

Rapid7 We obtained and used a single snapshot of *forward* DNS measurement data from Rapid7 [72] for the date January 28, 2022. The Rapid7 measurement is done from a US-based vantage point and also includes some domain names that the other sources do not include, which is mostly due to differences in ccTLD coverage and special zones, *e.g.*, `.blogspot.com`. We used the Rapid7 snapshot to verify that inferences made on the basis of data collected from a US-based vantage point results in similar observations about the numbers of parked domains.

[1] Not all zone files are accessible, see Park *et al.* [176].

[2] Due to a system change, no data is available for week 25 & 26 of 2021. This does not influence later weeks due to the independence of scans.

Table 7.1: Top 10 parking services with the number of parked domains and covered eTLDs based on all three input sources on January 28, 2022.

		TUM Scans		OpenINTEL		Rapid7	
Service	Category	Domains	eTLDs	Domains	eTLDs	Domains	eTLDs
GoDaddy (Free Parking)	Parking	29.95M	571	29.89M	510	28.92M	792
HugeDomains.com	Marketplace	4.62M	6	4.61M	21	4.61M	6
Sedo	Mixed	2.97M	671	2.51M	573	3.09M	877
Skenzo	Parking	2.76M	574	2.73M	502	2.77M	832
GoDaddy (CashParking)	Parking	2.19M	521	2.17M	469	2.16M	587
dan.com	Marketplace	2.14M	640	1.97M	552	2.47M	879
ParkingCrew	Parking	1.62M	725	1.04M	576	1.56M	1031
Bodis	Parking	1.08M	638	1.00M	552	1.13M	795
survey-smiles.com	Parking	1.04M	352	1.04M	310	1.17M	376
AfterNic	Marketplace	0.96M	438	0.95M	401	0.99M	477

Table 7.2: Top 10 parking services with the used web hosting locations based on A/AAAA records for parked domains.

		Web Hosting	
Service	Category	ASN	Organization
GoDaddy (Free Parking)	Parking	15169	Google
HugeDomains.com	Marketplace	16509	Amazon
Sedo	Mixed	47846	Sedo
Skenzo	Parking	40034	Confluence Networks
GoDaddy (CashParking)	Parking	15169	Google
dan.com	Marketplace	16509	Amazon
ParkingCrew	Parking	61969	Team Internet
Bodis	Parking	16509	Amazon
survey-smiles.com	Parking	60781[1]	Leaseweb
AfterNic	Marketplace	16509	Amazon

[1] Leaseweb hosts only 23% of parked domains for survey-smiles.com. 15 further ASes host at least 10k domains see Section 7.4.2.

Public Suffix List We used the Public Suffix List [177] (PSL) to account for effective TLD (eTLD) specific statistics. To see why PSL data is needed for this, consider that `example.com.br` and `example.edu.br` are both registered names under the TLD `.br`, but have different eTLDs.

In summary, two of the DNS data sets involve longitudinal data, measured at regular intervals, and covering 13 months. By combining the three sources we can compare results from different vantage points. In general, sources seed the active measurement from multiple input sources and resolve at least the A, AAAA and NS records of domains. While data sets overlap in their seed, each data set includes unique domains extending our view. For example, one project primarily uses full TLD zones as seed, while the other uses partial seeds which extends coverage into other ccTLDs. It is worth noting that some collection efforts include, to some extent, fully qualified domain names. To increase comparability of the input data, we focus on registered domains only. In doing so, our analysis will produce a comparable lower bound of parked domains.

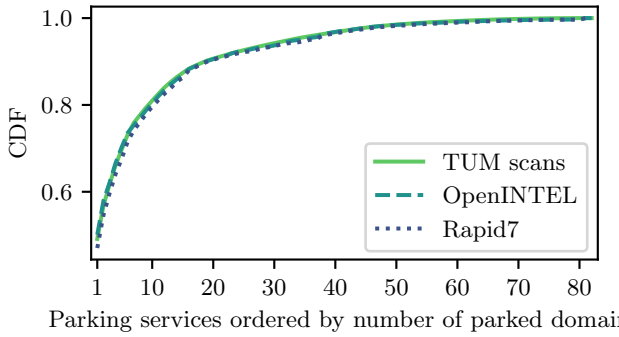

Figure 7.1: Distribution of ~60 M parked domains across services on January 28, 2022. Note the y-axis does not start at 0.

7.4 Analysis of Domain Parking

We initially used snapshots from all three data sources for January 28, 2022 and evaluated the prevalence and impact of parking services on resolved domains. In the TUM scan data, 60.7M of 267M successfully resolved domains (23%) could be mapped to parking services. Using OpenINTEL data, we inferred 59.7M of 228M (26%) domains as parked. Finally, Rapid7 data reveals 61M out of 332M domains were parked (18%). Rapid7 contained more domain names because of its coverage (see Section 7.3). However, the number of parked domains we inferred is on a par. The additional domains involved 35M AWS domains and 11M `.blogspot.com`, among others. These can not be used freely in combination with external infrastructure and thus cannot be parked. We stress that the number of parked domains we infer forms a lower bound, because our list of parking services does not encase every possible provider, and the DNS data is not without end. However, we provide an overview at parking prevalence as of 2022 and report nearly eight times more parked domains than Vissers *et al.* [86], who inferred 8M parked domains for 15 services in 2015. Note that their work relied on a historic, cumulative data set covering registered domains from two years and the number of total registered domains during their validation of identified parked domains is not available. Thus, we can not argue about the fraction of parked domains in this comparison. Also, the total number of existing domains is expected to have increased since 2015.

Figure 7.1 shows the presence and distribution of parked domains under the 82 services considered. Minor input differences aside, the results are largely similar for all three DNS data sources, which shows that our inferences were consistent among data collected at different vantage points.

Table 7.1 shows the 10 most prevalent parking services, along with their category, number of parked domains, number of eTLDs (*i.e.*, public suffixes) Table 7.2 additionally indicates the used hosting organization by the top 10. GoDaddy was the predominant service with more than 32M domains. We bisect GoDaddy into free and paid *CashParking* services, managing 30M and 2M parked domains respectively. GoDaddy increased their parking operations over the years by acquiring other services, *e.g.*, AfterNic and SmartName [178]. While AfterNic

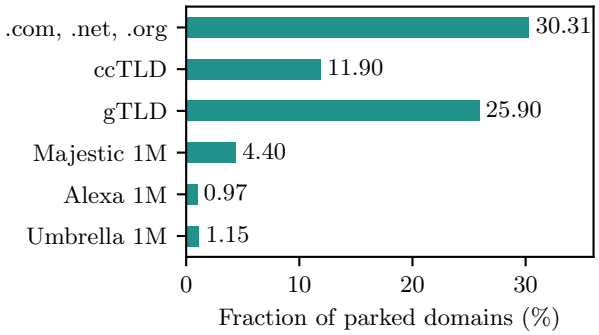

Figure 7.2: Fraction of parked in relation to resolved domains per input on January 28, 2022.

Table 7.3: Top 10 eTLDs with more than 500k resolving domains based on the percentage of parked domains on January 28, 2022.

		Parking	
eTLD	Total Domains	Domains	%
app	0.54M	0.29M	54.24
co	1.26M	0.59M	47.34
us	1.13M	0.45M	39.95
vip	0.52M	0.19M	37.10
club	0.77M	0.27M	35.25
info	3.23M	1.09M	33.89
in	0.87M	0.27M	31.37
com	138.58M	42.37M	30.57
me	0.51M	0.15M	29.76
org	9.06M	2.62M	29.00

was operated on discernible infrastructure, SmartName was run on the same infrastructure as GoDaddy (CashParking).

The second rank, *i.e.*, HugeDomains.com is a domain marketplace but does not provide selling services to customers and only sells its own names (see Section 7.2). We observe that they held more than 4.6M domains from a comparably small set of eTLDs. Additional top ranked marketplaces, dan.com and AfterNic provide selling services to customers and cover a large variety of eTLDs. The third-largest service, Sedo, a German domain parking service, is categorized as mixed. Besides general, advertisement-focused parking, it also offers a marketplace and domain brokerage service. Both ran on the same infrastructure, which hinders a clear distinction made solely based on DNS information. The leading services mainly showing placeholders (*i.e.*, no monetization) were Alibaba with 390k domains followed by 123 Reg with 270k domains (ranked 19 and 21). While 15 services already covered 85% of parked domains, differences between our services and the ones identified by Vissers *et al.* [86] show a significant shift in the domain parking ecosystem, *e.g.*, GoDaddy was not included in their list while Namedrive was bought by and integrated into ParkingCrew since 2015 [179].

We also investigated the impact of domain parking on different scan seeds and eTLDs. Figure 7.2 shows the fraction of parked domains in relation to all domains accounted for per input of the

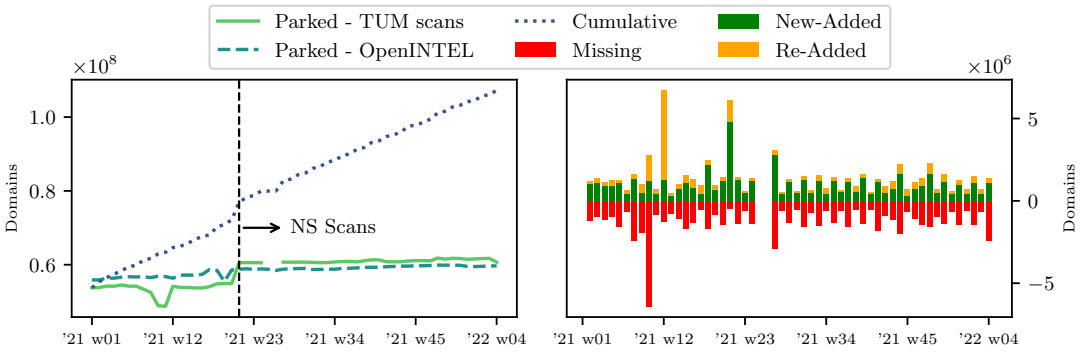

Figure 7.3: Development of parked domain names over time (January 2021 through January 2022). The gap in week 25 and 26 is due to scan system changes (see Section 7.3).

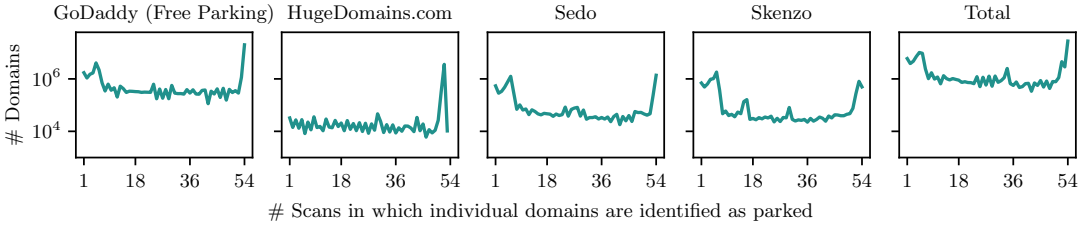

Figure 7.4: Lifetime of parked domains in number of scans for the Top 4 services and all 82 services combined from January 2021 until January 2022. All results are based on the TUM scans. Note the y-axis does not start at 0.

TUM scans as described in Section 7.3. In general, parking is most prevalent under the legacy gTLDs `.com`, `.net` and `.org` which contained roughly 30% parked domains, followed by other gTLDs and then ccTLDs. Interestingly, even top lists include parked domains. On January 28, 2022, the Majestic Top 1M [36] contained 40k (4.4%) parked domains, 15 of which had Top 10k ranks. In the Alexa Top 1M [129], 5.3k domains were parked. In general, most parked domains in top lists were on a rank above 100k. A large fraction of parked domains listed on the Majestic Top 1M was parked throughout most of our measurement period and simultaneously listed. A higher churn can be seen in combination with other top lists. We assume the Majestic list to be more broadly impacted due to differences in the generation of top lists. The Majestic Top 1M relies on web crawls and rates pages based on link metrics. This might be influenced due to efforts to showcase parked domains and increase revenue from page visits. Advertising a parked domain by placing a link onto a large variety of web pages to attract visits can result in a ranking, especially with the Majestic Top 1M. In contrast, the Alexa Top 1M requires active page visits by users recorded by its toolbar and the Umbrella Top 1M is based on visible DNS requests and includes automatically generated API requests. This shows that domain parking even affects the large amount of studies solely relying on top lists.

The domains in TUM input data involve 4095 eTLDs in total. We observed one or more parked domains under 901. We study the prevalence of parked domains under eTLDs with more than 500k domains in total and rank by percentage. Table 7.3 shows the results. `.app` contained

the highest percentage of parked domains: 54.23% (298k of 550k). `.com` accounted for most parked domains overall with 42.4M. The eTLD `.realty`, which is not tabulated because its size was 13.2k, even contained 91.4% (12k) parked domains. A majority of these was registered and parked by a single service, *i.e.*, epik.

Key take-away: *Parked domains are present in large numbers (~60M) and as significant fractions of a variety of eTLDs (e.g., 31% of .com). Furthermore, they are administered by a few services, which can drastically influence the appearance of the Internet.*

7.4.1 Development of Domain Parking over Time

We analyze the development of parking services and parked domains over a 13-month period (January 2021 through January 2022). Figure 7.3 shows the total number of parked domains in data from TUM and OpenINTEL for each week, and the cumulative number of distinct parked domains seen in between January 2021 and each scan. The second part shows the differences between two consecutive scans, *i.e.*, how many domains were not parked anymore or were newly parked. The latter is further divided into domains parked before and new, previously unknown parked domains, based on TUM data. No general reason why domains were not parked anymore can be given. They were evenly distributed across three reasons: *(i)* they were not part of zone files used as input anymore; *(ii)* they did not resolve during that respective scan, and *(iii)* they resolved but are classified as not parked. The daily number of parked domains remained relatively stable (1-2% change between scans similar to the overall change in input domains), but the overall parked names (cumulative) learned kept growing. The trends observed in TUM scan data are confirmed by OpenINTEL data (dashed line). The rise in daily parked domains in May 2021 was due to improvements to TUM scans (see Section 7.3) that increased the lower bound inference. We see a brief drop in parked domains in our scan data for week 10 & 11 of 2021. As this is not visible in the OpenINTEL data, we assume that this was a scan artifact.

While the total number of parked domains remained relatively stable, individual services can involve noticeable change. Figure 7.4 shows the number of scans during which individual domains were parked with a service. In general, most domain names could be identified as parked throughout many scans up to the complete measurement period (54 scans), thus they remained parked with their parking provider for extensive periods. For example, most domains from HugeDomains.com were parked throughout all but two scans. Similarly, 19.9M (66.6%) of domains parked with GoDaddy (Free Parking) were visible throughout the 13-month period. However, we observe occasional changes in the portfolios of parking providers, resulting in domains only identified as parked in few scans. For GoDaddy (Free Parking), these accounted for 12.3M distinct domains, identified as parked in at most six scans.

Domains, identified as parked only for a short amount of time, are often parked by drop-catch services (cf. Lauinger *et al.* [92]) shortly after expiration or by registrars themselves after cancellation by a customer until final expiration. This effect is more clearly visible for, *e.g.*, Sedo and Skenzo compared to HugeDomains.com, resulting in a lower stability of parked domains (cf. Figure 7.4). Our published list of services allows to analyze the life cycle of domains in more detail in the future.

Table 7.4: Top-10 ASes with web hosting based on number of parked domains. The general rank is based on the total number of domains relying on web hosting in the AS. R_c is a cleaned rank if parked domains are excluded.

		General		Parking		
ASN	Organization	Rank	Domains	Domains	%	R_c
15169	Google	1	40.73M	32.33M	79.4	3
16509	Amazon	2	24.71M	8.76M	35.5	1
14618	Amazon AES	7	5.53M	2.94M	53.1	12
40034	Confluence N.	14	2.91M	2.88M	98.9	427
19324	DOSArrest	43	0.84M	0.84M	99.6	1140
29873	Newfold	15	2.29M	0.76M	33.2	19
46606	Unified Layer	5	5.74M	0.72M	12.7	8
20857	Trans IP	37	0.96M	0.51M	53.7	61
24940	Hetzner	11	3.23M	0.47M	14.7	11
63949	Linode	28	1.22M	0.41M	33.8	38

Key take-away: *The overall number of parked domains remains stable in both long-running data sources and the majority of domains is parked throughout the complete period. However, a frequent change of 1-2% of domains per week is visible.*

7.4.2 Service Infrastructure

We argue that domain parking mainly affects three infrastructure components: *(i)* Web hosting; *(ii)* the DNS; and *(iii)* monetization means (*e.g.*, advertisement systems). We focus on the first two components. The latter was analyzed by Alrwais *et al.* [85].

Note, that due to the TUM scan spanning more than 24h, distributing resolved domains and records evenly, some domains identified using specific records might not be parked during the scan of all remaining records anymore/yet. However, as shown in Section 7.4.1, most domains remain stable over time and were seen in multiple scans.

Web hosting Parking services can use their own infrastructure to host parked pages or rely on external, large providers such as Google or Amazon. Table 7.4 shows the Top 10 ASes ranked by the number of hosted parking pages, and the total number of hosted domains and associated ranking.

Multiple prominent hosting locations were extensively used by domain parking services. All domains parked with GoDaddy (Free Parking) were hosted within Google and all of dan.com were within Amazon. In fact, for 72 services at least 95% of the parked pages were hosted in a single AS. Evaluating the effects of these ASes on Internet centralization without taking parking into account can bias results. For example, consider that Google (AS15169) was the most-used AS based on DNS resolutions in total and by parked domains. In TUM data, 40.7M domains in total resolved to a Google IP address, but GoDaddy relied on Google Cloud (see Table 7.2) and as such 32.1M (78.9%) were in fact parked. Such effects were recognized by Zembruzki *et al.* [93] who found that in recent years GoDaddy (Free Parking) switched from self-hosting to Google Cloud, which drastically changed their view on hosting centralization.

Amazon (AS16509) was second in rank and was used by multiple services, including HugeDomains.com, dan.com and Bodis (see Table 7.2). However, only 35.5% domains in Amazon (8.8M

Table 7.5: Top-10 ASes containing name servers based on number of parked domains. The general rank is based on the total number of domains delegated to a name server in the AS. R_c is a cleaned rank if parked domains are excluded.

		General		Parking		
ASN	Organization	Rank	Domains	Domains	%	R_c
44273	GoDaddy DNS	1	57.88M	33.08M	57.1	2
16509	Amazon	3	17.76M	8.25M	46.4	8
14618	Amazon AES	13	8.63M	6.44M	74.6	24
47846	Sedo	29	1.98M	1.87M	94.4	204
13335	Cloudflare	2	27.09M	1.59M	5.8	1
40034	Confluence N.	51	1.02M	0.96M	94.2	310
33438	Highwinds	52	0.97M	0.93M	96.2	408
397238	NeuStar	9	9.99M	0.68M	6.8	9
397220	NeuStar	10	9.95M	0.68M	6.8	10
397213	NeuStar	11	9.94M	0.68M	6.8	11

out of 24.7M) were parked. Confluence Networks, ranked 14 in terms of overall hosting, contained 98.9% parked domains. This is comparable to DOSArrest with 99.6% and overall rank 43.

While most services rely on a single hosting location, some services distribute domains over several ASes. The service, snparking.ru, hosted 42% of domains using Amazon, 28% using Serverel (AS50245) and another 28% using Sedo (AS47846). Note, the latter is a parking service itself, additionally enabling other services on their infrastructure. Based on the contained privacy policy of websites hosted within Amazon, snparking.ru likely relies on ParkingCrew besides Sedo. Among all parking services, survey-smiles.com showed the most distribution. They had 1M parked domains and hosted at least 10k domains using 16 different providers. The largest fraction was hosted within Private Layer (AS51852, 23%), followed by Nocix (AS33387, 22%) and NFOrce (AS43350, 12%). It is in general the most opaque service in our analysis, but with 1M parked domains a clearly visible one.

DNS provider The mode of operation of a parking service determines how parked domains need to configure their DNS, which for example involves name server delegation (see Section 7.2). Some services use their name servers not only for parked domains but also other resources. GoDaddy (Free Parking) is an example. We can use the NS records of parked domains to analyze the DNS infrastructure used by parked domains and how this infrastructure relates to parking services. Table 7.5 shows the Top 10 ASes in which authoritative name servers for parked domains were located, as well as the total number of domains that used this DNS infrastructure and the associated rank.

GoDaddy DNS was authoritative for the highest number of domains in the TUM data set in total with 57.8M domains out of 260M. However, 33.1M (57.1%) out of these domains were parked. The AS was mostly used by domains parked with GoDaddy itself, but also, *e.g.*, by 960k domains of AfterNic, which GoDaddy owns. In contrast, Cloudflare ranked second and was authoritative for 27M domains out of which *only* 1.6M (5.8%) were parked. Similarly to web hosting, Amazon was used by a variety of parking services, *e.g.*, dan.com (2M parked domains), Sedo (1.7M) and ParkingCrew (1.4M). Sedo (mostly used by Sedo itself), Confluence Networks

(mostly Skenzo) and Highwinds (mostly Bodis) were in the Top 60 of name server hosting ASes and were used almost exclusively by parked domains (94% to 96%).

Key take-away: *Domain parking accounts for large fractions of domains hosted within large providers and of domains delegated to authoritative name servers within well known DNS providers, e.g., Google, Amazon or Linode. Therefore, they directly influence the analysis of commonly selected, important organizations.*

7.4.3 Verification and Content Similarity

The validation of our results based on visual identification and the comparison of websites hosted on parked domains has been done by Steffen Deusch and Mattijs Jonker. It validates our findings but is not important for the detection and evaluation of protocol deployments as focused on in this work. Therefore, it is only summarized in this work.

The presented validation *(i)* accesses websites hosted on randomly sampled parked domains, takes screenshots and checks the visible content and *(ii)* compares the content of parked domains based on Common Crawl (CC) data [180]. The first methodology verified that only parking pages are visible for the random subset. The second methodology was used to reduce the required time to manually inspect websites and showed that most websites extracted from Common Crawl data are highly similar for each parking service.

The verification shows that our collected list of services and DNS indicators results in correctly identified parked domain names. It highlights the marginal importance to users due to the high similarity of content within each service.

7.4.4 Changes in the Parking Ecosystem

The following analysis is not part of the initial publication [10] but conducted one year later to re-evaluate the effectiveness of collected parking indicators and the state of the parking ecosystem.

We used a DNS scan from September 15th, 2023 (more than one year after the initial study [10]) to re-evaluate the state of parking. We rely on a scan conducted by GINO at TUM similar to the scans described in Section 7.3. The main difference is, that the Alexa Top 1M is not available anymore, but we added a new list, namely Cloudflare Radar [181].

Within this scan, 58.5M (17.5%) were parked out of 334M resolved domains. Furthermore, the fraction of parked domains within each source (*e.g.*, ccTLD or gTLD) was similar to results reported in Figure 7.2. Even the newly added Top 1M list, Cloudflare Radar, contained up to 26k parked domains.

While the quantitative results are stable, interesting changes can be seen when analyzing the used providers. Table 7.6 shows an updated list of providers hosting the most parked domains (based on A/AAAA records). In comparison to initial results in Table 7.4, another AS operated by Google (AS396982) hosted most parked domains in September 2023. This was still due to GoDaddy (Free Parking) relying on Google Cloud to host parked domains and Google announcing the used address from a different AS. However, GoDaddy (CashParking) changed its

Table 7.6: Top-10 ASes with web hosting based on number of parked domains based on a scan from September 2023. The general rank is based on the total number of domains relying on web hosting in the AS. R_c is a cleaned rank if parked domains are excluded.

		General		Parking		
ASN	Organization	Rank	Domains	Domains	%	R_c
396982	Google	2	32.21M	28.99M	90.0	13
16509	Amazon	1	36.22M	12.69M	35.0	2
14618	Amazon AES	5	7.77M	3.32M	42.8	8
40034	Confluence N.	21	1.67M	1.67M	99.9	4344
29873	Newfold	19	2.12M	0.64M	30.3	20
46606	Unified Layer	10	4.69M	0.54M	11.6	10
20857	Trans IP	42	0.94M	0.45M	47.7	59
32244	Liquid Web	35	1.06M	0.42M	39.1	50
15169	Google	4	9.50M	0.40M	4.3	3
197695	reg.ru	46	0.85M	0.33M	38.2	56

hosting provider from Google to Amazon. This increased the number of parked domains within Amazon by around 3M. Regarding DNS providers, no drastic changes were visible compared to the results from Table 7.5.

Key take-away: *This analysis shows that results are similar to our initial study and collected indicators are still valid and can be used to identify parked domains. We take this and the overall stability of parked domains over time as a sign that the DNS-based indicators that we developed (see Section 7.2) remain valid throughout. This also suggests that the list that we share will not quickly lose its value. It can of course be amended in the future. While the main indicators stay the same, parking services change their infrastructure and used providers. Therefore, future studies should rely on an up-to-date classification of parked domains and their impact on specifically used data.*

7.5 Discussion and Summary

Our results underpin the prevalence of parked domains. Taking `.com` as an example, up to 31% of names are parked. Evidently, evaluations that indiscriminately rely on domain names to identify infrastructure (*e.g.*, hosting or DNS) can inadvertently be biased by parked domains. So even though most parked domains are part of the web, we argue that their special characteristics need to be taken into account.

The hosting of advertisements and sales banners on parked pages results in large numbers of similar pages (see Section 7.4) that are centralized on the same infrastructure. For example, GoDaddy (Free Parking) uses only two IP addresses to host more than 29M domains. The content of these pages only has circumstantial benefits for visitors. Furthermore, our analysis of used DNS infrastructure (see Section 7.4.2) highlights potential impact. While more domains would be affected in total in case of a service disruption of GoDaddy DNS, 57.1% of these are parked. This, we argue, would not be as consequential as a Cloudflare disruption, which would affect considerably more non-parked (and thus user-centric) domains.

Due to their economical value and prevalence, parking services and parked domains should of course still be the subject of future research. However, we propose that, based on their specific appearance and client value, they need to be classified as a specific asset and evaluated as such.

A re-evaluation of findings based on our work can help to better understand the ecosystem and support future discussions regarding Internet consolidation. Studies, *e.g.*, on web [170], [172] and DNS [171], [182] consolidation often already rely on large DNS data sources but do not consider parked domains, leaving their role for future work, if recognized. Our collected parking indicators can be used by research to filter own DNS resolutions or in combination with available DNS data, *e.g.*, from OpenINTEL and Rapid7. Future research utilizing top lists should filter parked domains using our DNS indicators.

While results are not completely overturned if domain parking is considered properly, its effect on results is clearly visible, as seen in changes of ranks in Table 7.4 and Table 7.5. Even for top providers, changes are visible. As an example, Zembruzki *et al.*[93] saw a massive shift of hosted domains from GoDaddy to Google Cloud in between 2020 and 2021, and pinpoint this event to domains parked with GoDaddy. This finding is supported by our results in Section 7.4. However, they miss further effects of domain parking, *e.g.*, classifying Confluence Networks (AS40034) similar to other hosting providers. As shown in Table 7.4, its relevance drops significantly if parked domains are excluded.

Besides the evaluation of infrastructure and hosting providers, domain parking can bias evaluations of new protocol deployments, *e.g.*, regarding TLS 1.3 [41] or QUIC [9]. Both studies show that mainly large providers drive the deployment of new protocols. However, they do not consider the effect of domain parking. While the most influential service GoDaddy does neither support TLS 1.3 nor QUIC at the moment, it can change the deployment status drastically if these protocols are supported. In May 2021, we identified around 30M domains supporting QUIC (see Section 8.4). The domains parked with GoDaddy (Free Parking) alone could nearly double these findings, if GoDaddy starts to support QUIC.

Besides studies regarding Internet consolidation, the identification of parked domains can also be important during other research. During their analysis of malicious domains, Lloyd *et al.* [183] developed an approach to identify *active* domains. A major step of this approach is to identify and filter parked domains based on our indicators. This allowed to filter up to 30% of domains without reducing the number of false positives and the workload of humans in the loop.

In this chapter, we analyzed the prevalence of domain parking on the Internet. We systematically collected 82 parking services and used large-scale and longitudinal DNS data sets, representative for a sizable part of the global DNS namespace and collected at different vantage points, to identify 60M parked domains. Domain parking accounts for 20% to 30% of registered domains in most available gTLDs, including `.com`, `.net` and `.org`, and does not show any sign of decline throughout our 13-month observation period. Our findings show that most parked domains are concentrated on a few services, often consistently relying on single hosting and DNS infrastructure locations.

Furthermore, because parking services largely focus on advertisements and domain sales, the content of parked pages is highly similar, while only circumstantially relevant for visitors. This highlights our initial proposition, that while domain parking is of low importance for most users and the Internet in general, it represents a large part of the DNS and web ecosystem, and can introduce bias in evaluations that treat parked domains the same as any other. Major shares of domains hosted within, *e.g.*, Google Cloud or Amazon AWS or served by name servers within GoDaddy are only parked domains. Analyzing the impact of these on the Internet in general and its consolidation towards certain providers is drastically influenced by domain parking.

Part III
An Evaluation of QUIC Deployments

The third part of this thesis focuses on the third research goal (see Section 1.1): *Identification and Evaluation of QUIC Deployments and TLS Properties.* Standardized in 2021 [1], QUIC is a new protocol that combines functionality from different layers. It includes transport functionality similar to TCP, security features relying on TLS and the concept of streams. This increases the value of the protocol but also increases its complexity. Tools to identify and scan QUIC and evaluations of the ecosystem are important to understand its development. This part builds on the findings of previous parts and uses Internet-wide measurements to analyze the deployment of QUIC including TLS.

We focus on the Internet-wide detection of QUIC capable targets and their analysis. The latter includes the evaluation of involved operators, deployed QUIC libraries and configured QUIC and TLS parameters. Besides the results in this work, the insights into QUIC have directly influenced additional collaborations, *e.g.*, [17], [20], [22], [28]

Overview of the third part.

	Part III: An Evaluation of QUIC Deployments	
G3	Detection and evaluation of QUIC deployments	Chapter 8
	Identification of used QUIC libraries	Chapter 9

In Chapter 8 we present methodologies to identify QUIC deployments and a scanner to evaluate deployment specifics. We present an extensive view on QUIC deployments shortly before the standardization in 2021 and evaluate involved parties. We find that the deployment of QUIC was mostly driven by hypergiants. Even seemingly individual deployments in various ASes could be attributed to hypergiants as edge POPs. Nevertheless, we show a lot of early adoption and properly configured deployments.

QUIC is designed to be implemented in user space and runs on top of UDP. Therefore, a variety of implementations are available from different companies, written in different programming languages and supporting different functionality. While all implementations follow the same standards, differences are to be expected. Knowledge about used libraries is helpful to analyze QUIC measurements and traffic. In Chapter 9, we evaluate the impact of these differences on

scans, present a mechanism to identify the QUIC library of a server and analyze which libraries are actually deployed on the Internet.

Chapter 8

Finding QUIC Deployments

After nearly five years and 34 draft versions, standardization of the new connection oriented transport protocol QUIC was finalized in May 2021. Designed as a fundamental network protocol with increased complexity due to the combination of functionality from multiple network stack layers, it has the potential to drastically influence the Internet ecosystem. Even in early stages, the protocol attracted a variety of stakeholders including large providers.

In this chapter, we provide and evaluate a versatile tool set to identify QUIC capable hosts and their properties. Besides the stateful QScanner we present and analyze a newly implemented IPv4 and IPv6 ZMap module. We compare it to additional detection methods based on HTTP Alternative Service Header values from HTTP handshakes and DNS scans of the newly drafted `HTTPS` DNS resource record as of 2021. While each method revealed unique deployments the latter would allow lightweight scans to detect QUIC capable targets but is drastically biased towards Cloudflare.

This chapter shows that more than 2.3M IPv4 and 300k IPv6 addresses supported QUIC shortly before the standardization. They hosted more than 30M domains. We were able to successfully scan 26M targets. We show that TLS as an integral part is similarly configured between QUIC and TLS over TCP stacks for the same target. In comparison, we identified 45 widely varying transport parameter configurations, *e.g.*, with differences in the order of magnitudes for performance relevant parameters. Combining these configurations with HTTP Server header values and associated domains revealed two large edge deployments from Facebook and Google. Thus, while found QUIC deployments were located in 4667 ASes, numerous of these were again operated by large providers.

In our experience, IETF QUIC already saw an advanced deployment status mainly driven by large providers. We argue that the deployment state and diversity of existing implementations and seen configurations in 2021 solidifies the importance of QUIC as a future research topic.

This chapter is based on the following publication:

> J. Zirngibl, P. Buschmann, P. Sattler, B. Jaeger, J. Aulbach, and G. Carle, "It's over 9000: Analyzing early QUIC Deployments with the Standardization on the Horizon," in *Proc. ACM Internet Measurement Conference (IMC)*, 2021. DOI: 10.1145/3487552.3487826 [9]

The publication builds in parts on Philippe Buschmann's Master's thesis [34].

Author's Contributions: *The author of this work led the research by framing the original research question, laying out the methodology and by leading the analysis and synthesis into the publication. The author conducted all scans collecting required data as a foundation for this work.*

The QScanner and ZMap module was initially implemented by Philippe Buschmann based on the design from the author of this work. The author of this work maintained and improved the tools afterwards.

The comparison of QUIC with TLS over TCP was mainly done by Patrick Sattler. It is included in this work due to its relevance for understanding the state of QUIC deployments. The author of this work was mainly involved in discussions and the collection of results into the publication.

8.1 Motivation

QUIC, a new connection-oriented Internet protocol was finally standardized by the IETF in May 2021 [1]. The protocol was initially developed and implemented by Google and made public in 2013 [107]. Afterwards, the official standardization process was transferred to the IETF within its own working group[1]. Over the years, the base draft passed through 34 revisions before entering its last calls.

The QUIC protocol combines functionalities from different layers of the network stack, including the transport layer, security in the form of TLS and stream control to optimize higher layer applications. The integration of QUIC into the protocol stack and the comparison to TLS over TCP can be seen in Figure 8.1. This combination of functionality from multiple layers increases the overall complexity of a protocol and, therefore, increases the possibility of diverging implementations, potential errors or unintended behavior. In addition to the QUIC base protocol, HTTP Version 3 was drafted, specifically focusing on the deployment of HTTP on top of QUIC. HTTP/3 was finalized in June 2022 [184], one year after QUIC and this study.

As a new, fundamental network protocol, QUIC has the potential to drastically influence the Internet ecosystem. It attracted a variety of providers, developers and contributors even in its early stages. Before the final standardization, the QUIC working group already listed 22

[1] https://tools.ietf.org/wg/quic/

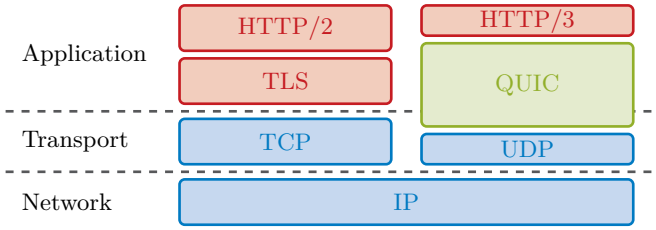

Figure 8.1: QUIC stack compared to TLS over TCP.

different implementations [185]. Additionally, with the initial contribution from Google, and as shown by Rüth *et al.* [42] in 2018 significant, productive deployment of Google QUIC was already visible on the Internet in early stages of the specification. By 2021, QUIC carried over a third of the Google traffic [186] and Facebook reported that QUIC is responsible for over 75% of its traffic [187]. Furthermore, in April 2021, Firefox officially announced QUIC support in its Nightly and Beta release [188].

Due to the attraction of the protocol in the community and within large providers but also due to the increased complexity of the protocol based on the combinations of functionality from multiple layers, thorough research of the protocol, its deployment and its effects on the Internet ecosystem are necessary. Therefore, research requires the means to identify QUIC capable targets. Furthermore, knowledge about the state of deployments, configurations but also involved parties provides a fundamental baseline for future research and developments regarding IETF QUIC.

In this chapter, we compare different methods to identify QUIC deployments and analyze the availability of QUIC shortly before the standardization based on an Internet-wide measurement study. This allows to identify how many QUIC deployments could already be found and whether deployments were well-prepared for the final standardization already supporting the latest versions. Besides the identification of deployments and supported versions, we examined characteristics in regard to successful handshakes, TLS behavior, transport parameters and HTTP/3 capabilities based on a newly implemented and shared QUIC scanner, namely the QScanner.

Our contributions in this chapter are:

(i) We compared different methodologies to identify IETF QUIC deployments. On one hand, we executed large scale ZMap based IPv4 and IPv6 scans which detect QUIC servers and their supported versions. On the other hand, we compared our findings to alternative service discovery methods that can reveal QUIC deployments. Therefore, we analyzed the HTTP ALT-SVC Header from TLS-over-TCP scans including HTTP requests and the new DNS RRs for *Service Binding*, `SVCB` and `HTTPS` [189]. We scanned in regular intervals over a period of three months to analyze the development during the final steps of the standardization.

(ii) We deployed stateful scans that attempt complete QUIC handshakes with found deployments and analyzed how many targets could be successfully connected to using QUIC. This

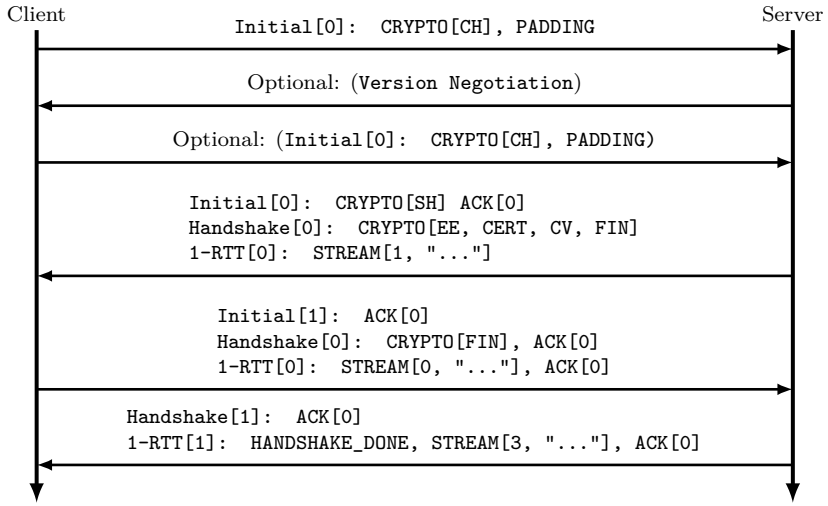

Figure 8.2: QUIC handshake including a version negotiation.

allows us and future research to analyze the configuration of targets in regard to their transport parameters, TLS and HTTP.

(iii) To support the community to conduct substantial research on QUIC, we publish our versatile tool set including the ZMap modules to detect IETF QUIC deployments and our stateful QUIC scanner: the QScanner.

8.2 Background

We explain important background covering relevant parts of the QUIC handshake and Version Negotiation (VN). Additionally, we cover the HTTP ALT-SVC Header and `HTTPS` DNS RR.

8.2.1 QUIC

QUIC is designed to be a general transport protocol combining features from different layers of the network stack. The protocol covers transport functionality, including reliable delivery and congestion control, allows network path independent connection migration, and integrates TLS 1.3 to provide confidentiality and integrity of data. To reduce latency, mainly during the handshake, it combines the exchange of transport and cryptographic parameters as shown in Figure 8.2.

As indicated in Figure 8.1, QUIC includes functionality, namely stream multiplexing and per stream flow control that is normally handled by higher layers, *e.g.*, HTTP/2. Therefore, HTTP/3 was standardized alongside QUIC [184] as HTTP derivative on top of QUIC. Furthermore, new ALPN values are specified to indicate the support of HTTP/3. During the standardization process, it included the draft version, *e.g.*, h3-29 for "Version 29" but was fixed to h3 [184].

Handshake: To begin a handshake, a client sends an *Initial* packet including a TLS 1.3 *Client Hello* and transport parameters. The latter are sent as TLS extension to offer integrity protection. Furthermore, the *Initial* packet needs to be padded to at least 1200 B. This ensures that a reasonable PMTU is supported and reduces the amplification factor to prevent potential misuse. If the server is able to pursue the handshake, it replies with an *Initial* packet containing the TLS 1.3 *Server Hello* in a *Crypto* frame, and further TLS messages in *Handshake* frames. The client needs to conclude the handshake with necessary *Acknowledgement* frames, a *Crypto* frame finishing the TLS handshake and can start data transmission using *Stream* frames.

Version Negotiation: Due to the possibility to implement QUIC in user space, different implementations are available [185] which can be quickly adapted to new versions [97]. Besides, Google advanced its own QUIC specification. This resulted in a multitude of existing versions [190] and their deployment in parallel. To allow successful handshakes without previous knowledge about supported versions by a server, a simple VN mechanism was specified alongside QUIC [1]. If the server does not support the initially offered version, it can reply to the client's *Initial* frame with a VN including its set of supported versions. A specific set of versions with the pattern `0x?a?a?a?a` is specified to enforce a VN [1]. This mechanism was extended in 2023 with more functionality [191]. Similar to Rüth *et al.* [42], we use this to discover QUIC capable hosts and their supported versions described in Section 8.3.1.

Transport Parameter: QUIC allows each client and server to specify an individual set of transport parameters [1]. To send these parameters early during the handshake and provide integrity protection, a new TLS extension was defined. The initial RFC [1] contained 17 different parameters, which can be extended in the future. For example, peers can set the initial size of the flow control window for the connection or streams, the maximum number of allowed streams, and options regarding connection migration. Some options are server-only and cannot be used by the client. Also, options set during the handshake can later be updated with corresponding QUIC frames. We look into deployment specific parameters on the Internet in more detail in Section 8.5.2.

8.2.2 Alternative Service Discovery

With the increase of different services sharing the same domain or even the same port and the necessity to reduce latency, different mechanisms exist to discover alternative service endpoints and their parameters.

HTTP Alternative Services: HTTP provides a functionality called *Alternative Services* [192], to enable servers to redirect clients. Alternative service endpoints are defined as ALPN protocol name, a host and port. They can be served using the HTTP ALT-SVC Header or in a dedicated ALTSVC frame using HTTP/2. The new HTTP/3 ALPN value can be added accordingly to indicate support. Due to the strict relation between HTTP/3 and QUIC, receiving an HTTP ALT-SVC Header with an ALPN value indicating HTTP/3 implies QUIC support.

SVCB and HTTPS DNS RR: Besides HTTP ALT-SVC Header, IETF standardized new DNS RRs, namely `SVCB` and `HTTPS` [7] to allow a client to learn about endpoints or additional information for a specific endpoint before an initial transport layer handshake using DNS. We provide detailed information about those records in Chapter 6. This chapter is based on early deployments as of Draft Version 5 [189]. We used addresses from these hints in combination with resolved domains for further analysis (see Section 8.4) and as targets for stateful scans (see Section 8.5).

8.3 Conducted Scans

This section explains our measurements and describes all implemented or used tools to conduct scans including QUIC, DNS and TLS over TCP. We published all implemented scanning tools to support the community with research regarding QUIC. For all scans, we apply ethical measures described in Section 2.3.

8.3.1 ZMap Scans

ZMap allows detecting targets supporting a specific transport protocol on a scanned port without previous knowledge about targets (at least for IPv4) [31]. Therefore, we implemented a ZMap module to detect QUIC capable hosts on the Internet as first scan. This module is similar to the work from Rüth *et al.* [42]. We are not able to directly reuse their published module, since the available code was built to detect Google QUIC versions and was neither up to date with the IETF drafts nor does it support IPv6.

The implemented ZMap module sends IETF draft conform QUIC packets enforcing a Version Negotiation packet as response. The remaining content is neither encrypted nor does it contain a *Client Hello* message. This is not necessary since the server must process the invalid version first and respond with a VN [1]. This reduces the computational overhead at the scanner and theoretically allows higher scanning rates. Furthermore, padding is added to reach the required 1200 B. Before scanning the Internet, we tested the module against local QUIC setups and officially provided test servers [185] and found no issues in 2021. However, more libraries became openly available, we improved our test environment (see Section 9.3) and updated our tools. We show in Section 9.4, that this module works but misses some deployments not following the RFC correctly.

We tested whether a VN can be triggered without padding and found a drastically lower response rate. Only 11.3% of IPv4 addresses found with padding responded and 95.4% of these were from a single AS. With a subset of targets, we tested to complete a handshake with an *Initial* packet without padding, but were unsuccessful in all tests. Thus, all tested deployments follow the specification at least for the *Initial* packet without a VN [1].

For IPv4, we scanned the complete address space filtering the local blocklist following the ethical considerations discussed in Section 2.3. For IPv6, we use `AAAA` records from domain resolutions explained in Section 8.3.2 in addition to the inputs provided by the IPv6 Hitlist service [8]. Combining both input sources we scanned 24.5M IPv6 addresses.

We show in Section 8.4 and 8.5, that a major disadvantage of these scans is missing information about domains resolving to found IP addresses supporting QUIC. During stateful scans, missing knowledge about associated domains results in a low success rate due to the strict integration of TLS into QUIC and the usage of SNI. We argue that statistics regarding the deployment of QUIC only based on ZMap scans need to be analyzed carefully.

8.3.2 DNS Scans

To circumvent the disadvantages of QUIC ZMap scans, mainly missing domains and the high bandwidth requirement, we evaluate additional means to discover QUIC deployments based on domains that can be used as SNI. In theory, the new `HTTPS` DNS RR provides a mechanism to quickly identify whether a service can be accessed using QUIC based on its domain. This allows to set up lightweight scans based on DNS resolution. In contrast to ZMap scans, previous knowledge about potential targets is required in the form of domains.

To analyze the effectiveness of this approach and the quality of found QUIC capable targets for further research, we actively resolved domain lists searching for their `SVCB` and `HTTPS` DNS RRs as explained in Section 8.2.2.

We used the DNS scan approach presented in Section 2.2. The following describes specific data used for this study. Resolved domain lists include the Alexa Top 1M [129], Majestic Top 1M [36] and Umbrella Top 1M [37]. Furthermore, we request and scan available zone files from the CZDS [38], including `com`, `net` and `org`. For this work, we used weekly scans between March 1, and May 9, 2021. While we successfully resolved `HTTPS` DNS RR of domains, none successfully resolved to an `SVCB` DNS RR. Therefore, we focus on results from `HTTPS` DNS RR scans throughout the remaining chapter. Nevertheless, we show in Section 8.4 that these scans still extend our view on the state of QUIC deployments. Our results in Chapter 6 show improved adoption since 2021 and further improvements provide research a lightweight mean to detect service information of domains.

We additionally relied on resolved `A` and `AAAA` DNS RRs for further scans including TLS and IPv6 ZMap scans. For the latter, we use `AAAA` records as input. Furthermore, DNS resolutions are combined with ZMap scans for stateful QUIC scans and TLS over TCP scans. The information from domain resolutions is used as SNI to increase the success rate of TLS handshakes.

8.3.3 TLS over TCP Scans

Due to the low success rate of `HTTPS` DNS RR scans (see Section 8.4), we additionally investigate the value of the HTTP ALT-SVC Header to detect QUIC capable targets. Compared to the DNS scans, collecting HTTP ALT-SVC Headers requires more costly scans but due to the advanced maturity of the methodology, we were able to detect more QUIC deployments as shown in Section 8.4. We rely on our regular TLS over TCP scans (see Section 2.2). This allowed us to collect HTTP ALT-SVC Headers, and thus potential QUIC deployments as explained in Section 8.2.2.

8.3.4 QScanner: A Stateful QUIC Scanner

The previous scans only allow to detect QUIC deployments and their supported versions. Collection of further information about the deployment, *e.g.*, whether complete handshakes succeed or information regarding its QUIC specifics, including TLS and HTTP properties is not possible. Therefore, we implemented a stateful scanner similar to the used Goscanner for TLS over TCP. It is based on the QUIC implementation *quic-go*[1] and *qtls*[2]. Both libraries are under active development and implement new draft versions quickly. The version used for scans analyzed in Section 8.5 supported draft 29, 32 and 34. However, it was updated shortly after the release of RFC9000 to support IETF version 1, usable with the published QScanner. According to the *Interop Runner* from Seemann *et al.* [96], it is compatible with most implementations. Therefore, we expected a high success rate with a scanner based on *quic-go* and are able to effectively parallelize our scan reducing the overall scan duration, while respecting the ethical considerations from Section 2.3. We only altered the respective QUIC and TLS libraries to expose information about QUIC, TLS and HTTP. The QScanner allows to either scan IP addresses individually, IPv4 as well as IPv6, or combined with a domain used as SNI.

We used the QScanner for stateful scans covering all found targets, *(i)* from ZMap scans in combination with DNS `A` and `AAAA` resolutions, *(ii)* from HTTP ALT-SVC Header data and *(iii)* from `HTTPS` DNS RR scans.

8.4 QUIC Deployments on the Internet

The following section analyzes the results from scans in regard to their detection rate of QUIC deployments, seen versions and potential biases towards providers. Most of the following analyses focus on the scans between May 3, until May 9, 2021 (calendar week 18). Table 8.1 provides a general overview about found targets from each source.

***ZMap results**:* With 2.1M IPv4 addresses reacting with a *Version Negotiation* packet, our ZMap scan resulted in the most targets based on IP addresses. Furthermore, these addresses were located in over 4.7k ASes. Compared to results from Rüth *et al.* [42], the number of addresses has tripled since 2018 and the involved ASes increased by 50%. Joining the result with our DNS resolution revealed 30M domains with potential QUIC support resolving to 10% of found IPv4 addresses, while no domain resolved to the remaining addresses in our scans. Most IP addresses without an associated domain belonged to large CDNs, mainly Cloudflare (AS13335) with 28%, Google (AS15169) with 22% but also Akamai (AS20940, 16.9%) and Fastly (AS54113, 12.3%). We argue that our resolved domain set could only be associated to a subset of their IP addresses due to load balancing mechanisms. Furthermore, our list of resolved domains is not exhaustive. Thus, for some IP addresses we might not be aware of associated domains.

[1] `https://github.com/quic-go/quic-go`

[2] `https://github.com/marten-seemann/qtls-go1-16/`

Table 8.1: Found QUIC targets (calendar week 18, 2021).

		Scanned	Results		
		Targets	Addresses	ASes	Domains
ZMap	IPv4	3.0B	2.1M	4.7k	31.0M[1]
	IPv6	24.4M	211.0k	1.7k	18.0M[2]
ALT-SVC[3]	IPv4	375.3M	232.6k	2.2k	36.9M
	IPv6	69.5M	283.2k	292	17.0M
HTTPS	IPv4	213.7M[4]	85.1k	1.3k	3.0M
	IPv6		69.7k	112	2.7M

[1] Join with DNS scan, 10% of IPv4 addresses map to a domain
[2] Join with DNS scan, 62% of IPv6 addresses map to a domain
[3] Extracted from TLS over TCP scans from GINO (see Chapter 2)
[4] A and AAAA records are additionally resolved to join with ZMap scans (see Section 8.3.2)

In contrast to IP addresses without associated domains, we are aware that not all joined domains might be QUIC enabled but offer different functionality. Especially for large CDNs, a domain resolving to an IP address does not necessarily mean it is used for QUIC. We evaluate the success rate of QUIC handshakes with these targets in more detail in Section 8.5.

For IPv6, we found considerably fewer QUIC capable targets and ASes. The ZMap scan resulted in 210k IPv6 addresses out of 24M probed targets (see Section 8.3). Furthermore, IPv6 addresses were located in 1.7k ASes compared to 4.7k for IPv4. On one hand, this difference was based on the fact that an IPv6 ZMap scan relies on an input and cannot scan the complete address space. On the other hand, IPv6 still lacks deployment, reducing potentially found targets. Similar differences could be seen for our TLS over TCP scans with 53M IPv4 but only 3M IPv6 addresses with an open port 443.

A domain could be found for 62% of IPv6 addresses. This was substantially higher than for IPv4 but one of the input sources for the IPv6 ZMap scans were the DNS scans. Similar to IPv4, most IPv6 addresses without domains were from CDNs (31.7% Google (AS15169), 28.5% Akamai (AS20940)).

Alternative service results: As shown in Figure 8.3, the overall success rate of `HTTPS` DNS RR per input was low with ~1% for `com/net/org` and up to 8% for top lists but increases over time. `HTTPS` DNS RRs for 2.9M domains contained information about an IPv4 QUIC deployment but hinted to only 85k distinct addresses located in 1.2k ASes. Regarding IPv6, 2.7M domains indicated QUIC support including 69.7k addresses located in only 112 ASes.

Extracting HTTP ALT-SVC Headers from TLS over TCP scans resulted in 232k QUIC capable IPv4 addresses located in 2k ASes. While this methodology revealed more QUIC deployments, it hinted to a magnitude less IPv4 addresses compared to the ZMap scan. Regarding IPv6, HTTP ALT-SVC Headers revealed a similar number of addresses, but they were located in considerably fewer ASes.

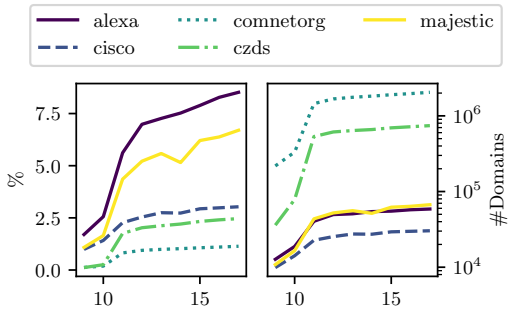

Figure 8.3: Success rate of HTTPS DNS RR scans over multiple calendar weeks (x-axis) in 2021. We resolve around 180 M domains from `com/net/org`. TLDs from CZDS (without `com/net/org`) yield in 31 M additional domains.

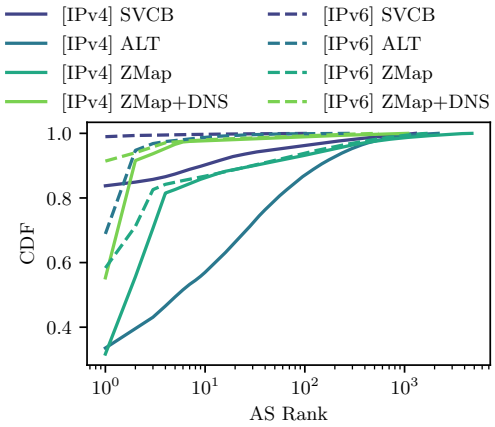

Figure 8.4: AS distribution of addresses indicating QUIC support during VN with ZMap or ALPN values (Note the y-axis does not start with 0).

We did not additionally scan addresses from these targets with ZMap but only with the QScanner (see Section 8.5). The following results regarding the distribution across ASes and supported versions were directly extracted from HTTP ALT-SVC Headers or from resolved domains and their `HTTPS` DNS RR.

Overlap between sources: All three input sources provided unique targets. The overlap of IPv4 addresses was 69.5k, while 146k IPv4 addresses were unique based on HTTP ALT-SVC Headers and 2M IPv4 addresses were only found by ZMap. Even though the success rate of `HTTPS` DNS RR scans was low (see Figure 8.3), it offered 12k unique addresses. While the overall number of IPv6 addresses was smaller, the overlap was similar with 68k addresses. Only 855 addresses were uniquely seen in `HTTPS` DNS RR and 136k addresses were from ZMap. The largest share of unique IPv6 addresses was from the HTTP ALT-SVC Header with 208k IPv6 addresses. While some addresses found by alternative service discovery mechanisms were missed by ZMap due to network events, most differences are due to deployments not implementing the VN mechanism as used by our ZMap module. Stateful scans presented in Section 8.5 show that even though found deployments did not react to the implemented ZMap module, the QScanner was able to communicate with multiple of these targets, either resulting in successful handshakes or QUIC specific alerts. In contrast, ZMap was able to detect further IP addresses supporting QUIC, missed by domain based scans from a single vantage point due to load balancing. With the later availability of new open-source libraries and test candidates, we improve our scanning approach in Chapter 9.

Key take-away: *Analyzing the found number of QUIC deployments based on the three described discovery methods illustrates their differences. Each methodology revealed unique QUIC deployments due to differences in QUIC implementations but also configurations and thus their behavior in respect to our scans. Research needs to rely on different sources of QUIC capable targets to allow a holistic analysis of QUIC and its deployment on the Internet. ZMap indicates*

Table 8.2: Top 5 providers hosting QUIC services based on each source: ZMap scans, `HTTPS` record resolutions and HTTP ALT-SVC Headers. Providers are based on ASes announcing the prefix for each IP address.

		IPv4				IPv6			
Source		Provider	AS	Addr.	#Domains	Provider	AS	Addr.	#Domains
ZMap	1	Cloudflare	13335	676.5k	23.8M	Cloudflare	13335	123.1k	17.9M
	2	Google	15169	510.5k	6.0M	Google	15169	27.2k	19.8k
	3	Akamai	20940	320.6k	23.2k	Akamai	20940	24.0k	12.7k
	4	Fastly	54113	232.8k	938.6k	Cloudflare L.	209242	3.4k	25.8k
	5	Cloudflare L.	209242	23.5k	62.0k	Jio	55836	1.4k	153
`HTTPS`	1	Cloudflare	13335	71.3k	2.9M	Cloudflare	13335	69.0k	2.7M
	2	DigitalOcean	14061	969	1.3k	Amazon	16509	263	48
	3	Google	15169	719	1.2k	DigitalOcean	14061	56	65
	4	Amazon	16509	709	814	Linode	63949	49	73
	5	OVH	16276	708	1.0k	1&1 IONOS	8560	38	42
ALT-SVC	1	Cloudflare	13335	78.0k	19.3M	Hostinger	55293	195.0k	195.0k
	2	OVH	16276	14.0k	1.7M	Cloudflare	13335	73.3k	16.0M
	3	GTS Telecom	5606	8.2k	234.1k	PrivateSystems	63410	5.9k	52.8k
	4	A2 Hosting	55293	8.1k	858.9k	EuroByte	210079	1.8k	12.4k
	5	DigitalOcean	14061	6.6k	135.9k	Synergy	45638	825	150.6k

QUIC support for most IPv4 addresses but based on our DNS scans, domains resolve to only 10% of found addresses.

We analyze whether successful handshakes are possible without a domain in Section 8.5. In comparison, alternative service discovery approaches revealed fewer addresses but HTTP ALT-SVC Headers revealed similar amounts of domains reachable using QUIC. The low success rate of `HTTPS` DNS RR scans drastically limited the utility of the scan in 2021. Similar results can be seen in 2023 as shown in Section 6.4. Most domains using these new records are operated by Cloudflare.

8.4.1 Who Deploys and Uses QUIC?

To evaluate the distribution of deployments in regard to providers, we analyzed which ASes announced ranges containing QUIC capable IP addresses. Figure 8.4 shows the distribution of addresses, which indicated QUIC support either based on a successful VN with ZMap, with an HTTP ALT-SVC Header during TLS over TCP scans or an `HTTPS` DNS RR, across ASes ranked by the number of addresses per AS. For IPv6 addresses, the overall number of hits and corresponding ASes was not only smaller, but the top AS already covered between 60% for ZMap and 99% for HTTPS DNS RR. Considering ZMap for IPv4, the top AS covered only 35%, but the top 4 already covered 80%. The most even distribution can be seen for HTTP ALT-SVC Headers where the top AS only covered 35%, and only after 100 ASes, an 80% coverage is reached.

Table 8.2 shows the top 5 ASes for IPv4 addresses. Cloudflare (AS13335) originated most addresses for all sources but IPv6 addresses in combination with their HTTP ALT-SVC Headers. The latter source was dominated by Hostinger (AS47583). Interestingly, Google (AS15169) as initial force behind QUIC was Rank 2 based on ZMap scans, Rank 3 based on `HTTPS` DNS RRs

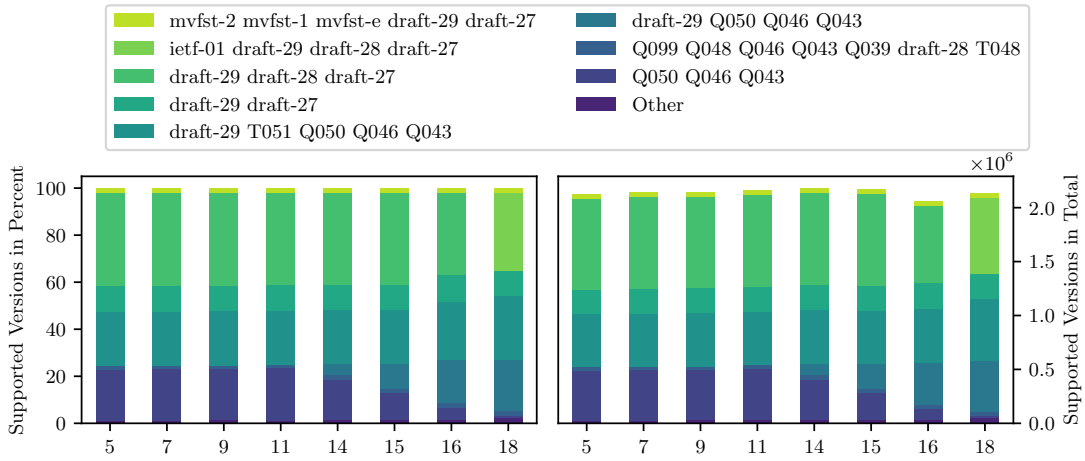

Figure 8.5: Supported QUIC version sets per IPv4 address from ZMap scans over several calendar weeks (x-axis) in 2021. *Other* combines all sets with an occurrence of less than 1 %.

and only Rank 6 based on HTTP ALT-SVC Headers. Most of the remaining top ASes were cloud providers like DigitalOcean (AS14061), Akamai (AS2094), or OVH (AS16276).

Key take-away: *The analysis shows that the early deployments could mainly be found in ASes operated by large providers. Nevertheless, deployments could be found in more than 4.7k different ASes. Using* `HTTPS` *DNS RRs to discover QUIC capable hosts were mainly limited to Cloudflare. We investigated whether deployments distributed across ASes are set up by individuals or operated by large providers in Section 8.5 based on stateful scans.*

8.4.2 Deployed Versions

Visible Versions in 2021: We used the VN results from ZMap scans, but also versions directly indicated as ALPN value in `HTTPS` DNS RRs and the HTTP ALT-SVC Headers. Figure 8.5 shows the distribution of version sets announced by servers in the VN packet during the IPv4 ZMap scans. *Other* combines 46 sets with a visibility of less than 1% each. Figure 8.6 shows the frequency of individual versions. Versions starting with Q and T indicate Google QUIC without and with TLS respectively and versions including mvfst are Facebook specific.

The sets solely consisting of IETF versions were primarily used by Cloudflare (AS13335). They were mainly responsible for the change of sets in week 18 to a new set including IETF "Version 1". At the end of our scanning period, IETF "Version 1" was seen in 95 different ASes. While the version was mentioned in draft 34, it was labeled as "do not deploy" [193].

Google (AS15169) announced the set consisting of Google QUIC versions (including T051) but also IETF draft-29. We show in Section 8.5 that this set was often inconsistent to the actual server behavior observable as version mismatches in our stateful scan. The set with only Google QUIC versions was mainly used by Akamai (AS2094) at the beginning, but they started to include IETF draft-29 throughout our measurement period. Regarding individual versions, Figure 8.6 shows on one hand, that 50% of found addresses still supported Google QUIC. On

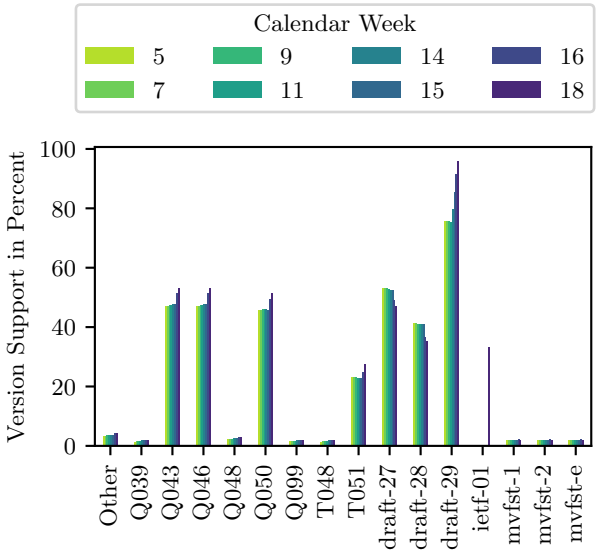

Figure 8.6: Supported individual QUIC versions from ZMap scans. *Other* combines versions with an occurrence of less than 1 %.

the other hand, the frequency of IETF draft-29 increased from 80% in February to 96% in May 2021. IETF draft-27 was seen more often than draft-28 mainly due to Fastly (AS54113) announcing the set, draft-29 and draft-27.

Results from HTTPS DNS RRs and HTTP ALT-SVC Headers do not contain exact QUIC versions but HTTP ALPN values including the draft version, *e.g.*, *h3-29* (see Section 8.2). Furthermore, the ALPN can be different for domains even when they share the same IP address.increased Therefore, the following analysis is based on targets as a combination of (Domain, IP address)-pairs.

Figure 8.7 shows the distribution of ALPN sets retrieved from HTTP ALT-SVC Headers. *Other* combines all sets with an occurrence of less than 1%. The remaining sets can be divided into three groups, *(i)* a set consisting only of IETF QUIC versions, *(ii)* sets including Google QUIC versions covering different ranges and *(iii)* a set only containing the string *quic*. The most common set only consisted of IETF QUIC versions, namely *h3-27,h3-28,h3-29*. It was mainly used by Cloudflare, besides 269 additional ASes, and thus already covered a majority of domains. Interestingly, the deployment of IETF "Version 1" from Cloudflare seen in Figure 8.5 could not be seen based on the HTTP ALT-SVC Header during our measurement period.

Even though Google itself was only the sixth common AS based on HTTP ALT-SVC Header data, the second most common set was *h3-25,h3-27,h3-Q043,h3-Q046,h3-Q050,quic*, containing Google QUIC versions and older IETF QUIC versions. It is used by 1.7k ASes. However, over time a slight shift towards a new set including ALPN *h3-27,h3-29 and h3-34* besides Google QUIC versions could be seen for targets in 444 ASes. The set only consisting of *quic* was used more often at the beginning of our measurement period but mainly lost its share towards the end.

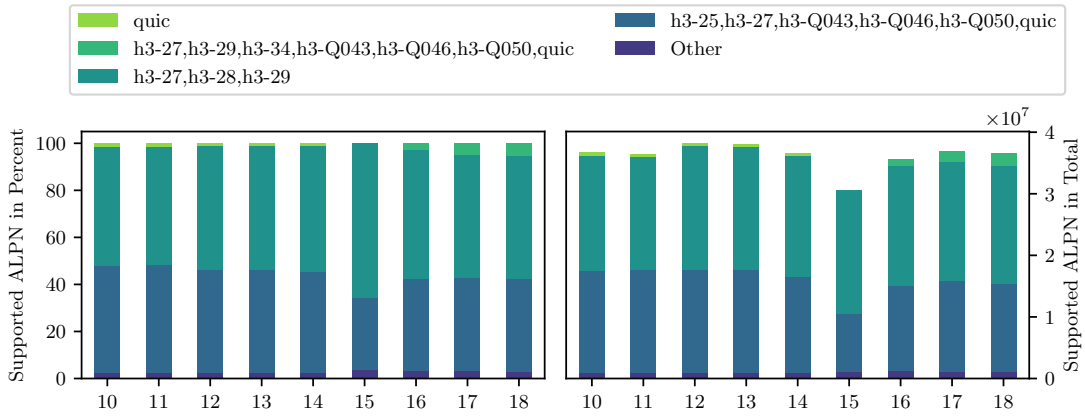

Figure 8.7: QUIC related ALPN values for domains from successful TLS-over-TCP IPv4 scans over several calendar weeks (x-axis) in 2021. *Other* combines all sets with an occurrence of less than 1 %.

Regarding the ALPN sets retrieved from `HTTPS` DNS RRs, 99.9% of the domains resolved to records including *h3-29, h3-28, h3-27*, the majorly used set from Cloudflare. As one of the driving forces behind the `HTTPS` DNS RR, Cloudflare was also dominating this dataset.

Visible Versions in 2023: We used a ZMap scan of the IPv4 address space from October 2023 to analyze the development of deployed versions. The scan identified 11.96M QUIC capable targets (see Chapter 9 for more details about the scan). 11.90M targets (99.5%) announced IETF "Version 1" as one of their supported versions and 9.9M targets (82.5%) even announced it as the only supported version. 456.0k targets (3.8%) already announced IETF "Version 2", which is identical to version 1 but some details [194]. It was specified to prevent ossification of QUIC stacks towards a single version and its details. These deployments could be found in 2.7k ASes including ISPs such as Sprint or Cogent but none of the hypergiants normally leading deployments yet. The combination of v1 and v2 as version set was also the second most common set.

Version sets including Google QUIC versions or specific versions from Facebook were still visible but with a drastically smaller share. The third most common set of versions in October 2023 consisted of three Google QUIC versions (Q043 Q046 Q050), draft 29 and IETF "Version 1".

Key take-away: *We find that based on announced versions, existing QUIC deployments in 2021 were well-prepared for the final standardization of QUIC. Throughout our initial measurement period, the support for draft 29, the final draft supposed to be deployed [193], increased to 96%. The activation of "Version 1" by Cloudflare but also other ASes even before the official conversion of the draft to an RFC shows that deployments were ready for the final standardization of IETF QUIC.*

Comparing results to data from October 2023 shows that a drastic increase of "Version 1" is visible, with up to 99.5% of deployments supporting it. First deployments even offer "Version 2" and fewer deployments offer Google QUIC or Facebook specific versions. The standardization IETF version is the dominating one.

Table 8.3: Stateful scan results of combined sources. For scans without SNI, targets are IP addresses. For scans with SNI targets are (IP address, domain)-pairs.

	IPv4 (%)		IPv6 (%)	
	no SNI	SNI	no SNI	SNI
Success	7.25	76.06	27.66	90.70
Timeout	34.50	11.09	12.35	6.01
Crypto Error (0x128)	48.26	5.73	58.85	1.90
Version Mismatch	8.83	5.77	0.74	0.99
Other	1.16	1.35	0.40	0.39
Total Targets	2M	17M	210k	14M

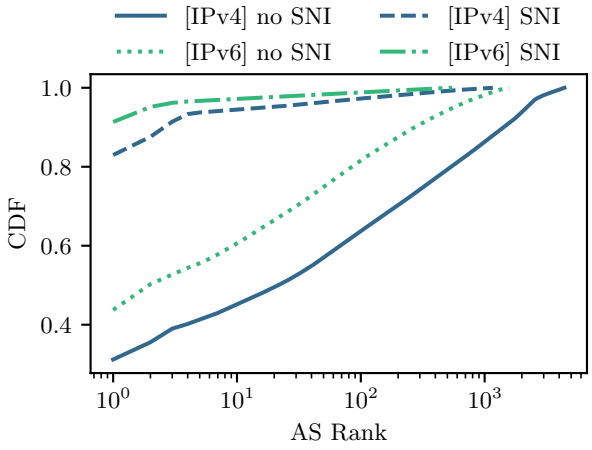

Figure 8.8: AS distribution of successfully scanned targets. For scans without SNI, targets are IP addresses. For scans with SNI targets are (IP address, domain)-pairs. Note the y-axis does not start at 0.

8.5 The State of QUIC Deployments

To analyze found QUIC deployments in more detail, we used the QScanner to complete QUIC handshakes with targets from ZMap, HTTP ALT-SVC Headers and `HTTPS` DNS RRs. We extracted QUIC specific parameters, TLS configurations and HTTP headers.

For HTTP ALT-SVC Headers and `HTTPS` DNS RRs we only scanned with SNI as we expected higher success rates and knew at least one domain per IP address that indicates reachability using QUIC. As explained in Section 8.4, for 90% of IPv4 and 38% of IPv6 addresses, no domains were found by joining the data with our DNS scans. Therefore, we scanned addresses from ZMap also without SNIs to test whether a QUIC connection can be established and to check the default behavior of targets without SNI.

To reduce the scan overhead, we only selected targets that announced a version compatible with the QScanner as of 2021, namely draft 29, 32 and 34 (see Section 8.3.4). Similar to the results from Section 8.4 we used scans from May 3, 2021 until May 09, 2021 (calendar week 18). General results are shown in Table 8.3.

NO SNI Scan: We scanned 2.0M IPv4 (95.9% of found IP addresses with ZMap) and 209.7k IPv6 addresses (99.4%). For IPv4, 7.25% or 148.3k connection attempts were successful, 34.5% timed out and 50% resulted in the generic QUIC Alert `0x128`, more precisely the generic TLS Alert `0x28` [195]. For IPv6, 27.7% or 58.0k connection attempts were successful, only 12.3% timed out and 60% resulted in the QUIC Alert `0x128`. Messages for alert `0x128` differed between scanned targets but did not reveal the exact reason why the handshake failed. We identified that the exact wording of the error message depends on their corresponding implementation, *e.g.*, we found the most prominent error message in the QUIC implementation from Cloudflare and the second most prominent string in the library from Google. We use this information to identify specific libraries in Chapter 9.

Interestingly, the handshake failed for 180k (9%) IPv4 addresses and 1.5k (0.7%) IPv6 addresses due to a version mismatch. While these addresses indicated during the ZMap scan to support a version compatible to the QScanner, the stateful scan failed with a version mismatch, *i.e.*, none of the offered versions were supported. 99% of these were part of AS15169 or AS396982, both operated by Google. We re-scanned a subset of these targets with the ZMap module and the QScanner and found that the mismatch between version negotiation and QUIC handshake was reproducible and constant over a period of time. However, in August 2021, the behavior changed, and version mismatches could not be seen anymore. We talked about our observation with Google and concluded that observed inconsistencies were most likely due to an iterative roll-out of IETF QUIC within the Google network. Due to the complete deployment of IETF QUIC by most Google services, these inconsistencies were not seen during new scans anymore.

As shown in Figure 8.8, IPv4 addresses with a successful scan still covered more than 4.4k ASes (93.1% of all seen ASes) even though the success rate was 7.25%. Successful IPv6 address scans reached a similar coverage with at least one successful target in 92.6% of the seen ASes.

SNI Scans: We combined found targets from all three sources for our scans with SNI. Besides filtering for targets with supported versions, we reduced the number of scanned domains per IP address to a maximum of 100 domains from each source for SNI scans as described in Section 2.3. Therefore, we scanned 17.4M IPv4 targets consisting of 417.7k addresses in combination with 13.3M domains and 14.2M IPv6 targets consisting of 344.4k addresses and 10.2M domains.

The total IPv4 scan reached a success rate of 76.1% while 11.1% of connection attempts timed out, 5.7% resulted in the QUIC alert `0x128` and another 5.8% failed due to a version mismatch similar to the scan without SNI (see Table 8.3). The 13M successful targets only accounted for 110k addresses, 26.5% of all scanned addresses. Furthermore, they were only part of 1.6k ASes and 82.3% were part of Cloudflare (AS13335).

Considering the IPv6 scan, the success rate was comparably high with 90.7%, but this only included 90k distinct addresses from 546 ASes. The most common errors were the QUIC alert `0x128`, a timeout and no compatible QUIC version in that order.

All three sources contributed to successful targets as shown in Table 8.4. We scanned 14M million targets with IPv4 but also with IPv6 addresses each, from *(i)* ZMap joined with DNS and *(ii)* HTTP ALT-SVC Header with a respective success rate of 85%. ZMap covered 105k

Table 8.4: Individual success rate per input. Due to an overlap between sources, targets do not sum up to total scanned targets. Targets are (IP address, domain)-pairs

	IPv4		IPv6	
Source	Targets	Success	Targets	Success
ZMAP + DNS	14.4M	85.6%	14.1M	85.3%
ALT-SVC	14.1M	85.2%	13.7M	84.9%
HTTPS	6.2M	77.6%	6.0M	77.0%

and HTTP ALT-SVC Headers covered 85k distinct IPv4 addresses respectively. `HTTPS` DNS RR resulted in only 6.2M targets based on IPv4 and IPv6 addresses respectively and reached success rates of 77%.

Key take-away: *While many QUIC deployments can be found using stateless measures from Section 8.4, successful handshakes can only be established to a subset of found hosts. The most unexpected error was the version mismatch for many targets from Google. While IETF versions were announced during the version negotiation, successful handshakes failed. After a discussion with Google, we were able to link the observed behavior to an iterative roll-out of IETF QUIC throughout Google network services. While connection attempts have been impacted throughout the roll-out period, it was only temporary and a resolution of the error is visible with the deployment of the finally standardized version. We argue that the large number of timeouts is either due to load balancers or due to the high duration of ZMap scans (see Section 8.3.1) and thus the large interval between version negotiation and stateful handshake for some targets. If* `HTTPS` *DNS RRs are deployed more widely in the future, it offers a reliable method to quickly detect QUIC service endpoints in the future and reduce the overall scan overhead for further studies drastically.*

8.5.1 QUIC TLS Behavior Compared to TLS over TCP

TLS is an intrinsic part of QUIC. Due to changed requirements, *e.g.*, the new Transport Parameters extension and the necessity to use TLS 1.3, many QUIC implementations rely on custom or patched TLS libraries. As a consequence, services reachable using QUIC and TLS over TCP might use different TLS stacks and configurations. Therefore, we evaluated the deployment of TLS as part of QUIC and compared it to TLS over TCP measurements for the same targets. As reported in Table 8.3, the stateful no SNI QUIC scans exhibited a low success rate. Our TLS over TCP scans could perform a successful TLS handshake for 43% IPv4 and 50% of the IPv6 targets. We analyzed this substantial difference and found that it was caused by only a handful of providers. Google, Akamai, Cloudflare, and Fastly are nearly evenly responsible for more than 80% (600k) of cases where the TLS over TCP scan succeeded but the QUIC scan failed. Evaluating QUIC scan errors hinted towards version mismatches, *e.g.*, as reported for Google or CDN artifacts. We discuss the impact of missing SNI values, especially for CDNs in more detail in Section 9.4. We found a small number (less than 0.5%) of targets successfully completing a QUIC scan but resulting in an error on the TLS over TCP scan. As this does not represent a relevant share and could also be caused by the two scans not running in parallel, we did not investigate it further.

Table 8.5: Share of hosts using the same TLS properties on TLS over TCP and QUIC. All properties after the TLS version are made on targets where the TLS over TCP scan also performed a TLS 1.3 handshake.

	IPv4 (%)		IPv6 (%)	
	no SNI	SNI	no SNI	SNI
Certificate	31.7	98.1	17.7	98.2
TLS Version	99.6	99.7	99.8	99.7
Key Exchange Group	100.0	100.0	100.0	100.0
Cipher	99.2	100.0	100.0	100.0
Extensions	67.3	99.9	56.4	99.9

In Table 8.5, we compare the TLS over TCP scans with the QUIC results. We evaluated whether the certificate collected by the QUIC scan was used for the same set of domains as with the TLS over TCP scan. With the SNI scan we found more than 98% of all targets returning the same certificate for QUIC but also TLS over TCP. Some certificates differed due to the delay between QUIC and TLS over TCP scans. However, we saw a different picture for the no SNI scans. Only 31.7% and 17.7% of the targets for IPv4 and IPv6 respectively returned the same certificates. Our evaluation of this artifact revealed that Google returned a self-signed certificate with the common name indicating an error due to the SNI missing on TLS over TCP. When a QUIC handshake is performed to the same target, it returned a valid certificate. Moreover, we saw the effect of Google rolling its certificates about weekly [196] which produces further certificate mismatches between QUIC and TLS over TCP.

We found few IP addresses which only offer TLS 1.2 over TCP while QUIC uses at least TLS 1.3 [195]. As QUIC implementations need a modified TLS library they often include it into their software while traditional HTTPS over TCP web servers usually allow the configuration of a TLS library. Therefore, the versions and exact deployment configurations can differ. We found that 99.7% of all targets used the same TLS version (*i.e.*, TLS 1.3). The single most contributor to differing TLS versions was Cloudflare. We investigated this and found that using Cloudflare, it was possible to disable TLS 1.3 but enable QUIC. Moreover, we found that Cloudflare enables QUIC by default, which might have been a reason for this behavior. To our knowledge there is no other reason to disable TLS 1.3 but enable QUIC.

As QUIC requires TLS 1.3, a comparison of TLS ciphers, key exchange groups and extensions is only useful if the chosen TLS over TCP version is the same. In order to have comparable results we made sure to send the same TLS *Client Hello* with our QUIC and TLS over TCP scanner. We offered the X25519 key exchange group which is accepted by close to all targets (*e.g.*, 206 of IPv4 SNI targets chose other curves). Only among those who did not choose X25519 there is in part a discrepancy between QUIC and TLS over TCP. Although the vast majority used the offered key exchange group, this shows on a small scale the effect of two different TLS deployments. We got a similar result for the chosen cipher. Currently, TLS 1.3 has five allowed cipher suites, limited to four by the QUIC draft [195] and only three specified as required to be supported, hence it provides less choice compared to TLS 1.2. Most servers chose `TLS_AES_128_GCM_SHA256` in both scans.

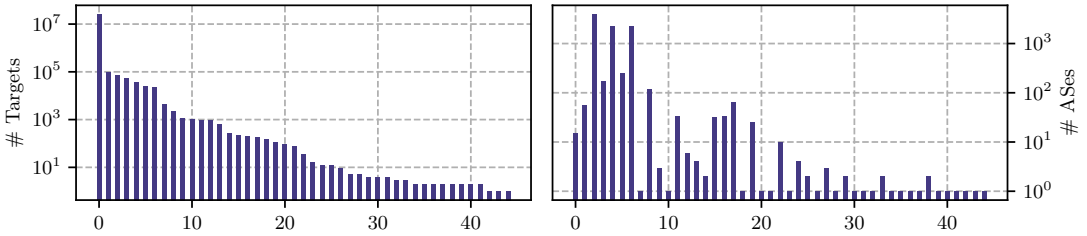

Figure 8.9: Distribution of transport parameter configurations ranked by number of targets.

QUIC requires a new TLS extension in order to transmit its transport parameters. We exclude that from our comparison to TLS over TCP to guarantee comparability. Furthermore, the latest draft of QUIC also requires using ALPN for application protocol negotiation unless there is another mechanism [197]. ALPN was also the single most significant extension which is missing in TLS over TCP no SNI scans while it was present in the QUIC scan. Again, Google and its edge deployments were the root cause for this observation.

The mismatch for the SNI scans was caused by a missing SNI extension on the TLS over TCP scans. According to RFC6066 [198] the server is required to return the extension if it used the name for certificate selection. The RFC does not actively forbid sending the extension if the value from the client is not used. As it is unlikely that a target on TCP 443 only serves a single domain but multiple on QUIC, we assume this uncritical gap in the standard led to this observation.

Key take-away: *We find that the TLS deployments on QUIC enabled hosts were very similar to the ones over TCP even though QUIC deployments are often based on new, dedicated implementations or forked TLS libraries. For a majority of QUIC deployments, the respective TLS over TCP deployments used TLS 1.3 with similar configurations. All major differences could be explained with new requirements by QUIC. As QUIC requires TLS 1.3 and most QUIC deployments were by large providers the overall state seems solid. We expect more diversity in deployments when adoption increases.*

8.5.2 QUIC Configurations and Setups

We evaluate the previously omitted and newly specified TLS extension to transmit transport parameters [195] in more detail individually. While some transport parameters are session specific, *e.g.*, *stateless reset token*, others are implementation and / or configuration specific [195] and can be used to analyze deployments, thus we ignore options which contain tokens or connection IDs. In total, we saw 45 different configurations, combining all parameters. Figure 8.9 depicts the usage of each configuration based on targets and ASes.

Most of the scanned domains reside within Cloudflare (AS13335), which all share the same parameter configuration (0 in Figure 8.9). The configuration was used by targets in 15 ASes in total. It mostly consists of the default values as specified in draft 34 [193], an *initial stream data* of 1 048 576 B and *initial max data* of a magnitude larger.

Table 8.6: Top 5 HTTP Server values by the number of ASes with at least one target returning the value. #Parameters displays the number of distinct transport parameter configurations seen in combination with the Server value.

Server Value	#ASes	#Targets	#Parameters
proxygen-bolt	2224	46.4k	4
gvs 1.0	1537	5.7k	1
LiteSpeed	238	23.8k	2
nginx	156	10.5k	16
Caddy	105	1.5k	1

Twenty configurations were only used by a single AS each and in 50% of the seen ASes, we only saw a single configuration. Analyzing specific values reveals that while for the parameters *active conn id limit*, *max ack delay* and *ack delay exponent* default values were mostly used, some parameters differed widely. For example, in the QUIC specification *max udp payload size* is by default set to the maximum UDP payload size (65 527 B). This value was used by 12 configurations, however 12 further configurations used 1500 B and overall, 10 different values could be seen. Furthermore, data transmission related parameters varied within multiple orders of magnitudes. While some deployments only promoted 8192 B of *initial max data*, others supported up to 16 777 216 B. For initial stream data, values still ranged in between 32k and 10M. However, these values can be updated throughout the connection.

Edge POPs: Interestingly, targets in 42.2% of the ASes used three configurations. To better understand these results, we include results from HTTP additionally collected by the QScanner. The HTTP HEAD request was successful for:

- IPv4 with SNI: 12.6M (95.8%)
- IPv4 without SNI: 104k (70.4%)
- IPv6 with SNI: 12.3M (96.1%)
- IPv6 without SNI: 36k (62.2%)

In total, we collected more than 8k different HTTP Headers but focus on the HTTP Server header in the following. Even though it is often recommended to reduce information included in the Server header, it can contain hints about implementations. Besides the most frequent value, *Cloudflare*, some values could be seen from targets located in a wide variety of ASes.

As shown in Table 8.6, we saw the value *proxygen-bolt* from targets in 2.2k ASes. It indicates usage of the Facebook HTTP Library *Proxygen* [199] which provides QUIC based on the mvfst [200] implementation. The value was from successful connections with 50k IP addresses, out of which we were able to associate 7.5k addresses with a domain and scan with SNI. 95% of domains contained either `fbcdn.net` or `cdninstagram.com`. Furthermore, they shared four combinations of QUIC transport parameters not seen in combination with other HTTP server values. Two configurations were only used by targets located in the Facebook AS (AS32934). They allow a relatively high initial value for all stream data parameters with 10 485 760. They differ only in the `max_udp_payload_size` parameter with 1500 B and 1404 B respectively. The remaining two configurations were mostly responsible for two out of three configurations seen in 42.2% ASes as mentioned earlier. They respectively announced both payload sizes as well

but further differ in the initial value for all stream data parameters with 67 584. We assume, the latter are edge POPs part of the Facebook CDN providing content close to the user [201]. Therefore, while these deployments were not hosted in Facebook ASes directly, they were most likely set up by Facebook as large provider.

The value *gvs 1.0* behaved similarly with 8.5k IP addresses in 1.7k ASes. While in most cases, we could not associate a domain with these IP addresses, they all used the same set of transport parameters not seen with any other HTTP Server header values. Furthermore, this value is the third configuration, seen besides the values above for many ASes. 14% of IP addresses were part of Google (AS15169) indicating a similar deployment with edge POPs.

This shows that results about the deployment state of QUIC as shown in Section 8.4 can be misleading if further information about deployments is neglected. Combining the unique information collected by the QScanner allows to investigate QUIC in more detail throughout the early stages of the protocol. These additional insights show that the deployment status in 2021 was mainly dominated by large providers, not only in their own networks, but also due to POPs in external networks.

Diversity within single ASes: Single ASes do not necessarily need to provide a unique setup, *e.g.*, from cloud providers, a variety of configurations could be visible due to individual setups from customers. The highest number of different configurations in a single AS was 11 seen at Google (AS15169), Amazon (AS16509) and DigitalOcean (AS14061). All three offer cloud computing services allowing setups from customers. The configurations differed in most values and single configurations dominated for most of the hosts while the remaining ones are rarely seen. Considering HTTP Server header values reveals a large diversity, with 44 different values at Google, including *e.g.*, different NGINX versions or *Python/3.7 aiohttp/3.7.2*. For Amazon, only 12 values could be seen and 9 for DigitialOcean. While some of these deployments and corresponding server values were most likely set up by the provider themselves, we argue that others imply individual setups inside the cloud computing services. We further analyze used QUIC libraries in Chapter 9 and show that these ASes also show the highest diversity of deployed libraries.

Additional HTTP Server Values: Besides the earlier mentioned values, the third most common value was *LiteSpeed* (see Table 8.6) indicating a deployment based on LSQUIC [202]. It was used by 24k domains in combination with 1.3k IP addresses and 240 ASes. While most targets shared the same configuration, no relation between domains or ASes could be found. Thus, in our data it was the most seen implementation not deployed by a single large provider.

The value NGINX was present for 15k targets but also as sub-string of the HTTP Server header value for 16k targets on 7.8k scanned IP addresses in combination with 17 different transport parameter combinations. Besides only NGINX, *yunjiasu-nginx* was used by 15k targets and the remaining values included a variety of different versions between 1.13.12 and 1.20.0. This reflects that besides the official QUIC branch from NGINX starting after the release of version 1.17.8 [203], others have based HTTP3 implementations on NGINX forks, *e.g.*, Cloudflare [204].

Searching for further implementations listed by the QUIC working group [185] only reveals a few hits, *e.g.*, *h2o* used by 12 targets in five ASes including different commit hashes. Most implementations were either infrequently used or not revealed by the header value.

***Parameters in 2023**:* Based on the scan from 2023 to analyze QUIC versions in Section 8.4 (see Chapter 9 for more details about the scan), we re-evaluated identified transport parameters. While we saw 45 different configurations, combining all parameters, in 2021, the number increased by nearly a magnitude to 388 in 2023. Most parameters could be seen with a variety of different values resulting in the diversity of values overall, *e.g.*, 106 *initial max data* values and 70 *max idle timeout* values.

Google and Facebook show the same behavior as in 2021 with many, identifiable edge POPs. Facebook still used four different configurations in 2023. Two were mostly used within their ASes and indicate larger initial data values, while two were often seen in other ASes and smaller data values.

***Key take-away:** Using the QUIC specific Transport Parameter TLS extension allows analyzing and identifying deployments in more detail. Due to the variety of used configurations and individual parameters, we were able to identify deployments located in a multitude of ASes as edge POPs of large providers similar to the work from Gigis* et al. *[109] but based on a differing methodology. Taking these edge POPs into account, reveals that the deployment state of QUIC was even more focused towards large providers than shown in Section 8.4, solely based on originating ASes.*

Furthermore, we argue that advertised transport parameters can be used to analyze deployments and their differences in more detail in the future. The availability of server preferences of relevant parameters regarding connection properties allows analyzing setups and the impact of different parameters on QUIC connections.

8.6 Discussion and Summary

As a foundation for future research regarding the newly standardized, fundamental network protocol QUIC, this chapter and the corresponding work provides a versatile tool set, to identify QUIC capable hosts and their properties. We presented an extensive analysis of different methodologies to detect the QUIC deployment state on the Internet shortly before the standardization. We verified that IETF QUIC already gained relevant traction before its final standardization, and we showed widespread QUIC deployment. Based on ZMap scans, `HTTPS` DNS RRs and HTTP ALT-SVC Header, we found deployments in more than 4.7k ASes and were able to conduct successful QUIC handshakes with more than 26M targets using the QScanner. We argue that QUIC has the potential to change the Internet ecosystem drastically and highlight its importance to future Internet studies due to the extensive deployment by large network providers.

Stateful scans with the QScanner revealed that TLS as an integral part was similarly configured between QUIC and TLS over TCP stacks for the same target. In contrast, different implemen-

tations and configurations, with 45 transport parameter sets could be found on the Internet. A thorough analysis of these differences and their impact on network communications and especially user experience in the future is necessary to improve the Internet, support long-term deployment of QUIC and allow the evaluation of design decisions from the protocol specification.

The Dominance of CDNs: Similar to related work from Rüth *et al.* [42], a small group of providers dominated the deployment of QUIC. While their research reported in 2018, that a majority of found deployments could be associated with Google, our work shows that the state of IETF QUIC in 2021 was mainly dominated by Cloudflare. Google was still highly involved in the development of QUIC but deployed its own version of QUIC in parallel. The dominance of large providers during the deployment of IETF drafts in their early years has also been shown by Holz *et al.* [41]. Mainly Cloudflare, but also Google, Akamai and Mozilla were the driving forces behind the quick deployment of TLS 1.3 on the Internet. While our work shows that QUIC capable hosts could be found in more than 4.7k ASes and successful connections could be established with targets in 4.4k ASes, the analysis of transport parameters and HTTP Server Header values indicates, that many of these were orchestrated by large CDNs as edge POPs similar to the work from Gigis *et al.* [109].

We argue that it has to be considered carefully by research in the future and leads to substantial centralization. Measurement studies are easily biased towards these providers. Operators cannot solely be identified based on ASes but might be responsible for distributed deployments. Nevertheless, it can be seen that QUIC as a new protocol was used by individuals even before the QUIC draft was finally standardized and even though prominent HTTP servers, *e.g.*, NGINX [203], only provided QUIC support on specific branches.

Fingerprinting QUIC: Based on presented results, we argue that the combination of functionality from multiple layers of the network stack into a single protocol increases the possibility to fingerprint specific deployments or implementations. As long as many QUIC stacks implement transport functionality, necessary TLS adaptations and HTTP servers on top individually, the number of parameters pointing towards a specific implementation is comparably higher than for traditional HTTP servers with exchangeable TLS libraries built on top of an independent TCP stack. As shown in Section 8.5 we found 45 sets of QUIC parameters, out of which, some were closely related to specific providers. Further adding TLS properties and HTTP results allowed us to identify edge POP deployments of specific providers. Whether this persists in the future or whether the standardization leads to a separation of functionality, *e.g.*, with TLS specific libraries adapting to new requirements should be evaluated. We provide means to identify specific libraries besides deployments in Section 9.5.

Chapter 9

Identifying QUIC Libraries in the Wild

The diversity of QUIC implementations poses challenges for Internet measurements and the analysis of the QUIC ecosystem. The new QUIC protocol can be implemented in user space, and various implementations already exist. While all implementations follow the same specification and there is general interoperability, differences in performance, functionality, but also security (*e.g.*, due to bugs) can be expected. Therefore, knowledge about the implementation of an endpoint on the Internet can help researchers, operators, and users to better analyze connections, performance, and security. In this chapter, we improved the detection rate of QUIC scans to find more deployments and provide an approach to effectively identify QUIC server libraries based on `CONNECTION_CLOSE` frames and transport parameter orders. We performed Internet-wide scans and identified at least one deployment for 18 QUIC libraries. In total, we can identify the libraries with 8.0M IPv4 and 2.5M IPv6 addresses. Our approach provides a comprehensive view of the landscape of competing QUIC libraries.

This chapter is based on the following publication:

> J. Zirngibl, F. Gebauer, P. Sattler, M. Sosnowski, and G. Carle, "QUIC Hunter: Finding QUIC Deployments and Identifying Server Libraries Across the Internet," in *Proc. Passive and Active Measurement (PAM)*, 2024. DOI: 10.1007/978-3-031-56252-5_13 [15]

The publication builds in parts on Florian Gebauer's Bachelor's thesis [205].

Author's Contributions: *The author of this work helped during the development of the approaches and extension of required scanning tools and helped to set up and maintain long-running measurement campaigns. Furthermore, the author led the research by formulating the original research question, specifying the methodology and by leading the analysis and synthesis into the publication.*

A central feature of QUIC is the strict inclusion of TLS into the protocol. TLS was initially developed as cryptographic protocol to secure and authenticate communication channels independent of the transport layer protocol. A major use case of TLS is to secure HTTP connections on top of TCP. The author of this work helped in the development of different approaches to fingerprint and evaluate TLS deployments [18], [19], [23]. These efforts are related to the following chapter. Interested readers are referred to the respective papers.

9.1 Motivation

Based on UDP, the new QUIC protocol [1] can be implemented in user space and has thus seen wide attention from various implementers [185]. A number of operators (*e.g.*, Cloudflare [206]), but also open source teams, started to implement QUIC libraries during the standardization process. They regularly update their implementations to follow new developments and improve their library.

Even though all libraries follow the same standard, implementation differences are to be expected. As related work has shown, these differences affect functionality [96], [98] and performance [106], [207], but they can also lead to different implementation errors and potential security issues. Knowledge about deployed libraries and their potential impact can help researchers, operators and users to analyze QUIC in the future. Furthermore, the ability to distinguish between distinct library deployments can also serve as an indicator of compromise in the event of a vulnerability affecting a library. Differentiating libraries helps to understand QUIC deployments, evaluate specifics of QUIC connections, and improve QUIC usage in the future. Therefore, means to properly scan and identify QUIC libraries and an overview of their deployment on the Internet are essential. Our key contributions in this chapter are:

(i) We analyzed QUIC scanning approaches and propose a new ZMap approach to identify more deployments. We evaluated the state of QUIC deployments as of October 2023 and analyzed the importance of SNI values.

(ii) We propose an approach to identify QUIC libraries based on error messages and transport parameters. We conducted Internet-wide scans and found at least one deployment for 18 libraries respectively and 12 different libraries in one AS.

(iii) We developed a test environment for scanners and our approach. We published all tools to evaluate future changes of QUIC libraries and scan configurations, and to update our approach to identify libraries in case of changes: `https://github.com/quic-hunter/libraries`

9.2 Background

QUIC combines functionality from different layers, *e.g.*, transport functionality, but also security through TLS [1]. It is used on top of UDP and implemented in user space. This has led to many QUIC libraries. The QUIC working group lists more than 24 different libraries [185], of which 16 are tested by the QUIC Interop Runner [96].

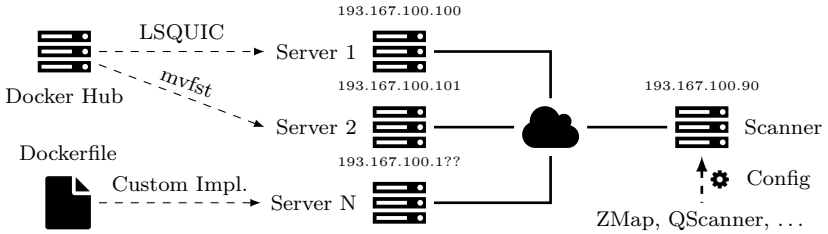

Figure 9.1: Test environment Docker setup. Each server implementation is hosted within its own container and thus isolated. Public (*e.g.*, from the QUIC Interop Runner) or self-built containers can be used.

The QUIC handshake combines the transport handshake with a TLS handshake and exchanges various information. QUIC restricts TLS to version 1.3 but adds a new extension to TLS [195], namely `quic_transport_parameters`. This extension allows the peers to exchange transport parameters during the QUIC handshake. These include values like maximum timeouts or limits on the amount of data that may be sent in this connection [1]. Thus, their content is authenticated with the successful completion of the TLS handshake during the connection establishment [195].

Furthermore, relying on TLS allows QUIC clients to send a domain as SNI value during the handshake, indicating the requested service. SNI is part of the TLS 1.3 standard and allows a server with multiple domains to select the correct certificate for authentication. Similarly, TLS offers an extension called ALPN, which allows the negotiation of an application layer protocol during the handshake (*e.g.*, `h3` to request HTTP version 3). The client sends a list of supported and requested values, while the server selects an offered value or terminates the connection early with an error.

In case of an error, QUIC peers should send a `CONNECTION_CLOSE` frame and terminate the connection. This frame may contain an error code to explicitly indicate the associated error if there was one. Interestingly, the frame may include an arbitrary reason phrase provided by the peer and encoded as a string [1]. We explain in Section 9.5 how this can be used to identify QUIC libraries.

We rely on information exchanged during the `CONNECTION_CLOSE` frame and QUIC handshake to identify implementations (see Section 9.5).

9.3 Test Environment

QUIC scanners are typically built on a single library and assume the correct implementation of the QUIC protocol as defined by the RFCs. However, due to the number of available implementations, some implement standards incorrectly, interpret parts of it differently or miss some functionality. While the QUIC interoperability runner [96] covers a variety of test cases between libraries, no test environment was available for scanners. Therefore, we developed a local test environment to evaluate QUIC libraries and their behavior to various requests. It allows for ethical evaluations without interfering with the network and existing deployments in case of unexpected behavior in edge-case scenarios.

The environment is based on Docker and isolates different servers from the scanners (see Figure 9.1). Each implementation is running in its own Docker container and is reachable via an individual IP address. An additional container executes scanners, *i.e.*, ZMap and the QScanner. While individual containers can be added, we include all 16 existing server implementations of the QUIC Interop Runner [96] as of October 2023. The library developers themselves mainly provide and regularly update them. Our test implementation supports general QUIC handshakes and HTTP/3 requests.

We used this environment *(i)* to test different scanners and configurations to improve QUIC scans (Section 9.4) and *(ii)* to develop an approach to identify QUIC libraries (Section 9.5). Furthermore, we publish this environment [208] to allow others to reproduce our findings and update our identification approach in the future in case of library changes. Updated QUIC servers can easily be integrated using new and up-to-date containers published by the library maintainers or built individually. Other researchers can use it to test scanners or scan configurations.

Besides our local environment, we collected a set of servers explicitly announced as test servers or operated by the developers of specific libraries. We only used these servers for tests with a high success rate in the local environment to confirm findings. They allow us to verify findings with real deployments on the Internet and to analyze additional libraries where no QUIC server is available for the QUIC Interop Runner (*e.g.*, google.com for Google Quiche [209]). While only a little information about a QUIC library by Akamai is available, we confirmed with Akamai that they maintain their library originally forked from Google Quiche.

Table 9.1 lists all QUIC servers integrated in our local test environment. We use already available Docker images for each implementation provided to the QUIC Interop Runner [96]. Furthermore, it lists QUIC targets on the Internet used besides our local environment to extend and verify our insights. These targets are either listed as test servers for implementations by the IETF working group [185] or official company pages by the developers of libraries.

9.4 Scanning for QUIC Deployments

Scanning for QUIC deployments is often a two-step process: *(i)* identifying targets with a stateless scan and *(ii)* evaluating the capabilities of targets with a stateful scan completing a handshake. We deployed this approach on large scale, improved the relevant components, and evaluated the state of QUIC servers on the Internet.

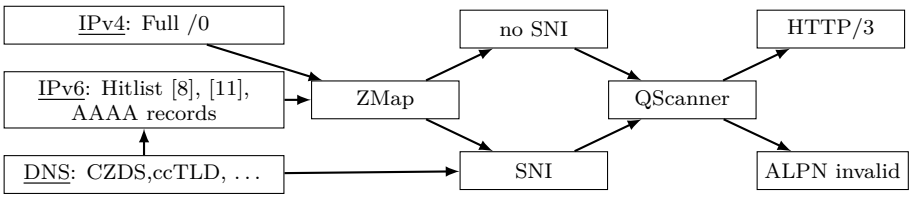

Figure 9.2: Scan setup to identify QUIC deployments and used libraries. The ALPN `invalid` scan is explained in Section 9.5.

Table 9.1: Functional servers available in our test environment (Env). GH indicates a Github repository. The listed domains can additionally be used as test servers for respective libraries on the Internet. We confirmed with Akamai that they maintain their own library forked from Google Quiche. The last two columns indicate which methodology can be used to identify the respective library (see Section 9.5). The transport parameter (TP) order allows identifying more libraries but requires a successful handshake.

	Implementation	Env	Domain	ALPN	TP
quic-go	GH: `quic-go/quic-go`	✓	`interop.seemann.io`		✓
ngtcp2	GH: `ngtcp2/ngtcp2`	✓	`nghttp2.org`		✓
Quant	GH: `NTAP/quant`	✓	`quant.eggert.org`	✓	✓
mvfst	GH: `facebookincubator/mvfst`	✓	`www.facebook.com`	✓	✓
quiche	GH: `cloudflare/quiche`	✓	`cloudflare-quic.com`		✓
Kwik	GH: `ptrd/kwik`	✓		✓	✓
picoquic	GH: `private-octopus/picoquic`	✓	`test.privateoctopus.com`		✓
aioquic	GH: `aiortc/aioquic`	✓	`quic.aiortc.org`	✓	✓
Neqo	GH: `mozilla/neqo`	✓			✓
NGINX	`quic.nginx.org/`	✓	`quic.nginx.org`	✓	✓
MsQuic	GH: `microsoft/msquic`	~[1]	`[*].sharepoint.com`		✓
XQUIC	GH: `alibaba/xquic`	✓			✓
LSQUIC	GH: `litespeedtech/lsquic`	✓	`www.litespeedtech.com`	✓	✓
HAProxy	GH: `haproxytech/quic-dev`	✓	`www.haproxy.org`		~[2]
Quinn	GH: `quinn-rs/quinn`	✓			~[3]
s2n-quic	GH: `aws/s2n-quic`	✓	`[*].cloudfront.net`	✓	✓
Haskell QUIC	GH: `kazu-yamamoto/quic`		`mew.org`		✓
Google Quiche	GH: `google/quiche`		`google.com`	✓	~[3]
Akamai QUIC			`akaquic.com`	✓	~[2]

[1] The container only supports `hq-interop` handshakes but no `h3`.
[2] The TP order of HAProxy can collide with one permutation of Akamai QUIC (see Section 9.5).
[3] The TP order of Quinn can collide with one permutation of Google Quiche (see Section 9.5).

We used Internet-wide scans set up during the work for Chapter 8 and maintained by GINO to analyze current QUIC deployments, focusing on used libraries. Our approach follows scan approaches based on ZMap to identify potential targets and DNS resolutions to find potential SNI values, followed by stateful QUIC scans (see Figure 9.2).

We seeded our DNS scans with more than 400M domains, *e.g.*, from the CZDS, a list of ccTLD domains and domains extracted from Certificate Transparency Logs. We resolved all domains to A and AAAA records for IPv4 and IPv6 addresses.

For IPv4, we seeded ZMap with the entire address space. Regarding IPv6, we used the IPv6 Hitlist [8], [11] and added IPv6 addresses extracted from AAAA records. The latter ensures that we include IPv6 addresses of large CDNs in our dataset and mitigate the impact of fully responsive prefixes as explained in Section 4.4. We used the QScanner [9] for the stateful scan. We conducted a QUIC handshake to collect transport parameters and used *h3* as ALPN value. After a successful handshake, we sent an HTTP request to collect HTTP header information.

In this chapter, we do not rely on `HTTPS` DNS resource records which could indicate QUIC support via an HTTP/3 ALPN value [7]. As shown in Section 6.4.4 records are mostly used by Cloudflare as of 2023. Furthermore, we do not use Alt-Svc HTTP headers from TLS over TCP scans, an otherwise recommended approach to negotiate HTTP/3 support [210]. They are impacted by missing SNI values as well and do not necessarily reveal additional targets. Our approach solely relies on QUIC probes sent out using ZMap and is independent of other scans.

9.4.1 ZMap: Version Negotiation

The first important step is a ZMap scan, designed to effectively evaluate on a large scale whether a target IP address is offering a service on a given port [31]. While TCP-based protocols are identified based on the 3-way handshake (sending a SYN packet and expecting a SYN-ACK), the UDP-based QUIC protocol requires a meaningful payload during the first flight. Our approach presented in Section 8.3 and used previously [42] relied on a payload that triggers a VN by the server, *i.e.*, sending a QUIC packet with an unsupported version. The QUIC standard [1] explicitly reserves versions following the pattern `0x?a?a?a?a` to be used to trigger a VN.

The module, introduced in Section 8.3, sets a reserved version and a valid size but omits most of the remaining information, *e.g.*, a valid Client Hello. Servers should first parse the version field and directly send a VN in case of an unsupported version [1], allowing to identify QUIC servers in a stateless manner. We tested this module in our test environment and found that three implementations do not respond to the probe, namely Amazon's s2n-quic, LSQUIC and aioquic. They try to parse the entire initial packet first, only check the version afterward and do not respond to the existing module. Therefore, deployments based on these libraries are not detected by the ZMap module and were missed by related work and our previous study.

In contrast, sending a QUIC initial packet including a valid Client Hello results in a response by all three implementations in our test environment. However, it triggers a handshake and key exchange, thus creating state on the server. To reduce state during ZMap scans, we extracted a valid initial packet from a QUIC communication as raw bytes, but manually set a reserved

Table 9.2: Found QUIC deployments based on ZMap with three different probes in October 2023.

	VN		Initial		Initial-VN	
IP	Addr.	ASes	Addr.	ASes	Addr.	ASes
v4	9.2M	7.5k	9.3M	8.7k	11.9M	8.7k
v6	460.7k	2.7k	483.6k	2.9k	7.9M	2.9k

Table 9.3: Successful handshakes based on the QScanner. For SNI, IP addresses with at least one successful domain are counted.

	No SNI		SNI		Total	
IP	Addr.	ASes	Addr.	ASes	Addr.	ASes
v4	522.6k	6.0k	411.9k	3.9k	876.6k	7.8k
v6	118.4k	2.4k	2.1M	1.3k	2.2M	2.8k

version. This extracted and customized packet can be used as a pre-defined payload in combination with the existing UDP ZMap module. It triggers a VN which can be implemented without state on the server (cf. [1]) and works with all tested libraries.

We compared the three possibilities within an Internet-wide scan: *(i)* a VN with zero bytes, *(ii)* a QUIC initial with version 1, and *(iii)* a QUIC initial with a reserved version. Table 9.2 shows results from a scan during October 2023. The approaches with zero bytes or a proper initial packet resulted in similar sets of identified deployments. However, our newly proposed approach with a valid QUIC initial and a reserved version identified 2.6M more IPv4 and 7.4M IPv6 targets. These targets were operated by few ASes, mainly Amazon. While s2n-quic responded to initial packets with a correct version in our test environment, Amazon deployments do not respond but probes time out. This is due to missing SNI values as shown in the next section.

Key take-away: *Properly selecting a QUIC probe during ZMap scans can drastically influence the number of identified deployments. Our test environment helps to identify high-quality probes and can be used in the future to adapt scans to potential library or protocol changes.*

9.4.2 The Importance of SNI

Servers often host multiple domains on the same IP address. The client can add the desired domain as SNI to the Client Hello during the TLS handshake. Therefore, the server can use the correct certificate and authenticate itself. In case no SNI is provided, the server can *(i)* serve a default certificate, *(ii)* respond with an error, or *(iii)* time out the connection attempt.

Analyzing uniquely identified deployments by the third ZMap approach (see Table 9.2) reveals that implementations require a proper QUIC initial packet, respond with a version negotiation in case of an incorrect version, but do not respond to a QUIC initial without SNI. Amazon owns most of these addresses. Thus, we searched for a domain hosted by them and conducted a handshake with 1k IP addresses revealed by ZMap and the domain set as SNI. In this scenario, the handshake was successful with 97.7% of the addresses. Missing values resulted in timeouts and not necessarily QUIC errors that can be analyzed. This reinforces the importance of SNI values during QUIC handshakes.

To further investigate this, we scanned all identified QUIC deployments with the QScanner without SNI values and all addresses we could map to a domain based on DNS with SNI. The latter set covers fewer IP addresses because we could not map domains to all addresses. For IPv4, ZMap discovered 11.8M addresses that reacted to our new probe. Using the QScanner, handshakes with 522.6k addresses without SNI were successful (see Table 9.3). Considering the

scan with SNI values 601.9k addresses were tested. At least one handshake with 411.9k (68.4%) addresses was successful.

As indicated by the total (836.2k), the overlap of successful addresses between SNI and no SNI scans was small. Out of the 522.6k IPv4 addresses with a successful handshake without SNI, we could map 63.1k to a domain. The handshake with an SNI was also successful for 58.0k (91.0%) out of these addresses. In contrast, only 14.1% out of 411.9k addresses with a successful SNI handshake were also successful without an SNI value. Without SNI, they often resulted in generic TLS errors or timeouts. This shows the importance of SNI for many deployments.

We used addresses from AAAA records besides the IPv6 Hitlist during our scans, following the findings from Section 4.4. Therefore, we can identify more QUIC-capable targets than the hitlist (7.9M compared to 320k). Similarly to IPv4, scans without SNI resulted in more timeouts. For SNI, the increased number of IPv6 addresses compared to IPv4 was mostly due to Amazon which often responds with eight AAAA records compared to at most four A records per domain. The distribution of IPv6 addresses is more biased toward a few ASes. 73.7% of addresses were within Amazon (AS16509), the main contributor to fully responsive prefixes (see Section 4.4).

Comparing these results to previous findings from 2021 (Chapter 8) shows a significant increase in found QUIC deployments (from 2.1M to 11.8M), their AS distribution (from 4.7k to 8.1k) and the number of successful handshakes (from 148.2k to 503.1k). The comparison was made based on IPv4 ZMap scans and handshakes without SNI, but the same effects can be seen for IPv6 and scans with SNI. Two deployments mostly dominate the new addresses. *(i)* Akamai (AS16625) with 5.6M addresses, a 20-fold increase, and *(ii)* Amazon (AS16509) with 2.2M addresses not identified in 2021. The deployment by Amazon was not necessarily a new deployment, but only detected in this study due to the different ZMap probe. In comparison, the large deployment by Akamai could also be detected with the previously existing ZMap QUIC module and was thus mostly new [211].

Key take-away: *Scanning with or without SNI drastically influences results. SNI values are essential with some operators, especially CDNs hosting multiple domains on the same IP address, to guarantee successful handshakes. In some cases, e.g., Amazon, incorrect or missing SNI values result in timeouts. This leads to incorrect conclusions regarding the existence of QUIC deployments.*

9.5 Library Identification

As shown in Chapter 8, section 9.4 and by related work [20], [98], different QUIC libraries show different behavior even though they implement the same standards. Based on these differences, we developed an approach to actively identify QUIC libraries.

9.5.1 Identification Methodology

Our goal was to identify libraries with few packets and a high success rate. Therefore, we used our test environment to evaluate different scanner configurations (*e.g.*, different TLS parameters)

Table 9.4: Error messages in response to the ALPN value `invalid`. The first column indicates who formats this exact error message.

	Implementation	Code	Message
Scanner	quiche, quic-go, HAProxy ngtcp2, MsQuic, picoquic,	`0x178`	tls: no application protocol
	Neqo	`0x178`	(frame type: 0x6): tls: no application protocol
	XQUIC	`0x178`	ALPN negotiation failed. Server didn't offer any protocols[1]
Server library	aioquic	`0x128`	(frame type: 0x6): No common ALPN protocols
	Kwik	`0x178`	unsupported application protocol: invalid
	LSQUIC	`0x178`	no suitable application protocol
		`0x150`	TLS alert 80[2]
	NGINX	`0x178`	handshake failed
	Quant	`0x178`	(frame type: 0x6): PTLS error 120 (NO_APPLICATION_PROTOCOL)
	Quinn	`0x178`	peer doesn't support any known protocol
	Google Quiche	`0x178`	(frame type: 0x6): 28:TLS handshake failure (ENCRYPTION_INITIAL) 120: no application protocol
	Haskell QUIC	`0x178`	no supported application protocols
	Akamai QUIC	`0x150`	200:TLS handshake failure (ENCRYPTION_INITIAL) 80: internal error
		`--`	PROTOCOL_VIOLATION: 28:No known ALPN provided by client
	mvfst	`0x178`	(frame type: 0x1c): fizz::FizzException: Unable to negotiate ALPN, as required by policy. policy=AlpnMode::Required[3]

[1] XQUIC continues with the handshake and the QScanner terminates with this error.
[2] LSQUIC checks SNI before ALPN and sends this error if SNI is required/incorrect.
[3] mvfst did not respond in the test environment. However, a manual test with `facebook.com` revealed a unique response, as well.

and the individual behavior of server implementations. Our tests resulted in an approach relying on QUIC-specific properties, namely `CONNECTION_CLOSE` frames and transport parameters.

Error Message The `CONNECTION_CLOSE` frame can provide a detailed error message, *i.e.*, a string without a specified structure. We expect that individual deployments do not change this behavior, and the respective string is specific for a library. We identified an unusual ALPN value as a reliable method to trigger an error during the handshake. ALPN values are simple strings without limitations and thus allow us to send *invalid*, a value we argue is highly unlikely used.

We found that nine implementations send a unique error message as response to unknown ALPN values. Table 9.4 shows error messages of some implementations in our test environment and from public servers. While all libraries indicate the same error, they send different text formats and slightly different content. Seven implementations do not send a specific error message, and the QScanner outputs a message from quic-go internally. Interestingly, XQUIC continues with the handshake but does not include an ALPN extension, resulting in the QScanner terminating the connection. The mvfst container image does not send an error but only indicates an error in internal logs. However, testing against `facebook.com` results in a unique error message. The same error is visible in the server logs in our local environment but most likely not sent due to a different configuration. LSQUIC checks the SNI value first and ALPN second. However,

Table 9.5: Transport parameter [213] orders and TLS extensions [214] to identify QUIC libraries. The last four randomize their order.

Implementation	Extension	Transport Parameters
s2n-quic	43-51	4-6-7-8-0-f
	51-43	4-6-7-8-0-f
LSQUIC	51-43	4-6-7-8-0-f-2
ngtcp2	43-51	0-2-f-6-7-4-8
XQUIC	43-51	0-3-4-6-7-8-f
Haskell QUIC	51-43	0-3-4-6-7-8-f
HAProxy	43-51	0-2-f-3-4-6-7-8
Quinn	51-43	3-4-6-7-8-2-0-f
quic-go	43-51	6-7-4-8-3-b-2-0-f
picoquic	43-51	4-8-3-6-7-b-f-0-2
quicly	43-51	3-6-7-4-0-f-2-8-a

Implementation	Extension	Transport Parameters
mvfst	43-51	0-6-7-4-8-a-3-2-f
quiche	51-43	0-3-4-6-7-8-a-b-f
aioquic	43-51	0-2-4-6-7-8-a-b-f
NGINX	51-43	4-8-6-7-3-b-a-0-f-2
	43-51	4-8-6-7-3-b-a-0-f-2
MsQuic	43-51	0-2-3-4-6-7-8-a-b-f
Google Q.	51-43	set(0, 2, 3, 4, 6, 7, 8, f)
Akamai Q.	43-51	set(0, 2, 3, 4, 6, 7, 8, f)
Quant	43-51	set(0, 2, 3, 4, 6, 8, f)
Neqo	51-43	set(0, 6, 4, f, 8, 7)

both error messages are unique compared to others in our experience and identify LSQUIC. We saw matching error messages for the remaining implementations with a server in our local environment and a test server on the Internet.

We verified the respective messages are created within the published code and found that messages have not been changed within the last three to five years for most libraries. For example, the error message within LSQUIC has not been updated for at least five years as of January 2024 [212]. Thus, we expect them to remain stable in the future as well.

Transport Parameters Besides the error message usable for nine libraries, we use the order of transport parameters. Our test environment revealed that most implementations send specific transport parameters in a particular order. The explicit transport parameters are implemented and set by individual QUIC libraries and are not necessarily part of the TLS library, *e.g.*, BoringSSL. We create an identifier for each QUIC library representing the parameters and their order. We remove their values as they can be more easily configured. In 2021, on- and off-net deployments of Facebook and Google sent the same set of parameters but with different values (see Section 8.5.2).

Table 9.5 shows the order for all libraries included in this study. While the order of most implementations is stable, we found that four libraries randomize the order of a constant set (*e.g.*, Quant [215] and Google Quiche [216]). However, the set alone can not be used as an identifier. One permutation of Google Quiche and Akamai QUIC collides with Quinn and HAProxy respectively. Nevertheless, the latter libraries do not randomize the order. Therefore, additional handshakes can be used to check whether permutations are visible or the identifier remains stable. Furthermore, for some implementations, the set of transport parameters and their order is the same, *e.g.*, for XQUIC and Haskell QUIC. Therefore, we add the order of the *key share* and *supported versions* extensions from the received Server Hello. Both extensions must be present in TLS 1.3, and thus do not depend on a deployment-specific configuration.

Additional Considerations During the development of our approach, we tested the following features but excluded them: Additional TLS parameters could be used. However, QUIC relies solely on TLS 1.3, thus a simplified feature set compared to TCP/TLS stacks. No specific features from earlier TLS versions and only four instead of hundreds of ciphers can be used [195],

[217]. Furthermore, features are often unique to the TLS library. Some libraries are shared by implementations and information is not necessarily unique. Furthermore, information can be hardware (*e.g.*, preferred cipher suites) or configuration dependent (*e.g.*, supported groups). While this can be used as information to differentiate specific deployments, it can not be used to identify libraries. Different QUIC features are not visible during the initial handshake (*e.g.*, the acknowledgement frequency or a path MTU discovery) or not necessarily static. For example, we find 48 different configurations of transport parameter values across all quic-go deployments but all with the same order. We leave an extension to our approach to future work.

Key take-away: *QUIC libraries send different error messages or transport parameter orders and can therefore be identified. While the identification based on ALPN values can only be done for nine libraries, it does not require a successful handshake. This is especially helpful for deployments that require an SNI value as shown in Section 9.4. Table 9.1 provides information which approach can identify which library. While these identifiers can change in the future, our published test environment can be used to identify new error messages testing against updated server implementations.*

9.5.2 Library Classification on the Internet

We updated our scans (see Section 9.4) and used the QScanner with two configurations: *(i)* sending a valid `h3` ALPN and *(ii)* sending `invalid`. We focused on the classification of libraries per IP address and expect a consistent QUIC deployment on a reachable target.

Stability of Identified Libraries We verified this assumption based on the stability of our identification. Some targets can have multiple identifications due to multiple domain names mapping to a single IP address. Within a single scan, we saw no inconsistent libraries per IP address, even though up to 100 domains were used as SNI values for each IP address. Furthermore, we evaluate the stability of our identification over nine months comparing our library identification of four scans, one from April, July, October and December 2023. 6.6M IPv4 targets were visible in all four scans and were identified in at least one. 6.3M were identified in all four scans; for 99.99% of them, the identified library was consistent. It shows that the identification is consistent and stable and not influenced by network or timing events, *e.g.*, caching. Only for 215 IPv4 addresses changes were visible. They are distributed across 53 different ASes and consist of different library combinations. We argue that our identification is stable and not influenced by network or timing effects. Changes are due to actual deployment changes. For the remaining evaluation, we focused on the scan from October 2023.

Resolving Collisions For 462 IPv4 addresses, the order of transport parameters either matches HAProxy/Quinn or the respective permutation of Akamai QUIC/Google Quiche, which randomize the order (see Table 9.5). However, for 146 (31.6%) targets, the *Error Message*-based approach identified the library as Quinn and directly resolved the collision. For the remaining 316 targets indicating a collision between HAProxy and Akamai QUIC, we tried to conduct three additional handshakes and succeeded for 172. The remaining targets were not responsive anymore because the scans were conducted one week later. Given the randomization, it is highly unlikely that multiple consecutive handshakes result in the same order. Thus, for 171 targets,

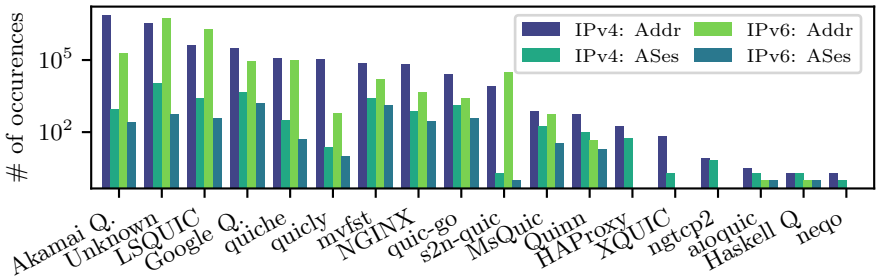

Figure 9.3: Libraries on the Internet based on both scans (with/without SNI) and our approach. Handshakes are not necessarily successful with all targets.

the order remained consistent, and we identified the library as HAProxy. One target randomized the order during all three scans and was classified as Akamai QUIC.

Approach Effectiveness In the IPv4 scan without SNI, we could identify the library for 7.9M targets based on the *Error Messages* and 519.7k based on the transport parameters. This shows the utility of the first approach. We could identify a majority of target libraries because no successful handshake is required. As shown in Section 9.4, many targets hosted by CDNs require a valid SNI, *e.g.*, Akamai. However, they terminate with a usable *Error Message* in case of an invalid ALPN value. While the *Error Messages* approach identifies libraries on more targets, the transport parameter approach allows the identification of more distinct libraries. It identified the library for 167.2k previously unknown targets. For 352.5k targets, both approaches resulted in a matching classification, and in total 8.0M (67.1%) targets could be classified. Using a domain as SNI allowed identifying libraries for 433.4k targets. In this scenario, fewer results are based on the *Error Messages* (289.8k) than the transport parameters (411.4k). Combining both the results from the scan with and without SNI slightly improved the results, allowing us to identify the library on 8.2M targets, 68.0% of initially identified deployments from ZMap.

The unknown category consists of 3.9M IPv4 addresses. However, 2.2M (56.7%) are hosted by Amazon and 658.2k (17.2%) by Cloudflare. For both, scans without the correct SNI resulted in a timeout or a generic TLS error and due to load balancing, we can only map domains to few IP addresses. We tested 1k random IP addresses from Amazon with a valid SNI again (see Section 9.4), can conduct handshakes with 977 targets and identified the library as s2n-quic. Therefore, we could identify more libraries, but opted to not use the same SNI with multiple millions of targets within the same provider for ethical reasons. The remaining targets with unknown libraries mostly run into timeouts thus no information is available (another 5.8%), result in generic TLS errors or can not provide a certificate (*e.g.*, for scans without SNI). We found one group of targets covering around 4.4k ASes which seem to be *Google Edge Caches*. They show similar behavior in our scans, however no handshake is possible. However, the error message in respect to our ALPN *invalid* scan is different to Google Quiche. Furthermore, no TLS over TCP handshake is possible, which is normally possible with other Google edge deployments. Plain HTTP HEAD requests over TCP are possible and return a server header indicating *Google-Edge-Cache*. In general for most unknown targets, we argue that for most of

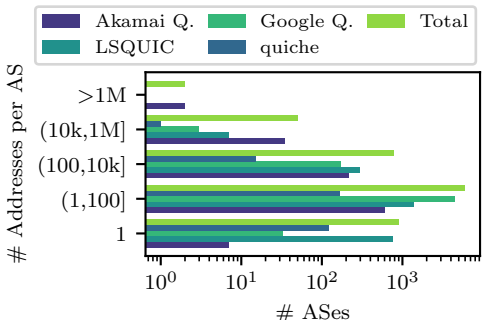

Figure 9.4: Distribution of identified libraries in IPv4 targets across ASes. Note the log x-axis.

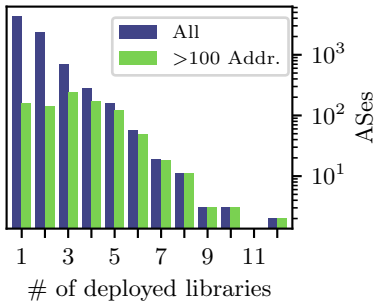

Figure 9.5: Number of distinct libraries within ASes. Note the log y-axis.

these targets, a valid SNI and thus a successful handshake results in required information to identify the library based on our approach.

Deployed Libraries Figure 9.3 shows the classification for the IPv4 and IPv6 scans, including the number of visible ASes. We found all libraries at least once, but Kwik. Figure 9.4 shows the distribution of libraries across ASes. Most ASes contain between 1 and 100 QUIC deployments we can identify, but ASes with more than 100 or even 10k deployments are visible. The most visible implementation for IPv4 is the library from Akamai, which is also the most present operator based on IP addresses deploying QUIC in general. It is the only library with more than 1M deployments in two ASes each (both operated by Akamai itself). However, it is only visible in 864 ASes, compared to the libraries of other large so-called hypergiants [218], *e.g.*, Google (4.7k ASes) and Facebook (2.5k ASes). Interestingly, quic-go (1.3k ASes) and LSQUIC (2.5k ASes) show the largest distribution across different networks besides the libraries of hypergiants. They are used for production-ready QUIC servers in Caddy and LiteSpeed. Therefore, deployments based on these products can easily use QUIC, *e.g.*, [219], [220].

In most ASes only one (4.2k, 54.3%) or two (2.3k, 30.1%) implementations are visible. However, Figure 9.5 shows up to 12 libraries within a single AS. Within Amazon (AS16509) (66.9% s2n-quic, 23.7% LSQUIC) and Digital Ocean (AS14061) (93.3% LSQUIC, 3.9% quic-go) the highest variety of implementations is visible. Thus, identifying an AS operator does not necessarily reveal the used library. Amazon develops its own library, but visible servers also rely on different QUIC libraries. Our methodology is required to correctly identify an implementation.

HTTP Server Header We compare our identification to HTTP Server headers. The header should represent the HTTP implementation, but QUIC libraries often directly include HTTP/3. However, the value is only available for 699.2k IPv4 addresses, 79.8% of targets with a successful QUIC handshake and 8.5% of targets we can identify a library of. Thus, it is no replacement for our approach, and we only used it to cross-check our identification. Our scans show that if available, the Server header value often matches or relates to the identified library. Thus, for most targets relying on Akamai QUIC, the Server header is *AkamaiGHost*, while *gvs* is used with Google Quiche and *proxygen* with Facebook.

Table 9.6: Most common HTTP Server header value for top 10 implementations. The header is only available after successful handshakes and HTTP requests. An asterisk indicates aggregated values containing the leading substring.

		HTTP Server Header		
Impl.	Target	Value	Targets	%
quiche	213 034	gvs 1.0	109 571	51.43
		gws	39 821	18.69
LSQUIC	194 953	LiteSpeed	193 411	99.21
		MCP_VCLOUD_LIVE	283	0.15
quiche	121 756	cloudflare	104 973	86.22
		nginx	9126	7.50
quicly	91 602	Varnish	88 923	97.08
		Cloudinary	822	0.90
NGINX	48 875	nginx	28 160	57.62
		GreyWS	4906	10.04
quic-go	12 526	Caddy	11 325	90.41
		nginx	426	3.40
s2n-quic	7509	AmazonS3	1640	21.84
		CloudFront	1071	14.26
mvfst	5317	proxygen[*]	5316	99.98
		gunicorn, proxygen[*]	1	0.02
Akamai QUIC	2376	Akamai[*]	1653	69.57
		TLB	87	3.66
MsQuic	927	Microsoft[*]	805	86.84
		Kestrel, Microsoft[*]	52	5.61

Table 9.6 shows the two most common HTTP Server header values for each implementation and IPv4 target. The header can only be collected for targets with successful QUIC handshakes and a successive HTTP request. The analysis combines results from the scan without and with SNI. For the latter, different server headers can be received and are combined as comma separated list. We mainly focused on deployments outside the AS of the respective developers and found the same results supporting the correctness of our approach. However, the value does not always reveal the library and is only available after a successful handshake.

Key take-away: *Our approach is stable and works for the majority of targets. Compared to a more incomplete approach based on HTTP Server headers, our error message-based approach succeeds because no successful handshake is required. We find deployments for 18 different QUIC libraries on the Internet and 12 different libraries within a single AS, showing the QUIC ecosystem diversity.*

9.6 Discussion and Summary

We analyzed QUIC deployments and used server libraries on the Internet. We evaluated QUIC scanning approaches and are able to detect previously unseen deployments based on a new ZMap scan approach. Furthermore, we developed an effective approach based on *Error Messages* and the order of transport parameters to identify QUIC libraries, both able to identify different

libraries in different scenarios (*e.g.*, only if a successful handshake is possible). Eighteen different libraries are in use with at least one target on the Internet, and up to 12 libraries are visible within a single AS. We can identify the library used by more than 8.0M IPv4 and 2.5M IPv6 addresses. *Network analysts and researchers need to be aware of this and consider potential differences based on different QUIC stacks.*

Impact on Research The variety of seen QUIC implementations shows that key goals of the standardization have been met, and libraries can be quickly developed and deployed. However, this diversity potentially increases the complexity of the network and might influence research regarding performance and security. Our library identification could extend recent studies, *e.g.*, to evaluate the impact of libraries on performance [22], to analyze the spin bit [221] or to evaluate ECN in QUIC [222] instead of relying on more incomplete HTTP server headers.

Test Environment We presented and published [208] an environment for testing scanners against different QUIC libraries. During the development of our scan and library identification methodology, we mainly limited ourselves to this environment to reduce the impact on the network and deployed QUIC servers. The environment helps to easily test scanners against a variety of implementations without impacting real-world deployments. It can easily be used in the future to adapt our approach to potential library changes.

Malicious Use Our identification approach can expose vulnerable deployments, thereby offering more effective means to identify exploitable systems. We have contacted developers of QUIC libraries about our findings and the potential to identify their specific implementations.

Part IV
Conclusion and Future Work

Chapter 10

Conclusion

In this chapter, we recapitulate and summarize our findings. The thesis followed the overall goal to provide the infrastructure, means, and knowledge to analyze deployments of a new, important protocol, namely QUIC. To reach this goal, three sub-goals were defined. Each sub-goal targets an important component of the Internet ecosystem related to QUIC deployments.

Table 10.1: Structure of this thesis and a mapping of chapters to research goals.

	Part I: IP Measurements	
G1	State of the IPv6 Hitlist	Chapter 4
	Target generation algorithms	Chapter 5
	Part II: DNS Measurements: A General Scan Foundation	
G2	Potential of new Domain Name System Resource Records	Chapter 6
	Influence of domain parking	Chapter 7
	Part III: An Evaluation of QUIC Deployments	
G3	Detection and evaluation of QUIC deployments	Chapter 8
	Identification of used QUIC libraries	Chapter 9

G1: Methodologies to Identify IPv6 Deployments.
We set out to improve the possibility of scanning and analyzing IPv6 deployments. Due to the size of the address space, it is not feasible to scan it completely and thus find deployments of any service. We used the *IPv6 Hitlist* as a foundation, analyzed its development, cleaned it from the impact of DNS injections by the GFW, and evaluated the impact of fully responsive prefixes. We further compared the latter effect to the IPv4 ecosystem and showed that the impact of fully responsive prefixes on scans needs to be considered.

Finally, we improved the *IPv6 Hitlist* in multiple steps, adding new sources, address candidates, and responsive addresses. Throughout the work on this thesis, improvements to the *IPv6 Hitlist* increased responsive addresses from 2.9M to 19.3M.

The *IPv6 Hitlist* was used as a foundation for the overall goal of this thesis. All improvements were also directly integrated into the ongoing service and are thus available to the community.

G2: Evaluation of the Impact of DNS on Internet-wide Scans.
Domains are an important resource for scans, *e.g.*, to identify IPv6 capable targets or to be used as SNI value during QUIC scans (cf. Chapter 9). We analyzed the impact of DNS on scans from two different perspectives in Part II.

(i) We evaluated whether new resource records can improve Internet-wide scans (see Chapter 6). We found that the new `HTTPS` DNS RR is used by a variety of domains, as well as web and DNS providers. Provided information besides a *simple* mapping of domains to IP addresses, *e.g.*, ports or ALPN values, can be used by scans. These records and additional information provided within DNS can be a valuable source of information for research and scans in the future.

(ii) We quantified the impact of domain parking on large-scale DNS datasets (see Chapter 7). Due to Centralized Zone Data Service and other sources, it is relatively easy to collect large domain lists and resolve them with MassDNS or ZDNS. However, not all domains are equally relevant, and biases are easily induced. We showed that up to 30% of domains, independent of their source or TLD, are parked. Furthermore, they are often hosted by prominent operators and served by large DNS providers. DNS based research should consider parked domains as different category and evaluate a potential impact.

G3: Identification and Evaluation of QUIC Deployments and TLS Properties.
To accomplish the final goal of the thesis, we developed different tools to identify and analyze QUIC deployments. We analyzed the state of deployments between 2021 (shortly before the release of RFC 9000 [1]) and 2023.

We developed the following tools during this work. All tools are open source and can be used and extended by the community in the future to build on this work.

(i) A ZMap module and different scan approaches to identify QUIC-capable targets.

(ii) The QScanner to conduct full QUIC handshakes and evaluate their configurations (*e.g.*, transport parameters and TLS properties).

(iii) A local environment to test scanners and their configuration with various QUIC server implementations.

(iv) An approach to identify the QUIC library of servers based on `CONNECTION_CLOSE` frames and transport parameters.

Our evaluation shortly before the release of the final QUIC RFC, shows that the protocol was supported by a large variety of providers, some of which already supported version 1 of the protocol. However, even though QUIC deployments were visible in more than 4.7k ASes, a more detailed analysis of deployments showed that many of these are edge POPs of Google and Facebook. Two years later, the number of QUIC deployments and the variety of providers increased.

We could further show that 18 different server libraries were used by at least one QUIC deployment each. While this diversity and ease of implementation have a positive effect on the ecosystem, they need to be considered by future research. Our evaluations in Section 9.4 already showed the potential impact of this diversity on scans. The provided library identification could extend recent studies, *e.g.*, to evaluate the impact of libraries on performance [22], to analyze the spin bit [221] or to evaluate ECN in QUIC [222] instead of relying on more incomplete HTTP server headers.

How to Measure the Internet?

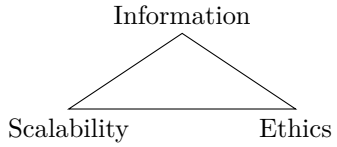

FIGURE 10.1: Measurement Properties Conflict

Actively measuring the Internet is a challenge that requires caution and planning. On one hand, it is important to maximize the gained information, *e.g.*, to detect problems or malicious actors quickly and to gain a holistic view on the Internet. On the other hand, scalability and ethics of measurements are important properties that need to be considered. Figure 10.1 depicts this conflict. Measurements can not maximize towards all properties. Thus, depending on the goals, tradeoffs have to be made.

This work offers insights and different arguments to evaluate and improve measurements. For example, full IPv6 address space scans are not feasible, but the ecosystem is steadily growing and can not be neglected by future measurement studies. The *IPv6 Hitlist* provides means to include IPv6 in measurement studies. In contrast, DNS resolutions for more than 500M domains are possible daily. However, as shown in Chapter 7, many domains are parked and thus have little value. Using them as input for higher-layer scans (*e.g.*, as SNI value during QUIC scans) might not be important for all or regular studies. Excluding these domains allows to reduce the overall scan load and focus on relevant parts of the Internet. Including important targets, *e.g.*, based on IPv6, but also excluding irrelevant targets, *e.g.*, parked domains, can be important to set up valuable, scalable, and ethical measurements.

In general, GINO relies on different scan intervals and rates for different measurements, as explained in Chapter 2. The goal is to provide a baseline of the Internet, covering the IPv6, DNS, TLS and QUIC ecosystem. The *IPv6 Hitlist* is conducted as often as possible. DNS scans for established records are conducted on a daily basis to track changes on the Internet and the ecosystem in general. The new `HTTPS` and `SVCB` records are *only* scanned on a weekly basis. As of May 2024, these records are mainly deployed by Cloudflare, and until more widespread deployment is visible, we argue fewer scans are more ethical but still valuable. Internet-wide TLS over TCP and QUIC scans are conducted on a weekly basis. They are stateful scans that induce more load on servers and the network. Therefore, for the scalability and ethics of the scans, and because the ecosystem is relatively stable, the information gained offers a good general baseline.

In contrast, for specific studies focusing on increased information gain, *e.g.*, to identify specific deployments, the scan behavior and rate towards each target can be adapted. During some studies [18], [19], [23], we relied on in-depth measurements of TLS deployments to identify and

differentiate malicious actors. We increased the gained information for individual targets by testing server behavior with multiple TLS Client Hellos. However, this decreases the scalability to Internet-wide scans and increases the impact on respective targets. Therefore, we focused on popular targets based on top lists and potentially malicious deployments based on blocklists.

General Conclusion

We have developed means to identify and analyze QUIC deployments and evaluated the ecosystem on the Internet between 2021 and 2023. However, additional protocols must be considered, their impact examined, and possibly induced biases must be understood to accomplish this goal. Our work has improved the means to cover the IPv6 ecosystem alongside IPv4 and has evaluated the impact of fully responsive prefixes. The latter are especially impactful in IPv6 but induce biases to IPv4 as well. Furthermore, we have shown the dependency of the QUIC scans on domains (as SNI values) and the potential of new records (`SVCB` and `HTTPS`) for future scans. Again, biases need to be considered, *e.g.*, the effect of domain parking, as highlighted in this work.

In summary, this work provides a holistic view on Internet-wide scans for QUIC deployments, including IP layer and DNS ecosystem peculiarities. However, the findings of this work are not limited to QUIC but should be applied to other (new) network protocols as well. Given the complexity of the Internet, the interplay of protocols, and the increasing fuzziness of protocol layer placement in the Open Systems Interconnection (OSI) model, an evaluation of new protocols can not focus on the individual protocol itself, but needs to take a look at its integration into the ecosystem.

Chapter 11

Future Work

We structure suggestions for future work alongside the three goals of this thesis.

G1: Methodologies to Identify IPv6 Deployments.
The *IPv6 Hitlist* is an important tool for IPv6 scans and used by various researchers. However, it does not claim to be complete. Future extensions, *e.g.*, based on target generation algorithms or new data sources, are important. Furthermore, the current *IPv6 Hitlist* focuses on a specific web focused target set. Besides Internet Control Message Protocol (ICMP) scans, it targets DNS (UDP/53) and HTTP (TCP/80), and HTTPS (TCP/443 and QUIC/443). Different ports must be considered for other research, *e.g.*, targeting Internet of Things (IoT) or mail, and different addresses might be responsive. Therefore, a different composition of input sources might be of interest.

Furthermore, as an ongoing service, regular supervision is required to make sure it is still of value to the community. Other factors besides fully responsive prefixes or injections by the GFW might impact the quality of the list.

Finally, aliased prefixes have long been considered an issue in IPv6, but we have shown that they rarely represent actual aliases but are often artifacts of CDNs. Instead, they should be considered fully responsive prefixes and need to be included in scans, *e.g.*, if addresses are visible in AAAA records. Future work could investigate these prefixes, their effect on scans and proper strategies to select addresses in more detail.

Furthermore, we showed that fully responsive prefixes are also visible in IPv4. While they can still be scanned in feasible time, they induce biases, and higher-layer protocol scans can be improved if they are considered. Future research could compare this effect in IPv4 and IPv6 in more detail, identifying potential sibling deployments or networks.

G2: Evaluation of the Impact of DNS on Internet-wide Scans.
The DNS ecosystem is ever-growing with the assignment of new TLDs [223] and the addition of functionality. The latter includes new records, status codes or extended signaling, *e.g.*, extended

error codes or other options. The new general `SVCB` DNS RR can be adapted for various future use cases to transport information within DNS. We have already shown the potential of the new `HTTPS` record to identify QUIC deployments. However, many specified parameters have not yet been used. A more comprehensive and diverse deployment in the future provides further possibilities for research and scans. This also extends to new protocol extensions supported by those parameters, such as Encrypted Client Hello for TLS.

Current drafts to use the record for further information besides HTTP endpoint information illustrate that these records open up a promising field for future research.

G3: Identification and Evaluation of QUIC Deployments and TLS Properties. QUIC, as a new protocol, offers a variety of new research opportunities. While this work provided tools to identify and evaluate deployments and initially analyzed the current state, long-term evaluation of the protocol, its deployment and diversity are important. Currently, a large diversity of libraries is visible. Whether this converges to a smaller set of libraries or remains diverse is an open question that needs to be answered in the future.

Furthermore, while QUIC relies on and combines a set of well-known and tested features (*e.g.*, TLS, 0-RTT, congestion control) the new composition and implementations are error-prone. While general concepts are adapted, the impact of specific differences should be evaluated. An example is the new approach to send acknowledgments within QUIC compared to the established mechanism in TCP. Allowing multiple acknowledgment ranges and more flexibility in the acknowledgment frequency can drastically influence traffic and the user experience. The diversity of implementations amplifies this. While they all follow the same standard, they might interpret and implement specifics differently. We showed this impact on active scans and in local tests [20]. However, the impact on the Internet ecosystem and traffic is still a wide and open research area.

Chapter A

Appendix

A.1 List of Advised Theses

The following theses have been advised during the work on this thesis. For each work, the main advisor is indicated by an asterisk (*).

A.1.1 QUIC

- M. Kempf, "Evaluation of the QUIC Spin Bit for RTT Estimation," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2019
- L. Keller, "Packet Pacing with the QUIC Protocol," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2020
- P. Buschmann, "Analyzing QUIC in the Wild," Advised by Johannes Zirngibl*, Patrick Sattler, Benedikt Jaeger and Juliane Aulbach, Master's Thesis, Technical University of Munich, 2020
- M. Mussner, "In Depth Analysis of QUIC's Lack of Kernel Optimizations," Advised by Benedikt Jaeger* and Johannes Zirngibl, Master's Thesis, Technical University of Munich, 2020
- S. Voit, "Bringing QUIC to High-speed Networks," Advised by Benedikt Jaeger* and Johannes Zirngibl, Master's Thesis, Technical University of Munich, 2021
- D. Hegedüs, "The First Year of QUIC v1 Deployment," Advised by Johannes Zirngibl*, Patrick Sattler, Benedikt Jaeger and Juliane Aulbach, Bachelor's Thesis, Technical University of Munich, 2021
- K. Ploch, "QUIC Performance on 10G Links," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2022

- M. Kutter, "Evaluation of Scalability and Limitations of HTTP/3," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2022
- M. Kempf, "Analysis of Performance Limitations in QUIC Implementations," Advised by Benedikt Jaeger* and Johannes Zirngibl, Master's Thesis, Technical University of Munich, 2022
- F. Gebauer, "Evaluating Different QUIC Scan Approaches," Advised by Johannes Zirngibl* and Patrick Sattler, Bachelor's Thesis, Technical University of Munich, 2022
- S. K. Guayana, "Analyzing the Effect of Transport Parameters on QUIC's Performance," Advised by Johannes Zirngibl* and Benedikt Jaeger, Bachelor's Thesis, Technical University of Munich, 2022
- L. Otting, "Improving QUIC with User Space Networking," Advised by Kilian Holzinger*, Benedikt Jaeger and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2023
- N. Gauder, "Performance Evaluation of Cryptography in QUIC," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2023
- R. Stadler, "QUICkly Reaching Maximum Throughput: A Comparative Evaluation of QUIC Implementations," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2023
- J. Späth, "QUIC Performance Improvements Using DPDK," Advised by Johannes Zirngibl*, Benedikt Jaeger and Kilian Holzinger, Master's Thesis, Technical University of Munich, 2023
- N. Beck, "Root Cause Analysis for Throughput Limitations of QUIC Connections," Advised by Simon Bauer* and Johannes Zirngibl, Master's Thesis, Technical University of Munich, 2023
- M. Buhl, "QUIC Kernel: an In-Kernel Port and Socket Abstraction Layer," Advised by Johannes Zirngibl*, Benedikt Jaeger and Kilian Holzinger, Interdisciplinary Project, Technical University of Munich, 2023
- M. Haid, "Impact of a TEE on QUIC Performance," Advised by Marcel Kempf* and Filip Rezabek and Johannes Zirngibl and Benedikt Jaeger, Bachelor's Thesis, Technical University of Munich, 2023

A.1.2 DNS

- D. Kreutzer, "Nameserver Rate Limits - Dynamic Adjustment of Scan Behavior," Advised by Johannes Zirngibl* and Johannes Naab, Bachelor's Thesis, Technical University of Munich, 2019
- L. Schedelbeck, "rDNS Leaks - Disclosing the Real Infrastructure of Shadowed Services," Advised by Johannes Zirngibl* and Patrick Sattler, Bachelor's Thesis, Technical University of Munich, 2019

- R. B. Reif, "Analysis of EDNS Client-Subnet Load Balancing," Advised by Patrick Sattler* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2019
- C. Wahl, "Analyzing the Stability and Expressiveness of Large-Scale DNS Scans," Advised by Patrick Sattler*, Johannes Zirngibl and Juliane Aulbach, Master's Thesis, Technical University of Munich, 2020
- Z. Sonkaya, "Development of an Efficient Large Scale DNS Scanning Pipeline," Advised by Patrick Sattler* and Johannes Zirngibl, Interdisciplinary Project, Technical University of Munich, 2021
- S. Deusch, "Analyzing the Effect of Domain Parking on DNS Based Research," Advised by Johannes Zirngibl*, Patrick Sattler and Juliane Aulbach, Bachelor's Thesis, Technical University of Munich, 2021
- P. Großmann, "Extended Usage Analysis of EDNS Client Subnet," Advised by Patrick Sattler*, Johannes Zirngibl and Lion Steger, Bachelor's Thesis, Technical University of Munich, 2022
- C. B. Dietze, "Tracking the Lifetime of Domains," Advised by Johannes Zirngibl* and Patrick Sattler, Master's Thesis, Technical University of Munich, 2023

A.1.3 IP and BGP

- R. B. Reif, "Detecting BGP Hijacking in Real Time," Advised by Patrick Sattler* and Johannes Zirngibl, Interdisciplinary Project, Technical University of Munich, 2020
- P. Henschke, "Analyzing BGP as a Graph," Advised by Johannes Zirngibl*, Patrick Sattler and Juliane Aulbach, Bachelor's Thesis, Technical University of Munich, 2021
- R. Schmid, "ROV + IRR: Are Authorized Routes Registered?" Advised by Johannes Zirngibl*, Patrick Sattler and Juliane Aulbach, Bachelor's Thesis, Technical University of Munich, 2021
- K. Ilse, "IPv6 Deployment Analysis using BGP Announcements," Advised by Patrick Sattler*, Johannes Zirngibl and Juliane Aulbach, Bachelor's Thesis, Technical University of Munich, 2021
- L. Steger, "Revisiting IPv6 Hitlists," Advised by Johannes Zirngibl*, Patrick Sattler and Juliane Aulbach, Master's Thesis, Technical University of Munich, 2021
- T. Wothge, "Industrial Control Systems (ICS) Protocol Detection," Advised by Patrick Sattler*, Lars Wüstrich and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2021
- L. Kuang, "Target Generation for IPv6 Hitlists," Advised by Lion Steger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2022
- Z. Lu, "Structural Analysis of the Great Firewall of China," Advised by Lion Steger* and Johannes Zirngibl, Master's Thesis, Technical University of Munich, 2022

- I. Varas, "Autonomous System Models using BGP Data and GNNs," Advised by Max Helm*, Benedikt Jaeger, Johannes Zirngibl and Patrick Sattler, Bachelor's Thesis, Technical University of Munich, 2023
- F. Bauernschmitt, "Evalution of Network Categorization Strategies," Advised by Lion Steger*, Johannes Zirngibl and Patrick Sattler, Bachelor's Thesis, Technical University of Munich, 2022
- B. R. Schaschko, "Inferring AS Links from a Tier 1 Dataset," Advised by Johannes Zirngibl* and Patrick Sattler, Bachelor's Thesis, Technical University of Munich, 2023

A.1.4 Other Theses

- N. Buchner, "Influence of Network Conditions on PTP Accuracy," Advised by Max Helm*, Henning Stubbe and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2019
- C. Kilb, "Blocklists: What is blocked and why?" Advised by Johannes Zirngibl*, Patrick Sattler and Markus Sosnowski, Interdisciplinary Project, Technical University of Munich, 2020
- S. H. Kappes, "An Analysis of the Development and Early Deployment of Encrypted SNI," Advised by Johannes Zirngibl*, Max Helm and Patrick Sattler, Bachelor's Thesis, Technical University of Munich, 2020
- R. Dillitz, "Uncovering PTP Master Clocks in the Wild," Advised by Johannes Zirngibl*, Max Helm and Henning Stubbe, Bachelor's Thesis, Technical University of Munich, 2020
- J. von der Heidt, "Analyzing PTP Master Clocks in the Wild," Advised by Johannes Zirngibl*, Max Helm and Henning Stubbe, Bachelor's Thesis, Technical University of Munich, 2020
- B. Riegel, "Assessing Link Utilization From Passive Datasets," Advised by Simon Bauer* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2021
- L. Lehle, "Efficient Processing of Large Network Captures," Advised by Lars Wüstrich*, Johannes Zirngibl and Christian Lübben, Bachelor's Thesis, Technical University of Munich, 2021
- F. Myhsok, "Blocklists: Who is blocked?" Advised by Johannes Zirngibl*, Patrick Sattler and Markus Sosnowski, Bachelor's Thesis, Technical University of Munich, 2021
- Y. Wibowo, "Analysis of Blocklisted TLS Servers," Advised by Johannes Zirngibl* and Patrick Sattler, Bachelor's Thesis, Technical University of Munich, 2022
- R. Dillitz, "Transformation and Evaluation of TLS Behavior Graphs," Advised by Johannes Zirngibl*, Benedikt Jaeger and Markus Sosnowski, Master's Thesis, Technical University of Munich, 2022

- T. Gräbner, "Setup and Deployment of a Large Scale Certificate Scan Database," Advised by Patrick Sattler* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2022
- M. S. Shaukat, "Measuring the Impact of Transport Layer Protocols and Their Configuration on the Performance of Connections," Advised by Simon Bauer*, Patrick Sattler and Johannes Zirngibl, Master's Thesis, Technical University of Munich, 2022
- C. B. Dietze, "Setup and Deployment of a Resilient Internet Scanning Infrastructure," Advised by Patrick Sattler* and Johannes Zirngibl, Interdisciplinary Project, Technical University of Munich, 2022
- D. Weissmann, "The Impact of iCloud Private Relay on Networks," Advised by Patrick Sattler* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2023
- T. Wothge, "Egress Node Behavior in iCloud Private Relay," Advised by Patrick Sattler*, Lars Wüstrich, Johannes Zirngibl and Lion Steger, Interdisciplinary Project, Technical University of Munich, 2023
- T. Zierl, "Evaluating Domain Presence in Certificate Transparency Logs," Advised by Patrick Sattler* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2023
- L. Pydde, "TLS Certificate Usage Evaluation," Advised by Patrick Sattler* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2023
- T. Wasner, "Continuous Monitoring and Quality Assessment of Internet-wide Scans," Advised by Patrick Sattler and Johannes Zirngibl, Interdisciplinary Project, Technical University of Munich, 2023
- M. Kirstein, "Happy Eyeballs: A Comprehensive Analysis of the Deployment and Configuration Across Various Versions and Implementations," Advised by Patrick Sattler*, Johannes Zirngibl and Lars Wüstrich, Bachelor's Thesis, Technical University of Munich, 2024

A.2 List of Figures

A.3 List of Tables

A.4 List of Acronyms

ALPN Application-Layer Protocol Negotiation

ALT-SVC Alternative Service

APD Aliased Prefix Detection

AS Autonomous System

BGP Border Gateway Protocol

ccTLD country-code TLD

CDN Content Delivery Network

CPE Customer-premises equipment

CT Certificate Transparency

CT log Certificate Transparency Log

CZDS Centralized Zone Data Service

DNS Domain Name System

DNS RR Domain Name System Resource Record

ECH Encrypted Client Hello

eTLD effective TLD

GFW Great Firewall of China

GINO Global INternet Observatory

gTLD generic TLD

HRP highly responsive prefix

HSTS HTTP Strict Transport Security

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

ICMP Internet Control Message Protocol

IETF Internet Engineering Task Force

IID IPv6 Interface ID

IoT Internet of Things

ISP Internet Service Provider

MSS Maximum Segment Size

NTP Network Time Protocol

OSI Open Systems Interconnection

OUI Organizationally Unique Identifier

PMTU Path Maximum Transmission Unit

POP Point of Presence

PPC Pay-Per-Click

PPR Pay-Per-Redirect

RIR Regional Internet Registry

SNI Server Name Indication

TBT Too Big Trick

TCP Transmission Control Protocol

TGA Target Generation Algorithm

TLD Top Level Domain

TLS Transport Layer Security

UDP User Datagram Protocol

VN Version Negotiation

Bibliography

[1] J. Iyengar and M. Thomson, *QUIC: A UDP-Based Multiplexed and Secure Transport*, RFC 9000, May 2021. DOI: 10.17487/RFC9000. [Online]. Available: https://rfc-editor.org/rfc/rfc9000.txt.

[2] Y. Liu, Y. Ma, Q. D. Coninck, O. Bonaventure, C. Huitema, and M. Kühlewind, "Multipath Extension for QUIC," Internet Engineering Task Force, Internet-Draft draft-ietf-quic-multipath-06, Oct. 2023, Work in Progress, 33 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-quic-multipath/06/.

[3] D. Schinazi, *Proxying UDP in HTTP*, RFC 9298, Aug. 2022. DOI: 10.17487/RFC9298. [Online]. Available: https://www.rfc-editor.org/info/rfc9298.

[4] T. Pauly, D. Schinazi, A. Chernyakhovsky, M. Kühlewind, and M. Westerlund, *Proxying IP in HTTP*, RFC 9484, Oct. 2023. DOI: 10.17487/RFC9484. [Online]. Available: https://www.rfc-editor.org/info/rfc9484.

[5] J. Gruessing and S. Dawkins, "Media Over QUIC - Use Cases and Requirements for Media Transport Protocol Design," Internet Engineering Task Force, Internet-Draft draft-ietf-moq-requirements-02, Sep. 2023, Work in Progress, 27 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-moq-requirements/02/.

[6] B. Hinden and D. S. E. Deering, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460, Dec. 1998. DOI: 10.17487/RFC2460. [Online]. Available: https://www.rfc-editor.org/info/rfc2460.

[7] B. M. Schwartz, M. Bishop, and E. Nygren, *Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records)*, RFC 9460, Nov. 2023. DOI: 10.17487/RFC9460. [Online]. Available: https://www.rfc-editor.org/info/rfc9460.

[8] O. Gasser, Q. Scheitle, P. Foremski, Q. Lone, M. Korczynski, S. D. Strowes, L. Hendriks, and G. Carle, "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists," in *Proc. ACM Internet Measurement Conference (IMC)*, 2018. DOI: 10.1145/3278532.3278564.

[9] J. Zirngibl, P. Buschmann, P. Sattler, B. Jaeger, J. Aulbach, and G. Carle, "It's over 9000: Analyzing early QUIC Deployments with the Standardization on the Horizon," in *Proc. ACM Internet Measurement Conference (IMC)*, 2021. DOI: 10.1145/3487552.3487826.

[10] J. Zirngibl, S. Deusch, P. Sattler, J. Aulbach, G. Carle, and M. Jonker, "Domain Parking: Largely Present, Rarely Considered!" In *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2022.

[11] J. Zirngibl, L. Steger, P. Sattler, O. Gasser, and G. Carle, "Rusty Clusters? Dusting an IPv6 Research Foundation," in *Proc. ACM Internet Measurement Conference (IMC)*, 2022. DOI: 10.1145/3517745.3561440.

[12] J. Zirngibl, P. Sattler, and G. Carle, "A First Look at SVCB and HTTPS DNS Resource Records in the Wild," in *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2023. DOI: 10.1109/EuroSPW59978.2023.00058.

[13] L. Steger, L. Kuang, J. Zirngibl, G. Carle, and O. Gasser, "Target Acquired? Evaluating Target Generation Algorithms for IPv6," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2023. DOI: 10.23919/TMA58422.2023.10199073.

[14] P. Sattler, J. Zirngibl, M. Jonker, O. Gasser, G. Carle, and R. Holz, "Packed to the Brim: Investigating the Impact of Highly Responsive Prefixes on Internet-wide Measurement Campaigns," *Proc. ACM Netw.*, 2023. DOI: 10.1145/3629146.

[15] J. Zirngibl, F. Gebauer, P. Sattler, M. Sosnowski, and G. Carle, "QUIC Hunter: Finding QUIC Deployments and Identifying Server Libraries Across the Internet," in *Proc. Passive and Active Measurement (PAM)*, 2024. DOI: 10.1007/978-3-031-56252-5_13.

[16] F. Franzen, L. Steger, J. Zirngibl, and P. Sattler, "Looking for Honey Once Again: Detecting RDP and SMB Honeypots on the Internet," in *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2022. DOI: 10.1109/EuroSPW55150.2022.00033.

[17] P. Sattler, J. Aulbach, J. Zirngibl, and G. Carle, "Towards a Tectonic Traffic Shift? Investigating Apple's New Relay Network," in *Proc. ACM Internet Measurement Conference (IMC)*, 2022. DOI: 10.1145/3517745.3561426.

[18] M. Sosnowski, J. Zirngibl, P. Sattler, G. Carle, C. Grohnfeldt, M. Russo, and D. Sgandurra, "Active TLS Stack Fingerprinting: Characterizing TLS Server Deployments at Scale," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2022.

[19] M. Sosnowski, J. Zirngibl, P. Sattler, and G. Carle, "DissecTLS: A Scalable Active Scanner for TLS Server Configurations, Capabilities, and TLS Fingerprinting," in *Proc. Passive and Active Measurement (PAM)*, 2023. DOI: 10.1007/978-3-031-28486-1_6.

[20] B. Jaeger, J. Zirngibl, M. Kempf, K. Ploch, and G. Carle, "QUIC on the Highway: Evaluating Performance on High-Rate Links," in *IFIP Networking Conference (IFIP Networking)*, 2023. DOI: 10.23919/IFIPNetworking57963.2023.10186365.

[21] J. Naab, P. Sattler, J. Zirngibl, S. Günther, and G. Carle, "Gotta Query 'Em All, Again! Repeatable Name Resolution with Full Dependency Provenance," in *Proceedings of the Applied Networking Research Workshop*, 2023. DOI: 10.1145/3606464.3606478.

[22] S. Bauer, P. Sattler, J. Zirngibl, C. Schwarzenberg, and G. Carle, "Evaluating the Benefits: Quantifying the Effects of TCP Options, QUIC, and CDNs on Throughput," in *Proceedings of the Applied Networking Research Workshop*, 2023. DOI: 10.1145/3606464.3606474.

[23] M. Sosnowski, J. Zirngibl, P. Sattler, G. Carle, C. Grohnfeldt, M. Russo, and D. Sgandurra, "EFACTLS: Effective Active TLS Fingerprinting for Large-scale Server Deployment Characterization," *IEEE Transactions on Network and Service Management*, 2024. DOI: 10.1109/TNSM.2024.3364526.

[24] M. Sosnowski, P. Sattler, J. Zirngibl, T. Betzer, and G. Carle, "Propagating Threat Scores With a TLS Ecosystem Graph Model Derived by Active Measurements," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, May 2024.

[25] M. Kempf, N. Gauder, B. Jaeger, J. Zirngibl, and G. Carle, "A Quantum of QUIC: Dissecting Cryptography with Post-Quantum Insights," in *IFIP Networking Conference (IFIP Networking)*, 2024.

[26] M. Sosnowski, J. Zirngibl, P. Sattler, J. Aulbach, J. Lang, and G. Carle, "An Internet-wide View on HTTPS Certificate Revocations: Observing the Revival of CRLs via Active TLS Scans," in *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2024.

[27] M. Kempf, B. Jaeger, J. Zirngibl, K. Ploch, and G. Carle, "QUIC on the Fast Lane: Extending Performance Evaluations on High-rate Links," *Computer Communications*, vol. 223, 2024. DOI: https://doi.org/10.1016/j.comcom.2024.04.038.

[28] J. Mücke, M. Nawrocki, R. Hiesgen, P. Sattler, J. Zirngibl, G. Carle, T. C. Schmidt, and M. Wählisch, *Waiting for QUIC: On the Opportunities of Passive Measurements to Understand QUIC Deployments*, 2022. [Online]. Available: https://arxiv.org/abs/2209.00965.

[29] O. Gasser. "ZMapv6: Internet Scanner with IPv6 capabilities." (2022), [Online]. Available: https://github.com/tumi8/zmap (visited on 01/05/2024).

[30] O. Gasser, M. Sosnowski, P. Sattler, and J. Zirngibl. "Goscanner." (2023), [Online]. Available: https://github.com/tumi8/goscanner (visited on 03/15/2023).

[31] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide Scanning and Its Security Applications," in *Proc. USENIX Security Symposium*, 2013, ISBN: 9781931971034.

[32] J. Amann, O. Gasser, Q. Scheitle, L. Brent, G. Carle, and R. Holz, "Mission Accomplished? HTTPS Security after Diginotar," in *Proc. ACM Internet Measurement Conference (IMC)*, 2017. DOI: 10.1145/3131365.3131401.

[33] M. Seemann. "quic-go: A QUIC implementation in pure Go." (2022), [Online]. Available: https://github.com/lucas-clemente/quic-go (visited on 11/04/2022).

[34] P. Buschmann, "Analyzing QUIC in the Wild," Advised by Johannes Zirngibl*, Patrick Sattler, Benedikt Jaeger and Juliane Aulbach, Master's Thesis, Technical University of Munich, 2020.

[35] O. Gasser, Q. Scheitle, S. Gebhard, and G. Carle, "Scanning the IPv6 Internet: Towards a Comprehensive Hitlist," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2016.

[36] Majestic. "The Majestic Million." (2022), [Online]. Available: https://majestic.com/reports/majestic-million/ (visited on 01/05/2024).

[37] Cisco. "Umbrella Top 1M List." (2022), [Online]. Available: https://umbrella.cisco.com/blog/cisco-umbrella-1-million (visited on 01/05/2024).

[38] ICANN. "Centralized Zone Data Service." (2021), [Online]. Available: https://czds.icann.org/home (visited on 01/05/2024).

[39] D. Dittrich, E. Kenneally, *et al.*, "The Menlo Report: Ethical principles guiding information and communication technology research," *US Department of Homeland Security*, 2012.

[40] C. Partridge and M. Allman, "Addressing Ethical Considerations in Network Measurement Papers," *Communications of the ACM*, vol. 59, no. 10, Oct. 2016.

[41] R. Holz, J. Hiller, J. Amann, A. Razaghpanah, T. Jost, N. Vallina-Rodriguez, and O. Hohlfeld, "Tracking the Deployment of TLS 1.3 on the Web: A Story of Experimentation and Centralization," *ACM SIGCOMM Computer Communication Review*, 2020. DOI: 10.1145/3411740.3411742.

[42] J. Rüth, I. Poese, C. Dietzel, and O. Hohlfeld, "A First Look at QUIC in the Wild," in *Proc. Passive and Active Measurement (PAM)*, 2018. DOI: 10.1007/978-3-319-76481-8_19.

[43] T. Fiebig, K. Borgolte, S. Hao, C. Kruegel, and G. Vigna, "Something from Nothing (There): Collecting Global IPv6 Datasets from DNS," in *Proc. Passive and Active Measurement (PAM)*, 2017. DOI: 10.1007/978-3-319-54328-4_3.

[44] P. Foremski, D. Plonka, and A. Berger, "Entropy/IP: Uncovering Structure in IPv6 Addresses," in *Proc. ACM Internet Measurement Conference (IMC)*, 2016. DOI: 10.1145/2987443.2987445.

[45] A. Murdock, F. Li, P. Bramsen, Z. Durumeric, and V. Paxson, "Target Generation for Internet-Wide IPv6 Scanning," in *Proc. ACM Internet Measurement Conference (IMC)*, London, United Kingdom, 2017.

[46] E. C. Rye and R. Beverly, "Discovering the IPv6 Network Periphery," in *Proc. Passive and Active Measurement (PAM)*, 2020. DOI: 10.1007/978-3-030-44081-7_1.

[47] E. Rye, R. Beverly, and K. C. Claffy, "Follow the Scent: Defeating IPv6 Prefix Rotation Privacy," in *Proc. ACM Internet Measurement Conference (IMC)*, 2021. DOI: 10.1145/3487552.3487829.

[48] E. Rye and D. Levin, "IPv6 Hitlists at Scale: Be Careful What You Wish For," in *Proc. ACM SIGCOMM*, 2023. DOI: 10.1145/3603269.3604829.

[49] F. Gont and T. Chown, *Network Reconnaissance in IPv6 Networks*, RFC 7707, Mar. 2016. DOI: 10.17487/RFC7707. [Online]. Available: https://www.rfc-editor.org/info/rfc7707.

[50] J. Ullrich, P. Kieseberg, K. Krombholz, and E. Weippl, "On Reconnaissance with IPv6: A Pattern-Based Scanning Approach," in *10th International Conference on Availability, Reliability and Security*, 2015. DOI: 10.1109/ares.2015.48.

[51] T. Cui, G. Gou, and G. Xiong, "6GCVAE: Gated Convolutional Variational Autoencoder for IPv6 Target Generation," in *Advances in Knowledge Discovery and Data Mining*, 2020. DOI: 10.1007/978-3-030-47426-3_47.

[52] T. Cui, G. Gou, G. Xiong, C. Liu, P. Fu, and Z. Li, "6GAN: IPv6 Multi-Pattern Target Generation via Generative Adversarial Nets with Reinforcement Learning," in *Proc. IEEE Int. Conference on Computer Communications (INFOCOM)*, 2021. DOI: 10.1109/INFOCOM42981.2021.9488912.

[53] T. Cui, G. Xiong, G. Gou, J. Shi, and W. Xia, "6VecLM: Language Modeling in Vector Space for IPv6 Target Generation," in *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track*, 2021. DOI: 10.1007/978-3-030-67667-4_12.

[54] B. Hou, Z. Cai, K. Wu, J. Su, and Y. Xiong, "6Hit: A Reinforcement Learning-based Approach to Target Generation for Internet-wide IPv6 Scanning," in *Proc. IEEE Int. Conference on Computer Communications (INFOCOM)*, 2021. DOI: 10.1109/INFOCOM42981.2021.9488794.

[55] Z. Liu, Y. Xiong, X. Liu, W. Xie, and P. Zhu, "6Tree: Efficient dynamic discovery of active addresses in the IPv6 address space," *Computer Networks*, 2019. DOI: 10.1016/j.comnet.2019.03.010.

[56] T. Yang, B. Hou, Z. Cai, K. Wu, T. Zhou, and C. Wang, "6Graph: A graph-theoretic approach to address pattern mining for Internet-wide IPv6 scanning," *Computer Networks*, 2022. DOI: 10.1016/j.comnet.2021.108666.

[57] G. Song, J. Yang, Z. Wang, L. He, J. Lin, L. Pan, C. Duan, and X. Quan, "DET: Enabling Efficient Probing of IPv6 Active Addresses," *IEEE/ACM Transactions on Networking*, 2022. DOI: 10.1109/TNET.2022.3145040.

[58] T. Yang, Z. Cai, B. Hou, and T. Zhou, "6Forest: An Ensemble Learning-based Approach to Target Generation for Internet-wide IPv6 Scanning," in *Proc. IEEE Int. Conference on Computer Communications (INFOCOM)*, 2022. DOI: 10.1109/infocom48880.2022.9796925.

[59] B. Hou, Z. Cai, K. Wu, T. Yang, and T. Zhou, "6Scan: A High-Efficiency Dynamic Internet-Wide IPv6 Scanner With Regional Encoding," *IEEE/ACM Transactions on Networking*, 2023. DOI: 10.1109/tnet.2023.3233953.

[60] X. Li, B. Liu, X. Zheng, H. Duan, Q. Li, and Y. Huang, "Fast IPv6 Network Periphery Discovery and Security Implications," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2021. DOI: 10.1109/dsn48987.2021.00025.

[61] M. Luckie, R. Beverly, W. Brinkmeyer, and K. C. Claffy, "Speedtrap: Internet-Scale IPv6 Alias Resolution," in *Proc. ACM Internet Measurement Conference (IMC)*, 2013. DOI: 10.1145/2504730.2504759.

[62] R. Beverly, W. Brinkmeyer, M. Luckie, and J. P. Rohrer, "IPv6 Alias Resolution via Induced Fragmentation," in *Proc. Passive and Active Measurement (PAM)*, 2013. DOI: 10.1007/978-3-642-36516-4_16.

[63] A. Marder, "APPLE: Alias Pruning by Path Length Estimation," in *Proc. Passive and Active Measurement (PAM)*, 2020. DOI: 10.1007/978-3-030-44081-7_15.

[64] K. Vermeulen, B. Ljuma, V. Addanki, M. Gouel, O. Fourmaux, T. Friedman, and R. Rejaie, "Alias Resolution Based on ICMP Rate Limiting," in *Proc. Passive and Active Measurement (PAM)*, 2020. DOI: 10.1007/978-3-030-44081-7_14.

[65] R. Padmanabhan, Z. Li, D. Levin, and N. Spring, "UAv6: Alias Resolution in IPv6 Using Unused Addresses," in *Proc. Passive and Active Measurement (PAM)*, 2015. DOI: 10.1007/978-3-319-15509-8_11.

[66] Q. Scheitle, O. Gasser, M. Rouhi, and G. Carle, "Large-Scale Classification of IPv6-IPv4 Siblings with Variable Clock Skew," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2017. DOI: 10.23919/TMA.2017.8002901.

[67] O. Farnan, A. Darer, and J. Wright, "Poisoning the Well: Exploring the Great Firewall's Poisoned DNS Responses," in *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, 2016. DOI: 10.1145/2994620.2994636.

[68] Anonymous, "Towards a Comprehensive Picture of the Great Firewall's DNS Censorship," in *4th USENIX Workshop on Free and Open Communications on the Internet (FOCI 14)*, 2014.

[69] Anonymous, A. A. Niaki, N. P. Hoang, P. Gill, and A. Houmansadr, "Triplet Censors: Demystifying Great Firewall's DNS Censorship Behavior," in *10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*, 2020.

[70] F. Zhang, C. Lu, B. Liu, H. Duan, and Y. Liu, "Measuring the Practical Effect of DNS Root Server Instances: A China-Wide Case Study," in *Proc. Passive and Active Measurement (PAM)*, 2022. DOI: 10.1007/978-3-030-98785-5_11.

[71] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras, "A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements," *IEEE Journal on Selected Areas in Communications*, 2016. DOI: 10.1109/JSAC.2016.2558918.

[72] Rapid7 Labs. "Forward DNS (FDNS)." (2022), [Online]. Available: https://opendata.rapid7.com/sonar.fdns_v2/ (visited on 02/08/2022).

[73] M. Trevisan, D. Giordano, I. Drago, and A. S. Khatouni, "Measuring HTTP/3: Adoption and Performance," in *19th Mediterranean Communication and Computer Networking Conference (MedComNet)*, 2021. DOI: 10.1109/MedComNet52149.2021.9501274.

[74] Ralf Weber. "DNS HTTP RR: a bright new future?" (2021), [Online]. Available: https://indico.dns-oarc.net/event/37/contributions/810/attachments/784/1413/dns-https-rr-final.pdf (visited on 01/25/2024).

[75] C. Aguilar-Melchor, T. Bailleux, J. Goertzen, D. Joseph, and D. Stebila, *TurboTLS: TLS connection establishment with 1 less round trip*, 2023. DOI: 10.48550/ARXIV.2302.05311.

[76] Z. Chai, A. Ghafari, and A. Houmansadr, "On the Importance of Encrypted-SNI (ESNI) to Censorship Circumvention," in *9th USENIX Workshop on Free and Open Communications on the Internet, FOCI*, 2019.

[77] Z. Tsiatsikas, G. Karopoulos, and G. Kambourakis, "Measuring the Adoption of TLS Encrypted Client Hello Extension and Its Forebear in the Wild," in *Computer Security. ESORICS 2022 International Workshops*, 2023. DOI: 10.1007/978-3-031-25460-4_10.

[78] N. P. Hoang, M. Polychronakis, and P. Gill, "Measuring the Accessibility of Domain Name Encryption and Its Impact on Internet Filtering," in *Proc. Passive and Active Measurement (PAM)*, 2022. DOI: 10.1007/978-3-030-98785-5_23.

[79] K. Bhargavan, V. Cheval, and C. Wood, "A Symbolic Analysis of Privacy for TLS 1.3 with Encrypted Client Hello," in *Proc. ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022. DOI: 10.1145/3548606.3559360.

[80] D. Shamsimukhametov, A. Kurapov, M. Liubogoshchev, and E. Khorov, "Is Encrypted ClientHello a Challenge for Traffic Classification?" *IEEE Access*, 2022. DOI: 10.1109/ACCESS.2022.3191431.

[81] M. Kosek, T. V. Doan, S. Huber, and V. Bajpai, "Measuring DNS over TCP in the Era of Increasing DNS Response Sizes: A View from the Edge," *ACM SIGCOMM Computer Communication Review*, 2022. DOI: 10.1145/3544912.3544918.

[82] T. Böttger, F. Cuadrado, G. Antichi, E. L. Fernandes, G. Tyson, I. Castro, and S. Uhlig, "An Empirical Study of the Cost of DNS-over-HTTPS," in *Proc. ACM Internet Measurement Conference (IMC)*, 2019. DOI: 10.1145/3355369.3355575.

[83] M. Kosek, L. Schumann, R. Marx, T. V. Doan, and V. Bajpai, "DNS Privacy with Speed? Evaluating DNS over QUIC and Its Impact on Web Performance," in *Proc. ACM Internet Measurement Conference (IMC)*, 2022. DOI: 10.1145/3517745.3561445.

[84] T. V. Doan, I. Tsareva, and V. Bajpai, "Measuring DNS over TLS from the Edge: Adoption, Reliability, and Response Times," in *Proc. Passive and Active Measurement (PAM)*, 2021. DOI: 10.1007/978-3-030-72582-2_12.

[85] S. Alrwais, K. Yuan, E. Alowaisheq, Z. Li, and X. Wang, "Understanding the Dark Side of Domain Parking," in *Proc. USENIX Security Symposium*, 2014.

[86] T. Vissers, W. Joosen, and N. Nikiforakis, "Parking Sensors: Analyzing and Detecting Parked Domains," in *Proc. Network and Distributed System Security Symposium (NDSS)*, 2015. DOI: 10.14722/ndss.2015.23053.

[87] M. Kührer, C. Rossow, and T. Holz, "Paint It Black: Evaluating the Effectiveness of Malware Blacklists," in *Research in Attacks, Intrusions and Defenses*, 2014. DOI: 10.1007/978-3-319-11379-1_1.

[88] T. Halvorson, M. F. Der, I. Foster, S. Savage, L. K. Saul, and G. M. Voelker, "From .Academy to .Zone: An Analysis of the New TLD Land Rush," in *Proc. ACM Internet Measurement Conference (IMC)*, 2015. DOI: 10.1145/2815675.2815696.

[89] T. Halvorson, K. Levchenko, S. Savage, and G. M. Voelker, "XXXtortion? Inferring Registration Intent in the .XXX TLD," in *Proceedings of the 23rd International Conference on World Wide Web*, 2014. DOI: 10.1145/2566486.2567995.

[90] T. Halvorson, J. Szurdi, G. Maier, M. Felegyhazi, C. Kreibich, N. Weaver, K. Levchenko, and V. Paxson, "The BIZ Top-Level Domain: Ten Years Later," in *Proc. Passive and Active Measurement (PAM)*, 2012. DOI: 10.1007/978-3-642-28537-0_22.

[91] T. Alexander, "Domain Name Registrars: Are They Part of the Domain Name Fraud Problem?" In *Proceedings of the 3rd Annual Conference on Information Security Curriculum Development*, 2006. DOI: 10.1145/1231047.1231071.

[92] T. Lauinger, A. Chaabane, A. S. Buyukkayhan, K. Onarlioglu, and W. Robertson, "Game of Registrars: An Empirical Analysis of Post-Expiration Domain Name Takeovers," in *Proc. USENIX Security Symposium*, 2017.

[93] L. Zembruzki, R. Sommese, L. Z. Granville, A. Selle Jacobs, M. Jonker, and G. C. M. Moura, "Hosting Industry Centralization and Consolidation," in *IEEE/IFIP Network Operations and Management Symposium*, 2022. DOI: 10.1109/NOMS54207.2022.9789881.

[94] M. Fischlin and F. Günther, "Multi-Stage Key Exchange and the Case of Google's QUIC Protocol," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014. DOI: 10.1145/2660267.2660308.

[95] R. Lychev, S. Jero, A. Boldyreva, and C. Nita-Rotaru, "How Secure and Quick is QUIC? Provable Security and Performance Analyses," in *Proc. IEEE Symposium on Security and Privacy (S&P)*, 2015. DOI: 10.1109/SP.2015.21.

[96] M. Seemann and J. Iyengar, "Automating QUIC Interoperability Testing," in *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, 2020. DOI: 10.1145/3405796.3405826.

[97] M. Piraux, Q. De Coninck, and O. Bonaventure, "Observing the Evolution of QUIC Implementations," in *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, 2018. DOI: 10.1145/3284850.3284852.

[98] R. Marx, J. Herbots, W. Lamotte, and P. Quax, "Same Standards, Different Decisions: A Study of QUIC and HTTP/3 Implementation Diversity," in *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, 2020. DOI: 10.1145/3405796.3405828.

[99] A. M. Kakhki, S. Jero, D. Choffnes, C. Nita-Rotaru, and A. Mislove, "Taking a Long Look at QUIC: An Approach for Rigorous Evaluation of Rapidly Evolving Transport Protocols," in *Proc. ACM Internet Measurement Conference (IMC)*, 2017. DOI: 10.1145/3131365.3131368.

[100] K. Nepomuceno, I. N. d. Oliveira, R. R. Aschoff, D. Bezerra, M. S. Ito, W. Melo, D. Sadok, and G. Szabó, "QUIC and TCP: A Performance Evaluation," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018. DOI: 10.1109/ISCC.2018.8538687.

[101] K. Wolsing, J. Rüth, K. Wehrle, and O. Hohlfeld, "A Performance Perspective on Web Optimized Protocol Stacks: TCP+TLS+HTTP/2 vs. QUIC," in *Proceedings of the Applied Networking Research Workshop*, 2019. DOI: 10.1145/3340301.3341123.

[102] X. Yang, L. Eggert, J. Ott, S. Uhlig, Z. Sun, and G. Antichi, "Making QUIC Quicker With NIC Offload," in *Proc. of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, 2020. DOI: 10.1145/3405796.3405827.

[103] E. Volodina and E. P. Rathgeb, "Impact of ACK Scaling Policies on QUIC Performance," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, 2021. DOI: 10.1109/LCN52139.2021.9524947.

[104] T. Shreedhar, R. Panda, S. Podanev, and V. Bajpai, "Evaluating QUIC Performance Over Web, Cloud Storage, and Video Workloads," *IEEE Transactions on Network and Service Management*, 2022. DOI: 10.1109/TNSM.2021.3134562.

[105] C. Sander, I. Kunze, and K. Wehlre, "Analyzing the Influence of Resource Prioritization on HTTP/3 HOL Blocking and Performance," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2022.

[106] N. Tyunyayev, M. Piraux, O. Bonaventure, and T. Barbette, "A High-Speed QUIC Implementation," in *Proc. of the 3rd International CoNEXT Student Workshop*, 2022. DOI: 10.1145/3565477.3569154.

[107] J. Roskind. "Experimenting with QUIC." (2013), [Online]. Available: https://blog.chromium.org/2013/06/experimenting-with-quic.html (visited on 01/25/2024).

[108] P. Kotzias, A. Razaghpanah, J. Amann, K. G. Paterson, N. Vallina-Rodriguez, and J. Caballero, "Coming of Age: A Longitudinal Study of TLS Deployment," in *Proc. ACM Internet Measurement Conference (IMC)*, 2018. DOI: 10.1145/3278532.3278568.

[109] P. Gigis, M. Calder, L. Manassakis, G. Nomikos, V. Kotronis, X. Dimitropoulos, E. Katz-Bassett, and G. Smaragdakis, "Seven Years in the Life of Hypergiants' off-Nets," in *Proc. ACM SIGCOMM*, 2021. DOI: 10.1145/3452296.3472928.

[110] J. Althouse, A. Smart, R. Nunnally Jr., and M. Brady. "Easily Identify Malicious Servers on the Internet with JARM." (2020), [Online]. Available: https://engineering.salesforce.com/

easily-identify-malicious-servers-on-the-internet-with-jarm-e095edac525a (visited on 01/25/2024).

[111] G. F. Lyon, *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. 2009, ISBN: 978-0-9799587-1-7.

[112] Z. Shamsi, A. Nandwani, D. Leonard, and D. Loguinov, "Hershel: Single-Packet OS Fingerprinting," *IEEE/ACM Transactions on Networking*, 2016. DOI: 10.1145/2637364.2591972.

[113] M. Nawrocki, P. F. Tehrani, R. Hiesgen, J. Mücke, T. C. Schmidt, and M. Wählisch, "On the Interplay between TLS Certificates and QUIC Performance," in *Proc. ACM Int. Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2022. DOI: 10.1145/3555050.3569123.

[114] M. Nawrocki, R. Hiesgen, T. C. Schmidt, and M. Wählisch, "QUICsand: Quantifying QUIC Reconnaissance Scans and DoS Flooding Events," in *Proc. ACM Internet Measurement Conference (IMC)*, 2021. DOI: 10.1145/3487552.3487840.

[115] Y. Gbur and F. Tschorsch, "QUICforge: Client-side Request Forgery in QUIC," in *Proc. Network and Distributed System Security Symposium (NDSS)*, 2023. DOI: 10.14722/ndss.2023.23072.

[116] Y. Nosyk, M. Korczyński, Q. Lone, M. Skwarek, B. Jonglez, and A. Duda, "The Closed Resolver Project: Measuring the Deployment of Inbound Source Address Validation," *IEEE/ACM Transactions on Networking*, 2023. DOI: 10.1109/TNET.2023.3257413.

[117] N. Liu, C. Jia, B. Hou, C. Hou, Y. Chen, and Z. Cai, "6Search: A reinforcement learning-based traceroute approach for efficient IPv6 topology discovery," *Computer Networks*, 2023. DOI: 10.1016/j.comnet.2023.109987.

[118] A. Hsu, F. Li, and P. Pearce, "Fiat Lux: Illuminating IPv6 Apportionment with Different Datasets," *Proc. ACM Meas. Anal. Comput. Syst.*, 2023. DOI: 10.1145/3579334.

[119] A. Hilton, J. Hirschmann, and C. Deccio, "Beware of IPs in Sheep's Clothing: Measurement and Disclosure of IP Spoofing Vulnerabilities," *IEEE/ACM Transactions on Networking*, 2022. DOI: 10.1109/TNET.2022.3149011.

[120] L. Steger, "Revisiting IPv6 Hitlists," Advised by Johannes Zirngibl*, Patrick Sattler and Juliane Aulbach, Master's Thesis, Technical University of Munich, 2021.

[121] RIPE NCC. "Total IPv6 Allocations and Assignments." (), [Online]. Available: https://www.ripe.net/analyse/statistics/amount-of-ipv6-addresses-allocated-and-assigned/ (visited on 01/25/2024).

[122] Google. "Statistics: IPv6 Adoption." (), [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html (visited on 01/25/2024).

[123] APNIC. "IPv6 Capable Rate by country (%)." (), [Online]. Available: https://stats.labs.apnic.net/ipv6 (visited on 01/25/2024).

[124] F. Aschenbrenner, T. Shreedhar, O. Gasser, N. Mohan, and J. Ott, "From Single Lane to Highways: Analyzing the Adoption of Multipath TCP in the Internet," in *2021 IFIP Networking Conference (IFIP Networking)*, 2021. DOI: 10.23919/IFIPNetworking52078.2021.9472785.

[125] Y. Nosyk, M. Korczyński, and A. Duda, "Routing Loops as Mega Amplifiers for DNS-Based DDoS Attacks," in *Proc. Passive and Active Measurement (PAM)*, 2022. DOI: 10.1007/978-3-030-98785-5_28.

[126] R. Almeida, í. Cunha, R. Teixeira, D. Veitch, and C. Diot, "Classification of Load Balancing in the Internet," in *Proc. IEEE Int. Conference on Computer Communications (INFOCOM)*, 2020. DOI: 10.1109/INFOCOM41043.2020.9155387.

[127] C. Deccio, A. Hilton, M. Briggs, T. Avery, and R. Richardson, "Behind Closed Doors: A Network Tale of Spoofing, Intrusion, and False DNS Security," in *Proc. ACM Internet Measurement Conference (IMC)*, 2020. DOI: 10.1145/3419394.3423649.

[128] N. Rodday, L. Kaltenbach, I. Cunha, R. Bush, E. Katz-Bassett, G. D. Rodosek, T. C. Schmidt, and M. Wählisch, "On the Deployment of Default Routes in Inter-Domain Routing," ser. TAURIN'21, 2021. DOI: 10.1145/3472951.3473505.

[129] Alexa. "Top 1M sites." The Alexa Top 1M was retired in 2022: https://web.archive.org/web/20211208204723/https://support.alexa.com/hc/en-us/articles/4410503838999. Updates to the list were available for one more year. (2022), [Online]. Available: https://web.archive.org/web/20220404013924/https://www.alexa.com/topsites (visited on 01/05/2024).

[130] R. Beverly, "Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery," in *Proc. ACM Internet Measurement Conference (IMC)*, Santa Monica, California, USA, 2016.

[131] RIPE NCC. "Routing Information Service (RIS)." (), [Online]. Available: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris (visited on 01/25/2024).

[132] C. Huitema, *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)*, RFC 4380, Feb. 2006. DOI: 10.17487/RFC4380. [Online]. Available: https://www.rfc-editor.org/info/rfc4380.

[133] Maxmind. "GeoLite2 Free Geolocation Data." (2022), [Online]. Available: https://dev.maxmind.com/geoip/geolite2-free-geolocation-data?lang=en (visited on 01/05/2024).

[134] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, "IP Geolocation Databases: Unreliable?" *ACM SIGCOMM Computer Communication Review*, 2011. DOI: 10.1145/1971162.1971171.

[135] Q. Scheitle, O. Gasser, P. Sattler, and G. Carle, "HLOC: Hints-Based Geolocation Leveraging Multiple Measurement Frameworks," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2017. DOI: 10.23919/TMA.2017.8002903.

[136] L. Izhikevich, R. Teixeira, and Z. Durumeric, "LZR: Identifying unexpected internet services," in *Proc. USENIX Security Symposium*, Aug. 2021. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/izhikevich.

[137] L. Alt, R. Beverly, and A. Dainotti, "Uncovering Network Tarpits with Degreaser," in *Proceedings of the 30th Annual Computer Security Applications Conference*, 2014. DOI: 10.1145/2664243.2664285.

[138] A. Mukaddam, I. Elhajj, A. Kayssi, and A. Chehab, "IP Spoofing Detection Using Modified Hop Count," in *IEEE International Conference on Advanced Information Networking and Applications*, 2014. DOI: 10.1109/AINA.2014.62.

[139] M. Backes, T. Holz, C. Rossow, T. Rytilahti, M. Simeonovski, and B. Stock, "On the Feasibility of TTL-Based Filtering for DRDoS Mitigation," in *Research in Attacks, Intrusions, and Defenses*, 2016. DOI: 10.1007/978-3-319-45719-2_14.

[140] Rapid7 Project Sonar. "Open Data." (2023), [Online]. Available: https://opendata.rapid7.com/ (visited on 01/05/2024).

[141] K. Z. Sediqi, L. Prehn, and O. Gasser, "Hyper-Specific Prefixes: Gotta Enjoy the Little Things in Interdomain Routing," *ACM SIGCOMM Computer Communication Review*, 2022. DOI: 10.1145/3544912.3544916.

[142] J. Durand, I. Pepelnjak, and G. Döring, *BGP Operations and Security*, RFC 7454, Feb. 2015. DOI: 10.17487/RFC7454. [Online]. Available: https://www.rfc-editor.org/info/rfc7454.

[143] MANRS. "Prefix filter configuration tools." (), [Online]. Available: https://www.manrs.org/isps/guide/filtering/ (visited on 01/25/2024).

[144] P. Sattler, J. Zirngibl, M. Jonker, O. Gasser, G. Carle, and R. Holz. "HRP Website with data." (2023), [Online]. Available: https://hrp-stats.github.io/ (visited on 10/05/2023).

[145] "University of Oregon Route Views Project." (2023), [Online]. Available: http://www.routeviews.org/routeviews/ (visited on 03/25/2023).

[146] J. Sitnicki. "It's crowded in here!" (2019), [Online]. Available: https://blog.cloudflare.com/its-crowded-in-here/ (visited on 01/25/2024).

[147] Cloudflare. "Cloudflare Spectrum - Network ports." (), [Online]. Available: https://developers.cloudflare.com/fundamentals/get-started/reference/network-ports/ (visited on 01/25/2024).

[148] ——, "Cloudflare Spectrum." (), [Online]. Available: https://www.cloudflare.com/products/cloudflare-spectrum/ (visited on 01/25/2024).

[149] M. Fayed, L. Bauer, V. Giotsas, S. Kerola, M. Majkowski, P. Odintsov, J. Sitnicki, T. Chung, D. Levin, A. Mislove, C. A. Wood, and N. Sullivan, "The Ties That Un-Bind: Decoupling IP from Web Services and Sockets for Robust Addressing Agility at CDN-Scale," in *Proc. ACM SIGCOMM*, 2021. DOI: 10.1145/3452296.3472922.

[150] M. Fayed. "Unbuckling the narrow waist of IP: Addressing Agility for Names and Web Services." (2021), [Online]. Available: https://blog.cloudflare.com/addressing-agility/ (visited on 01/25/2024).

[151] J. Zirngibl, L. Steger, P. Sattler, O. Gasser, and G. Carle. "Data and Analysis at TUM University Library: Rusty Clusters? Dusting an IPv6 Research Foundation." (2022), [Online]. Available: https://mediatum.ub.tum.de/1686542.

[152] L. Kuang, "Target Generation for IPv6 Hitlists," Advised by Lion Steger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2022.

[153] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman, "Zippier ZMap: Internet-Wide Scanning at 10 Gbps," in *WOOT*, 2014.

[154] CAIDA. "Ark IPv6 Topology Dataset." (2022), [Online]. Available: https://catalog.caida.org/details/dataset/ipv6_allpref_topology (visited on 01/25/2024).

[155] IPinfo.io. "The trusted source for IP address data." (2023), [Online]. Available: https://ipinfo.io/ (visited on 01/05/2024).

[156] D. T. Narten, T. Jinmei, and D. S. Thomson, *IPv6 Stateless Address Autoconfiguration*, RFC 4862, Sep. 2007. DOI: 10.17487/RFC4862. [Online]. Available: https://www.rfc-editor.org/info/rfc4862.

[157] E. Rescorla, K. Oku, N. Sullivan, and C. A. Wood, "TLS Encrypted Client Hello," Internet Engineering Task Force, Internet-Draft draft-ietf-tls-esni-17, Oct. 2023, Work in Progress, 48 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-esni/17/.

[158] A. Ghedini. "Speeding up HTTPS and HTTP/3 negotiation with... DNS." (2020), [Online]. Available: https://blog.cloudflare.com/speeding-up-https-and-http-3-negotiation-with-dns/ (visited on 01/25/2024).

[159] Akamai. "New SVCB & HTTPS Resource Records in the wild." (2020), [Online]. Available: https://community.akamai.com/customers/s/article/NetworkOperatorCommunityNewSVCBHTTPSResourceRecordsinthewild20201128135350 (visited on 01/25/2024).

[160] T. Pauly. "DNS HTTPS/SVCB record type support in iOS 14." (2020), [Online]. Available: https://mailarchive.ietf.org/arch/msg/dnsop/ldaCto09yaOuSXM92HgJhGqmPJw/ (visited on 01/25/2024).

[161] Chrome Platform Status. "Feature: HTTP -> HTTPS redirect for HTTPS DNS records." (2021), [Online]. Available: https://chromestatus.com/feature/5485544526053376 (visited on 01/25/2024).

[162] D. Crocker, *Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves*, RFC 8552, Mar. 2019. DOI: 10.17487/RFC8552. [Online]. Available: https://www.rfc-editor.org/info/rfc8552.

[163] W. Kumari. "Breaking the logjam that is draft-ietf-dnsop-svcb-https." (2023), [Online]. Available: https://mailarchive.ietf.org/arch/msg/dnsop/5aiWtJbmAoqj7-5oD03Rgw1PEoo/ (visited on 01/25/2024).

[164] B. M. Schwartz, *Service Binding Mapping for DNS Servers*, RFC 9461, Nov. 2023. DOI: 10.17487/RFC9461. [Online]. Available: https://www.rfc-editor.org/info/rfc9461.

[165] ——, "Service Binding Mapping for DNS Servers," Internet Engineering Task Force, Internet-Draft draft-ietf-add-svcb-dns-09, Jun. 2023, Work in Progress, 12 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-add-svcb-dns/09/.

[166] M. Prince. "Introducing: I'm Under Attack Mode." (2012), [Online]. Available: https://blog.cloudflare.com/introducing-im-under-attack-mode/ (visited on 01/25/2024).

[167] A. van der Mandele, A. Ghedini, C. Wood, and R. Mehra. "Encrypted Client Hello - the last puzzle piece to privacy." (2023), [Online]. Available: https://blog.cloudflare.com/announcing-encrypted-client-hello (visited on 01/25/2024).

[168] Cloudflare. "Early Hints and Encrypted Client Hello (ECH) are currently disabled globally." (2023), [Online]. Available: https://community.cloudflare.com/t/early-hints-and-encrypted-client-hello-ech-are-currently-disabled-globally/567730 (visited on 01/25/2024).

[169] S. Deusch, "Analyzing the Effect of Domain Parking on DNS Based Research," Advised by Johannes Zirngibl*, Patrick Sattler and Juliane Aulbach, Bachelor's Thesis, Technical University of Munich, 2021.

[170] N. P. Hoang, A. A. Niaki, M. Polychronakis, and P. Gill, "The Web is Still Small after More than a Decade," *ACM SIGCOMM Computer Communication Review*, 2020. DOI: 10.1145/3402413.3402417.

[171] S. Wang, K. MacMillan, B. Schaffner, N. Feamster, and M. Chetty, "A First Look at the Consolidation of DNS and Web Hosting Providers," 2021. [Online]. Available: https://arxiv.org/abs/2110.15345.

[172] T. V. Doan, R. van Rijswijk-Deij, O. Hohlfeld, and V. Bajpai, "An Empirical View on Consolidation of the Web," *ACM Trans. Internet Technol.*, 2022. DOI: 10.1145/3503158.

[173] Internet Society. "Global Internet Report 2019: Consolidation in the Internet Economy." (2019), [Online]. Available: https://future.internetsociety.org/2019/ (visited on 01/25/2024).

[174] GoDaddy. "Park a domain registered with GoDaddy." (2022), [Online]. Available: https://www.godaddy.com/help/park-a-domain-registered-with-godaddy-23936.

[175] J. Zirngibl, S. Deusch, P. Sattler, J. Aulbach, G. Carle, and M. Jonker. "Data: Domain Parking: Largely Present, Rarely Considered!" (2022), [Online]. Available: https://tma22-parking.github.io.

[176] J. Park, J. Choi, D. Nyang, and A. Mohaisen, "Transparency in the New gTLD Era: Evaluating the DNS Centralized Zone Data Service," *IEEE Transactions on Network and Service Management*, 2019. DOI: 10.1109/HotWeb.2016.18.

[177] Mozilla Foundation. "Public Suffic List." (2022), [Online]. Available: https://publicsuffix.org/ (visited on 02/22/2022).

[178] GoDaddy. "GoDaddy Acquires Afternic - Primes Domain Aftermarket for New TLDs Move Also Delivers Improved Service, Selection & Speed to More Aftermarket Customers." (2013), [Online]. Available: https://aboutus.godaddy.net/newsroom/press-releases/press-release-details/2013/GoDaddy-Acquires-Afternic---Primes-Domain-Aftermarket-for-New-

TLDs-Move-Also-Delivers-Improved-Service-Selection--Speed-to-More-Aftermarket-Customers/default.aspx.

[179] Domain Name Wire. "ParkingCrew acquires NameDrive." (2015), [Online]. Available: https://domainnamewire.com/2015/07/16/parkingcrew-acquires-namedrive/.

[180] Common Crawl. "The Common Crawl Corpus." (2022), [Online]. Available: https://commoncrawl.org/ (visited on 01/05/2024).

[181] Cloudflare. "Cloudflare Radar." (2023), [Online]. Available: https://radar.cloudflare.com/ (visited on 01/05/2024).

[182] M. Allman, "Comments on DNS Robustness," in *Proc. ACM Internet Measurement Conference (IMC)*, 2018. DOI: 10.1145/3278532.3278541.

[183] S. Lloyd, C. Hernandez-Gañan, and S. Tajalizadehkhoob, "Towards more rigorous domain-based metrics: quantifying the prevalence and implications of "Active" Domains," in *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2023. DOI: 10.1109/EuroSPW59978.2023.00066.

[184] M. Bishop, *HTTP/3*, RFC 9114, Jun. 2022. DOI: 10.17487/RFC9114. [Online]. Available: https://www.rfc-editor.org/info/rfc9114.

[185] IETF QUIC Working Group. "Implementations." (2023), [Online]. Available: https://github.com/quicwg/base-drafts/wiki/Implementations (visited on 03/25/2023).

[186] Schinazi, David and Yang, Fan and Swett, Ian. "Chrome is deploying http/3 and ietf quic." (2020), [Online]. Available: https://blog.chromium.org/2020/10/chrome-is-deploying-http3-and-ietf-quic.html (visited on 01/25/2024).

[187] M. Joras and Y. Chi. "How Facebook is bringing QUIC to billions." (2020), [Online]. Available: https://engineering.fb.com/2020/10/21/networking-traffic/how-facebook-is-bringing-quic-to-billions/ (visited on 01/25/2024).

[188] D. Damjanovic. "QUIC and HTTP/3 Support now in Firefox Nightly and Beta." (2021), [Online]. Available: https://hacks.mozilla.org/2021/04/quic-and-http-3-support-now-in-firefox-nightly-and-beta/ (visited on 01/25/2024).

[189] B. M. Schwartz, M. Bishop, and E. Nygren, "Service binding and parameter specification via the DNS (DNS SVCB and HTTPS RRs)," Internet Engineering Task Force, Internet-Draft draft-ietf-dnsop-svcb-https-05, Apr. 2021, Work in Progress, 56 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-dnsop-svcb-https/05/.

[190] IETF QUIC Working Group. "QUIC Versions." (), [Online]. Available: https://github.com/quicwg/base-drafts/wiki/QUIC-Versions (visited on 01/25/2024).

[191] D. Schinazi and E. Rescorla, *Compatible Version Negotiation for QUIC*, RFC 9368, May 2023. DOI: 10.17487/RFC9368. [Online]. Available: https://www.rfc-editor.org/info/rfc9368.

[192] M. Nottingham, P. McManus, and J. Reschke, *HTTP Alternative Services*, RFC 7838, Apr. 2016. DOI: 10.17487/RFC7838. [Online]. Available: https://www.rfc-editor.org/info/rfc7838.

[193] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," Internet Engineering Task Force, Internet-Draft draft-ietf-quic-transport-34, Jan. 2021, Work in Progress, 151 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-quic-transport/34/.

[194] M. Duke, *QUIC Version 2*, RFC 9369, May 2023. DOI: 10.17487/RFC9369. [Online]. Available: https://www.rfc-editor.org/info/rfc9369.

[195] M. Thomson and S. Turner, *Using TLS to Secure QUIC*, RFC 9001, May 2021. DOI: 10.17487/RFC9001. [Online]. Available: https://rfc-editor.org/rfc/rfc9001.txt.

[196] crt.sh. "Certificates for Google Video in CT Log." Relevant Snapshot: https://web.archive.org/web/20210526164544/https://crt.sh/?q=googlevideo.com. (2021), [Online]. Available: https://crt.sh/?q=googlevideo.com (visited on 01/25/2024).

[197] M. Thomson and S. Turner, "Using TLS to Secure QUIC," Internet Engineering Task Force, Internet-Draft draft-ietf-quic-tls-34, Jan. 2021, Work in Progress, 66 pp. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-quic-tls-34.

[198] D. E. E. 3rd, *Transport Layer Security (TLS) Extensions: Extension Definitions*, RFC 6066, Jan. 2011. DOI: 10.17487/RFC6066. [Online]. Available: https://www.rfc-editor.org/info/rfc6066.

[199] Facbook. "Proxygen: Facebook's C++ HTTP Libraries." (2023), [Online]. Available: https://github.com/facebook/proxygen.

[200] Meta Platforms. "mvfst." (2023), [Online]. Available: https://github.com/facebookincubator/mvfst.

[201] Kim, Hyojeong and Zeng, James Hongyi. "Steering oceans of content to the world." (2017), [Online]. Available: https://engineering.fb.com/2017/08/21/networking-traffic/steering-oceans-of-content-to-the-world/ (visited on 01/25/2024).

[202] LiteSpeed Technologies. "LiteSpeed QUIC (LSQUIC) Library." (2023), [Online]. Available: https://github.com/litespeedtech/lsquic.

[203] NGINX QUIC. "Welcome to the demo site for nginx-quic." (2023), [Online]. Available: https://quic.nginx.org/.

[204] G. Alessandro. "Experiment with HTTP/3 using NGINX and quiche." (2019), [Online]. Available: https://blog.cloudflare.com/experiment-with-http-3-using-nginx-and-quiche/ (visited on 01/25/2024).

[205] F. Gebauer, "Evaluating Different QUIC Scan Approaches," Advised by Johannes Zirngibl* and Patrick Sattler, Bachelor's Thesis, Technical University of Munich, 2022.

[206] Cloudflare. "quiche." (2023), [Online]. Available: https://github.com/cloudflare/quiche.

[207] A. Yu and T. A. Benson, "Dissecting Performance of Production QUIC," in *Proc. of the Web Conference*, 2021. DOI: 10.1145/3442381.3450103.

[208] J. Zirngibl, F. Gebauer, P. Sattler, M. Sosnowski, and G. Carle. "Test Environment and Identification Tools." (2023), [Online]. Available: https://github.com/quic-hunter/libraries (visited on 01/05/2024).

[209] Google. "QUICHE." (2023), [Online]. Available: https://github.com/google/quiche.

[210] Web Almanac. "Negotiating HTTP/3." (2021), [Online]. Available: https://almanac.httparchive.org/en/2021/http#negotiating-http3 (visited on 02/22/2024).

[211] J. C. Herrera. "Deliver Fast, Reliable, and Secure Web Experiences with HTTP/3." (2023), [Online]. Available: https://www.akamai.com/blog/performance/deliver-fast-reliable-secure-web-experiences-http3 (visited on 02/22/2024).

[212] LSQUIC. "LSQUIC ALPN error." (2024), [Online]. Available: https://github.com/litespeedtech/lsquic/blame/master/src/liblsquic/lsquic%5C_mini%5C_conn%5C_ietf.c%5C#L2157 (visited on 02/22/2024).

[213] IANA. "QUIC Transport Parameters." (2023), [Online]. Available: https://www.iana.org/assignments/quic/quic.xhtml#quic-transport (visited on 02/22/2024).

[214] ——, "Transport Layer Security (TLS) Extensions." (2023), [Online]. Available: https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml (visited on 02/22/2024).

[215] Quant. "Randomization of quic transport parameter order." (), [Online]. Available: https://github.com/NTAP/quant/blob/main/lib/src/tls.c%5C#L857 (visited on 01/25/2024).

[216] Google. "Randomization of QUIC Transport Parameter Order." (), [Online]. Available: https://github.com/google/quiche/blob/main/quiche/quic/core/crypto/transport%5C_parameters.cc%5C#L832 (visited on 01/25/2024).

[217] IANA Registry. "TLS Cipher Suites." (2023), [Online]. Available: https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-4 (visited on 02/22/2024).

[218] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet Inter-Domain Traffic," *ACM SIGCOMM Computer Communication Review*, 2010. DOI: 10.1145/1851275.1851194.

[219] A. Lapiene. "Improving Website Performance with LiteSpeed." (2019), [Online]. Available: https://www.hostinger.com/blog/introducing-litespeed (visited on 01/25/2024).

[220] A2 Hosting. "The Best LiteSpeed Web Server Hosting Solution." (), [Online]. Available: https://www.a2hosting.com/litespeed-hosting/ (visited on 01/25/2024).

[221] I. Kunze, C. Sander, and K. Wehrle, "Does It Spin? On the Adoption and Use of QUIC's Spin Bit," in *Proc. ACM Internet Measurement Conference (IMC)*, 2023. DOI: 10.1145/3618257.3624844.

[222] C. Sander, I. Kunze, L. Blöcher, M. Kosek, and K. Wehrle, "ECN with QUIC: Challenges in the Wild," in *Proc. ACM Internet Measurement Conference (IMC)*, 2023. DOI: 10.1145/3618257.3624821.

[223] ICANN. "The new generic top-level domains program: Next round." (2023), [Online]. Available: https://newgtlds.icann.org/en/next-round (visited on 03/27/2024).

[224] M. Kempf, "Evaluation of the QUIC Spin Bit for RTT Estimation," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2019.

[225] L. Keller, "Packet Pacing with the QUIC Protocol," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2020.

[226] M. Mussner, "In Depth Analysis of QUIC's Lack of Kernel Optimizations," Advised by Benedikt Jaeger* and Johannes Zirngibl, Master's Thesis, Technical University of Munich, 2020.

[227] S. Voit, "Bringing QUIC to High-speed Networks," Advised by Benedikt Jaeger* and Johannes Zirngibl, Master's Thesis, Technical University of Munich, 2021.

[228] D. Hegedüs, "The First Year of QUIC v1 Deployment," Advised by Johannes Zirngibl*, Patrick Sattler, Benedikt Jaeger and Juliane Aulbach, Bachelor's Thesis, Technical University of Munich, 2021.

[229] K. Ploch, "QUIC Performance on 10G Links," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2022.

[230] M. Kutter, "Evaluation of Scalability and Limitations of HTTP/3," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2022.

[231] M. Kempf, "Analysis of Performance Limitations in QUIC Implementations," Advised by Benedikt Jaeger* and Johannes Zirngibl, Master's Thesis, Technical University of Munich, 2022.

[232] S. K. Guayana, "Analyzing the Effect of Transport Parameters on QUIC's Performance," Advised by Johannes Zirngibl* and Benedikt Jaeger, Bachelor's Thesis, Technical University of Munich, 2022.

[233] L. Otting, "Improving QUIC with User Space Networking," Advised by Kilian Holzinger*, Benedikt Jaeger and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2023.

[234] N. Gauder, "Performance Evaluation of Cryptography in QUIC," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2023.

[235] R. Stadler, "QUICkly Reaching Maximum Throughput: A Comparative Evaluation of QUIC Implementations," Advised by Benedikt Jaeger* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2023.

[236] J. Späth, "QUIC Performance Improvements Using DPDK," Advised by Johannes Zirngibl*, Benedikt Jaeger and Kilian Holzinger, Master's Thesis, Technical University of Munich, 2023.

[237] N. Beck, "Root Cause Analysis for Throughput Limitations of QUIC Connections," Advised by Simon Bauer* and Johannes Zirngibl, Master's Thesis, Technical University of Munich, 2023.

[238] M. Buhl, "QUIC Kernel: an In-Kernel Port and Socket Abstraction Layer," Advised by Johannes Zirngibl*, Benedikt Jaeger and Kilian Holzinger, Interdisciplinary Project, Technical University of Munich, 2023.

[239] M. Haid, "Impact of a TEE on QUIC Performance," Advised by Marcel Kempf* and Filip Rezabek and Johannes Zirngibl and Benedikt Jaeger, Bachelor's Thesis, Technical University of Munich, 2023.

[240] D. Kreutzer, "Nameserver Rate Limits - Dynamic Adjustment of Scan Behavior," Advised by Johannes Zirngibl* and Johannes Naab, Bachelor's Thesis, Technical University of Munich, 2019.

[241] L. Schedelbeck, "rDNS Leaks - Disclosing the Real Infrastructure of Shadowed Services," Advised by Johannes Zirngibl* and Patrick Sattler, Bachelor's Thesis, Technical University of Munich, 2019.

[242] R. B. Reif, "Analysis of EDNS Client-Subnet Load Balancing," Advised by Patrick Sattler* and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2019.

[243] C. Wahl, "Analyzing the Stability and Expressiveness of Large-Scale DNS Scans," Advised by Patrick Sattler*, Johannes Zirngibl and Juliane Aulbach, Master's Thesis, Technical University of Munich, 2020.

[244] Z. Sonkaya, "Development of an Efficient Large Scale DNS Scanning Pipeline," Advised by Patrick Sattler* and Johannes Zirngibl, Interdisciplinary Project, Technical University of Munich, 2021.

[245] P. Großmann, "Extended Usage Analysis of EDNS Client Subnet," Advised by Patrick Sattler*, Johannes Zirngibl and Lion Steger, Bachelor's Thesis, Technical University of Munich, 2022.

[246] C. B. Dietze, "Tracking the Lifetime of Domains," Advised by Johannes Zirngibl* and Patrick Sattler, Master's Thesis, Technical University of Munich, 2023.

[247] R. B. Reif, "Detecting BGP Hijacking in Real Time," Advised by Patrick Sattler* and Johannes Zirngibl, Interdisciplinary Project, Technical University of Munich, 2020.

[248] P. Henschke, "Analyzing BGP as a Graph," Advised by Johannes Zirngibl*, Patrick Sattler and Juliane Aulbach, Bachelor's Thesis, Technical University of Munich, 2021.

[249] R. Schmid, "ROV + IRR: Are Authorized Routes Registered?" Advised by Johannes Zirngibl*, Patrick Sattler and Juliane Aulbach, Bachelor's Thesis, Technical University of Munich, 2021.

[250] K. Ilse, "IPv6 Deployment Analysis using BGP Announcements," Advised by Patrick Sattler*, Johannes Zirngibl and Juliane Aulbach, Bachelor's Thesis, Technical University of Munich, 2021.

[251] T. Wothge, "Industrial Control Systems (ICS) Protocol Detection," Advised by Patrick Sattler*, Lars Wüstrich and Johannes Zirngibl, Bachelor's Thesis, Technical University of Munich, 2021.

[252] Z. Lu, "Structural Analysis of the Great Firewall of China," Advised by Lion Steger* and Johannes Zirngibl, Master's Thesis, Technical University of Munich, 2022.

[253] I. Varas, "Autonomous System Models using BGP Data and GNNs," Advised by Max Helm*, Benedikt Jaeger, Johannes Zirngibl and Patrick Sattler, Bachelor's Thesis, Technical University of Munich, 2023.

[254] F. Bauernschmitt, "Evalution of Network Categorization Strategies," Advised by Lion Steger*, Johannes Zirngibl and Patrick Sattler, Bachelor's Thesis, Technical University of Munich, 2022.

[255] B. R. Schaschko, “Inferring AS Links from a Tier 1 Dataset,” Advised by Johannes Zirngibl* and Patrick Sattler, Bachelor’s Thesis, Technical University of Munich, 2023.

[256] N. Buchner, “Influence of Network Conditions on PTP Accuracy,” Advised by Max Helm*, Henning Stubbe and Johannes Zirngibl, Bachelor’s Thesis, Technical University of Munich, 2019.

[257] C. Kilb, “Blocklists: What is blocked and why?” Advised by Johannes Zirngibl*, Patrick Sattler and Markus Sosnowski, Interdisciplinary Project, Technical University of Munich, 2020.

[258] S. H. Kappes, “An Analysis of the Development and Early Deployment of Encrypted SNI,” Advised by Johannes Zirngibl*, Max Helm and Patrick Sattler, Bachelor’s Thesis, Technical University of Munich, 2020.

[259] R. Dillitz, “Uncovering PTP Master Clocks in the Wild,” Advised by Johannes Zirngibl*, Max Helm and Henning Stubbe, Bachelor’s Thesis, Technical University of Munich, 2020.

[260] J. von der Heidt, “Analyzing PTP Master Clocks in the Wild,” Advised by Johannes Zirngibl*, Max Helm and Henning Stubbe, Bachelor’s Thesis, Technical University of Munich, 2020.

[261] B. Riegel, “Assessing Link Utilization From Passive Datasets,” Advised by Simon Bauer* and Johannes Zirngibl, Bachelor’s Thesis, Technical University of Munich, 2021.

[262] L. Lehle, “Efficient Processing of Large Network Captures,” Advised by Lars Wüstrich*, Johannes Zirngibl and Christian Lübben, Bachelor’s Thesis, Technical University of Munich, 2021.

[263] F. Myhsok, “Blocklists: Who is blocked?” Advised by Johannes Zirngibl*, Patrick Sattler and Markus Sosnowski, Bachelor’s Thesis, Technical University of Munich, 2021.

[264] Y. Wibowo, “Analysis of Blocklisted TLS Servers,” Advised by Johannes Zirngibl* and Patrick Sattler, Bachelor’s Thesis, Technical University of Munich, 2022.

[265] R. Dillitz, “Transformation and Evaluation of TLS Behavior Graphs,” Advised by Johannes Zirngibl*, Benedikt Jaeger and Markus Sosnowski, Master’s Thesis, Technical University of Munich, 2022.

[266] T. Gräbner, “Setup and Deployment of a Large Scale Certificate Scan Database,” Advised by Patrick Sattler* and Johannes Zirngibl, Bachelor’s Thesis, Technical University of Munich, 2022.

[267] M. S. Shaukat, “Measuring the Impact of Transport Layer Protocols and Their Configuration on the Performance of Connections,” Advised by Simon Bauer*, Patrick Sattler and Johannes Zirngibl, Master’s Thesis, Technical University of Munich, 2022.

[268] C. B. Dietze, “Setup and Deployment of a Resilient Internet Scanning Infrastructure,” Advised by Patrick Sattler* and Johannes Zirngibl, Interdisciplinary Project, Technical University of Munich, 2022.

[269] D. Weissmann, “The Impact of iCloud Private Relay on Networks,” Advised by Patrick Sattler* and Johannes Zirngibl, Bachelor’s Thesis, Technical University of Munich, 2023.

[270] T. Wothge, “Egress Node Behavior in iCloud Private Relay,” Advised by Patrick Sattler*, Lars Wüstrich, Johannes Zirngibl and Lion Steger, Interdisciplinary Project, Technical University of Munich, 2023.

[271] T. Zierl, “Evaluating Domain Presence in Certificate Transparency Logs,” Advised by Patrick Sattler* and Johannes Zirngibl, Bachelor’s Thesis, Technical University of Munich, 2023.

[272] L. Pydde, “TLS Certificate Usage Evaluation,” Advised by Patrick Sattler* and Johannes Zirngibl, Bachelor’s Thesis, Technical University of Munich, 2023.

[273] T. Wasner, “Continuous Monitoring and Quality Assessment of Internet-wide Scans,” Advised by Patrick Sattler and Johannes Zirngibl, Interdisciplinary Project, Technical University of Munich, 2023.

[274] M. Kirstein, “Happy Eyeballs: A Comprehensive Analysis of the Deployment and Configuration Across Various Versions and Implementations,” Advised by Patrick Sattler*, Johannes Zirngibl and Lars Wüstrich, Bachelor’s Thesis, Technical University of Munich, 2024.